CIS AWS Compute Services Benchmark

v1.1.0 - 01-30-2025

Terms of Use

Please see the below link for our current terms of use:

https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/

For information on referencing and/or citing CIS Benchmarks in 3rd party documentation (including using portions of Benchmark Recommendations) please contact CIS Legal (CISLegal@cisecurity.org) and request guidance on copyright usage.

NOTE: It is ***NEVER*** acceptable to host a CIS Benchmark in ***ANY*** format (PDF, etc.) on a 3rd party (non-CIS owned) site.

Table of Contents

Overview

All CIS Benchmarks™ (Benchmarks) focus on technical configuration settings used to maintain and/or increase the security of the addressed technology, and they should be used in **conjunction** with other essential cyber hygiene tasks like:

- Monitoring the base operating system and applications for vulnerabilities and quickly updating with the latest security patches.
- End-point protection (Antivirus software, Endpoint Detection and Response (EDR), etc.).
- Logging and monitoring user and system activity.

In the end, the Benchmarks are designed to be a key **component** of a comprehensive cybersecurity program.

Important Usage Information

All Benchmarks are available free for non-commercial use from the CIS Website. They can be used to manually assess and remediate systems and applications. In lieu of manual assessment and remediation, there are several tools available to assist with assessment:

- CIS Configuration Assessment Tool (CIS-CAT® Pro Assessor)
- CIS Benchmarks™ Certified 3rd Party Tooling

These tools make the hardening process much more scalable for large numbers of systems and applications.

NOTE: Some tooling focuses only on the Benchmark Recommendations that can be fully automated (skipping ones marked **Manual**). It is important that ***ALL*** Recommendations (**Automated** and **Manual**) be addressed since all are important for properly securing systems and are typically in scope for audits.

Key Stakeholders

Cybersecurity is a collaborative effort, and cross functional cooperation is imperative within an organization to discuss, test, and deploy Benchmarks in an effective and efficient way. The Benchmarks are developed to be best practice configuration guidelines applicable to a wide range of use cases. In some organizations, exceptions to specific Recommendations will be needed, and this team should work to prioritize the problematic Recommendations based on several factors like risk, time, cost, and labor. These exceptions should be properly categorized and documented for auditing purposes.

Apply the Correct Version of a Benchmark

Benchmarks are developed and tested for a specific set of products and versions and applying an incorrect Benchmark to a system can cause the resulting pass/fail score to be incorrect. This is due to the assessment of settings that do not apply to the target systems. To assure the correct Benchmark is being assessed:

- **Deploy the Benchmark applicable to the way settings are managed in the environment:** An example of this is the Microsoft Windows family of Benchmarks, which have separate Benchmarks for Group Policy, Intune, and Stand-alone systems based upon how system management is deployed. Applying the wrong Benchmark in this case will give invalid results.

- **Use the most recent version of a Benchmark**: This is true for all Benchmarks, but especially true for cloud technologies. Cloud technologies change frequently and using an older version of a Benchmark may have invalid methods for auditing and remediation.

Exceptions

The guidance items in the Benchmarks are called recommendations and not requirements, and exceptions to some of them are expected and acceptable. The Benchmarks strive to be a secure baseline, or starting point, for a specific technology, with known issues identified during Benchmark development are documented in the Impact section of each Recommendation. In addition, organizational, system specific requirements, or local site policy may require changes as well, or an exception to a Recommendation or group of Recommendations (e.g. A Benchmark could Recommend that a Web server not be installed on the system, but if a system's primary purpose is to function as a Webserver, there should be a documented exception to this Recommendation for that specific server).

In the end, exceptions to some Benchmark Recommendations are common and acceptable, and should be handled as follows:

- The reasons for the exception should be reviewed cross-functionally and be well documented for audit purposes.
- A plan should be developed for mitigating, or eliminating, the exception in the future, if applicable.
- If the organization decides to accept the risk of this exception (not work toward mitigation or elimination), this should be documented for audit purposes.

It is the responsibility of the organization to determine their overall security policy, and which settings are applicable to their unique needs based on the overall risk profile for the organization.

Remediation

CIS has developed Build Kits for many technologies to assist in the automation of hardening systems. Build Kits are designed to correspond to Benchmark's "Remediation" section, which provides the manual remediation steps necessary to make that Recommendation compliant to the Benchmark.

When remediating systems (changing configuration settings on deployed systems as per the Benchmark's Recommendations), please approach this with caution and test thoroughly.

The following is a reasonable remediation approach to follow:

- CIS Build Kits, or internally developed remediation methods should never be applied to production systems without proper testing.
- Proper testing consists of the following:
  - Understand the configuration (including installed applications) of the targeted systems. Various parts of the organization may need different configurations (e.g., software developers vs standard office workers).
  - Read the Impact section of the given Recommendation to help determine if there might be an issue with the targeted systems.
  - Test the configuration changes with representative lab system(s). If issues arise during testing, they can be resolved prior to deploying to any production systems.
  - When testing is complete, initially deploy to a small sub-set of production systems and monitor closely for issues. If there are issues, they can be resolved prior to deploying more broadly.
  - When the initial deployment above is completes successfully, iteratively deploy to additional systems and monitor closely for issues. Repeat this process until the full deployment is complete.

Summary

Using the Benchmarks Certified tools, working as a team with key stakeholders, being selective with exceptions, and being careful with remediation deployment, it is possible to harden large numbers of deployed systems in a cost effective, efficient, and safe manner.

NOTE: As previously stated, the PDF versions of the CIS Benchmarks™ are available for free, non-commercial use on the CIS Website. All other formats of the CIS Benchmarks™ (MS Word, Excel, and Build Kits) are available for CIS SecureSuite® members.

CIS-CAT® Pro is also available to CIS SecureSuite® members.

Target Technology Details

This document provides prescriptive guidance for configuring security options for the services within the Compute category in AWS. This Benchmark is intended to be used in conjunction with the CIS Amazon Web Services Foundations Benchmark. For more information about this approach see the Introduction section of this document.
The specific AWS Services in scope for this document include:

- Amazon Elastic Cloud Compute (EC2)
- Amazon Lightsail
- AWS Lambda
- AWS Batch
- AWS Elastic Beanstalk
- AWS Serverless Application Repository
- AWS Outposts
- EC2 Image Builder
- AWS App Runner
- AWS SimSpace Weaver

To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at benchmarkinfo@cisecurity.org.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, platform deployment, and/or DevOps personnel who plan to develop, deploy, assess, or secure solutions in Amazon Web Services.

Consensus Guidance

This CIS Benchmark™ was created using a consensus review process comprised of a global community of subject matter experts. The process combines real world experience with data-based information to create technology specific guidance to assist users to secure their environments. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS Benchmark undergoes two phases of consensus review. The first phase occurs during initial Benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the Benchmark. This discussion occurs until consensus has been reached on Benchmark recommendations. The second phase begins after the Benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the Benchmark. If you are interested in participating in the consensus process, please visit https://workbench.cisecurity.org/.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
`Monospace font`	Used for inline code, commands, UI/Menu selections or examples. Text should be interpreted exactly as presented.
`<Monospace font in brackets>`	Text set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to reference other relevant settings, CIS Benchmarks and/or Benchmark Communities. Also, used to denote the title of a book, article, or other publication.
Bold font	Additional information or caveats things like **Notes**, **Warnings**, or **Cautions** (usually just the word itself and the rest of the text normal).

Recommendation Definitions

The following defines the various components included in a CIS recommendation as applicable. If any of the components are not applicable it will be noted, or the component will not be included in the recommendation.

Title

Concise description for the recommendation's intended configuration.

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

Profile

A collection of recommendations for securing a technology or a supporting platform. Most benchmarks include at least a Level 1 and Level 2 Profile. Level 2 extends Level 1 recommendations and is not a standalone profile. The Profile Definitions section in the benchmark provides the definitions as they pertain to the recommendations included for the technology.

Description

Detailed information pertaining to the setting with which the recommendation is concerned. In some cases, the description will include the recommended value.

Rationale Statement

Detailed reasoning for the recommendation to provide the user a clear and concise understanding on the importance of the recommendation.

Impact Statement

Any security, functionality, or operational consequences that can result from following the recommendation.

Audit Procedure

Systematic instructions for determining if the target system complies with the recommendation.

Remediation Procedure

Systematic instructions for applying recommendations to the target system to bring it into compliance according to the recommendation.

Default Value

Default value for the given setting in this recommendation, if known. If not known, either not configured or not defined will be applied.

References

Additional documentation relative to the recommendation.

CIS Critical Security Controls® (CIS Controls®)

The mapping between a recommendation and the CIS Controls is organized by CIS Controls version, Safeguard, and Implementation Group (IG). The Benchmark in its entirety addresses the CIS Controls safeguards of (v7) “5.1 - Establish Secure Configurations” and (v8) '4.1 - Establish and Maintain a Secure Configuration Process” so individual recommendations will not be mapped to these safeguards.

Additional Information

Supplementary information that does not correspond to any other field but may be useful to the user.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1**

  Items in this profile intend to:

  - be practical and prudent;
  - provide security-focused best practice hardening of a technology; and
  - limit the impact to the utility of the technology beyond acceptable means.

- **Level 2**

  This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is more critical than manageability and usability
  - acts as a defense in depth measure
  - may impact the utility or performance of the technology
  - may include additional licensing, cost, or addition of third-party software

Acknowledgements

This Benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Gregory Carpenter, Mike Wicks, Michelle Peterson, Chantel Duckworth, Rachel Rice

Author
Gregory Carpenter
Michelle Peterson

Contributor
Mike Wicks
Jason Kao
Harshal Khachane
Rachel Rice
Chantel Duckworth

Recommendations

1 Introduction

Benchmark Approach:

The suggested approach for securing your cloud environment is to start with the CIS Amazon Web Services Foundations Benchmark found here: https://www.cisecurity.org/benchmark/amazon_web_services/. The CIS Foundations benchmark provides prescriptive guidance for configuring a subset of Amazon Web Services with an emphasis on foundational, testable, and architecture agnostic settings including:

- AWS Identity and Access Management (IAM)
- AWS Config
- AWS CloudTrail
- AWS CloudWatch
- AWS Simple Notification Service (SNS)
- AWS Simple Storage Service (S3)
- AWS VPC (Default)

The Amazon Web Services Foundation Benchmark is what you should start with when setting up your AWS environment. It is also the foundation for which all other AWS service based benchmarks are built on so that as you grow your cloud presence and usage of the services offered you have the necessary guidance to securely configure your environment as it fits with your company's policy.

After configuring your environment to the CIS Amazon Web Services Foundations Benchmark, we suggest implementing the necessary configurations for the services utilized as defined in the associated product and service level benchmarks. The CIS Compute Benchmark provides prescriptive guidance for configuring security options for the services within Compute in AWS. The specific AWS Services in scope for this document include:

- Amazon EC2
- Amazon Lightsail
- AWS Lambda
- AWS Batch
- AWS Elastic Beanstalk
- AWS Serverless Application Repository
- AWS Outposts
- Amazon EC2 Image Builder
- AWS App Runner
- AWS SimSpace Weaver

All CIS Benchmarks are created and maintained through consensus-based collaboration. Should you have feedback, suggested changes, or just like to get involved in the continued maintenance and development of CIS Amazon Web Services Benchmarks, please register on CIS WorkBench at https://workbench.cisecurity.org and join the CIS Amazon Web Services Benchmarks community.

2 Amazon Elastic Cloud Compute (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. This section will contain recommendations for configuring your compute resources within EC2. Some of the security settings and related options might be applied differently depending on how you are using other EC2 services and functionality.

2.1 Amazon Machine Images (AMI)

This section contains recommendations for the security of Amazon Machine Images (AMI's) that you could utilize within the AWS EC2 Service. An Amazon Machine Image (AMI) is a image provided by AWS and its Partners. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you require multiple instances with the same configuration. You can use different AMIs to launch instances when you require instances with different configurations.

2.1.1 Ensure Consistent Naming Convention is used for Organizational AMI (Manual)

Profile Applicability:

- Level 1

Description:

The naming convention for AMI (Amazon Machine Images) should be documented and followed for any AMI's created.

Rationale:

The majority of AWS resources can be named and tagged. Most organizations have already created standardize naming conventions, and have existing rules in effect. They simply need to extend that for all AWS cloud resources to include Amazon Machine Images (AMI)

Audit:

Perform the following to determine what AMI's are created:
From the Console:

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane, under `Images`, click `AMIs`.
3. Review the list of AMIs.
4. Confirm that the AMI Name matches the organizational image naming policy.

From the Command Line:

1. Run aws ec2 describe-images.

```
aws ec2 describe-images --owner self --region us-west-2
```

2. Review the list of AMIs.
3. Confirm that the AMI Name matches the organizational image naming policy.

If any of the AMI Name's do not match the Organization policy refer to the remediation below.

Remediation:

If the AMI Name for an AMI doesn't follow Organization policy
Perform the following to copy and rename the AMI:
From the Console:

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.

2. In the left pane click `Images`, click `AMIs`.
3. Select the AMI that does not comply to the naming policy.
4. Click on `Actions`.
5. Click on `Copy AMI`.

```
Destination region - Select the region the AMI is in.

Name - `Enter the new Name`

Description - `Enter the new description`

Encryption - `Select` if it matches your image policy
```

6. Click on Copy AMI

```
Once the AMI has finished copying.
```

7. Select the AMI that does not comply to the naming policy.
8. Click on `Actions`.
9. Click on `Deregister`

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-images.html
2. https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/AMIs.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

2.1.2 Ensure Amazon Machine Images (AMIs) are encrypted (Automated)

Profile Applicability:

- Level 1

Description:

Amazon Machine Images should utilize EBS Encrypted snapshots

Rationale:

AMIs backed by EBS snapshots should use EBS encryption. Snapshot volumes can be encrypted and attached to an AMI.

Audit:

Perform the following to determine AMIs are encrypted:
From the Console:

1. Login to the IAM console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane click `Instances`, click `AMIs`.
3. In the `Details` tab.
4. Review the 'Block Devices'
5. Confirm that it ends with `encrypted`.

If it doesn't end with encrypted, refer to the remediation below.
From the Command Line:

1. Run the aws ec2 describe-images command

```
aws ec2 describe-images --region us-east-1 --owner self --filter "Name=block-
device-mapping.encrypted,Values=false" --query "Images[*].[ImageId]"
```

2. If this produces a list of AMI's make note as these are not encrypted, then refer to the remediation below.

Remediation:

Perform the following to encrypt AMI EBS Snapshots:
From the Console:

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane click on `AMIs`.
3. Select the AMI that does not comply to the encryption policy.
4. Click on `Actions`.
5. Click on `Copy AMI`.

```
    Destination region - `Select the region the AMI is in`.

    Name - `Enter the new Name`

    Description - `Enter the new description`

    Encryption - `Select` Encrypt target EBS snapshots
```

6. Click on Copy AMI

Once the AMI has finished copying.

7. Select the AMI that does not have encrypted EBS snapshots.
8. Click on `Actions`.
9. Click on `Deregister`

From the Command Line:

1. Run the aws ec2 copy-image command to copy AMI with encrypted block device

```
aws ec2 copy-image --name <New_AMI_Name> --source-image-id <Image-ID> --
source-region <region> --encrypted
```

2. Run aws ec2 deregister-image to deregister older AMIs

```
aws ec2 deregister-image --image-id <Image-ID>
```

References:

1. https://aws.amazon.com/premiumsupport/knowledge-center/view-ami-snapshot-encryption-details/
2. https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/AMIEncryption.html
3. https://docs.aws.amazon.com/cli/latest/reference/ec2/copy-image.html
4. https://docs.aws.amazon.com/cli/latest/reference/ec2/deregister-image.html
5. https://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_DeregisterImage.html
6. https://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_CopyImage.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

2.1.3 Ensure Only Approved Amazon Machine Images (AMIs) are Used (Manual)

Profile Applicability:

- Level 1

Description:

Ensure that all base AMIs utilized are approved for use by your organization.

Rationale:

An approved AMI is a base EC2 machine image that is a pre-configured OS configured to run your application. Using approved AMIs helps enforce consistency and security.

Audit:

Perform the following to confirm only approved AMIs are being used.
From the Console:

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane click on `Images`.
3. Then choose `AMIs`
4. Confirm that `Owned by me` is selected
5. Review the list of AMIs.
6. Confirm that the AMIs listed are all approved for use
7. In the left pane click on `Instances`
8. Then choose `Instances`
9. Select the EC2 instance for review.
10. In the Details tab review:

```
AMI Name

AMI location
```

11. Confirm that the AMI name matches an approved AMI and the AMI location is within your account.
12. Repeat steps 9 – 11 to verify the AMI is approved

Repeat the process for all other regions.
If any of the AMIs are not approved refer to the remediation below.

Remediation:

Perform the following to remove unauthorized AMIs.

From the Console:

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane click on `Images`.
3. Then choose `AMIs`
4. Confirm that `Owned by me` is selected
5. Review the list of AMIs.
6. Confirm that the AMIs listed are all approved for use
7. If an AMI is listed that is not approved select it.
8. Click on `Actions` and choose `Deregister`

After all unauthorized AMIs have been De-registered review all EC2 instances.

1. Click on `Instances`
2. Then choose `Instances`
3. Select the `EC2 instance` for review.
4. In the `Details` tab review:

```
    AMI Name

    AMI location]
```

5. If this information is listed as not available this instance was built with an unauthorized AMI.
6. Follow organization steps to secure this instance and replace it with an instance built from an approved AMI if applicable.
7. Repeat steps 3 – 6 to verify all instance have been created with approved AMIs

Repeat the process for all other regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

2.1.4 Ensure Images (AMI) are not older than 90 days (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that your AMIs are not older than 90 days.

Rationale:

Using up-to-date AMIs will provide many benefits from OS updates and security patches helping to ensure reliability, security and compliance.

Audit:

Perform the following to determine the age of an AMI.
From the Console

1. Login to the EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left pane, under `Images`, click `AMIs`.
3. Select the AMI for review.
4. Under the `Details` tab
5. Review the `Creation date`.

If the age of the selected AMI is greater than 90 days, the AMI is considered outdated and it should be updated.

6. Repeat steps no. 3 – 5 to verify the date of the other approved AMIs available.

Repeat all steps for the other regions.
Refer to the remediation procedure below to update the AMI.
From the Command Line:
Run the aws ec2 describe-images command

```
aws ec2 describe-images \
    --region <region> \
    --image-ids <image-ID>
```

Look for CreationDate in response.
If the age of the selected AMI is greater than 90 days, the AMI is considered outdated and it should be updated.

Remediation:

Perform these steps if the Creation date is older than 90 days.

From the Console

1. Login to the EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left pane, under `Images`, click `AMIs`.
3. Select the AMI to be updated.
4. Click on Launch
5. Go through the EC2 Instance creation process.
6. Apply all system, security and application updates that are applicable to the EC2 instance.
7. Once completed click on `Instance state`, `Stop instance1.
8. Click on `Actions`, `Image and templates`, `Create image`
9. Once the image process has complete return to the AMI list but clicking on `Images`, `AMIs`
10. Select the AMI that is older than 90 days.
11. Click on `Actions`, `Deregister`

Repeat these steps for any other AMIs older than 90 days.

References:

1. https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-images.html
2. https://docs.aws.amazon.com/cli/latest/reference/ec2/deregister-image.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	2.2 Ensure Authorized Software is Currently Supported Ensure that only currently supported software is designated as authorized in the software inventory for enterprise assets. If software is unsupported, yet necessary for the fulfillment of the enterprise's mission, document an exception detailing mitigating controls and residual risk acceptance. For any unsupported software without an exception documentation, designate as unauthorized. Review the software list to verify software support at least monthly, or more frequently.	●	●	●
v7	2.2 Ensure Software is Supported by Vendor Ensure that only software applications or operating systems currently supported by the software's vendor are added to the organization's authorized software inventory. Unsupported software should be tagged as unsupported in the inventory system.	●	●	●

2.1.5 Ensure Images are not Publicly Available (Manual)

Profile Applicability:

- Level 1

Description:

EC2 allows you to make an AMI public, sharing it with all AWS accounts.

Rationale:

Publicly sharing an AMI with all AWS accounts could expose organizational data and configuration information.

Audit:

Perform the steps below to determine if any AMIs are shared with all AWS accounts.
From the Console

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane, under `Images`, click `AMIs`.
3. Confirm the `Owned by me` is set.
4. Select the AMI from the list.
5. Click on the `Permissions` Tab
6. If this reads `This image is currently Public`.

Please refer to the remediation below.

Remediation:

Perform the steps below to set an AMIs to Private.

From the Console

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane, under `Images`, click `AMIs`.
3. Confirm the `Owned by me` is set.
4. Select the AMI from the list.
5. Click on the `Permissions` Tab
6. Click on `Edit`
7. Click on the radio button `Private`

Add AWS Account Number if you have a need to share with other Internal AWS accounts that your Organization owns.

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sharing-amis.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.3 Protect Recovery Data Protect recovery data with equivalent controls to the original data. Reference encryption or data separation, based on requirements.	●	●	●
v7	5.3 Securely Store Master Images Store the master images and templates on securely configured servers, validated with integrity monitoring tools, to ensure that only authorized changes to the images are possible.		●	●

2.2 Elastic Block Storage (EBS)

This section contains guidance for Amazon Elastic Block Store (EBS) which is a high performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2).

2.2.1 Ensure EBS volume encryption is enabled (Automated)

Profile Applicability:

- Level 1

Description:

Elastic Compute Cloud (EC2) supports encryption at rest when using the Elastic Block Store (EBS) service. While disabled by default, forcing encryption at EBS volume creation is supported.

Rationale:

Encrypting data at rest reduces the likelihood that it is unintentionally exposed and can nullify the impact of disclosure if the encryption remains unbroken.

Audit:

From Console:

1. Login to the EC2 console using https://console.aws.amazon.com/ec2/
2. Under `Account attributes`, click `EBS encryption`.
3. Verify `Always encrypt new EBS volumes` displays `Enabled`.
4. Review every region in-use.

Note: EBS volume encryption is configured per region.

From Command Line:

1. Run

```
aws --region <region> ec2 get-ebs-encryption-by-default
```

2. Verify that `"EbsEncryptionByDefault": true` is displayed.
3. Review every region in-use.

Note: EBS volume encryption is configured per region.

From Console:

1. Login to the EC2 console using https://console.aws.amazon.com/ec2/
2. Under `Account attributes`, click `EBS encryption`.
3. Click `Manage`.
4. Click the `Enable` checkbox.
5. Click `Update EBS encryption`
6. Repeat for every region requiring the change.

Note: EBS volume encryption is configured per region.

From Command Line:

1. Run

```
aws --region <region> ec2 enable-ebs-encryption-by-default
```

2. Verify that `"EbsEncryptionByDefault": true` is displayed.
3. Repeat every region requiring the change.

Note: EBS volume encryption is configured per region.

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html
2. https://aws.amazon.com/blogs/aws/new-opt-in-to-default-encryption-for-new-ebs-volumes/
3. AWS Config rule - ec2_ebs_encryption_by_default

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

2.2.2 Ensure Public Access to EBS Snapshots is Disabled (Automated)

Profile Applicability:

- Level 1

Description:

To protect your data disable the public mode of EBS snapshots.

Rationale:

This protects your data so that it is not accessible to all AWS accounts preventing accidental access and leaks.

Audit:

Perform the following to determine if a snapshot is shared publicly:
From the Console

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane click `Snapshots`.
3. Select the `snapshot` then click `Actions`, `Modify Permissions`.
4. Confirm that the snapshot is set to `Private`
5. Repeat for any additional Snapshots, Regions and AWS accounts.

If the snapshot is set to public refer to the remediation below.
From the CLI

1. For each snapshot, run

```
aws ec2 describe-snapshot-attribute \
    --snapshot-id <snapshot-ID> \
    --attribute createVolumePermission
```

2. Validate `Group` is not set to all.

Remediation:

Perform the following to set a snapshot to private:

From the Console

1. Login to the EC2 console at `https://console.aws.amazon.com/ec2/`.
2. In the left pane click `Snapshots`.
3. Select the `snapshot` then click 'Actions, Modify Permissions`.

4. Click the radio button for `Private`
5. Click `Save`
6. Repeat for any additional Snapshots, Regions and AWS accounts.

From the CLI

1. For each snapshot, run

```
aws ec2 modify-snapshot-attribute \
   --snapshot-id <snapshot-ID> \
   --attribute createVolumePermission \
   --operation remove --group-name all
```

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-snapshot-attribute.html

Additional Information:

1. Snapshots are constrained to the Region in which they were created. To share a snapshot with another Region, copy the snapshot to that Region.
2. AWS prevents you from sharing snapshots that were encrypted with your default CMK. Snapshots that you intend to share must instead be encrypted with a customer managed CMK.
3. The public option is not valid for encrypted snapshots or snapshots with an AWS Marketplace product code.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

2.2.3 Ensure EBS volume snapshots are encrypted (Automated)

Profile Applicability:

- Level 1

Description:

Elastic Compute Cloud (EC2) supports encryption at rest when using the Elastic Block Store (EBS) service.

Rationale:

Encrypting data at rest reduces the likelihood that it is unintentionally exposed and can nullify the impact of disclosure if the encryption remains unbroken.

Audit:

From Console:

1. Login to the EC2 console using `https://console.aws.amazon.com/ec2/`
2. Under `Elastic Block Store`, click `Snapshots`.
3. Click the snapshot you want to review.
4. Select the `Description` tab.
5. Review the `Encryption` setting.
6. If it reads `encrypted` you are all set.

If it is set to `Not Encrypted` refer to the remediation below.
Note: EBS snapshot volume encryption is configured per snapshot.

From Command Line:

1. Run describe-snapshots

```
aws ec2 describe-snapshots --owner-ids <account number> --filter
Name=status,Values=completed --query "Snapshots[*].{ID:SnapshotId}"
```

2. This will provide a list of all the snapshots associated with that account in the region.
3. For every snapshot listed - Run - describe-snapshots

```
aws ec2 describe-snapshots --snapshot-id <snap-name> --query
"Snapshots[*].{Encrypt:Encrypted}"
```

4. If the output reads `"Encrypt": true`, Encryption is set on the snapshot.

If the output reads `"Encrypt": false` refer to the remediation below.
Note: EBS snapshot volume encryption is configured per snapshot.

Remediation:

From Console:

1. Login to the EC2 console using `https://console.aws.amazon.com/ec2/`
2. Under `Elastic Block Store, click` Snapshots`.
3. Select the snapshot you want to encrypt.
4. Click on `Actions` select `Copy`.

```
Confirm `Snapshot ID`
Set the `Destination Region`
Update the `Description`
Select the check box for `Encryption`
```

5. Check the box for `Encrypt this snapshot`
6. Set the `Master Key`
7. Click on `Copy`
8. Repeat steps 3-7 for the snapshots that need to be encrypted.
9. Delete any of the unencrypted snapshots that are not longer needed.

Note: EBS snapshot volume encryption is configured per snapshot.
From Command Line:
Using the snapshot ids gathered from the Audit section

1. Run - copy-snapshot

```
aws ec2 copy-snapshot --source-region  <region> --source-snapshot-id <snap-
id> --description "Name of the new snapshot" --encrypted
```

2. This will copy the existing unencrypted snapshot and set it to encrypted
   The output will show the new SnapshotId
3. Run - describe-snapshots

```
aws ec2 describe-snapshots --owner-ids <account id> --filter
Name=status,Values=completed --query "Snapshots[*].{ID:SnapshotId}"
```

Once the new Snapshot shows in the list confirm encryption is set

4. Run - describe-snapshots

```
aws ec2 describe-snapshots --snapshot-id <snap-name> --query
"Snapshots[*].{Encrypt:Encrypted}"
```

5.Repeat steps 1-4 for the snapshots that need to be encrypted.
Delete snapshots that are no longer needed.

6. Run - delete-snapshot

```
aws ec2 delete-snapshot --snapshot-id <snap-name>
```

7. Repeat for all unencrypted snapshots that have been copied and encrypted.

Note: EBS snapshot volume encryption is configured per snapshot.

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-snapshots.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/delete-snapshot.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/copy-snapshot.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

2.2.4 Ensure unused EBS volumes are removed (Manual)

Profile Applicability:

- Level 1

Description:

Identify any unused Elastic Block Store (EBS) volumes in your AWS account and remove them.

Rationale:

Any Elastic Block Store volume created in your AWS account contains data, regardless of being used or not. If you have EBS volumes (other than root volumes) that are unattached to an EC2 instance they should be removed to prevent unauthorized access or data leak to any sensitive data on these volumes.

Impact:

Once a EBS volume is deleted, the data will be lost. If this is data that you need to archive, create an encrypted EBS snapshot before deleting them.

Audit:

From Console:

1. Login to the EC2 console using `https://console.aws.amazon.com/ec2/`
2. Under `Elastic Block Store`, click `Volumes`.
3. Find the `State` column
4. Sort by `Available`
5. Any `Volumes` listed as Available can be deleted as that is the indication the volume is not attached to an instance.
   Capture this list of volume names and refer to the remediation below.

Note: EBS volumes can be in different regions. Make sure to review all the regions being utilized.
From Command Line:

1. Run describe-volumes

```
aws ec2 describe-volumes --filter Name=status,Values=available --query
"Volumes[*].{ID:VolumeId}"
```

2. This will provide a list of all the volumes not attached to an instance

Capture this list of volume names and refer to the remediation below.
Note: EBS volumes can be in different regions. Make sure to review all the regions being utilized.

Remediation:

From Console:

1. Login to the EC2 console using `https://console.aws.amazon.com/ec2/`
2. Under `Elastic Block Store`, click `Volumes`.
3. Find the `State` column
4. Sort by `Available`
5. Select the Volume that you want to delete.
6. Click `Actions`, `Delete volume`, `Yes, Delete`

Note: EBS volumes can be in different regions. Make sure to review all the regions being utilized.
From Command Line:
Using the list of `available volumes` identified in the Audit above

1. Run the delete-volume command

```
aws ec2 delete-volume --volume-id <vol-name>
```

2. This will delete the volume identified.

Note: Using this command will not prompt you for confirmation. It will delete the volume and you will not be able to recover it.
Please make sure you have the correct volume and that you have created a snapshot if it is something that needs to be archived.
Note: EBS volumes can be in different regions. Make sure to review all the regions being utilized.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-volumes.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/delete-volume.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

2.3 Ensure Tag Policies are Enabled (Manual)

Profile Applicability:

- Level 1

Description:

Tag policies help you standardize tags on all tagged resources across your organization.

Rationale:

You can use tag policies to define tag keys (including how they should be capitalized) and their allowed values.

Audit:

From the Console

1. Login to AWS Organizations using `https://console.aws.amazon.com/organizations/`
2. In the left pane click on `Policies`
3. Confirm `Tag policies` status is `enabled`.

If `Tag policies` status is disabled refer to the remediation below.

From the CLI

1. Run the `list-policies` command

```
aws organizations list-policies --filter TAG_POLICY
```

2. If information displays it means you have a tagging policy in place.
3. If empty brackets display [] refer to the remediation below.

Remediation:

From the Console:

You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.

1. Login to AWS Organizations using `https://console.aws.amazon.com/organizations/`
2. In the Left pane click on `Policies`
3. Click on `Tag policies`
4. Click on `Enable Tag Policies`
5. The page is update with a list of the Available policies and the ability to create one.

From the Command Line:
You must use an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.

1. Run the `enable-policy-type` command

```
aws organizations enable-policy-type --root-id <RootID>  --policy-type
TAG_POLICIES
```

The list of PolicyTypes in the output will now include the specified policy type with the Status of ENABLED.

References:

1. https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_enable-disable.html#enable-policy-type

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.5 Maintain Asset Inventory Information Ensure that the hardware asset inventory records the network address, hardware address, machine name, data asset owner, and department for each asset and whether the hardware asset has been approved to connect to the network.		●	●

2.4 Ensure an Organizational EC2 Tag Policy has been Created (Manual)

Profile Applicability:

- Level 1

Description:

A tag policy enables you to define tag compliance rules to help you maintain consistency in the tags attached to your organization's resources.

Rationale:

You can use an EC2 tag policy to enforce your tag strategy across all of your EC2 resources.

Audit:

From the Console:

1. Login to the AWS Organizations using `https://console.aws.amazon.com/organizations/`
2. On the left click `Policies`
3. Click on `Tag policies`
4. Confirm that a policy name exists with a description
5. Click on the policy for EC2 Tagging as indicated in the name, description or both.
6. Click on `Edit policy`
7. Confirm that `Tag key capitalization compliance` is checked
8. Confirm that `Prevent non-compliant operations for this tag` is checked.
9. Confirm that `ec2:image`, `ec2:instance` and `ec2:reserved-instances` are listed.

If the tag policy does not exist with the settings listed above refer to the remediation below.

Remediation:

From the Console:
You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization’s management account.
To create a tag policy

1. Login to the AWS Organizations using `https://console.aws.amazon.com/organizations/`
2. Left hand side Click on `Policies`
3. Under `Support policy types` click on `Tag policies`
4. Under `Available policies` click on `Create policy`

5. Enter policy name
6. Enter policy description (Indicate this is the EC2 tag policy)
7. For New tag key 1, specify the name of a tag key to add.
8. For `Tag key capitalization compliance` select the box for Use the capitalization to enable this option mandating a specific capitalization for the tag key
   using this policy.
9. For `Resource types to enforce` check the box for `Prevent non-compliant operations for this tag`
10. Click on `Specify resource types`
11. Expand EC2
12. Select ec2:image, ec2:instance, ec2:reserved-instances
13. Click `Save changes`
14. Click `Create policy`

References:

1. https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_tag-policies-create.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.5 Maintain Asset Inventory Information Ensure that the hardware asset inventory records the network address, hardware address, machine name, data asset owner, and department for each asset and whether the hardware asset has been approved to connect to the network.		●	●

2.5 Ensure no AWS EC2 Instances are Older than 180 days (Manual)

Profile Applicability:

- Level 1

Description:

Identify any running AWS EC2 instances older than 180 days.

Rationale:

An EC2 instance is not supposed to run indefinitely and having instance older than 180 days can increase the risk of problems and issues.

Audit:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `INSTANCES`, click `Instances`.
3. Select the `EC2 instance`. The Instance State must be 'running'.
4. Select the `Description` tab.
5. Check the `Launch time`.
6. Determine the `instance active age`.
7. If the selected EC2 instance active age is greater than 180 days, refer to the remediation below.
8. Repeat steps no. 3 – 7 to verify the launch date for all instances.
9. Go through the other `AWS regions` and repeat the audit process.

From the CLI

1. Run the describe-instances command

```
aws ec2 describe-instances --region us-east-1 --output json --filters
"Name=instance-state-code,Values=16" --query
"Reservations[*].Instances[*].{Instance:InstanceId}"
```

2 The output should look like this:

```
[
    [
        {
            "Instance": "i-1234567abcdefghi0"
        }
    ],
    [
        {
            "Instance": "i-1234567abcdefghi0"
        }
    ],
    [
        {
            "Instance": "i-1234567abcdefghi0"
        }
    ],
    [
        {
            "Instance": "i-1234567abcdefghi0"
        }
    ]
]
```

3 Run the describe-instances command for each instance ID listed:

```
aws ec2 describe-instances --region us-east-1 --instance-ids i-
1234567abcdefghi0 --query "Reservations[*].Instances[*].LaunchTime"
```

4. The command output should return the instance launch date in human readable format:

```
 "2021-06-11T15:04:52+00:00"
``
5. If the selected instance was launched more than 180 days ago, refer to the
remediation below.

6. Repeat steps 3 and 4 to verify the launch date for all instances listed.

7. Repeat steps 1 – 6 for the other AWS regions.
```

Remediation:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `INSTANCES`, click `Instances`.
3. Select the `EC2 instance` identified above in the audit. The Instance State must be 'running'.
4. Click `Actions`, click `Instance State`, click `Stop`.
5. Wait for the Instance State to read 'stopped'.
6. Click 'Actions' click 'Instance State', click 'Start'

7. Select the Description tab.
8. Check the Launch time.

Confirm that the instance active age is now set to today's date and time.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

2.6 Ensure detailed monitoring is enable for production EC2 Instances (Manual)

Profile Applicability:

- Level 2

Description:

Ensure that detailed monitoring is enabled for your Amazon EC2 instances.

Rationale:

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon EC2 instances

Impact:

Data is available in 1-minute periods. For the instances where you've enabled detailed monitoring, you can also get aggregated data across groups of similar instances. You are charged per metric that is sent to CloudWatch. You are not charged for data storage. Due to this added cost it is recommended that you only enable this on critical instances.

Audit:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `INSTANCES`, click `Instances`.
3. Select the `EC2 instance` you want to review.
4. Select the `Description` tab.
5. Check the `Launch time`.
6. Determine the level of monitoring by reviewing the 'Monitoring attribute'.
7. If the value is set to `basic` refer to the remediation below.
8. Repeat steps no. 3 – 7 to verify the monitoring level for all instances.
9. Go through the other `AWS regions` and repeat the audit process.

From the CLI

1. Run the describe-instances command

```
aws ec2 describe-instances --region us-east-1 --output json --filters
"Name=monitoring-state,Values=disabled" --query
"Reservations[*].Instances[*].{Instance:InstanceId}"
```

2. The output should be a list of running instances that have enhanced monitoring disabled.
3. Based on this list of instance ids refer to the remediation below.

Remediation:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `INSTANCES`, click `Instances`.
3. Select the `EC2 instance` you want to review.
4. Select the `Monitoring` tab.
5. Click on 'Enable Detailed Monitoring`
6. Click on `Yes, Enable`
7. Repeat steps no. 3 – 6 for any other instances that require detailed monitoring to be enabled.

From the CLI

1. Run the monitor-instances command using the list of instances collected in the audit.

```
aws ec2 monitor-instances --instance-ids <i-instancename>
```

2. The output will show 'state: pending'
3. Wait a few minutes and run the same command again for that instance and it will show enabled.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

2.7 Ensure Default EC2 Security groups are not being used. (Manual)

Profile Applicability:

- Level 1

Description:

When an EC2 instance is launched a specified custom security group should be assigned to the instance.

Rationale:

When an EC2 Instance is launched the default security group is automatically assigned. In error a lot of instances are launched in this way, and if the default security group is configured to allow unrestricted access, it will increase the attack footprint allowing the opportunity for malicious activity.

Audit:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `INSTANCES`, click `Instances`.
3. On the EC2 Instances page, click inside the attributes filter box
4. Click the Security Group Name from the dropdown list
5. Type `default` for the attribute value. (This filter will detect the EC2 instances currently associated with the default security group)
6. Refer to the remediation below using list of Ec2 Instance ids captured.

NOTE Repeat the audit process for all other regions used.
From the CLI

1. Run the describe-instances command

```
aws ec2 describe-instances --region us-east-1 --output json --filters
"Name=instance.group-name,Values=default" --query
"Reservations[*].Instances[*].{Instance:InstanceId}"
```

2. The command output should return an empty list if the default security group is not being used.
3. If there is a list of instance IDs then the default security group is currently attached to those EC2 instances.
4. Refer to the remediation below using list of EC2 Instance ids captured.

NOTE Repeat the audit process for all other regions used.

Remediation:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `Network & Security`, click `Security Groups`.
3. Select `Security Groups`
4. Click on the `default Security Group` you want to review.
5. Click `Actions`, `View details`.
6. Select the `Inbound rules` tab
7. Click on `Edit inbound rules`
8. Click on `Delete` for all the rules listed
9. Once there are no rules listed click on 'Save rules`
10. Repeat steps no. 3 – 8 for any other default security groups listed.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-security-groups.html
2. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/default-custom-security-groups.html#default-security-group

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

2.8 Ensure the Use of IMDSv2 is Enforced on All Existing Instances (Manual)

Profile Applicability:

- Level 2

Description:

Ensure the Instance Metadata Service Version 2 (IMDSv2) method is enabled on all running instances.

Rationale:

The IMDSv2 method uses session-based controls to help protect access and control of Amazon Elastic Compute Cloud (Amazon EC2) instance metadata. With IMDSv2, controls can be implemented to restrict changes to instance metadata.

Impact:

Once you enforce IMDSv2, then IMDSv1 no longer works, and applications that use IMDSv1 might not function correctly. Before enforcing IMDSv2, verify that any applications that use Amazon EC2 metadata are upgraded to a version that supports IMDSv2.

Audit:

From the Console:

1. At this time the instance metadata setting for existing instances can only be reviewed and confirmed using AWS CLI.

From the CLI

1. Run the describe-instances command

```
aws ec2 describe-instances --region us-east-1 --output text --filter
"Name=metadata-options.http-tokens,Values=optional" --query
"Reservations[*].Instances[*].{Instance:InstanceId}"
```

2 The output should look like this:

```
i-1234567abcdefghi0
i-1234567abcdefghi0
i-1234567abcdefghi0
```

The list above contains all the instances that have the metadata version set to optional which means either IMDSv1 or INDSv2 an be used. Refer to the remediation below. Repeat steps 1 – 2 for the other AWS regions.

Remediation:

From the Console:

1. At this time the instance metadata setting for existing instances can only be changed using AWS CLI.

From the CLI

1. Run the modify-instance-metadata-options command using the list of Instances collect in the audit

```
aws ec2 modify-instance-metadata-options --instance-id i-1234567abcdefghi0 --
http-tokens required --http-endpoint enabled
```

2. The output should show the information for the instance and the metadata changes:

```
{
    "InstanceId": "i-1234567abcdefghi0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled"
    }
}
```

3. Repeat for the other instances and regions collected during the audit.

References:

1. https://aws.amazon.com/premiumsupport/knowledge-center/ssm-ec2-enforce-imdsv2/
2. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/configuring-instance-metadata-service.html
3. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/configuring-instance-metadata-options.html
4. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html#configure_instance_details_step
5. https://docs.aws.amazon.com/config/latest/developerguide/ec2-imdsv2-check.html
6. https://docs.aws.amazon.com/systems-manager-automation-runbooks/latest/userguide/automation-aws-enforce-ec2-imdsv2.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

2.9 Ensure use of AWS Systems Manager to manage EC2 instances (Manual)

Profile Applicability:

- Level 2

Description:

An inventory and management of Amazon Elastic Compute Cloud (Amazon EC2) instances is made possible with AWS Systems Manager.

Rationale:

Use AWS Systems Manager to provide detailed system configurations, operating system patch levels, services name and type, software installations, application name, publisher and version, and other details about your environment.

Audit:

From the Console

1. Login to EC2 using `https://console.aws.amazon.com/systems-manager/`
2. On the left Click `Node Management`, click `Inventory`.
3. On the Dashboard confirm that all of your Instances are listed as part of your inventory.

If any instances are missing or AWS Systems Manager is not configured, refer to the remediation below.

Remediation:

From the Console
These directions already assume your AWS account is setup.
They will walk you through how to create non-Admin IAM users and groups for System Manager.
**Note - There is additional guidance provided by AWS on the process.

1. Create a `user group`
   a. Login to IAM using `https://console.aws.amazon.com/iam/`
   b. On the left Click `Access management`, click `User groups`, and then click `Create Group`.
   c. On the `Create user group` page, enter a name for the group
   d. Select and add the users required to the Group
   e. Attach permissions policies by selecting `ResourceGroupsandTagEditorFullAccess` policy.
   f. Then for Full access to Systems Manager console, click the `AmazonSSMFullAccess` policy.

- OR
  g. For access to view Systems Manager data, and not create or update resources, click the AmazonSSMReadOnlyAccess policy.
  h. For access to the Built-In Insights and Dashboard by CloudWatch pages in the Systems Manager console, add these policies:
- AWSHealthFullAccess
- AWSConfigUserAccess
- CloudWatchReadOnlyAccess
  i. Click Create group.

If you need additional users follow the next step. If not skip to Step 3.

2. Create users and assign permissions.
   a. Login to IAM using https://console.aws.amazon.com/iam/
   b. On the left Click Access management, click Users, and then click Add users.
   c. User name, enter the name that the user will use to sign in to AWS Systems Manager.
   d. To allow the user access to development tools, select the check box next to Access key - Programmatic access. This creates an access key for the new user. You can view or download the access keys when you get to the Final page.
   e. To allow user access to the AWS Management Console, select the check box next to AWS Management Console access. If you click Custom password, enter an initial password for the user. You can optionally select Require password reset to force the user to create a new password the next time the user signs in.
   f. Click Next: Permissions.
   g. To Set permissions, click Add user to group.
   h. In the group list, click the group you created in step 1
   i. Then click Next: Tags.
   j. (Optional) Add one or more tags
   k. Click Next: Review to see the list of group memberships that the new user is joining.
   l. Click Create user.
3. To add permissions for an existing user
   a. In the IAM console, click Users.
   b. click the name of the user to add to a group
   c. Then click Add permission.
   d. For Add user to group, select the box next to the group to add the user to
   e. Add any other available permission policies to assign to the user.
   f. Click Next: Review to see the list of group memberships that will be added to the new user.
   g. Click Add permissions.
4. Create an IAM instance profile for Systems Manager
   a. Login to the IAM console at https://console.aws.amazon.com/iam/
   b. In the left, click Roles, and then click Create role.
   c. Under Select type of trusted entity, click AWS service.
   d. Click EC2, and then click Next: Permissions.

e. On the Attach permissions policies page, do the following:
Use the Search field to locate the AmazonSSMManagedInstanceCore. Select the box next to it.
The console retains your selection even if you search for other policies.

- **Note - If you plan to join instances to an Active Directory managed by AWS Directory Service, search for AmazonSSMDirectoryServiceAccess and select the box next to its name
- **Note - If you plan to use EventBridge or CloudWatch Logs to manage or monitor your instance, search for CloudWatchAgentServerPolicy and select the box next to its name.
f. Click Next: Tags.
g. Add one or more tag-key value pairs to organize, track, or control access for this role,
h. Click Next: Review.
i. For Role name, enter a name for your new instance profile
- **Note - Make a note of the role name. You will click this role when you create new instances that you want to manage by using Systems Manager.
j. For Role description, enter a description for this instance profile.
k. Click Create role.

The system returns you to the Roles page.

5. Attach the Systems Manager instance profile to an existing instance
a. Login to the EC2 console at https://console.aws.amazon.com/ec2/
b. In he left pane, under Instances, click Instances.
c. Select the instance from the list.
d. In the Actions menu, click Security, Modify IAM role.
e. Select the instance profile you created using the procedure in Step 4.
f. Click Save.
6. Attach an IAM instance profile to an Amazon EC2 instance
a. Login to the EC2 console at https://console.aws.amazon.com/ec2/
b. Select or confirm the AWS Region for the instance.
c. Click Launch Instance.
d. Locate the AMI for the instance type you want to create, and then click Select.
e. Select the type of instance to launch, then click Next: Configure Instance Details.
f. On the Configure Instance Details page, in the IAM role dropdown list, select the instance profile you created in Step 4
g. For other options on the page, make selections that meet your requirements for the instance.
h. Complete the wizard.

If you create other instances that you want to configure using Systems Manager, specify the instance profile for each instance

References:

1. https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-setting-up.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

2.10 Ensure unused ENIs are removed (Manual)

Profile Applicability:

- Level 1

Description:

Identify and delete any unused Amazon AWS Elastic Network Interfaces in order to adhere to best practices and to avoid reaching the service limit. An AWS Elastic Network Interface (ENI) is pronounced unused when is not attached anymore to an EC2 instance.

Rationale:

Audit:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `NETWORK & SECURITY`, click `Network Interfaces`.
3. Select the ENI that you want to review
4. Go to the Details tab
5. Check the value set for the Status attribute
6. If it says `available`, refer to the remediation below.
7. Repeat steps 3 - 6 to determine the current status for any other `ENIs` within the current region.

NOTE Repeat the audit process for all other regions used.
From the CLI

1. Run describe-network-interfaces command

```
aws ec2 describe-network-interfaces --region us-east-1 --output json --
filters Name=status,Values=available --query
"NetworkInterfaces[*].{ENI:NetworkInterfaceId}"
```

2. The command output should return an empty list if the default security group is not being used.
3. If there is a list of ENI IDs then refer to the remediation below.
4. Repeat steps 1 - 3 to determine the current status for any other `ENIs` within the current region.

NOTE Repeat the audit process for all other regions used.

Remediation:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `NETWORK & SECURITY`, click `Network Interfaces`.
3. Select the ENI that you want to remove
4. Click 'Actions', then 'delete'
5. Click `Delete`
6. Repeat steps 3 - 5 any other `ENIs` listed in the audit within the current region.

NOTE Repeat the audit process for all other regions used.
From the CLI

1. Run the delete-network-interface command with the ENI names collected above in the audit.

```
aws ec2 delete-network-interface --region us-east-1 --network-interface-id
eni-1234abcd
```

2. This will remove the ENI that is not being used.
3. Repeat steps 1 - 2 for any `ENIs` within the current region.

NOTE Repeat the audit process for all other regions used.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	12.2 Establish and Maintain a Secure Network Architecture Establish and maintain a secure network architecture. A secure network architecture must address segmentation, least privilege, and availability, at a minimum.		●	●
v7	11.1 Maintain Standard Security Configurations for Network Devices Maintain standard, documented security configuration standards for all authorized network devices.		●	●

2.11 Ensure instances stopped for over 90 days are removed (Manual)

Profile Applicability:

- Level 1

Description:

Enable this rule to help with the baseline configuration of Amazon Elastic Compute Cloud (Amazon EC2) instances by checking whether Amazon EC2 instances have been stopped for more than the allowed number of days, according to your organization's standards.

Rationale:

Audit:

From the Console

1. Login to the EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left pane, click `Instances`, click `Instances`.
3. Select the Instance for review.
4. Under the `Details` tab
5. Review the `Launch time`.

If the `Launch time` of the selected Instance is greater than 90 days, the Instance has been offline and is considered outdated.

6. Repeat steps no. 3 – 5 to verify the Launch date for the other instances.

Repeat all steps for the other regions.
Refer to the remediation procedure below if any of the `Launch times` are over 90 days.

Remediation:

From the Console

1. Login to the EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left pane, click `Instances`, click `Instances`.
3. Select the Instance for that hasn't been used for over 90 days.
4. Under the `Details` tab
5. Click `Instance state`, click `Terminate instance`.
6. Click `Terminate`.

7.Repeat steps no. 3 – 6 the other instances with a launch date equal to or over 90 days.
Repeat all steps for the other regions.

References:

1. https://docs.aws.amazon.com/config/latest/developerguide/ec2-stopped-instance.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

2.12 Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination (Manual)

Profile Applicability:

- Level 1

Description:

This rule ensures that Amazon Elastic Block Store volumes that are attached to Amazon Elastic Compute Cloud (Amazon EC2) instances are marked for deletion when an instance is terminated. If an Amazon EBS volume isn't deleted when the instance that it's attached to is terminated, it may violate the concept of least functionality.

Rationale:

Audit:

From the Console:

1. Login to EC2 using `https://console.aws.amazon.com/ec2/`
2. On the left Click `INSTANCES`, click `Instances`.
3. Select the `EC2 instance` you want to review.
4. Select the `Storage` tab.
5. Scroll down until you reach the 'Volume ID' and review the setting for 'Delete on termination'
6. If the value is set to `No` refer to the remediation below.
7. Repeat steps no. 3 – 6 to verify the setting.
8. Go through the other `AWS regions` and repeat the audit process for all instances.

From the CLI

1. Run the describe-instances command

```
aws ec2 describe-instances --region us-east-1 --output json --filters
"Name=block-device-mapping.delete-on-termination,Values=false" --query
"Reservations[*].Instances[*].{Instance:InstanceId}"
```

2. The output should be a list of instances that have not set 'Delete on termination'.
3. Make note of the list of instance ids and refer to the remediation below.
4. Repeat steps no. 1 -3 with the other `AWS regions`.

Remediation:

From the Console:

1. At this time the `delete on termination` setting for existing instances can only be changed using AWS CLI.

From the CLI

1. Run the modify-instance-attribute command using the list of instances collected in the audit.

```
aws ec2 modify-instance-attribute --instance-id i-123456abcdefghi0 --block-
device-mappings "[{\"DeviceName\":
\"/dev/sda\",\"Ebs\":{\"DeleteOnTermination\":true}}]"
```

2. Repeat steps no. 1 with the other instances discovered in all `AWS regions`.

**Note - If you get any errors running the modify-instance-attribute command confirm the instance id and the Device Name for that instance is correct. The above command is referencing the typical default device name.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/modify-instance-attribute.html
2. https://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_DescribeInstances.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

2.13 Ensure Secrets and Sensitive Data are not stored directly in EC2 User Data (Manual)

Profile Applicability:

- Level 1

Description:

User Data can be specified when launching an ec2 instance. Examples include specifying parameters for configuring the instance or including a simple script.

Rationale:

The user data is not protected by authentication or cryptographic methods. Therefore, sensitive data, such as passwords or long-lived encryption keys should not be stored as user data.

Impact:

Anyone who has access to the instance and configuration can view the user data.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services` and click `EC2` under Compute.
3. Click on `Instances`.
4. For each instance, click `Actions` -> `Instance Settings` -> `Edit user data`
5. For each instance, review the user data to ensure there are no secrets or sensitive data stored.
6. If secrets or sensitive data is found, refer to the remediation below.
7. Repeat steps 2-7 for all regions used.

From the Command line

1. Run `aws ec2 describe-instances` to retrieve information about all instances in the AWS region. The output will include instance ids.
2. Run `aws ec2 describe-instance-attribute` for each instance in AWS account.

```
aws ec2 describe-instance-attribute
--instance-id "ID of instance" --attribute userData
```

Note: User Data may be Base64 encoded. Decode the output as necessary.

3. Review user data to ensure no secrets or sensitive information are stored.
4. Repeat the Audit for all the other AWS regions.

Remediation:

From the Console

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services` and click `EC2` under Compute.
3. Click on `Instances`.
4. If the instance is currently running, stop the instance first.

Note: ensure there is no negative impact from stopping the instance prior to stopping the instance.

5. For each instance, click `Actions` -> `Instance Settings` -> `Edit user data`
6. For each instance, edit the user data to ensure there are no secrets or sensitive data stored. A Secret Management solution such as AWS Secrets Manager can be used here as a more secure mechanism of storing necessary sensitive data.
7. Repeat this remediation for all the other AWS regions.

Note: If the ec2 instances are created via automation or infrastructure-as-code, edit the user data in those pipelines and code.

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.2 Establish and Maintain a Data Inventory Establish and maintain a data inventory, based on the enterprise's data management process. Inventory sensitive data, at a minimum. Review and update inventory annually, at a minimum, with a priority on sensitive data.	●	●	●
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	1.5 Maintain Asset Inventory Information Ensure that the hardware asset inventory records the network address, hardware address, machine name, data asset owner, and department for each asset and whether the hardware asset has been approved to connect to the network.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	5.2 Maintain Secure Images Maintain secure images or templates for all systems in the enterprise based on the organization's approved configuration standards. Any new system deployment or existing system that becomes compromised should be imaged using one of those images or templates.		●	●

2.14 Ensure EC2 Auto Scaling Groups Propagate Tags to EC2 Instances that it launches (Automated)

Profile Applicability:

- Level 1

Description:

Tags can help with managing, identifying, organizing, searching for, and filtering resources. Additionally, tags can help with security and compliance. Tags can be propagated from an Auto Scaling group to the EC2 instances that it launches.

Rationale:

Without tags, EC2 instances created via Auto Scaling can be without tags and could be out of compliance with security policy.

Audit:

AWS Console

1. Login to AWS Console using https://console.aws.amazon.com
2. Click All services and click EC2 under Compute.
3. Select Auto Scaling Groups.
4. For each Auto Scaling Group's Details, ensure that all tags have `Tag new instances` set to `Yes`.
5. Repeat Steps 1-4 for each AWS Region used.

AWS CLI

1. Run `aws autoscaling describe-auto-scaling-groups`.
2. Ensure `PropogateAtLaunch` is `true` under `Tags` for each Tag for the Auto Scaling Group.
3. Repeat Steps 1-2 for each AWS Region used.

Remediation:

AWS Console

1. Login to AWS Console using https://console.aws.amazon.com
2. Click All services and click EC2 under Compute.
3. Select Auto Scaling Groups.
4. Click `Edit` for each Auto Scaling Group.
5. Check the `Tag new instances` Box for the Auto Scaling Group.
6. Click `Update`.
7. Repeat Steps 1-6 for each AWS Region used.

AWS CLI

1. Run `aws autoscaling create-or-update-tags` for tags that are not set to `PropogateAtLaunch` for each Auto Scaling Group that does not have this property set to true.

```
aws autoscaling create-or-update-tags \
    --tags ResourceId=example-autoscaling-group,ResourceType=auto-scaling-
group,Key=TagKey,Value=TagValue,PropagateAtLaunch=true
```

2. Repeat Step 1 for each AWS Region used.

References:

1. https://docs.aws.amazon.com/autoscaling/ec2/userguide/ec2-auto-scaling-tagging.html

Additional Information:

Note: Tags may be specified via the launch template. The tag values for instances from the launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.5 Maintain Asset Inventory Information Ensure that the hardware asset inventory records the network address, hardware address, machine name, data asset owner, and department for each asset and whether the hardware asset has been approved to connect to the network.		●	●

3 Amazon Elastic Container Service (ECS)

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service that helps you to more efficiently deploy, manage, and scale containerized applications. It deeply integrates with the AWS environment to provide an easy-to-use solution for running container workloads in the cloud and on premises with advanced security features using Amazon ECS Anywhere.

3.1 Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that Amazon ECS task definitions using `host` network mode do not allow privileged or root user access. This protects the host container instance from unauthorized access and privilege escalation.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Combining host networking mode with privileged or root user access significantly increases the risk of container breakout, where a compromised container can gain control of the host system.

Impact:

There may be some administrative effort required to ensure Amazon ECS applications function as expected without privileged or root user access.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. If `Network mode` is set to `host`, click `JSON`.
6. For each element under `containerDefinitions`, ensure that `privileged` is set to `false` or is absent, and ensure that `user` is not set to `root` or is absent.
7. Repeat steps 1-6 for each task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command:

```
aws ecs describe-task-definition --task-definition <task-definition-arn>
```

If `networkMode` is set to `host`, ensure that for each element under `containerDefinitions`, `privileged` is set to `false` or is absent, and ensure that `user` is not set to `root` or is absent.
Repeat for each task definition.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision with JSON`.
7. For each element under `containerDefinitions`, set `privileged` to `false`, or remove `"privileged": true`.
8. For each element under `containerDefinitions`, set `user` to an appropriate non-root user, or remove `"user": "root"`.
9. Click `Create`.
10. Repeat steps 1-9 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

Default Value:

When creating a task definition with `host` network mode, the `privileged` and `user` parameters are unset by default.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definitions.html
2. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-definition.html

Additional Information:

- AWS recommends using the `awsvpc` network mode unless you have a specific need to use a different network mode.
- The `privileged` parameter is not supported for Windows containers or tasks using the Fargate launch type.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

3.2 Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that `assignPublicIp` is set to `DISABLED` for Amazon ECS services, to restrict direct exposure of containers to the public internet.

Rationale:

Enabling public IP assignment could expose container application servers to unintended or unauthorized access.

Impact:

Disabling `assignPublicIp` introduces administrative, operational, and potential cost overhead due to the need to configure and maintain private networking and associated resources.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Under `Services`, click the name of a service.
5. Click `Configuration and networking`.
6. Under `Network configuration`, ensure `Auto-assign public IP` is set to `Turned off`.
7. Repeat steps 1-6 for each ECS cluster and service.

From Command Line:

Run the following command to list clusters:

```
aws ecs list-clusters
```

Run the following command to list services in a cluster:

```
aws ecs list-services --cluster <cluster-arn>
```

Run the following command to view the details of a service:

```
aws ecs describe-services --cluster <cluster-arn> --services <service-arn>
```

Under `networkConfiguration` > `awsvpcConfiguration`, ensure `assignPublicIp` is set to `DISABLED`.
Repeat for each cluster and service.

Remediation:

From Command Line:
For each service requiring remediation, run the following command to set `assignPublicIp` to `DISABLED`:

```
aws ecs update-service --cluster <cluster-arn> --service <service-arn> --
network-configuration '{"awsvpcConfiguration":{"subnets":["<subnet-
id>"],"securityGroups":["<security-group-id>"],"assignPublicIp":"DISABLED"}}'
```

Default Value:

By default, `assignPublicIp` is set to `ENABLED`.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-clusters.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-services.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-services.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/update-service.html
5. https://docs.aws.amazon.com/AmazonECS/latest/APIReference/API_AwsVpcConfiguration.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.3 Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host' (Automated)

Profile Applicability:

- Level 1

Description:

A process ID (PID) namespace isolates processes, preventing system processes from being visible to containers and allowing for PID reuse. Ensure that Amazon ECS task definitions are not configured to share a host's process namespace with their containers.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Setting the `pidMode` parameter to `host` shares the host's PID namespace with containers, allowing them to view and interact with all host processes. This reduces isolation and may lead to unauthorized access and manipulation of host processes by malicious or compromised containers.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `JSON`.
6. If the `pidMode` parameter is present, ensure it is not set to `host`.
7. Repeat steps 1-6 for each task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command:

```
aws ecs describe-task-definition --task-definition <task-definition-arn> --
query 'taskDefinition.pidMode'
```

Ensure that the command does not return `"host"`.
Repeat for each task definition.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision with JSON`.
7. Set `pidMode` to `task`, or remove the `pidMode` parameter as appropriate.
8. Click `Create`.
9. Repeat steps 1-8 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

Default Value:

If no value is specified, the default is a private namespace for each container.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-task-definitions.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-definition.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.1 Establish and Maintain a Secure Configuration Process Establish and maintain a secure configuration process for enterprise assets (end-user devices, including portable and mobile, non-computing/IoT devices, and servers) and software (operating systems and applications). Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	5.1 Establish Secure Configurations Maintain documented, standard security configuration standards for all authorized operating systems and software.	●	●	●

3.4 Ensure Amazon ECS task definitions do not have 'privileged' set to 'true' (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that Amazon ECS task definitions do not grant privileged access to the host container instance.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Restricting privileged access enhances security of the host container instance by maintaining isolation and reducing the risk of privilege escalation.

Impact:

There may be some administrative effort required to ensure Amazon ECS applications function as expected without privileged access.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `JSON`.
6. For each element under `containerDefinitions`, ensure that `privileged` is set to `false` or is absent.
7. Repeat steps 1-6 for each task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command:

```
aws ecs describe-task-definition --task-definition <task-definition-arn> --
query 'taskDefinition.containerDefinitions[*].privileged'
```

Ensure that the command does not return `true`.
Repeat for each task definition.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision with JSON`.
7. For each element under `containerDefinitions`, set `privileged` to `false`, or remove `"privileged": true`.
8. Click `Create`.
9. Repeat steps 1-8 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

Default Value:

By default, `privileged` is set to `false`.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-task-definitions.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-definition.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

3.5 Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions (Automated)

Profile Applicability:

- Level 1

Description:

Ensure the `readonlyRootFilesystem` parameter is enabled in Amazon ECS task definitions to restrict write access to the filesystem.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Enabling `readonlyRootFilesystem` minimizes security risks by ensuring the container's filesystem cannot be altered unless specific read-write permissions are granted.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `JSON`.
6. For each element under `containerDefinitions`, ensure `readonlyRootFilesystem` is set to `true`.
7. Repeat steps 1-6 for each task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command:

```
aws ecs describe-task-definition --task-definition <task-definition-arn> --
query 'taskDefinition.containerDefinitions[*].readonlyRootFilesystem'
```

Ensure that the command returns only `true`.
Repeat for each task definition.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision with JSON`.
7. For each element under `containerDefinitions`, set `readonlyRootFilesystem` to `true`.
8. Click Create.
9. Repeat steps 1-8 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

Default Value:

By default, `readonlyRootFilesystem` is set to `false`.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-task-definitions.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-definition.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.1 Establish and Maintain a Secure Configuration Process Establish and maintain a secure configuration process for enterprise assets (end-user devices, including portable and mobile, non-computing/IoT devices, and servers) and software (operating systems and applications). Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	5.1 Establish Secure Configurations Maintain documented, standard security configuration standards for all authorized operating systems and software.	●	●	●

3.6 Ensure secrets are not passed as container environment variables in Amazon ECS task definitions (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that sensitive secrets, such as `AWS_ACCESS_KEY_ID`, are not passed as environment variables in Amazon ECS task definitions. Use more secure methods, such as secrets management services like AWS Secrets Manager or AWS Systems Manager Parameter Store, to inject these credentials into containers.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Passing secrets as environment variables exposes them to potential compromise, as they can be easily accessed by any process running within the container or by unauthorized users. This practice can lead to the unintended leakage of sensitive information.

Impact:

There is some administrative overhead involved in configuring and integrating secrets management solutions.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `JSON`.
6. For each element under `containerDefinitions`, ensure that the `environment` parameter does not contain `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `ECS_ENGINE_AUTH_DATA`.
7. Repeat steps 1-6 for each task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command:

```
aws ecs describe-task-definition --task-definition <task-definition-arn> --
query 'taskDefinition.containerDefinitions[*].environment[*].name'
```

Ensure that the command does not contain `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `ECS_ENGINE_AUTH_DATA`.
Repeat for each task definition.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision with JSON`.
7. For each element under `containerDefinitions`, in the `environment` parameter, remove any objects with a `name` matching `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `ECS_ENGINE_AUTH_DATA`.
8. Click `Create`.
9. Repeat steps 1-8 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

Default Value:

By default, secrets are not present as container environment variables in task definitions.

References:

1. https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-paramstore-su-create.html
2. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/specifying-sensitive-data.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-task-definitions.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-definition.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.1 Establish and Maintain a Data Management Process Establish and maintain a data management process. In the process, address data sensitivity, data owner, handling of data, data retention limits, and disposal requirements, based on sensitivity and retention standards for the enterprise. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v8	3.2 Establish and Maintain a Data Inventory Establish and maintain a data inventory, based on the enterprise's data management process. Inventory sensitive data, at a minimum. Review and update inventory annually, at a minimum, with a priority on sensitive data.	●	●	●
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.		●	●
v7	16.5 Encrypt Transmittal of Username and Authentication Credentials Ensure that all account usernames and authentication credentials are transmitted across networks using encrypted channels.		●	●

3.7 Ensure logging is configured for Amazon ECS task definitions (Automated)

Profile Applicability:

- Level 1

Description:

Configure logging for Amazon ECS task definitions to capture detailed application and container activity.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Logging facilitates effective monitoring, troubleshooting, and incident response, improving security and enabling rapid threat detection.

Impact:

Logging can incur costs for storage and processing, along with initial configuration and ongoing management.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `JSON`.
6. Ensure that at least one element under `containerDefinitions` has a `logConfiguration` property defined, and that the value for `logDriver` is not `null`.
7. Repeat steps 1-6 for each task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command:

```
aws ecs describe-task-definition --task-definition <task-definition-arn> --
query 'taskDefinition.containerDefinitions[*].logConfiguration'
```

Ensure that the command returns at least one `logConfiguration` object, and that the value for `logDriver` is not `null`.
Repeat for each task definition.

Remediation:

From Console

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision` again.
7. For at least one container, under `Logging` > `Log collection`, check the box next to `Use log collection` and further configure the log collection options as needed.
8. Click Create.
9. Repeat steps 1-8 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

Default Value:

Logging is enabled by default when a task definition is created via the console.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_definition_parameters.htm
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-task-definitions.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-definition.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

3.8 Ensure Amazon ECS Fargate services are using the latest Fargate platform version (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that Amazon ECS Fargate services use the latest Fargate platform version to benefit from the latest security enhancements, performance improvements, and feature updates.

Rationale:

Using the latest Fargate platform version ensures services benefit from up-to-date security patches and features.

Impact:

Updating to the latest Fargate platform version may require minor operational effort.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Under `Services`, from the `Filter launch type` drop-down menu, select `FARGATE`.
5. Click the name of a service.
6. Click `Configuration and networking`.
7. Under `Service configuration`, ensure `Platform version` is set to `1.4.0` or `LATEST` for Linux, or `1.0.0` or `LATEST` for Windows.
8. Repeat steps 1-7 for each ECS cluster and Fargate service.

From Command Line:

Run the following command to list clusters:

```
aws ecs list-clusters
```

Run the following command to list services in a cluster:

```
aws ecs list-services --cluster <cluster-arn>
```

Run the following command to view the details of a service:

```
aws ecs describe-services --cluster <cluster-arn> --services <service-arn> --
query 'services[*].[platformFamily,platformVersion]' --output table
```

Where `platformFamily` is `Linux`, ensure `platformVersion` is `1.4.0` or `LATEST`.
Where `platformFamily` is `Windows`, ensure `platformVersion` is `1.0.0` or `LATEST`.
Repeat for each cluster and service.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Under `Services`, from the `Filter launch type` drop-down menu, select `FARGATE`.
5. Click the name of a service.
6. Click `Update service`.
7. Expand the `Compute configuration (advanced)` section.
8. Under `Platform version`, select `LATEST` from the drop-down menu.
9. Click `Update`.
10. Repeat steps 1-9 for each ECS cluster and Fargate service requiring remediation.

From Command Line:
For each service requiring remediation, run the following command to set `platformVersion` to `LATEST`:

```
aws ecs update-service --cluster <cluster-arn> --service <service-arn> --
platform-version LATEST
```

Default Value:

The platform version for Fargate services is `LATEST` by default.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/platform-fargate.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-clusters.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-services.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-services.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	2.2 Ensure Authorized Software is Currently Supported Ensure that only currently supported software is designated as authorized in the software inventory for enterprise assets. If software is unsupported, yet necessary for the fulfillment of the enterprise's mission, document an exception detailing mitigating controls and residual risk acceptance. For any unsupported software without an exception documentation, designate as unauthorized. Review the software list to verify software support at least monthly, or more frequently.	●	●	●
v7	2.2 Ensure Software is Supported by Vendor Ensure that only software applications or operating systems currently supported by the software's vendor are added to the organization's authorized software inventory. Unsupported software should be tagged as unsupported in the inventory system.	●	●	●

3.9 Ensure monitoring is enabled for Amazon ECS clusters (Automated)

Profile Applicability:

- Level 2

Description:

Enable AWS CloudWatch Container Insights for Amazon ECS clusters to monitor resource usage, performance, and application health through metrics and logs.

Rationale:

Monitoring ECS clusters with Container Insights improves visibility, supports faster issue detection, and enhances security by identifying anomalies and resource bottlenecks.

Impact:

Enabling AWS CloudWatch Container Insights for ECS clusters incurs costs for metrics, log ingestion, storage, and alarms.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. For each cluster listed in the `CloudWatch monitoring` column, ensure that `Container Insights` is displayed.

From Command Line:
Run the following command to list clusters:

```
aws ecs list-clusters
```

Run the following command to view the settings for a cluster:

```
aws ecs describe-clusters --clusters <cluster-arn> --include SETTINGS --query
'clusters[*].settings'
```

Ensure `containerInsights` is set to `enabled` or `enhanced`.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Click `Update cluster`.

5. Under `Monitoring`, select the radio button next to `Container Insights` or `Container Insights with enhanced observability.
6. Click `Update`.
7. Repeat steps 1-6 for each ECS cluster requiring remediation.

From Command Line:
For each cluster requiring remediation, run the following command to enable `containerInsights`:

```
aws ecs update-cluster-settings --cluster <cluster-arn> --settings
name=containerInsights,value=enabled
```

Default Value:

Monitoring is disabled by default for Amazon ECS clusters.

References:

1. https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ContainerInsights.html
2. https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/container-insights-detailed-ecs-metrics.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

3.10 Ensure Amazon ECS services are tagged (Automated)

Profile Applicability:

- Level 1

Description:

Ensure all Amazon ECS services have resource tags to facilitate asset management, tracking, and compliance.

Rationale:

Consistent tagging supports compliance and helps identify unauthorized or misconfigured resources.

Impact:

There is minimal administrative overhead associated with implementing and maintaining resource tags.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Under `Services`, click the name of a service.
5. Click `Tags`.
6. Ensure at least one tag is listed that does not begin with `aws:`. Tags prefixed with `aws:` are AWS-managed.
7. Repeat steps 1-6 for each ECS cluster and service.

From Command Line:

Run the following command to list clusters:

```
aws ecs list-clusters
```

Run the following command to list services in a cluster:

```
aws ecs list-services --cluster <cluster-arn>
```

Run the following command to view the tags for a service:

```
aws ecs list-tags-for-resource --resource-arn <service-arn>
```

Ensure that tags are returned that do not begin with `aws:`. Tags prefixed with `aws:` are AWS-managed.
Repeat for each cluster and service.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Under `Services`, click the name of a service.
5. Click `Tags`.
6. Click `Manage tags`.
7. Click `Add tag`.
8. Provide a `Key` and optional `Value` for the tag.
9. Click `Save`.
10. Repeat steps 1-9 for each ECS cluster and service requiring remediation.

Default Value:

By default, Amazon ECS services are not tagged.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-using-tags.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-clusters.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-services.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-tags-for-resource.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

3.11 Ensure Amazon ECS clusters are tagged (Automated)

Profile Applicability:

- Level 1

Description:

Ensure all Amazon ECS clusters have resource tags to facilitate asset management, tracking, and compliance.

Rationale:

Consistent tagging supports compliance and helps identify unauthorized or misconfigured resources.

Impact:

There is minimal administrative overhead associated with implementing and maintaining resource tags.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Click `Tags`.
5. Ensure at least one tag is listed that does not begin with `aws:`. Tags prefixed with `aws:` are AWS-managed.
6. Repeat steps 1-5 for each ECS cluster.

From Command Line:
Run the following command to list clusters:

```
aws ecs list-clusters
```

Run the following command to view the tags for a cluster:

```
aws ecs list-tags-for-resource --resource-arn <service-arn>
```

Ensure that tags are returned that do not begin with `aws:`. Tags prefixed with `aws:` are AWS-managed.
Repeat for each cluster.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.

3. Click the name of a cluster.
4. Click `Tags`.
5. Click `Manage tags`.
6. Click `Add tag`.
7. Provide a `Key` and optional `Value` for the tag.
8. Click `Save`.
9. Repeat steps 1-8 for each ECS cluster requiring remediation.

Default Value:

By default, Amazon ECS clusters have only AWS-managed tags.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-using-tags.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-clusters.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-tags-for-resource.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

3.12 Ensure Amazon ECS task definitions are tagged (Automated)

Profile Applicability:

- Level 1

Description:

Ensure all Amazon ECS task definitions have resource tags to facilitate asset management, tracking, and compliance.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Consistent tagging supports compliance and helps identify unauthorized or misconfigured resources.

Impact:

There is minimal administrative overhead associated with implementing and maintaining resource tags.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Tags`.
6. Ensure at least one tag is listed that does not begin with `aws:`. Tags prefixed with `aws:` are AWS-managed.
7. Repeat steps 1-6 for each ECS task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command to view the tags:

```
aws ecs list-tags-for-resource --resource-arn <task-definition-arn>
```

Ensure that tags are returned that do not begin with `aws:`. Tags prefixed with `aws:` are AWS-managed.
Repeat for each task definition.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision` again.
7. Expand the `Tags` section.
8. Click `Add tag`.
9. Provide a `Key` and optional `Value` for the tag.
10. Click `Create`.
11. Repeat steps 1-10 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

Default Value:

By default, Amazon ECS task definitions are not tagged.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-using-tags.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-task-definitions.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-tags-for-resource.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

3.13 Ensure only trusted images are used with Amazon ECS (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that only trusted container images from verified sources or private repositories are used with Amazon ECS to maintain the integrity and security of workloads.

Note: This recommendation assumes that only the latest active revision of a task definition is in use. If older revisions are in use, apply the audit and remediation procedures to those revisions as needed.

Rationale:

Using trusted images reduces the risk of vulnerabilities, malware, or unauthorized modifications compromising ECS tasks.

Impact:

Minor costs for scanning, storage, and administrative effort to enforce policies and manage approved images.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `JSON`.
6. For each element under `containerDefinitions`, ensure that `image` is set to an image trusted by your organization.
7. Repeat steps 1-6 for each task definition.

From Command Line:
Run the following command to list task definitions:

```
aws ecs list-task-definitions
```

For the latest revision of a task definition, run the following command:

```
aws ecs describe-task-definition --task-definition <task-definition-arn> --
query 'taskDefinition.containerDefinitions[*].image'
```

Ensure that the command returns only images trusted by your organization.
Repeat for each task definition.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Task definitions`.
3. Click the name of a task definition.
4. Click on the latest active revision of the task definition.
5. Click `Create new revision`.
6. Click `Create new revision with JSON`.
7. For each element under `containerDefinitions`, set `image` to an appropriate image trusted by your organization.
8. Repeat steps 1-7 for each task definition requiring remediation.

Note: When a task definition is updated, running tasks launched from the previous task definition remain unchanged. Updating a running task requires redeploying it with the new task definition.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/container-considerations.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/list-task-definitions.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-definition.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	2.2 Ensure Authorized Software is Currently Supported Ensure that only currently supported software is designated as authorized in the software inventory for enterprise assets. If software is unsupported, yet necessary for the fulfillment of the enterprise's mission, document an exception detailing mitigating controls and residual risk acceptance. For any unsupported software without an exception documentation, designate as unauthorized. Review the software list to verify software support at least monthly, or more frequently.	●	●	●
v7	2.2 Ensure Software is Supported by Vendor Ensure that only software applications or operating systems currently supported by the software's vendor are added to the organization's authorized software inventory. Unsupported software should be tagged as unsupported in the inventory system.	●	●	●

3.14 Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that `assignPublicIp` is set to `DISABLED` for Amazon ECS task sets, to prevent task sets from being publicly accessible.

Rationale:

Enabling public IP assignment could expose task sets to unintended or unauthorized access.

Impact:

Disabling `assignPublicIp` introduces administrative, operational, and potential cost overhead due to the need to configure and maintain private networking and associated resources.

Audit:

From Command Line:
Run the following command to list clusters:

```
aws ecs list-clusters
```

Run the following command to list services in a cluster:

```
aws ecs list-services --cluster <cluster-arn>
```

Run the following command to view the task sets for a service:

```
aws ecs describe-task-sets --cluster <cluster-arn> --service <service-arn>
```

For each task set, under `networkConfiguration` > `awsvpcConfiguration`, ensure `assignPublicIp` is set to `DISABLED`.
Repeat for each cluster and service.

Remediation:

Default Value:

By default, `assignPublicIp` is set to `ENABLED`.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecs/describe-task-sets.html

2. https://docs.aws.amazon.com/AmazonECS/latest/APIReference/API_TaskSet.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

4 Amazon Elastic Kubernetes Service (EKS) (Reference)

Amazon Elastic Kubernetes Service (Amazon EKS) is a fully managed Kubernetes service that enables you to run Kubernetes seamlessly in both AWS Cloud and on-premises data centers. In the cloud, Amazon EKS automates Kubernetes cluster infrastructure management. This is essential for scheduling containers, managing application availability, dynamically scaling resources, optimizing compute, storing cluster data, and performing other critical functions. With Amazon EKS, you can leverage the robust performance, scalability, reliability, and availability of AWS infrastructure, as well as natively integrate with AWS networking, security, and storage services. To simplify running Kubernetes in on-premises environments, you can use the same Amazon EKS clusters, features, and tools to run nodes on AWS Outposts or your own infrastructure, or you can use Amazon EKS Anywhere for self-contained air-gapped environments.

At this time all Security Best practices for Amazon Elastic Kubernetes Service fall under the CIS Amazon Elastic Kubernetes Service (EKS) Benchmark in the CIS Kubernetes Benchmarks Community. Please refer to that Community and Benchmark for recommendations related to Amazon EKS.

CIS Kubernetes Benchmarks Community

5 Amazon Lightsail

Amazon Lightsail offers easy-to-use virtual private server (VPS) instances, containers, storage, databases, to create a website or application in just a few clicks. Automatically configure networking, access, and security environments.

Easily scale as you grow—or migrate your resources to the broader AWS ecosystem, such as Amazon EC2.

5.1 Apply updates to any apps running in Lightsail (Manual)

Profile Applicability:

- Level 1

Description:

Amazon Lightsail is a virtual private server (VPS) provider and is the easiest way to get started with AWS for developers, small businesses, students, and other users who need a solution to build and host their applications on cloud.

Rationale:

Lightsail offers a range of operating system and application templates that are automatically installed when you create a new Lightsail instance. Application templates include WordPress, Drupal, Joomla!, Ghost, Magento, Redmine, LAMP, Nginx (LEMP), MEAN, Node.js, Django, and more. You can install additional software on your instances by using the in-browser SSH or your own SSH client.

Audit:

To confirm that you are running the latest version of the application you are using is a manual process. Often dependent on the application itself and the operating system you are utilizing for the Lightsail instance.
From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Instance` you want to review.
5. Make sure the instance status is `running`.
6. Connect to the `instance`.
7. Depending on the instance OS and the application you are running determine what version it is and if there are any updates.
8. If there are updates refer to the remediation below.
9. Repeat steps no. 4 – 8 to verify if any Lightsail instances require application updates.

Remediation:

To process and apply the latest updates for the application you are using is a manual process. Often dependent on the application itself and the operating system you are utilizing for the Lightsail instance.
From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.

3. This will open up the Lightsail console.
4. Select the `Instance` you want to update.
5. Make sure the instance status is `running`.
6. Click on `Snapshots`
7. Under `Manual snapshots` click on `+ Create snapshot`
8. Give it a name you will recognize
9. Click on `create`

```
while in process it will show 'Snapshotting...'
```

10. Once the date and time and snapshot name appears it is completed.
11. Click on `Connect`
12. Run the updates for the application discovered above in the Audit.
13. Repeat steps no. 4 – 12 to apply any application updates required on the Lightsail instances that you are running.

References:

1. https://lightsail.aws.amazon.com/ls/docs/en_us/overview
2. https://aws.amazon.com/lightsail/features/?opdp2=features/?pg=ln&sec=hs

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	7.4 Perform Automated Application Patch Management Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.	●	●	●
v7	3.5 Deploy Automated Software Patch Management Tools Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.	●	●	●

5.2 Change default Administrator login names and passwords for applications (Manual)

Profile Applicability:

- Level 1

Description:

Change the default settings for the administrator login names and passwords of the application software that you install on Lightsail instances.

Rationale:

Default administrator login names and passwords for applications used on Lightsail instances can be used by hackers and individuals to break into your servers.

Audit:

To confirm that you have updated or changed the default administrator name and password for any application you are using is a manual process. Often dependent on the application itself and the operating system you are utilizing for the Lightsail instance.
From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Instance` you want to review.
5. Make sure the instance status is `running`.
6. Connect to the `instance`.
7. Depending on the instance OS and the application you are running determine what the default administrator name is set to and what the password is.
8. If the `default administrator` username and or password is still at the default settings please refer to the remediation below.
9. Repeat steps no. 4 – 8 to verify if any Lightsail instances require application updates.

Remediation:

To process and apply the latest updates for the application you are using is a manual process. Often dependent on the application itself and the operating system you are utilizing for the Lightsail instance.
From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.

4. Select the `Instance` you want to update the `default administrator` settings.
5. Make sure the instance status is `running`.
6. Click on `Snapshots`
7. Under `Manual snapshots` click on `+ Create snapshot`
8. Give it a name you will recognize
9. Click on `create`

```
while in process it will show Snapshotting...
```

10. Once the date and time and snapshot name appears it is completed.
11. Click on `Connect`
12. Run the process to change either the `default administrator` name or password or both.
13. Repeat steps no. 4 – 12 to apply any application `default administrator` changes required on the Lightsail instances that you are running.

References:

1. https://lightsail.aws.amazon.com/ls/docs/en_us/all

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	4.2 Change Default Passwords Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.	●	●	●

5.3 Disable SSH and RDP ports for Lightsail instances when not needed. (Manual)

Profile Applicability:

- Level 1

Description:

Any ports enable within Lightsail by default are open and exposed to the world. For SSH and RDP access you should remove and disable these ports when not is use.

Rationale:

Any ports enable within Lightsail by default are open and exposed to the world. This can result in outside traffic trying to access or even deny access to the Lightsail instances. Removing and disabling a protocol when not in use even if restricted by IP address is the safest solution especially when it is not required for access.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows or Linux Instance` you want to review.
5. Go to the Networking section.
6. If it is a Windows instance confirm that SSH has been removed. If it is a Linux instance confirm RDP has been removed.
7. If either still exists in the IPV4 Firewall list refer to the remediation below.
8. If the server needs HTTP, TCP Port 80 confirm that the application forwards Port 80 to HTTPS, TCP Port 443.
9. If the server does not need HTTP refer to the remediation below.
10. Confirm that there are no other unused or unneeded ports.
11. If the system has other ports that are not required or in use refer to the remediation below.

From the Command Line:

1. Run `aws lightsail get-instances`

```
aws lightsail get-instances --query "instances[*].name"
```

This command will provide a list of Instance names.

```
"WordPress-1",
"Windows_Server_2019-1"
```

2. Run `aws lightsail get-instance-port-states` for each instance listed above

```
aws lightsail get-instance-port-states --instance-name <instance_name>
```

This command will provide a list of available Ports for the Instance name.

```
"portStates": [
        {
            "fromPort": 80,
            "toPort": 80,
            "protocol": "tcp",
            "state": "open",
            "cidrs": [
                "0.0.0.0/0"
            ],
            "cidrListAliases": []
        },
        {
            "fromPort": 22,
            "toPort": 22,
            "protocol": "tcp",
            "state": "open",
            "cidrs": [
                "0.0.0.0/0"
            ],
            "cidrListAliases": []
        },
        {
            "fromPort": 443,
            "toPort": 443,
            "protocol": "tcp",
            "state": "open",
            "cidrs": [
                "0.0.0.0/0"
            ],
            "cidrListAliases": []
```

If it is a Linux host and has Port 3398 listed, HTTP Port 80 listed or any other ports listed that are not required refer to the remediation below.
If it is a Windows host and has Port 22 listed, HTTP Port 80 listed or any other ports listed that are not required refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows or Linux Instance` you want to review.

5. Go to the Networking section.
6. If it is a Windows instance confirm that SSH has been removed. If it is a Linux instance confirm RDP has been removed.
7. If either ssh(Port 22) is in the Windows system and RDP(Port 3389) is in the Linux system click the bucket icon to delete it.
8. If the server needs HTTP, TCP Port 80 confirm that the application forwards Port 80 to HTTPS, TCP Port 443.
9. If the server does not need HTTP click the bucket icon to delete it.
10. Confirm that there are no other unused or unneeded ports.
11. If the system has other ports that are not required or in use click the bucket icon to delete it.

From the Command Line:

1. Run `aws lightsail close-instance-public-ports`

For Windows:

```
aws lightsail close-instance-public-ports --instance-name
<Windows_Instance_Name> --port-info fromPort=22,protocol=TCP,toPort=22
```

For Linux:

```
aws lightsail close-instance-public-ports --instance-name
<Linux_Instance_Name> --port-info fromPort=3389,protocol=TCP,toPort=3389
```

For HTTP:

```
aws lightsail close-instance-public-ports --instance-name <ANY_Instance_Name>
--port-info fromPort=80,protocol=TCP,toPort=80
```

2. Repeat for all instance names identified in the audit that have SSH, RDP or HTTP's open and are not required based on the OS or the use of the system.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/lightsail/index.html#cli-aws-lightsail

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	9.2 Ensure Only Approved Ports, Protocols and Services Are Running Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●

5.4 Ensure SSH is restricted to only IP address that should have this access. (Manual)

Profile Applicability:

- Level 1

Description:

Any ports enable within Lightsail by default are open and exposed to the world. For SSH and RDP access you should identify which IP address need access.

Rationale:

Any ports enable within Lightsail by default are open and exposed to the world. This can result in outside traffic trying to access or even deny access to the Lightsail instances. Removing and adding approved IP address required for access.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Linux Instance` you want to review.
5. Go to the Networking section.
6. Confirm that the SSH Port is restricted to an IP address

```
Application     Protocol        Port or range / Code    Restricted to
SSH             TCP             22                      101.221.11.11
```

7. If SSH is needed and it is open to `Any IPv4 address` refer to the remediation below.

From the Command Line:

1. Run `aws lightsail get-instances`

```
aws lightsail get-instances --query "instances[*].name"
```

This command will provide a list of Instance names.

```
"WordPress-1",
"Windows_Server_2019-1"
```

2. Run `aws lightsail get-instance-port-states` for any Linux instances listed

```
aws lightsail get-instance-port-states --instance-name <instance_name>
```

This command will provide a list of available Ports for the Instance name.

```
        {
            "fromPort": 22,
            "toPort": 22,
            "protocol": "tcp",
            "state": "open",
            "cidrs": [
                "0.0.0.0/0"
               "101.221.11.11/32"
            ],
            "cidrListAliases": []
        },
```

3. Review the Port 22 settings and confirm that the only IP Addresses that should have access to the instance are listed in the cidrs as shown above.
4. If it is open to all ports (0.0.0.0/0) of there is an IP address listed that shouldn't have access refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Linux Instance` you want to review.
5. Go to the Networking section.
6. Under IPv4 networking find the SSH rule as shown below.

```
Application     Protocol        Port or range / Code    Restricted to
SSH             TCP             22                      Any IPv4 address
```

7. Click on the edit icon
8. Click on the check box next to Restrict to IP address
9. Under `Source IP address (192.0.2.0) or range (192.0.2.0-192.0.2.255 or 192.0.2.0/24)` type the IP address' you want.

From the Command Line:

1. Run `aws lightsail put-ins`

```
aws lightsail put-instance-public-ports --instance-name <instance_name> --
port-info
fromPort=22,protocol=TCP,toPort=22,cidrs=110.111.221.100/32,110.111.221.202/3
2
```

This command will enter the IP addresses that should have access to the instances identified above in the Audit.

2. Run `aws lightsail get-instance-port-states` for the Linux instance to confirm the new setting.

```
aws lightsail get-instance-port-states --instance-name <instance_name>
```

This command will provide a list of available Ports and show how the cidr value for Port 22 is now set.

```
        {
            "fromPort": 22,
            "toPort": 22,
            "protocol": "tcp",
            "state": "open",
            "cidrs": [
                "110.111.221.100/32",
                "110.111.221.202/32"
            ],
            "cidrListAliases": []
        },
```

3. Repeat the remediation below for all other instances identified in the Audit.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/lightsail/index.html#cli-aws-lightsail

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.5 Ensure RDP is restricted to only IP address that should have this access. (Manual)

Profile Applicability:

- Level 1

Description:

Any ports enable within Lightsail by default are open and exposed to the world. For SSH and RDP access you should identify which IP address need access.

Rationale:

Any ports enable within Lightsail by default are open and exposed to the world. This can result in outside traffic trying to access or even deny access to the Lightsail instances. Removing and adding approved IP address required for access.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows Instance` you want to review.
5. Go to the Networking section.
6. Confirm that the RDP Port is restricted to an IP address

```
Application     Protocol        Port or range / Code    Restricted to
RDP             TCP             3389                    101.221.11.11
```

7. If RDP is needed and it is open to `Any IPv4 address` refer to the remediation below.

From the Command Line:

1. Run `aws lightsail get-instances`

```
aws lightsail get-instances --query "instances[*].name"
```

This command will provide a list of Instance names.

```
"WordPress-1",
"Windows_Server_2019-1"
```

2. Run `aws lightsail get-instance-port-states` for any Windows instances listed

```
aws lightsail get-instance-port-states --instance-name <instance_name>
```

This command will provide a list of available Ports for the Instance name.

```
        {
            "fromPort": 3389,
            "toPort": 3389,
            "protocol": "tcp",
            "state": "open",
            "cidrs": [
                "0.0.0.0/0"
            ],
            "cidrListAliases": []
        },
```

3. Review the Port 22 settings and confirm that the only IP Addresses that should have access to the instance are listed in the cidrs as shown above.
4. If it is open to all ports (0.0.0.0/0) of there is an IP address listed that shouldn't have access refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows Instance` you want to review.
5. Go to the Networking section.
6. Under IPv4 networking find the SSH rule as shown below.

```
Application     Protocol        Port or range / Code    Restricted to
RDP             TCP             3389                    Any IPv4 address
```

7. Click on the edit icon
8. Click on the check box next to Restrict to IP address
9. Under `Source IP address (192.0.2.0) or range (192.0.2.0-192.0.2.255 or 192.0.2.0/24)` type the IP address' you want.

From the Command Line:

1. Run `aws lightsail put-ins`

```
aws lightsail put-instance-public-ports --instance-name <instance_name> --
port-info
fromPort=3389,protocol=TCP,toPort=3389,cidrs=110.111.221.100/32,110.111.221.2
02/32
```

This command will enter the IP addresses that should have access to the instances identified above in the Audit.

2. Run `aws lightsail get-instance-port-states` for the Windows instance to confirm the new setting.

```
aws lightsail get-instance-port-states --instance-name <instance_name>
```

This command will provide a list of available Ports and show how the cidr value for Port 3389 is now set.

```
"portStates": [
        {
            "fromPort": 3389,
            "toPort": 3389,
            "protocol": "tcp",
            "state": "open",
            "cidrs": [
                "110.111.221.100/32",
                "110.111.221.202/32"
            ],
            "cidrListAliases": []
        }
```

3. Repeat the remediation below for all other Windows instances identified in the Audit.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/lightsail/index.html#cli-aws-lightsail

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.6 Disable IPv6 Networking if not in use within your organization. (Manual)

Profile Applicability:

- Level 1

Description:

Any protocols enable within Lightsail by default that aren't being used should be disabled.

Rationale:

Any ports enable within Lightsail by default are open and exposed to the world. This can result in outside traffic trying to access or even deny access to the Lightsail instances. Removing and disabling a protocol when not in use even if restricted by IP address is the safest solution especially when it is not required for access.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows or Linux Instance` you want to review.
5. Go to the Networking section.
6. Under IPv6 networking confirm that it reads `IPv6 networking is disabled`.
7. If it reads `IPv6 networking is enabled` refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows or Linux Instance` you want to review.
5. Go to the Networking section.
6. Under IPv6 networking click on the check mark next to `IPv6 networking is enabled`.
7. In the `Disable IPv6 for this instance?`
8. Click on `Yes, disable`

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●
v7	9.2 Ensure Only Approved Ports, Protocols and Services Are Running Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●

5.7 Ensure you are using an IAM policy to manage access to buckets in Lightsail. (Manual)

Profile Applicability:

- Level 1

Description:

The following policy grants a user access to manage a specific bucket in the Amazon Lightsail object storage service.

Rationale:

This policy grants access to buckets through the Lightsail console, the AWS Command Line Interface (AWS CLI), AWS API, and AWS SDKs.

Impact:

Users who don't have this policy will experience errors when viewing the Objects tab of the bucket management page in the Lightsail console.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `IAM` under Security, Identity, & Compliance.
3. Click `Policies`
4. Click in the `Filter policies by property or policy name and press enter`
5. Type `Lightsail` and press enter
6. Click on the policy that contains lightsail in the name
7. Make sure the `Permissions` tab is selected.
8. Confirm the policy looks like this

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "LightsailAccess",
 "Effect": "Allow",
 "Action": "lightsail:*",
 "Resource": "*"
 },
 {
 "Sid": "S3BucketAccess",
 "Effect": "Allow",
 "Action": "s3:*",
 "Resource": [
 "arn:aws:s3:::<BucketName>/*",
 "arn:aws:s3:::<BucketName>"
 ]
 }
 ]
}
```

9. If this policy is in place move to the next step. If it is not in any of the policies listed for `lightsail` refer to the remediation below.
10. Click on the `Policy usage` tab
11. Confirm that the correct Group and/or User is listed under Permissions. If there is no one listed here refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `IAM` under Security, Identity, & Compliance.
3. Click `Policies`
4. Click `Create policy`
5. Click on the JSON tab
6. Copy and paste the policy below into the JSON editor replacing the text in there and filling in the Lightsail bucket names.
   **You can find the Lightsail bucket name in the Lightsail console, Storage, Under buckets.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "LightsailAccess",
 "Effect": "Allow",
 "Action": "lightsail:*",
 "Resource": "*"
 },
 {
 "Sid": "S3BucketAccess",
 "Effect": "Allow",
 "Action": "s3:*",
 "Resource": [
 "arn:aws:s3:::<BucketName>/*",
 "arn:aws:s3:::<BucketName>"
 ]
 }
 ]
}
```

7. Click `Next tags`
8. Add tags based on your companies outlined Tagging policy that should be in place based on the AWS Foundations Benchmark.
9. Click `Next review`
10. Click in `Name*` and give it a name that contains "Lightsail"
11. Review the summary.
12. Click `Create policy`
13. Click in the `Filter policies by property or policy name and press enter`
14. Type `Lightsail` and press enter
15. Click on the Policy name that you just created.
16. Click on the `Policy usage` tab
17. Click `Attach`
18. Add in the Users or Group that should have this permission.
19. Click `Attach policy`

References:

1. https://lightsail.aws.amazon.com/ls/docs/en_us/articles/amazon-lightsail-bucket-management-policies

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.8 Ensure Lightsail instances are attached to the buckets (Manual)

Profile Applicability:

- Level 1

Description:

Attaching an Amazon Lightsail instance to a Lightsail storage bucket gives it full programmatic access to the bucket and its objects.

Rationale:

When you attach instances to buckets, you don't have to manage credentials like access keys. Resource access is ideal if you're configuring software or a plugin on your instance to upload files directly to your bucket. For example, if you want to configure a WordPress instance to store media files on a bucket configuration with bucket storage resource access allows for that securely.

Impact:

You can attach instances that are in a running state only. Additionally, the instances have to be in the same AWS Region as the bucket or the buckets have to be in the same region as the instances.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select `Storage`.
5. All Lightsail buckets are listed here.
6. Click on a bucket name
7. Click `Permissions`.
8. Scroll down to `Resource access` and confirm that your instance is attached.
9. If the instance using this Storage bucket is not attached refer to the remediation below.

From the Command Line:

1. Run `aws lightsail get-buckets`

```
aws lightsail get-buckets
```

This command will provide a list of Buckets tied to Lightsail.

2. If there are no buckets listed then refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Confirm that the `instance` you want to connect to the Storage bucket is in a `running` state
5. If it is move on to Step 6. If it is not click on the instance name, then click on `Start`. Wait for the status to read `Running`
6. Select `Storage`.
7. All Lightsail buckets are listed here.
8. Click on the bucket you want to associate with the instances.
9. Click `Permissions`.
10. Scroll down to `Resource access`.
11. Click on `Attach instance`
12. Click on `Choose an instance`
13. Select the instance
14. Click Attach
15. Repeat this for any other instances and buckets that need to be attached.

From the Command Line:

1. Run `aws lightsail create-bucket`

```
aws lightsail create-bucket --bucket-name test-cli-bucket2 --bundle-id
small_1_0
```

This command will create a bucket.
If you want to review the bundle size ids run this command.

```
aws lightsail get-bucket-bundles
"bundles": [
        {
            "bundleId": "small_1_0",
            "name": "Object Storage 5GB",
            "price": 1.0,
            "storagePerMonthInGb": 5,
            "transferPerMonthInGb": 25,
            "isActive": true
        },
        {
            "bundleId": "medium_1_0",
            "name": "Object Storage 100GB",
            "price": 3.0,
            "storagePerMonthInGb": 100,
            "transferPerMonthInGb": 250,
            "isActive": true
        },
        {
            "bundleId": "large_1_0",
            "name": "Object Storage 250GB",
            "price": 5.0,
            "storagePerMonthInGb": 250,
            "transferPerMonthInGb": 500,
            "isActive": true
```

Change the "bundleId" to the size of storage you need.
Repeat and create all the S3 buckets that you need for Lightsail.

References:

1. https://lightsail.aws.amazon.com/ls/docs/en_us/articles/amazon-lightsail-configuring-bucket-resource-access

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	1.7 Deploy Port Level Access Control Utilize port level access control, following 802.1x standards, to control which devices can authenticate to the network. The authentication system shall be tied into the hardware asset inventory data to ensure only authorized devices can connect to the network.		●	●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

5.9 Ensure that your Lightsail buckets are not publicly accessible (Manual)

Profile Applicability:

- Level 1

Description:

You can make all objects private, public (read-only) or private while making individual objects public (read-only). By default when creating a bucket the permissions are set to "All objects are private".

Rationale:

When the Bucket access permissions are set to All objects are public (read-only) – All objects in the bucket are readable by anyone on the internet through the URL of the bucket.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select `Storage`.
5. All Lightsail buckets are listed here.
6. Underneath the bucket name and size there are 3 possible statements:

```
All objects are private
All objects are public (read-Only)
Individual objects can be public
```

7. If any buckets are set to `All objects are public (read-Only)` and or 'Individual objects can be public' refer to the remediation below.

From the Command Line:

1. Run `aws lightsail get-buckets`

```
aws lightsail get-buckets
```

This command will provide a list of Buckets tied to Lightsail.

2. Review the accessRules, getobject and allowPublicOverrides.

```
"accessRules": {
                "getObject": "private",
                "allowPublicOverrides": false
```

4. If it reads "getObject": "public" or "allowPublicOverrides": true please make note "name" of the bucket also listed in the output.
5. Then refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select `Storage`.
5. All Lightsail buckets are listed here.
6. Click on the bucket name that has `All objects are public (read-Only)` listed.
7. Click on `Permissions`
8. Click on `Change permissions`
9. Select `All objects are private`
10. Click `Save`
11. Repeat for any other Buckets within Lightsail that are set with `All objects are public (read-Only)` and/or `Individual objects can be made public and read only`

From the Command Line:

1. Run `aws lightsail update-bucket`

```
aws lightsail update-bucket --bucket-name <name from list in audit> --access-
rules getObject="private",allowPublicOverrides=false
```

2. The confirmation that the change was made will print out after running that command.
3. Repeat for any other buckets listed in the audit.

References:

1. https://lightsail.aws.amazon.com/ls/docs/en_us/articles/amazon-lightsail-understanding-bucket-permissions

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.10 Enable storage bucket access logging (Manual)

Profile Applicability:

- Level 1

Description:

Access logging provides detailed records for the requests that are made to this bucket. This information can include the request type, the resources that are specified in the request, and the time and date that the request was processed. Access logs are useful for many applications.

Rationale:

Access log information is useful in security and access audits.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select `Storage`.
5. All Lightsail buckets are listed here.
6. Click on a bucket name
7. Click `Logging`.
8. Confirm that Access logging is set to active. If it is set to inactive refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select `Storage`.
5. All Lightsail buckets are listed here.
6. Click on a bucket name
7. Click `Logging`.
8. Click on the X next to `Access logging is inactive`
9. Select a different bucket specific to store the logging information.
10. Note the path or create a path that matches your organization style.
11. Click save
12. Click OK
13. Repeat steps 6-12 for all Lightsail buckets.

References:

1. https://lightsail.aws.amazon.com/ls/docs/en_us/articles/amazon-lightsail-enabling-bucket-access-logs

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

5.11 Ensure your Windows Server based lightsail instances are updated with the latest security patches. (Manual)

Profile Applicability:

- Level 1

Description:

Windows server based Lightsail instances are still managed by the consumer and any security updates or patches have to be installed and maintained by the user.

Rationale:

Windows Server-based Lightsail instances need to be updated with the latest security patches so they are not vulnerable to attacks. Be sure your server is configured to download and install updates.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows Instance` you want to review.
5. Make sure the instance status is `running`.
6. Connect to the `instance` using `Connect using RDP`.
7. Log in using the credentials you have set for this instance.
8. Open a command prompt
9. Type sconfig, and then press Enter.

```
Windows Update Settings are at number 5 and by default are set to Automatic.
```

If this is the current setting continue with step 10. If this is not the current setting refer to the remediation below and start at step 10.

10. To determine if any updates are required, type 6, and then press Enter.
11. Type A to search for (A)ll updates in the new command window, and then press Enter.

If any updates are required refer to the remediation below and start at step 14.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`

2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows Instance` you want to review.
5. Make sure the instance status is `running`.
6. Connect to the `instance` using `Connect using RDP`.
7. Log in using the credentials you have set for this instance.
8. Open a command prompt
9. Type sconfig, and then press Enter.

```
Windows Update Settings are at number 5 and by default are set to Automatic.
```

If this is not the current setting continue with step 10. If this is the current setting skip to step 12

10. Type 5, and then press Enter.
11. Type A for `Automatic` and then press Enter. Wait until the setting is saved and you return back to the server configuration menu.
12. Type 6, and then press Enter.
13. Type A to search for (A)ll updates in the new command window, and then press Enter.
14. Type A again to install (A)ll updates, and then press Enter.

When finished, you see a message with the installation results and more instructions (if those apply).

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	7.4 Perform Automated Application Patch Management Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.	●	●	●
v7	3.5 Deploy Automated Software Patch Management Tools Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.	●	●	●

5.12 Change the auto-generated password for Windows based instances. (Manual)

Profile Applicability:

- Level 1

Description:

When you create a Windows Server-based instance, Lightsail randomly generates a long password that is hard to guess. You use this password uniquely with your new instance. You can use the default password to connect quickly to your instance using remote desktop (RDP). You are always logged in as the Administrator on your Lightsail instance.

Rationale:

Like any password it should be changed from the default and over time. The randomly generated password can be hard to remember and if anyone gains access to your AWS Lightsail environment they can utilize that to access your instances. For this reason you should change the password to something you can remember.

Impact:

If you change your password from the unique, default password, be sure to use a strong password. You should avoid passwords that are based on names or dictionary words, or repeating sequences of characters.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows Instance` you want to review.
5. Make sure the instance status is `running`.
6. Connect to the `instance` using `Connect using RDP`.
7. Log in using the credentials provided within the Lightsail console set for this instance.
8. If you are successful and based on your password change policy it is required that you change/update the password refer to the remediation below.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`

2. Click `All services`, click `Lightsail` under Compute.
3. This will open up the Lightsail console.
4. Select the `Windows Instance` you want to review.
5. Make sure the instance status is `running`.
6. Connect to the `instance` using `Connect using RDP`.
7. Log in using the credentials provided within the Lightsail console set for this instance.
8. Use the Windows Server password manager to change your password securely by press `Ctrl + Alt + Del`
9. Then choose `Change a password`.
   ** Be sure to keep a record of your password, because Lightsail doesn't store the new password you are setting.
10. Type in the `New Password`
11. Click `Save`

Additional Information:

You can use either the Lightsail-generated password or your own custom password with the browser-based RDP client in Lightsail. If you use a custom password, you will be prompted for your password every time you log in. It can be easier but not necessarily more secure to use the Lightsail-generated default password with the browser-based RDP client if you want quick access to your instance.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	4.2 Change Default Passwords Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.	●	●	●

6 AWS App Runner

AWS App Runner is a fully managed service that makes it easy for developers to deploy from source code or container image directly to a scalable and secure web application.

At this time all Security Best practices for the customer for App Runner occurs with source code development or with the container image. So Best practices should be considered when developing the source code and/or container image and should happen prior to those being utilized with App Runner. There are no specific Security best practice recommendations related for AWS App Runner itself, but that does not alleviate the end user of this service of the shared responsibility model and the customer security requirements for service and data access and protection.

6.1 Ensure you are using VPC Endpoints for source code access (Manual)

Profile Applicability:

- Level 1

Description:

App Runner needs access to your application source, so it can't be encrypted. Therefore, be sure to secure the connection between your development or deployment environment and App Runner.

Rationale:

Client-side encryption isn't a valid method for protecting the source image or code that you provide to App Runner for deployment. Using a VPC endpoint, you can privately connect your VPC to supported AWS services and VPC endpoint services that are powered by AWS PrivateLink.

Note that this isn't required if you are deploying your app runner directly from an ECR image as ECR images can be independently encrypted.

Audit:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/vpc/`
2. On the left hand side, click Endpoints.
3. On the `Endpoints` page.
4. Review all the endpoints listed under name.
5. Locate the Endpoint assigned and configured for App Runner.
6. If there is no Endpoint set for App Runner refer to the remediation below.
7. Either click the check box, Actions, View Details or click on the VPC endpoint ID.
8. Confirm these settings

```
1. Service name - `com.amazonaws."region".apprunner`
**Note - "Region" will reflect the region that you are operating in.
2. Status - Available
3. VPC ID - correctly associated for use with the service
4. Subnets tab - correctly associated for use with the service
5. Security Groups tab - correctly associated for use with the service
6. Policy tab - correctly configured for use with the service
```

9. If the settings listed above are not correct refer to the remediation below.

Remediation:

To create an interface endpoint for an App Runner
From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/vpc/`
2. On the left hand side, click Endpoints.
3. Click `Create endpoint`.
4. Under Service category, choose AWS services.
5. For Service name, select `com.amazonaws."region".apprunner`. "Region" will reflect the region that your are operating in.
6. For VPC, select the VPC from which you'll access App Runner.
7. For Subnets, select one subnet per Availability Zone.
8. For Security group, select the security groups to associate with the App Runner endpoint network interfaces.
9. For Policy, select Custom to attach a VPC endpoint policy that controls the permissions that principals have for performing actions on resources over the VPC endpoint.
10. Click `Create endpoint`.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v8	3.12 Segment Data Processing and Storage Based on Sensitivity Segment data processing and storage based on the sensitivity of the data. Do not process sensitive data on enterprise assets intended for lower sensitivity data.		●	●
v7	5.2 Maintain Secure Images Maintain secure images or templates for all systems in the enterprise based on the organization's approved configuration standards. Any new system deployment or existing system that becomes compromised should be imaged using one of those images or templates.		●	●
v7	10.4 Ensure Protection of Backups Ensure that backups are properly protected via physical security or encryption when they are stored, as well as when they are moved across the network. This includes remote backups and cloud services.	●	●	●

7 AWS Auto Scaling

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it’s easy to setup application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you’re already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

AWS Auto Scaling is primarily intended for cost and performance optimization. There are currently no specific Security best practice recommendations related to AWS Auto Scaling itself, but that does not alleviate the end user of this service of the Shared responsibility model and the customer security requirements for service and data access and protection.

8 AWS Batch

AWS Batch enables developers, scientists, and engineers to easily and efficiently run hundreds of thousands of batch computing jobs on AWS. AWS Batch dynamically provisions the optimal quantity and type of compute resources (e.g., CPU or memory optimized instances) based on the volume and specific resource requirements of the batch jobs submitted. With AWS Batch, there is no need to install and manage batch computing software or server clusters that you use to run your jobs, allowing you to focus on analyzing results and solving problems.

8.1 Ensure AWS Batch is configured with AWS Cloudwatch Logs. (Manual)

Profile Applicability:

- Level 1

Description:

You can configure Batch jobs to send log information to CloudWatch Logs.

Rationale:

This enables you to view different logs from all your jobs in one convenient location.

Audit:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/batch/`
2. On the left hand side under `Console settings`, Click on `Permissions`
3. Under `Job logs` section
4. Confirm that `Authorize Batch to use Cloudwatch` is set with a green check.
5. If is is showing a red X refer to the remediation below.

Remediation:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/batch/`.
2. In the left column under Console settings, Click on `Permissions`
3. In the Job logs section click on `Edit`
4. Click the `Authorize Batch to use CloudWatch`
5. Click Save

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

8.2 Ensure Batch roles are configured for cross-service confused deputy prevention (Manual)

Profile Applicability:

- Level 1

Description:

The Cross-service confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action.

Rationale:

Cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the calling service) calls another service (the called service). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access.

Impact:

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust. IAM Roles are often organization named and organization based. Searching for and reviewing the roles for this recommendation is a manual process.

Audit:

From the Console

1. Login to the AWS Console using https://console.aws.amazon.com/iam/
2. On the left hand side under Access management, Click on `Roles`
3. Search for any roles related to `Batch`
4. Click on the role and the Assume Role Policy Document and confirm that the AssumeRole Action has a aws:SourceArn key that contains the full ARN of the Batch resource

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-
environment/testCE",
          ]
        }
      }
    }
  ]
}
```

5. If it is showing an * within the ARN or does not have this condition key specified, then the Batch process has access to all of the resources defined in that environment.

```
"arn:aws:batch:us-east-1:123456789012:compute-environment/*",
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Repeat for any roles assigned to Batch that have AssumeRole
7. Refer to the remediation below

Remediation:

From the Console

1. Login to the AWS Console using https://console.aws.amazon.com/iam/
2. On the left hand side under Access management, Click on `Roles`
3. Search for any roles identified above in the audit.
4. Click on the role and update the Action AssumeRole, aws:SourceArn to contain the full ARN of the resource

```
"aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-
environment/testCE",
```

5. Repeat for any roles defined in the Audit.

References:

1. https://docs.aws.amazon.com/batch/latest/userguide/cross-service-confused-deputy-prevention.html

Additional Information:

Note: Usage of the `aws:SourceAccount` condition key can be used to prevent cross service confused deputy impersonation from external accounts. This condition key is not as specific as using `aws:SourceArn` which can be used to limit access of the IAM Role for specific resources or a group of specific resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●

9 AWS Compute Optimizer

Provides recommendations to optimize your use of AWS resources.

AWS Compute Optimizer is primarily intended for cost and performance optimization. There are currently no specific Security best practice recommendations related to AWS Compute Optimizer itself, but that does not alleviate the end user of this service of the Shared responsibility model and the customer security requirements for service and data access and protection.

10 Elastic Beanstalk

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS. You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates.

10.1 Ensure Managed Platform updates is configured (Manual)

Profile Applicability:

- Level 1

Description:

AWS Elastic Beanstalk regularly releases platform updates to provide fixes, software updates, and new features. With managed platform updates, you can configure your environment to automatically upgrade to the latest version of a platform during a scheduled maintenance window.

Rationale:

Your application remains in service during the update process with no reduction in capacity. Managed updates are available on both single-instance and load-balanced environments. They also ensure you aren't introducing any vulnerabilities by running legacy systems that require updates and patches.

Audit:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/elasticbeanstalk
2. On the left hand side click `Environments`
3. Click on the `Environment name` that you want to review
4. Under the `environment_name-env` in the left column click `Configuration`
5. Scroll down under Configurations
6. Under category look for `Managed updates`
7. Confirm `Managed updates: enabled`
8. If status options reads `Managed updates: disabled` refer to the remediation below.
9. Repeat steps 3-8 for each environment within the current region.
10. Then repeat the Audit process for all other regions.

Remediation:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/elasticbeanstalk
2. On the left hand side click `Environments`
3. Click on the `Environment name` that you want to update
4. Under the `environment_name-env` in the left column click `Configuration`
5. Scroll down under Configurations
6. Under category look for `Managed updates`
7. Click on Edit
8. On the Managed Platform Updates page

```
Managed updates - click the Enable checkbox
Weekly update window - set preferred maintenance window
Update level- set it to Minor and patch
Instance replacement - click the Enabled checkbox
```

9. Click Apply
10. Repeat steps 3-8 for each environment within the current region that needs Managed updates set.
11. Then repeat the remediation process for all other regions identified in the Audit.

References:

1. https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environment-platform-update-managed.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	7.4 Perform Automated Application Patch Management Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.	●	●	●
v7	3.5 Deploy Automated Software Patch Management Tools Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.	●	●	●

10.2 Ensure Persistent logs is setup and configured to S3 (Manual)

Profile Applicability:

- Level 1

Description:

Elastic Beanstalk can be configured to automatically stream logs to the CloudWatch service.

Rationale:

With CloudWatch Logs, you can monitor and archive your Elastic Beanstalk application, system, and custom log files from Amazon EC2 instances of your environments.

Audit:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/elasticbeanstalk
2. On the left hand side click `Environments`
3. Click on the `Environment name` that you want to review
4. Under the `environment_name-env` in the left column click `Configuration`
5. Scroll down under Configurations
6. Under category look for `Softwares`
7. Confirm `Log streaming: enabled`
8. If status options reads `Log streaming: disabled` refer to the remediation below.
9. Repeat steps 3-8 for each environment within the current region.
10. Then repeat the Audit process for all other regions.

Remediation:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/elasticbeanstalk
2. On the left hand side click `Environments`
3. Click on the `Environment name` that you want to update
4. Under the `environment_name-env` in the left column click `Configuration`
5. Scroll down under Configurations
6. Under category look for `Software`
7. Click on Edit
8. On the Modify software page

```
Instance log streaming to CloudWatch Logs
Log streaming - click the Enabled checkbox
Set the required retention based on Organization requirements
Lifecycle - Keep logs after terminating environment
```

9. Click Apply
10. Repeat steps 3-8 for each environment within the current region that needs Managed updates set.

References:

1. https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/AWSHowTo.cloudwatchlogs.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

10.3 Ensure access logs are enabled. (Manual)

Profile Applicability:

- Level 1

Description:

When you enable load balancing, your AWS Elastic Beanstalk environment is equipped with an Elastic Load Balancing load balancer to distribute traffic among the instances in your environment

Rationale:

For security reasons it is important to have a record of all the access logs and this is enabled within the Load Balancer assigned to the Elastic Beanstalk environments.

Audit:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/ec2
2. On the left hand scroll down to Load Balancing and click on `Load Balancers`
3. Click on the Load balancer associated with the Elastic Beanstalk Environment

```
Typically they have AWSEB in the name.
If you utilized Elastic Beanstalk to create the Load balancer the Source
Security Group listed in the Description will reference `Elastic Beanstalk`
```

4. Under the `Description` tab scroll down to the `Attributes` section
5. Confirm `Access logs` is set to Enabled.
6. If status options reads `Disabled` refer to the remediation below.
7. Repeat steps 3-8 for each environment within the current region.
8. Then repeat the Audit process for all other regions.

Remediation:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/ec2
2. On the left hand scroll down to Load Balancing and click on `Load Balancers`
3. Click on the Load balancer associated with the Elastic Beanstalk Environment

```
Typically they have AWSEB in the name.
If you utilized Elastic Beanstalk to create the Load balancer the Source
Security Group listed in the Description will reference `Elastic Beanstalk~
```

4. Under the `Description` tab scroll down to the `Attributes` section
5. Under Access logs - Disabled click on Configure access logs.
6. Click the check box next to `Enable access logs`.
7. enter the se bucket name you have setup for the Elastic Beanstalk access logs. **Note - if you don't have a s3 bucket already created enter an organization name in accordance with policy and have it identify with Elastic Beanstalk. Then click the check box next to `Create this location for me`
8. Click `Save`
9. Scroll down under the description tab and confirm that the Access logs are set as described above.
10. Repeat steps 3-11 for each Load balancer created and used with Elastic Beanstalk environment within the current region.
11. Then repeat the remediation process for all other regions identified in the Audit.

References:

1. https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.managing.elb.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

10.4 Ensure that HTTPS is enabled on load balancer (Manual)

Profile Applicability:

- Level 1

Description:

The simplest way to use HTTPS with an Elastic Beanstalk environment is to assign a server certificate to your environment's load balancer.

Rationale:

When you configure your load balancer to terminate HTTPS, the connection between the client and the load balancer is secure.

Audit:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/elasticbeanstalk
2. On the left hand side click `Environments`
3. Click on the `Environment name` that you want to review
4. Under the "environment_name-env" in the left column click `Configuration`
5. Scroll down under Configurations
6. Under category look for `Load balancer`
7. Click `Edit`
8. Under the `Listeners` section
9. Check the Listeners section for any enabled listeners and make sure the Protocol is set to HTTPS and Enabled.
10. If the Listener is required for HTTP and is not set to HTTPS refer to the remediation below.
11. Repeat steps 3-10 for each environment within the current region.
12. Then repeat the Audit process for all other regions.

Remediation:

From the Console:

1. Login to AWS Console using https://console.aws.amazon.com/elasticbeanstalk
2. On the left hand side click `Environments`
3. Click on the `Environment name` that you want to review
4. Under the "environment_name-env" in the left column click `Configuration`
5. Scroll down under Configurations
6. Under category look for `Load balancer`
7. Click `Edit`
8. Under the `Listeners` section
9. Click `Add listener`

```
Set listener port
Set Listener protocol to HTTPS
Set Instance Port
Sent Instance protocol to HTTPS
Select your SSL certificate
```

10. Click `Add`
11. Make sure it is listed as enabled. If you have other listeners not using HTTPS make sure to turn off enabled
12. Click `Apply` to save the configuration changes.
13. Repeat steps 3-12 for each environment within the current region.
14. Then repeat the remediation for all other regions.

References:

1. https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/configuring-https.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

11 AWS Fargate

AWS Fargate is a serverless, pay-as-you-go compute engine that lets you focus on building applications without managing servers. Moving tasks such as server management, resource allocation, and scaling to AWS does not only improve your operational posture, but also accelerates the process of going from idea to production on the cloud, and lowers the total cost of ownership.

11.1 Ensure customer-managed keys are used to encrypt AWS Fargate ephemeral storage data for Amazon ECS (Automated)

Profile Applicability:

- Level 2

Description:

Use customer-managed AWS KMS keys to encrypt AWS Fargate ephemeral storage data for on Amazon ECS, ensuring that sensitive data remains protected during task execution.

Rationale:

Customer-managed KMS keys offer enhanced control over encryption, including key rotation, access policies, and audit trails.

Impact:

There are costs and configuration overhead associated with setting up and managing customer-managed keys.

Audit:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Ensure that `Fargate ephemeral storage` is not set to `-`.
5. Repeat steps 1-4 for each ECS cluster.

From Command Line:
Run the following command to list clusters:

```
aws ecs list-clusters
```

Run the following command to view the Fargate ephemeral storage KMS key ID configured for a cluster:

```
aws ecs describe-clusters --clusters <cluster-arn> --include CONFIGURATIONS --query
'clusters[*].configuration.managedStorageConfiguration.fargateEphemeralStorageKmsKeyId'
```

Ensure the command returns a customer-managed KMS key ARN.
Repeat for each cluster.

Remediation:

From Console:

1. Login to the ECS console using https://console.aws.amazon.com/ecs/.
2. In the left panel, click `Clusters`.
3. Click the name of a cluster.
4. Click `Update cluster`.
5. Expand the `Encryption` section.
6. Under `Fargate ephemeral storage`, select a customer-managed KMS key.
   Note: Ensure the KMS key has appropriate Fargate service permissions.
7. Click `Update`.
8. Repeat steps 1-7 for each ECS cluster requiring remediation.

Default Value:

AWS Fargate ephemeral storage data is encrypted by default.

References:

1. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/fargate-storage-encryption.html
2. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/fargate-create-storage-key.html
3. https://awscli.amazonaws.com/v2/documentation/api/2.0.33/reference/ecs/list-clusters.html
4. https://awscli.amazonaws.com/v2/documentation/api/2.0.33/reference/ecs/describe-clusters.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

12 AWS Lambda

Serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. Serverless addresses some of today's biggest security concerns as it removes infrastructure management tasks, such as operating system patching, updating binaries, etc. Although the attack surface is reduced compared to non-serverless architectures, the Open Web Application Security Project (OWASP) and application security best practices still apply.

12.1 Ensure AWS Config is Enabled for Lambda and Serverless (Manual)

Profile Applicability:

- Level 2

Description:

With AWS Config, you can track configuration changes to the Lambda functions (including deleted functions), runtime environments, tags, handler name, code size, memory allocation, timeout settings, and concurrency settings, along with Lambda IAM execution role, subnet, and security group associations.

Rationale:

This gives you a holistic view of the Lambda function's lifecycle and enables you to surface that data for potential audit and compliance requirements.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Config` under Management & Governance.
3. This will open up the Config dashboard.
4. Click `Conformance packs`
5. Review the list of conformance packs.
6. If `serverless` is listed or included in the conformance pack you built you meet this recommendation.
7. If `serverless` is not listed refer to the remediation below
8. If none, see remediation section below.
9. Repeat steps 3-7 for all regions used.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Config` under Management & Governance.
3. This will open up the Config dashboard.
4. Click `Conformance packs`
5. Click on `Deploy conformance pack`
6. Click on `Use sample template`
7. Click the down arrow under Sample template
8. Scroll down and click on Operational Best Practices for Serverless
9. Click Next

10. Give it a Conformance pack name `Serverless`.
11. Click Next
12. Click `Deploy conformance pack`
13. Click on `Deploy conformance pack`
14. Click on `Use sample template`
15. Click the down arrow under Sample template
16. Scroll down and click on Security Best Practices for Lambda
17. Click Next
18. Give it a Conformance pack name `LambaSecurity`.
19. Click Next
20. Click `Deploy conformance pack`
21. Repeat steps 2-20 for all regions used.

References:

1. https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

12.2 Ensure Cloudwatch Lambda insights is enabled (Manual)

Profile Applicability:

- Level 1

Description:

Ensure that Amazon CloudWatch Lambda Insights is enabled for your Amazon Lambda functions for enhanced monitoring.

Rationale:

Amazon CloudWatch Lambda Insights allows you to monitor, troubleshoot, and optimize your Lambda functions. The service collects system-level metrics and summarizes diagnostic information to help you identify issues with your Lambda functions and resolve them as soon as possible. CloudWatch Lambda Insights collects system-level metrics and emits a single performance log event for every invocation of that Lambda function.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com/lambda/`
2. Click `Functions`.
3. Click on the name of the function.
4. Click on the `Configuration tab`
5. Click on 'Monitoring and operations tools'.
6. In the Monitoring and operations tools section check the `Enhanced monitoring`
7. If set to Not enabled, refer to the remediation below.
8. Repeat steps 2-7 for each Lambda function within the current region.
9. Then repeat the Audit process for all other regions.

From the Command Line

1. Run `aws lambda list-functions`

```
aws lambda list-functions --output table --query "Functions[*].FunctionName"
```

This command will provide a table titled `ListFunction`

2. Run `aws lambda get-function`

```
aws lambda get-function --function-name "name_of_function" --query
"'Configuration.Layers[*].Arn"

This command should provide the requested ARN

3. If the list of ARNs does not contain the CloudWatch Lambda Insights
extension ARN, i.e. "arn:aws:lambda:<aws-
region>:12345678910:layer:LambdaInsightsExtension:<version>", the Enhanced
Monitoring feature is not enabled.  Refer to the remediation below.
```

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com/lambda/`
2. Click `Functions`.
3. Click on the name of the function.
4. Click on the `Configuration tab`
5. Click on 'Monitoring and operations tools'.
6. In the Monitoring and operations tools section click `Edit` to update the monitoring configuration
7. In the CloudWatch Lambda Insights section click the `Enhanced monitoring` button to enable
   ***Note - When you enable the feature using the AWS Management Console, Amazon Lambda adds the required permissions to your function's execution role.
8. Click Save
9. Repeat steps 2-8 for each Lambda function within the current region that fails the Audit.
10. Then repeat the Audit process for all other regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

12.3 Ensure AWS Secrets manager is configured and being used by Lambda for databases (Manual)

Profile Applicability:

- Level 1

Description:

Lambda functions often have to access a database or other services within your environment.

Rationale:

Credentials used to access databases and other AWS Services need to be managed and regularly rotated to keep access into critical systems secure. Keeping any credentials and manually updating the passwords would be cumbersome, but AWS Secrets Manager allows you to manage and rotate passwords.

Impact:

note - Lambda code should be checked for correct configuration to get the credentials from AWS Secrets Manager. This audit and remediation is only to confirm you have the credentials in Secrets manager.

Audit:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Secrets Manager` under Security, Identity and Compliance.
3. Click on `Secrets`.
4. Review the secrets listed
5. Confirm that the secret required for Lambda functions is included in the list.
6. If it is, review your code and confirm that you are calling the credentials during runtime.
7. If the credentials are not listed refer to the remediation below.
8. Repeat steps 2-7 for all regions used.

Remediation:

From the Console:

1. Login to AWS Console using `https://console.aws.amazon.com`
2. Click `All services`, click `Secrets Manager` under Security, Identity and Compliance.
3. Click on `Secrets`.

4. Click on `Store a new secret`
5. Select the `Secret type`
6. Enter the information

```
For the `3 db types` listed enter the credentials and select the database.
For `other database` enter the credentials, select the db type and enter the
connection parameters.
```

For `Other type of secret` (Lambda) create the keys and values used. - example Username yepyep Password yepyep
choose an encryption key or create a new one
if you add a new key it will take you to the KMS console. Once you create the new key you can then select it here.

7. Click `Next`
8. Give the secret a name associated with your organization style and lambda
9. Click `Next`
10. Configure the auto rotation

```
Rotation schedule leave as default
Select the lambda function you use to rotate the key
```

11. Click `Next`
12. Review all the settings
13. Click `Store`

References:

1. https://aws.amazon.com/blogs/security/how-to-securely-provide-database-credentials-to-lambda-functions-by-using-aws-secrets-manager/
2. https://docs.aws.amazon.com/lambda/latest/dg/welcome.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	2.5 Allowlist Authorized Software Use technical controls, such as application allowlisting, to ensure that only authorized software can execute or be accessed. Reassess bi-annually, or more frequently.		●	●
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	1.7 Deploy Port Level Access Control Utilize port level access control, following 802.1x standards, to control which devices can authenticate to the network. The authentication system shall be tied into the hardware asset inventory data to ensure only authorized devices can connect to the network.		●	●
v7	4.2 Change Default Passwords Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.	●	●	●

12.4 Ensure least privilege is used with Lambda function access (Manual)

Profile Applicability:

- Level 1

Description:

Lambda is fully integrated with IAM, allowing you to control precisely what each Lambda function can do within the AWS Cloud. As you develop a Lambda function, you expand the scope of this policy to enable access to other resources. For example, for a function that processes objects put into an S3 bucket, it requires read access to objects stored in that bucket. Do not grant the function broader permissions to write or delete data, or operate in other buckets.

Rationale:

You can use AWS Identity and Access Management (IAM) to manage access to the Lambda API and resources like functions and layers. For users and applications in your account that use Lambda, you manage permissions in a permissions policy that you can apply to IAM users, groups, or roles. To grant permissions to other accounts or AWS services that use your Lambda resources, you use a policy that applies to the resource itself.

Audit:

Determining the exact permissions required is a manual process and can be challenging, since IAM permissions are very granular and they control access to both the data plane and control plane.
Please refer to the references section below for useful documentation on developing the correct IAM policies for Lambda.

Remediation:

As building out the IAM permissions for Lambda here are some things to consider.

- Set granular IAM permissions for Lambda functions.
- Limit user access via IAM permissions to only necessary resources and operations.
- Remove unused or outdated IAM Users, Roles and Permissions.
- Periodically review and adjust IAM permissions.
- Do not allow all-access permissions for Lambda functions as a short cut."

References:

1. https://docs.aws.amazon.com/service-authorization/latest/reference/reference_policies_actions-resources-contextkeys.html
2. https://awspolicygen.s3.amazonaws.com/policygen.html
3. https://policysim.aws.amazon.com/home/index.jsp?#
4. https://github.com/aws-samples/aws-iamctl/
5. https://docs.aws.amazon.com/lambda/latest/operatorguide/least-privilege-iam.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v8	6.7 Centralize Access Control Centralize access control for all enterprise assets through a directory service or SSO provider, where supported.		●	●
v7	1.7 Deploy Port Level Access Control Utilize port level access control, following 802.1x standards, to control which devices can authenticate to the network. The authentication system shall be tied into the hardware asset inventory data to ensure only authorized devices can connect to the network.		●	●
v7	7.8 Implement DMARC and Enable Receiver-Side Verification To lower the chance of spoofed or modified emails from valid domains, implement Domain-based Message Authentication, Reporting and Conformance (DMARC) policy and verification, starting by implementing the Sender Policy Framework (SPF) and the DomainKeys Identified Mail(DKIM) standards.		●	●

12.5 Ensure every Lambda function has its own IAM Role (Manual)

Profile Applicability:

- Level 1

Description:

Every Lambda function should have a one to one IAM execution role and the roles should not be shared between functions.

Rationale:

The Principle of Least Privilege means that any Lambda function should have the minimal amount of access required to perform its tasks. In order to accomplish this Lambda functions should not share IAM Execution roles.

Audit:

From the Console

1. Login to the AWS console using `https://console.aws.amazon.com/lambda/`
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review.
4. Click the `Configuration` tab
5. Under General configuration on the left column, click `Permissions`.
6. Under the `Execution role` section, `Role name` not the name listed as this is the IAM is the role that defines the access permissions for the selected function.
7. Repeat steps 2 – 6 for all the Lambda functions listed within the AWS region.
8. If any Lambda functions share the same Execution role, refer to the remediation below.
9. Repeat this Audit for all the AWS Regions.

Remediation:

From the Console

1. Login to the AWS console using `https://console.aws.amazon.com/lambda/`
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to change/update.
4. Click the `Configuration` tab
5. Under General configuration on the left column, click `Permissions`.
6. Under the `Execution role` section, click `Edit`.
7. Scroll down to `Execution role`

To use an existing IAM role

```
- Click `Use an existing role`
- Select the role from the `Existing role` dropdown.
- The IAM role can't be associated with another Lambda function and must
follow the Principle of Least Privilege.
```

To use a new IAM role

```
- Click `Create a new role from AWS policy templates`
- Provide a unique name based on company policy in the `Role name`
- Select the policy templates from the `Policy templates` dropdown.
```

8. Click Save
9. Repeat steps 2 – 8 for all the Lambda functions listed within the AWS region that do not have a unique IAM Execution Role.
10. Repeat this remediation process for all the AWS Regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	8.3 Enable Operating System Anti-Exploitation Features/ Deploy Anti-Exploit Technologies Enable anti-exploitation features such as Data Execution Prevention (DEP) or Address Space Layout Randomization (ASLR) that are available in an operating system or deploy appropriate toolkits that can be configured to apply protection to a broader set of applications and executables.		●	●

12.6 Ensure Lambda functions are not exposed to everyone. (Manual)

Profile Applicability:

- Level 1

Description:

A publicly accessible Amazon Lambda function is open to the public and can be reviewed by anyone. To protect against unauthorized users that are sending requests to invoke these functions they need to be changed so they are not exposed to the public.

Rationale:

Allowing anyone to invoke and run your Amazon Lambda functions can lead to data exposure, data loss, and unexpected charges on your AWS bill.

Audit:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. In the left column, click `Permissions`.
6. In the `Resource-based policy` section, click `View policy document`
7. Review the Resource-based policy document box. Find the "Principal" element defined for each policy statement and check the element value. If the element has one of the following values: "*" or { "AWS": "*" }, it means it is set to "Allow", and if it does not contain a "Condition" clause to filter the access, the selected Amazon Lambda function is set to anonymous access.
8. If any of the Lambda functions have anonymous access set refer to the remediation below.
9. Repeat steps 2 – 7 for each Lambda function available within the current AWS region.
10. Repeat this Audit for all the other AWS regions.

From the Command line

1. Run 'aws lambda list-functions'

```
aws lambda list-functions --output table --query "Functions[*].FunctionName"
```

This command will provide a table titles ListFunctions

2. Run `aws lambda get-policy`

```
aws lambda get-policy --function-name "name_of_function" --output text --
query "Policy"
```

This will provide an output of the policy assigned to that function.

3. Find the "Principal" element defined for that function. If the element has one of the following values: "*" or { "AWS": "*" }, it means it is set to "Allow", and if it does not contain a "Condition" clause to filter the access, the selected Amazon Lambda function is set to anonymous access.
4. Make note of the Function name from step 1 and the Statement name from step 2 and refer to the remediation steps below.
5. Repeat steps 1 – 3 for each Lambda function listed within the current region.
6. Repeat this Audit for all the other AWS regions.

Remediation:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. In the left column, click `Permissions`.
6. In the `Resource-based policy` section, perform the following actions:

- Under Policy statements
- Select the policy statement that allows anonymous access
- Click Delete to remove the non-compliant statement from the resource-based policy attached
- Within the Delete statement confirmation box, click Remove
- Click Add permissions to add a new policy statement that grants permissions to a trusted entity only.
- On the Add permissions page configure the new policy statement to grant access to another AWS account, IAM user, IAM role, or to another AWS service.
- Click Save

7. Repeat steps no. 2 – 6 for each Lambda function that fails the Audit above, within the current region.
8. Repeat this Audit for all the other AWS regions.

From the Command line

1. Run 'aws lambda remove-permission'

```
aws lambda remove-permission --function-name "name_of_function" --statement-
id "SID_of_Statement"
```

This command will remove the access policy that is failing the audit for that function.

2. Run `aws lambda add-permission`

```
aws lambda add-permission --function-name "name_of_function" --statement-id
"correctaccess" --principal "012345678910" --action lambda:InvokeFunction
```

This adds a new policy to the function.

***Note The --principal parameter can be the The ID of the trusted AWS account, another AWS account, IAM user, IAM role, or another AWS service.

3. The command output should display the new policy created.
4. Repeat steps 1-2 for each Lambda function from the audit for all regions.

References:

1. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/lambda/index.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.2 Address Unauthorized Assets Ensure that a process exists to address unauthorized assets on a weekly basis. The enterprise may choose to remove the asset from the network, deny the asset from connecting remotely to the network, or quarantine the asset.	●	●	●
v7	1.6 Address Unauthorized Assets Ensure that unauthorized assets are either removed from the network, quarantined or the inventory is updated in a timely manner.	●	●	●
v7	2.6 Address unapproved software Ensure that unauthorized software is either removed or the inventory is updated in a timely manner	●	●	●

12.7 Ensure Lambda functions are referencing active execution roles. (Manual)

Profile Applicability:

- Level 1

Description:

In order to have the necessary permissions to access the AWS cloud services and resources Amazon Lambda functions should be associated with active(available) execution roles.

Rationale:

A Lambda function's execution role is an Identity and Access Management (IAM) role that grants the function permission to process and access specific AWS services and resources. When Amazon Lambda functions are not referencing active execution roles, the functions are losing the ability to perform critical operations securely.

Audit:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. In the left column, click `Permissions`.
6. In the `Resource summary` section, if it reads "The role with name <role_name> cannot be found. (Service: LambdaConsole; Status Code: 404; Error Code: NoSuchEntity; Request ID: e3f12a73-2988-4dd5-b2d1-237c800a27f4; Proxy: null) refer to the remediation below.
7. Repeat steps 2 – 6 for each Lambda function available within the current AWS region.
8. Repeat this Audit for all the other AWS regions.

From the Command line

1. Run `aws lambda list-functions`

```
aws lambda list-functions --output table --query "Functions[*].FunctionName"
```

This command will provide a table titled ListFunctions

2. Run `aws lambda get-function`

```
aws lambda get-function --function-name "name_of_function" --query
"Configuration.Role"
```

This will provide an output returning the role ARN assigned to that function.

3. Run `aws lambda get-role`

```
aws iam get-role --role-name "name_of_role"
```

This will return the requested configuration information.

4. The command output should return the requested configuration information:
5. If the command output returns a `An error occurred (NoSuchEntity) when calling the GetRole operation` error message instead of the role's configuration, the execution role associated with the selected Lambda function is no longer available. Refer to the remediation below.
6. Repeat steps 1-5 for each Lambda function available in the selected AWS region.

Perform the Audit process for other regions.

Remediation:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to update.
4. Click the Configuration tab
5. In the left column, click `Permissions`.
6. In the `Execution role` section, click Edit
7. In the `Edit basic settings` page, perform one of the following actions:

```
- Click Use an existing role if you already a execution role for the selected
Lambda function.
- Select the IAM role from the `Existing role` dropdown list.
- Click Save.
```

Or

```
- Click To create a custom role, go to the `IAM console`.
- Click AWS Service
- Click `Lambda`.
- Click `Next: Permissions
- Attach the permission policies needed
- Click Next: Tags
- Add tags (optional) based on your Organizational policy
- Click Next: Review
- Enter a Role name and a Role description so you can attach the policy to
the Lambda function
- Click `Create role`
- Refresh the Edit basic settings page
- Select the new IAM role you just created from the `Existing role` dropdown
list.
- Click Save.
```

8. Repeat steps 2 – 7 to update the execution role for each misconfigured Amazon Lambda function within the current AWS region.
9. Repeat this Audit for all the other AWS regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	1.7 Deploy Port Level Access Control Utilize port level access control, following 802.1x standards, to control which devices can authenticate to the network. The authentication system shall be tied into the hardware asset inventory data to ensure only authorized devices can connect to the network.		●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●
v7	14.7 Enforce Access Control to Data through Automated Tools Use an automated tool, such as host-based Data Loss Prevention, to enforce access controls to data even when data is copied off a system.			●

12.8 Ensure that Code Signing is enabled for Lambda functions. (Manual)

Profile Applicability:

- Level 1

Description:

Ensure that all your Amazon Lambda functions are configured to use the Code Signing feature in order to restrict the deployment of unverified code.

Rationale:

Code Signing, ensures that the function code is signed by an approved (trusted) source, and that it has not been altered since signing, and that the code signature has not expired or been revoked.

Audit:

From the Console

1. Login to the AWS console using `https://console.aws.amazon.com/lambda/`
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review.
4. Click the `Configuration` tab
5. Under General configuration on the left column, click `Code signing`.
6. Under the `Code signing configuration` section check for any code signing configurations created for the function.
7. If there are no code signing configurations available or listed is not enabled, refer to the remediation
8. Repeat steps 2-7 for each Lambda function within the current region.
9. Then repeat the Audit process for all other regions.

From the Command Line

1. Run `aws lambda list-functions`

```
aws lambda list-functions --output table --query "Functions[*].FunctionName"
```

This command will provide a table titled ListFunctions

2. Run `aws lambda get-function-code-signing-config`

```
aws lambda get-function-code-signing-config --function-name
"name_of_function" --query "CodeSigningConfigArn"

3. The command output should return an array with the requested ARN(s)

4. If the get-function-code-signing-config command output returns null, there
are no code signing configurations for the Lambda function.

5. Refer to the remediation below.

6. Repeat step 2-5 for each Lambda function available in the selected AWS
region.

7. Perform the Audit process for all other regions used.
```

Remediation:

From the Console

1. Login to the AWS console using `https://console.aws.amazon.com/signer`
2. Click on `Create Signing Profile` if none are set up. If you already have some created in the left panel click on `Signing Profiles`, `Create Signing Profile`.
   ***Note a Signing Profile is a trusted publisher and is analogous to the use of a digital signing certificate to generate signatures for your application code.
3. On the `Create Signing Profile` setup page provide

```
Profile name
Specify the Signature Validity period (6 months up to 12 months is
recomended)
```

4. Click on `Create Profile`
5. Go to the Amazon Lambda console https://console.aws.amazon.com/lambda/.
6. In the left panel, under Additional resources, click on `Code signing configurations`.
7. Click on `Create configuration`
8. On the `Create code signing configuration` setup page:

- Description box - provide a short description to identify this configuration
- Click inside the `Signing profile version ARN` box and select the Signing Profile created above.
- For `Signature validation policy`, click the signature validation policy suitable for your Lambda function.
  **Note - A signature check can fail if the code is not signed by an allowed Signing Profile, or if the signature has expired or has been revoked.
- Click Enforce – blocking the deployment of the code and also issue a warning.
- Click `Create configuration`

9. Go to the Amazon Lambda console https://console.aws.amazon.com/lambda/.
10. Click Functions.
11. Under Function name click on the name of the function that you want to review
12. Click the Configuration tab
13. In the left menu click Code signing.
14. Click Edit
15. On the `Edit code signing`, select the code signing configuration created above from the drop down
16. Click `Save`

The Lambda function is now configured to use code signing.

17. Next Upload a signed .zip file or provide an S3 URL of a signed .zip made by a signing job in AWS Signer.
18. To start a signing job, go to AWS Signer console at https://console.aws.amazon.com/signer.
19. In the left panel, click on Signing Jobs.
20. Start a Signing Job to generate a signature for your code package and place the signed code package in the specified destination path.
21. Start Signing Job setup page:

```
- Select the Signing Profile created in dropdown list.
- Code asset source location, specify the Amazon S3 location of the code
package (.zip file) to be signed. Only S3 buckets available in the current
region are displayed and can be used
- Signature destination path with prefix where the signed code package should
be uploaded.
- Start Job to deploy your new Signing Job
- Job status reads Succeeded, you can find the signed .zip package in your
assigned S3 bucket.
```

22. Publish the signed code package to the selected Lambda function.
23. Amazon Lambda will perform signature checks to verify that the code has not been altered since signing
    **Note - The service verifies if the code is signed by one of the allowed signing profiles available.
24. Repeat steps for each Lambda function that was captured in the Audit.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	2.7 Allowlist Authorized Scripts Use technical controls, such as digital signatures and version control, to ensure that only authorized scripts, such as specific .ps1, .py, etc., files, are allowed to execute. Block unauthorized scripts from executing. Reassess bi-annually, or more frequently.			●

Controls Version	Control	IG 1	IG 2	IG 3
v8	10.2 Configure Automatic Anti-Malware Signature Updates Configure automatic updates for anti-malware signature files on all enterprise assets.	●	●	●
v7	5.3 Securely Store Master Images Store the master images and templates on securely configured servers, validated with integrity monitoring tools, to ensure that only authorized changes to the images are possible.		●	●
v7	8.2 Ensure Anti-Malware Software and Signatures are Updated Ensure that the organization's anti-malware software updates its scanning engine and signature database on a regular basis.	●	●	●

12.9 Ensure there are no Lambda functions with admin privileges within your AWS account (Manual)

Profile Applicability:

- Level 1

Description:

Ensure that your Amazon Lambda functions don't have administrative permissions potentially giving the function access to all AWS cloud services and resources.

Rationale:

In order to promote the Principle of Least Privilege (POLP) and provide your functions the minimal amount of access required to perform their tasks the right IAM execution role associated with the function should be used. Instead of providing administrative permissions you should grant the role the necessary permissions that the function really needs.

Audit:

From the Console

1. Login in to the AWS Console using `https://console.aws.amazon.com/lambda/`
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. Click on `Permissions` in the left column.
6. In the Execution role section, click the `Role name` to access the IAM role details. **Note this will bring you to the IAM Console.
7. Select the Permissions tab to view the identity–based policies attached
8. In the Permissions policies section click on the Policy name.
9. Select the Permissions tab.
   **Note The policy summary should show below in JSON format.
10. Within the {} JSON policy, identify the "Action" element defined for each statement and check the value.
11. If any of the "Action" element values are set to "*" and the "Effect" element is set to "Allow", the role policy provides access to all the supported AWS cloud services and resources.
12. Repeat this step for each IAM policy attached to the selected execution role.

If one or more policies allow access to all AWS services and resources, the execution role provides administrative permissions. Refer to the remediation below.
Repeat steps for each Lambda function within the current region.
Then repeat the Audit process for all other regions.

Remediation:

From the Console

1. Login in to the AWS Console using `https://console.aws.amazon.com/lambda/`
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to remediate
4. Click the Configuration tab
5. Click on `Permissions` in the left column.
6. In the Execution role section, click the `Edit`
7. Edit basic settings configuration page:

```
- associate the function with an existing, compliant IAM role
- click Use an existing role from the Execution role
- select the required role from the Existing role dropdown
- click Save
```

OR

```
- apply a new execution role to your Lambda function
- click Create a new role from AWS policy templates
- Provide a name for the new role based on org policy
- select only the necessary permission set(s) from the Policy templates -
optional dropdown list.
- click Save
```

8. Repeat steps for each Lambda function within the current region that failed the Audit.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●
v7	1.7 Deploy Port Level Access Control Utilize port level access control, following 802.1x standards, to control which devices can authenticate to the network. The authentication system shall be tied into the hardware asset inventory data to ensure only authorized devices can connect to the network.		●	●

12.10 Ensure Lambda functions do not allow unknown cross account access via permission policies. (Manual)

Profile Applicability:

- Level 1

Description:

Ensure that all your Amazon Lambda functions are configured to allow access only to trusted AWS accounts in order to protect against unauthorized cross-account access.

Rationale:

Allowing unknown (unauthorized) AWS accounts to invoke your Amazon Lambda functions can lead to data exposure and data loss. To prevent any unauthorized invocation requests for your Lambda functions, restrict access only to trusted AWS accounts.

Audit:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. In the left column, click `Permissions`.
6. In the `Resource-based policy statements` section, click `View policy document`
7. Review the Resource-based policy document box. Find the "Principal" element and check the element value (ARN).
8. Confirm that each AWS account ARN is an approved AWS account. If one or more of the ARNs is not an AWS account defined within your organization, refer to the remediation below.
9. Repeat steps no. 2-8 for each Lambda function available within the current AWS region.
10. Repeat this Audit for all the other AWS regions.

From the Command Line

1 Run `aws lambda list-functions`

```
aws lambda list-functions --output table --query "Functions[*].FunctionName"
```

2 This command will provide a table titled ListFunctions
3 Run `aws lambda get-policy` on the functions listed

```
aws lambda get-policy --function-name "name_of_function" --output text --
query "Policy"
```

4. This will provide an output of the policy assigned to that function.
5. Identify the "Principal" element for each function for the ARN.
6. Confirm that each AWS account ARN is an approved AWS account. If one or more of the ARNs is not an AWS account defined within your organization, refer to the remediation below.
7. Repeat steps 2–5 for each Lambda function available.
8. Run the Audit in the other AWS cloud regions

Remediation:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. In the left column, click `Permissions`.
6. In the `Resource-based policy statements` section, select the policy statement that allows the unknown AWS Account cross-account access
7. Click Edit
8. On the `Edit permissions` page, replace or remove the AWS Account(s) ARN of the unauthorized principal in the Principal box
9. Click Save
10. Repeat steps for each Lambda function that failed the Audit

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●
v7	1.7 Deploy Port Level Access Control Utilize port level access control, following 802.1x standards, to control which devices can authenticate to the network. The authentication system shall be tied into the hardware asset inventory data to ensure only authorized devices can connect to the network.		●	●

12.11 Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates. (Manual)

Profile Applicability:

- Level 1

Description:

Always using a recent version of the execution environment configured for your Amazon Lambda functions adheres to best practices for the newest software features, the latest security patches and bug fixes, and performance and reliability.

Rationale:

When you execute your Lambda functions using recent versions of the implemented runtime environment, you should benefit from new features and enhancements, better security, along with performance and reliability.

Audit:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click Code tab
5. In the Runtime settings section, check the Runtime attribute value to determine the runtime version.
6. Compare the function runtime with the updated list of Amazon Lambda runtimes. Link is in the resource section
7. If the version you are using is not the latest or is on the EOL list, the selected Amazon Lambda function is using an old and deprecated runtime environment.
8. Refer to the remediation below.
9. Repeat steps 2-6 for each Lambda function within the current region.

Then repeat the Audit process for all other regions.

From the Command Line

1. Run `aws lambda list-functions`

```
aws lambda list-functions --output table --query 'Functions[*].FunctionName'
```

This command will provide a table titled ListFunctions

2. Run `aws lambda get-function-configuration` using the Function names returned in the table.

```
aws lambda get-function-configuration --function-name "name_of_fuunction" --
query 'Runtime'
```

3. The command output should return the execution environment
4. Compare the function runtime with the updated list of Amazon Lambda runtimes. Link is in the resource section
5. If the version you are using is not the latest or is on the EOL list, the selected Amazon Lambda function is using an old and deprecated runtime environment.
6. Refer to the remediation below.

Remediation:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click Code tab
5. Go to the Runtime settings section.
6. Click Edit
7. On the Edit runtime settings page, select the latest supported version of the runtime environment from the dropdown list.
   **Note - make sure the correct architecture is also selected.
8. Click Save
9. Select the Code tab
10. Click Test from the Code source section.
11. Once the testing is completed, the execution result of your Lambda function will be listed
12. Repeat steps for each Lambda function that failed the Audit within the current region.

From the Command Line

1. Run `aws lambda update-function-configuration` using the name of the Function you need to remediate

```
aws lambda update-function-configuration --output table --query
'Functions[*].FunctionName'
```

This command will provide a table titled ListFunctions

2. Run `aws lambda get-function-configuration` using the Function names returned in the table.

```
aws lambda get-function-configuration --function-name "name_of_fuunction" --
function-name "name_of_function" --runtime "python3.9"
```

3. The command output should return the metadata available for the reconfigured function.
4. Repeat steps 1-2 to upgrade the runtime environment for each Amazon Lambda function found in the Audit.

References:

1. https://docs.aws.amazon.com/lambda/latest/dg/lambda-runtimes.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	7.4 Perform Automated Application Patch Management Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.	●	●	●
v7	3.5 Deploy Automated Software Patch Management Tools Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.	●	●	●

12.12 Ensure encryption in transit is enabled for Lambda environment variables (Manual)

Profile Applicability:

- Level 1

Description:

As you can set your own environmental variables for Lambda it is important to also encrypt them for in transit protection.

Rationale:

Lambda environment variables should be encrypted in transit for client-side protection as they can store sensitive information.

Audit:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. In the left column, click `Environment variables`.
6. In the `Environment variables` section, click `Edit`
7. On the Edit environment variables page, review the Values. If they are a long value that resembles this:
   AQICAHhxbKJYcFAU16CbU4IVpzi5CwK
   Encryption is in place for that Key. If the value is in plain text refer to the remediation below.
8. Repeat steps 2 – 7 for each Lambda function available within the current AWS region.
9. Repeat this Audit for all the other AWS regions.

From the Command line

1. Run 'aws lambda list-functions'

```
aws lambda list-functions --output table --query "Functions[*].FunctionName"
```

This command will provide a table titled `ListFunctions`

2. Run `aws lambda get-function`

```
aws lambda get-function --function-name "name_of_function" --query
"Configuration.Environment"
```

This will provide an output of the environment variables created for that function.

3. Review the Values in the table. If they contain a long value that resembles this: AQICAHhxbKJYcFAU16CbU4IVpzi5CwK. Encryption is in place for that Key. If the value is in plain text refer to the remediation below.
4. Repeat steps 1 – 3 for each Lambda function listed within the current region.
5. Repeat this Audit for all the other AWS regions.

Remediation:

From the Console

1. Login to the AWS Console using `https://console.aws.amazon.com/lambda/`.
2. In the left column, under `AWS Lambda`, click `Functions`.
3. Under `Function name` click on the name of the function that you want to review
4. Click the Configuration tab
5. In the left column, click `Environment variables`.
6. In the `Environment variables` section, click `Edit`
7. Click the check box for `Enable helpers for encryption in transit`
8. Click the `Encrypt` option for all the variable that need to be encrypted.
9. Repeat steps 2 – 8 for each Lambda function identified in the Audit within the current AWS region.
10. Repeat this remediation for all the other AWS regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	10.4 Ensure Protection of Backups Ensure that backups are properly protected via physical security or encryption when they are stored, as well as when they are moved across the network. This includes remote backups and cloud services.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	12.11 Require All Remote Login to Use Multi-factor Authentication Require all remote login access to the organization's network to encrypt data in transit and use multi-factor authentication.		●	●

13 AWS Local Zones

AWS Local Zones enables you to run applications that require single-digit millisecond latency or local data processing by bringing AWS infrastructure closer to your end users and business centers. Assists in meeting data residency requirements for regulatory and compliance-sensitive workloads.

14 AWS Outposts

AWS Outposts brings native AWS services, infrastructure, and operating models to virtually any data center, co-location space, or on-premises facility.

At this time all Security Best practices for AWS Outposts fall under the AWS Foundations Benchmark and/or Service Specific CIS Benchmarks that are available. There are no specific Security best practice recommendations related for the AWS Outposts itself, but that does not alleviate the end user of this Service of the Shared responsibility model and the customer security requirements for service and data access and protection.

15 Serverless Application Repository

The AWS Serverless Application Repository is a managed repository for serverless applications. It enables teams, organizations, and individual developers to store and share reusable applications, and easily assemble and deploy serverless architectures in powerful new ways. Using the Serverless Application Repository, you don't need to clone, build, package, or publish source code to AWS before deploying it. Instead, you can use pre-built applications from the Serverless Application Repository in your serverless architectures, helping you and your teams reduce duplicated work and ensure organizational best practices.

At this time all Security Best practices for serverless applications fall under Lamba or other code building services within AWS. So Best practices should be considered when developing those serverless applications prior to adding them to the repository. There are no specific Security best practice recommendations related for the AWS Serverless Application Repository itself, but that does not alleviate the end user of this Service of the Shared responsibility model and the customer security requirements for service and data access and protection.

16 AWS SimSpace Weaver

AWS SimSpace Weaver is a service used to build and run dynamic, large-scale spatial simulations, such as city-scale digital twins and crowd simulations with millions of people and objects.

As part of the simulation you will often have Applications and Clients functioning as part of the simulation. Some of the best practice recommendations will be for the communication between these applications and clients as a separate function outside of SimSpace Weaver. There are no specific Security best practice recommendations related for SimSapce Weaver, but that does not alleviate the end user of this service of the Shared responsibility model and the customer security requirements for service and data access and protection.

16.1 Ensure communications between your applications and clients is encrypted. (Manual)

Profile Applicability:

- Level 1

Description:

SimSpace Weaver doesn't manage communications between your apps and the clients.

Rationale:

Be sure to implement some form of authentication and encryption for all client sessions while using SimSpace Weaver.

Audit:

There is no setting for encryption setup for your clients and applications within SimSpace Weaver service. For this audit you have to confirm that the communication is configured in the app and the client with encryption to protect that traffic.

Remediation:

Confirm that the communication you have configured between you application and clients that run inside of SimSpace Weaver are encrypted.

References:

1. https://docs.aws.amazon.com/simspaceweaver/latest/userguide/security_best-practices.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

17 EC2 Image Builder

EC2 Image Builder is a fully-managed service that makes it easy to build, customize and deploy OS images without writing scripts. The image pipeline in Image Builder defines all aspects of the process to customize images. It consists of the image recipe, infrastructure configuration, distribution, and test settings. Image Builder significantly reduces the effort of keeping images up-to-date and secure by providing a simple graphical interface, built-in automation, and AWS-provided security settings.

There are no specific Security best practice recommendations related for EC2 Image Builder itself, but that does not alleviate the end user of this service of the Shared responsibility model and the customer security requirements for service and data access and protection.

Appendix: Summary Table

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1	**Introduction**		
2	**Amazon Elastic Cloud Compute (EC2)**		
2.1	**Amazon Machine Images (AMI)**		
2.1.1	Ensure Consistent Naming Convention is used for Organizational AMI (Manual)	☐	☐
2.1.2	Ensure Amazon Machine Images (AMIs) are encrypted (Automated)	☐	☐
2.1.3	Ensure Only Approved Amazon Machine Images (AMIs) are Used (Manual)	☐	☐
2.1.4	Ensure Images (AMI) are not older than 90 days (Automated)	☐	☐
2.1.5	Ensure Images are not Publicly Available (Manual)	☐	☐
2.2	**Elastic Block Storage (EBS)**		
2.2.1	Ensure EBS volume encryption is enabled (Automated)	☐	☐
2.2.2	Ensure Public Access to EBS Snapshots is Disabled (Automated)	☐	☐
2.2.3	Ensure EBS volume snapshots are encrypted (Automated)	☐	☐
2.2.4	Ensure unused EBS volumes are removed (Manual)	☐	☐
2.3	Ensure Tag Policies are Enabled (Manual)	☐	☐
2.4	Ensure an Organizational EC2 Tag Policy has been Created (Manual)	☐	☐
2.5	Ensure no AWS EC2 Instances are Older than 180 days (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
2.6	Ensure detailed monitoring is enable for production EC2 Instances (Manual)	☐	☐
2.7	Ensure Default EC2 Security groups are not being used. (Manual)	☐	☐
2.8	Ensure the Use of IMDSv2 is Enforced on All Existing Instances (Manual)	☐	☐
2.9	Ensure use of AWS Systems Manager to manage EC2 instances (Manual)	☐	☐
2.10	Ensure unused ENIs are removed (Manual)	☐	☐
2.11	Ensure instances stopped for over 90 days are removed (Manual)	☐	☐
2.12	Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination (Manual)	☐	☐
2.13	Ensure Secrets and Sensitive Data are not stored directly in EC2 User Data (Manual)	☐	☐
2.14	Ensure EC2 Auto Scaling Groups Propagate Tags to EC2 Instances that it launches (Automated)	☐	☐
3	**Amazon Elastic Container Service (ECS)**		
3.1	Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host (Automated)	☐	☐
3.2	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services (Automated)	☐	☐
3.3	Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host' (Automated)	☐	☐
3.4	Ensure Amazon ECS task definitions do not have 'privileged' set to 'true' (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.5	Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions (Automated)	☐	☐
3.6	Ensure secrets are not passed as container environment variables in Amazon ECS task definitions (Automated)	☐	☐
3.7	Ensure logging is configured for Amazon ECS task definitions (Automated)	☐	☐
3.8	Ensure Amazon ECS Fargate services are using the latest Fargate platform version (Automated)	☐	☐
3.9	Ensure monitoring is enabled for Amazon ECS clusters (Automated)	☐	☐
3.10	Ensure Amazon ECS services are tagged (Automated)	☐	☐
3.11	Ensure Amazon ECS clusters are tagged (Automated)	☐	☐
3.12	Ensure Amazon ECS task definitions are tagged (Automated)	☐	☐
3.13	Ensure only trusted images are used with Amazon ECS (Automated)	☐	☐
3.14	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets (Automated)	☐	☐
4	**Amazon Elastic Kubernetes Service (EKS) (Reference)**		
5	**Amazon Lightsail**		
5.1	Apply updates to any apps running in Lightsail (Manual)	☐	☐
5.2	Change default Administrator login names and passwords for applications (Manual)	☐	☐
5.3	Disable SSH and RDP ports for Lightsail instances when not needed. (Manual)	☐	☐
5.4	Ensure SSH is restricted to only IP address that should have this access. (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
5.5	Ensure RDP is restricted to only IP address that should have this access. (Manual)	☐	☐
5.6	Disable IPv6 Networking if not in use within your organization. (Manual)	☐	☐
5.7	Ensure you are using an IAM policy to manage access to buckets in Lightsail. (Manual)	☐	☐
5.8	Ensure Lightsail instances are attached to the buckets (Manual)	☐	☐
5.9	Ensure that your Lightsail buckets are not publicly accessible (Manual)	☐	☐
5.10	Enable storage bucket access logging (Manual)	☐	☐
5.11	Ensure your Windows Server based lightsail instances are updated with the latest security patches. (Manual)	☐	☐
5.12	Change the auto-generated password for Windows based instances. (Manual)	☐	☐
6	**AWS App Runner**		
6.1	Ensure you are using VPC Endpoints for source code access (Manual)	☐	☐
7	**AWS Auto Scaling**		
8	**AWS Batch**		
8.1	Ensure AWS Batch is configured with AWS Cloudwatch Logs. (Manual)	☐	☐
8.2	Ensure Batch roles are configured for cross-service confused deputy prevention (Manual)	☐	☐
9	**AWS Compute Optimizer**		
10	**Elastic Beanstalk**		

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
10.1	Ensure Managed Platform updates is configured (Manual)	☐	☐
10.2	Ensure Persistent logs is setup and configured to S3 (Manual)	☐	☐
10.3	Ensure access logs are enabled. (Manual)	☐	☐
10.4	Ensure that HTTPS is enabled on load balancer (Manual)	☐	☐
11	**AWS Fargate**		
11.1	Ensure customer-managed keys are used to encrypt AWS Fargate ephemeral storage data for Amazon ECS (Automated)	☐	☐
12	**AWS Lambda**		
12.1	Ensure AWS Config is Enabled for Lambda and Serverless (Manual)	☐	☐
12.2	Ensure Cloudwatch Lambda insights is enabled (Manual)	☐	☐
12.3	Ensure AWS Secrets manager is configured and being used by Lambda for databases (Manual)	☐	☐
12.4	Ensure least privilege is used with Lambda function access (Manual)	☐	☐
12.5	Ensure every Lambda function has its own IAM Role (Manual)	☐	☐
12.6	Ensure Lambda functions are not exposed to everyone. (Manual)	☐	☐
12.7	Ensure Lambda functions are referencing active execution roles. (Manual)	☐	☐
12.8	Ensure that Code Signing is enabled for Lambda functions. (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
12.9	Ensure there are no Lambda functions with admin privileges within your AWS account (Manual)	☐	☐
12.10	Ensure Lambda functions do not allow unknown cross account access via permission policies. (Manual)	☐	☐
12.11	Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates. (Manual)	☐	☐
12.12	Ensure encryption in transit is enabled for Lambda environment variables (Manual)	☐	☐
13	**AWS Local Zones**		
14	**AWS Outposts**		
15	**Serverless Application Repository**		
16	**AWS SimSpace Weaver**		
16.1	Ensure communications between your applications and clients is encrypted. (Manual)	☐	☐
17	**EC2 Image Builder**		

Appendix: CIS Controls v7 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
2.1.1	Ensure Consistent Naming Convention is used for Organizational AMI	☐	☐
2.1.3	Ensure Only Approved Amazon Machine Images (AMIs) are Used	☐	☐
2.1.4	Ensure Images (AMI) are not older than 90 days	☐	☐
2.2.2	Ensure Public Access to EBS Snapshots is Disabled	☐	☐
2.2.4	Ensure unused EBS volumes are removed	☐	☐
2.5	Ensure no AWS EC2 Instances are Older than 180 days	☐	☐
2.6	Ensure detailed monitoring is enable for production EC2 Instances	☐	☐
2.7	Ensure Default EC2 Security groups are not being used.	☐	☐
2.8	Ensure the Use of IMDSv2 is Enforced on All Existing Instances	☐	☐
2.9	Ensure use of AWS Systems Manager to manage EC2 instances	☐	☐
2.11	Ensure instances stopped for over 90 days are removed	☐	☐
2.12	Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination	☐	☐
3.1	Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host	☐	☐
3.2	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services	☐	☐
3.3	Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host'	☐	☐
3.4	Ensure Amazon ECS task definitions do not have 'privileged' set to 'true'	☐	☐
3.5	Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions	☐	☐

Recommendation		Set Correctly	
		Yes	No
3.7	Ensure logging is configured for Amazon ECS task definitions	☐	☐
3.8	Ensure Amazon ECS Fargate services are using the latest Fargate platform version	☐	☐
3.10	Ensure Amazon ECS services are tagged	☐	☐
3.11	Ensure Amazon ECS clusters are tagged	☐	☐
3.12	Ensure Amazon ECS task definitions are tagged	☐	☐
3.13	Ensure only trusted images are used with Amazon ECS	☐	☐
3.14	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets	☐	☐
5.1	Apply updates to any apps running in Lightsail	☐	☐
5.2	Change default Administrator login names and passwords for applications	☐	☐
5.4	Ensure SSH is restricted to only IP address that should have this access.	☐	☐
5.5	Ensure RDP is restricted to only IP address that should have this access.	☐	☐
5.7	Ensure you are using an IAM policy to manage access to buckets in Lightsail.	☐	☐
5.8	Ensure Lightsail instances are attached to the buckets	☐	☐
5.9	Ensure that your Lightsail buckets are not publicly accessible	☐	☐
5.10	Enable storage bucket access logging	☐	☐
5.11	Ensure your Windows Server based lightsail instances are updated with the latest security patches.	☐	☐
5.12	Change the auto-generated password for Windows based instances.	☐	☐
6.1	Ensure you are using VPC Endpoints for source code access	☐	☐
8.1	Ensure AWS Batch is configured with AWS Cloudwatch Logs.	☐	☐
10.1	Ensure Managed Platform updates is configured	☐	☐
10.2	Ensure Persistent logs is setup and configured to S3	☐	☐
10.3	Ensure access logs are enabled.	☐	☐

Recommendation		Set Correctly	
		Yes	No
12.1	Ensure AWS Config is Enabled for Lambda and Serverless	☐	☐
12.2	Ensure Cloudwatch Lambda insights is enabled	☐	☐
12.3	Ensure AWS Secrets manager is configured and being used by Lambda for databases	☐	☐
12.6	Ensure Lambda functions are not exposed to everyone.	☐	☐
12.7	Ensure Lambda functions are referencing active execution roles.	☐	☐
12.8	Ensure that Code Signing is enabled for Lambda functions.	☐	☐
12.11	Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates.	☐	☐
12.12	Ensure encryption in transit is enabled for Lambda environment variables	☐	☐

Appendix: CIS Controls v7 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
2.1.1	Ensure Consistent Naming Convention is used for Organizational AMI	☐	☐
2.1.3	Ensure Only Approved Amazon Machine Images (AMIs) are Used	☐	☐
2.1.4	Ensure Images (AMI) are not older than 90 days	☐	☐
2.1.5	Ensure Images are not Publicly Available	☐	☐
2.2.2	Ensure Public Access to EBS Snapshots is Disabled	☐	☐
2.2.4	Ensure unused EBS volumes are removed	☐	☐
2.3	Ensure Tag Policies are Enabled	☐	☐
2.4	Ensure an Organizational EC2 Tag Policy has been Created	☐	☐
2.5	Ensure no AWS EC2 Instances are Older than 180 days	☐	☐
2.6	Ensure detailed monitoring is enable for production EC2 Instances	☐	☐
2.7	Ensure Default EC2 Security groups are not being used.	☐	☐
2.8	Ensure the Use of IMDSv2 is Enforced on All Existing Instances	☐	☐
2.9	Ensure use of AWS Systems Manager to manage EC2 instances	☐	☐
2.10	Ensure unused ENIs are removed	☐	☐
2.11	Ensure instances stopped for over 90 days are removed	☐	☐
2.12	Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination	☐	☐
2.13	Ensure Secrets and Sensitive Data are not stored directly in EC2 User Data	☐	☐
2.14	Ensure EC2 Auto Scaling Groups Propagate Tags to EC2 Instances that it launches	☐	☐
3.1	Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host	☐	☐

Recommendation		Set Correctly	
		Yes	No
3.2	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services	☐	☐
3.3	Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host'	☐	☐
3.4	Ensure Amazon ECS task definitions do not have 'privileged' set to 'true'	☐	☐
3.5	Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions	☐	☐
3.6	Ensure secrets are not passed as container environment variables in Amazon ECS task definitions	☐	☐
3.7	Ensure logging is configured for Amazon ECS task definitions	☐	☐
3.8	Ensure Amazon ECS Fargate services are using the latest Fargate platform version	☐	☐
3.9	Ensure monitoring is enabled for Amazon ECS clusters	☐	☐
3.10	Ensure Amazon ECS services are tagged	☐	☐
3.11	Ensure Amazon ECS clusters are tagged	☐	☐
3.12	Ensure Amazon ECS task definitions are tagged	☐	☐
3.13	Ensure only trusted images are used with Amazon ECS	☐	☐
3.14	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets	☐	☐
5.1	Apply updates to any apps running in Lightsail	☐	☐
5.2	Change default Administrator login names and passwords for applications	☐	☐
5.3	Disable SSH and RDP ports for Lightsail instances when not needed.	☐	☐
5.4	Ensure SSH is restricted to only IP address that should have this access.	☐	☐
5.5	Ensure RDP is restricted to only IP address that should have this access.	☐	☐
5.6	Disable IPv6 Networking if not in use within your organization.	☐	☐
5.7	Ensure you are using an IAM policy to manage access to buckets in Lightsail.	☐	☐
5.8	Ensure Lightsail instances are attached to the buckets	☐	☐

Recommendation		Set Correctly	
		Yes	No
5.9	Ensure that your Lightsail buckets are not publicly accessible	☐	☐
5.10	Enable storage bucket access logging	☐	☐
5.11	Ensure your Windows Server based lightsail instances are updated with the latest security patches.	☐	☐
5.12	Change the auto-generated password for Windows based instances.	☐	☐
6.1	Ensure you are using VPC Endpoints for source code access	☐	☐
8.1	Ensure AWS Batch is configured with AWS Cloudwatch Logs.	☐	☐
10.1	Ensure Managed Platform updates is configured	☐	☐
10.2	Ensure Persistent logs is setup and configured to S3	☐	☐
10.3	Ensure access logs are enabled.	☐	☐
10.4	Ensure that HTTPS is enabled on load balancer	☐	☐
12.1	Ensure AWS Config is Enabled for Lambda and Serverless	☐	☐
12.2	Ensure Cloudwatch Lambda insights is enabled	☐	☐
12.3	Ensure AWS Secrets manager is configured and being used by Lambda for databases	☐	☐
12.4	Ensure least privilege is used with Lambda function access	☐	☐
12.5	Ensure every Lambda function has its own IAM Role	☐	☐
12.6	Ensure Lambda functions are not exposed to everyone.	☐	☐
12.7	Ensure Lambda functions are referencing active execution roles.	☐	☐
12.8	Ensure that Code Signing is enabled for Lambda functions.	☐	☐
12.9	Ensure there are no Lambda functions with admin privileges within your AWS account	☐	☐
12.10	Ensure Lambda functions do not allow unknown cross account access via permission policies.	☐	☐
12.11	Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates.	☐	☐

Recommendation		Set Correctly	
		Yes	No
12.12	Ensure encryption in transit is enabled for Lambda environment variables	☐	☐
16.1	Ensure communications between your applications and clients is encrypted.	☐	☐

Appendix: CIS Controls v7 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
2.1.1	Ensure Consistent Naming Convention is used for Organizational AMI	☐	☐
2.1.2	Ensure Amazon Machine Images (AMIs) are encrypted	☐	☐
2.1.3	Ensure Only Approved Amazon Machine Images (AMIs) are Used	☐	☐
2.1.4	Ensure Images (AMI) are not older than 90 days	☐	☐
2.1.5	Ensure Images are not Publicly Available	☐	☐
2.2.1	Ensure EBS volume encryption is enabled	☐	☐
2.2.2	Ensure Public Access to EBS Snapshots is Disabled	☐	☐
2.2.3	Ensure EBS volume snapshots are encrypted	☐	☐
2.2.4	Ensure unused EBS volumes are removed	☐	☐
2.3	Ensure Tag Policies are Enabled	☐	☐
2.4	Ensure an Organizational EC2 Tag Policy has been Created	☐	☐
2.5	Ensure no AWS EC2 Instances are Older than 180 days	☐	☐
2.6	Ensure detailed monitoring is enable for production EC2 Instances	☐	☐
2.7	Ensure Default EC2 Security groups are not being used.	☐	☐
2.8	Ensure the Use of IMDSv2 is Enforced on All Existing Instances	☐	☐
2.9	Ensure use of AWS Systems Manager to manage EC2 instances	☐	☐
2.10	Ensure unused ENIs are removed	☐	☐
2.11	Ensure instances stopped for over 90 days are removed	☐	☐
2.12	Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination	☐	☐
2.13	Ensure Secrets and Sensitive Data are not stored directly in EC2 User Data	☐	☐

Recommendation		Set Correctly	
		Yes	No
2.14	Ensure EC2 Auto Scaling Groups Propagate Tags to EC2 Instances that it launches	☐	☐
3.1	Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host	☐	☐
3.2	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services	☐	☐
3.3	Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host'	☐	☐
3.4	Ensure Amazon ECS task definitions do not have 'privileged' set to 'true'	☐	☐
3.5	Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions	☐	☐
3.6	Ensure secrets are not passed as container environment variables in Amazon ECS task definitions	☐	☐
3.7	Ensure logging is configured for Amazon ECS task definitions	☐	☐
3.8	Ensure Amazon ECS Fargate services are using the latest Fargate platform version	☐	☐
3.9	Ensure monitoring is enabled for Amazon ECS clusters	☐	☐
3.10	Ensure Amazon ECS services are tagged	☐	☐
3.11	Ensure Amazon ECS clusters are tagged	☐	☐
3.12	Ensure Amazon ECS task definitions are tagged	☐	☐
3.13	Ensure only trusted images are used with Amazon ECS	☐	☐
3.14	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets	☐	☐
5.1	Apply updates to any apps running in Lightsail	☐	☐
5.2	Change default Administrator login names and passwords for applications	☐	☐
5.3	Disable SSH and RDP ports for Lightsail instances when not needed.	☐	☐
5.4	Ensure SSH is restricted to only IP address that should have this access.	☐	☐
5.5	Ensure RDP is restricted to only IP address that should have this access.	☐	☐

Recommendation		Set Correctly	
		Yes	No
5.6	Disable IPv6 Networking if not in use within your organization.	☐	☐
5.7	Ensure you are using an IAM policy to manage access to buckets in Lightsail.	☐	☐
5.8	Ensure Lightsail instances are attached to the buckets	☐	☐
5.9	Ensure that your Lightsail buckets are not publicly accessible	☐	☐
5.10	Enable storage bucket access logging	☐	☐
5.11	Ensure your Windows Server based lightsail instances are updated with the latest security patches.	☐	☐
5.12	Change the auto-generated password for Windows based instances.	☐	☐
6.1	Ensure you are using VPC Endpoints for source code access	☐	☐
8.1	Ensure AWS Batch is configured with AWS Cloudwatch Logs.	☐	☐
10.1	Ensure Managed Platform updates is configured	☐	☐
10.2	Ensure Persistent logs is setup and configured to S3	☐	☐
10.3	Ensure access logs are enabled.	☐	☐
10.4	Ensure that HTTPS is enabled on load balancer	☐	☐
11.1	Ensure customer-managed keys are used to encrypt AWS Fargate ephemeral storage data for Amazon ECS	☐	☐
12.1	Ensure AWS Config is Enabled for Lambda and Serverless	☐	☐
12.2	Ensure Cloudwatch Lambda insights is enabled	☐	☐
12.3	Ensure AWS Secrets manager is configured and being used by Lambda for databases	☐	☐
12.4	Ensure least privilege is used with Lambda function access	☐	☐
12.5	Ensure every Lambda function has its own IAM Role	☐	☐
12.6	Ensure Lambda functions are not exposed to everyone.	☐	☐
12.7	Ensure Lambda functions are referencing active execution roles.	☐	☐
12.8	Ensure that Code Signing is enabled for Lambda functions.	☐	☐

Recommendation		Set Correctly	
		Yes	No
12.9	Ensure there are no Lambda functions with admin privileges within your AWS account	☐	☐
12.10	Ensure Lambda functions do not allow unknown cross account access via permission policies.	☐	☐
12.11	Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates.	☐	☐
12.12	Ensure encryption in transit is enabled for Lambda environment variables	☐	☐
16.1	Ensure communications between your applications and clients is encrypted.	☐	☐

Appendix: CIS Controls v7 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
8.2	Ensure Batch roles are configured for cross-service confused deputy prevention	☐	☐

Appendix: CIS Controls v8 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
2.1.1	Ensure Consistent Naming Convention is used for Organizational AMI	☐	☐
2.1.3	Ensure Only Approved Amazon Machine Images (AMIs) are Used	☐	☐
2.1.4	Ensure Images (AMI) are not older than 90 days	☐	☐
2.1.5	Ensure Images are not Publicly Available	☐	☐
2.2.2	Ensure Public Access to EBS Snapshots is Disabled	☐	☐
2.2.4	Ensure unused EBS volumes are removed	☐	☐
2.3	Ensure Tag Policies are Enabled	☐	☐
2.4	Ensure an Organizational EC2 Tag Policy has been Created	☐	☐
2.5	Ensure no AWS EC2 Instances are Older than 180 days	☐	☐
2.6	Ensure detailed monitoring is enable for production EC2 Instances	☐	☐
2.7	Ensure Default EC2 Security groups are not being used.	☐	☐
2.8	Ensure the Use of IMDSv2 is Enforced on All Existing Instances	☐	☐
2.9	Ensure use of AWS Systems Manager to manage EC2 instances	☐	☐
2.11	Ensure instances stopped for over 90 days are removed	☐	☐
2.12	Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination	☐	☐
2.13	Ensure Secrets and Sensitive Data are not stored directly in EC2 User Data	☐	☐
2.14	Ensure EC2 Auto Scaling Groups Propagate Tags to EC2 Instances that it launches	☐	☐
3.1	Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host	☐	☐

Recommendation		Set Correctly	
		Yes	No
3.2	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services	☐	☐
3.3	Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host'	☐	☐
3.4	Ensure Amazon ECS task definitions do not have 'privileged' set to 'true'	☐	☐
3.5	Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions	☐	☐
3.6	Ensure secrets are not passed as container environment variables in Amazon ECS task definitions	☐	☐
3.7	Ensure logging is configured for Amazon ECS task definitions	☐	☐
3.8	Ensure Amazon ECS Fargate services are using the latest Fargate platform version	☐	☐
3.10	Ensure Amazon ECS services are tagged	☐	☐
3.11	Ensure Amazon ECS clusters are tagged	☐	☐
3.12	Ensure Amazon ECS task definitions are tagged	☐	☐
3.13	Ensure only trusted images are used with Amazon ECS	☐	☐
3.14	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets	☐	☐
5.1	Apply updates to any apps running in Lightsail	☐	☐
5.4	Ensure SSH is restricted to only IP address that should have this access.	☐	☐
5.5	Ensure RDP is restricted to only IP address that should have this access.	☐	☐
5.7	Ensure you are using an IAM policy to manage access to buckets in Lightsail.	☐	☐
5.8	Ensure Lightsail instances are attached to the buckets	☐	☐
5.9	Ensure that your Lightsail buckets are not publicly accessible	☐	☐
5.10	Enable storage bucket access logging	☐	☐
5.11	Ensure your Windows Server based lightsail instances are updated with the latest security patches.	☐	☐
8.1	Ensure AWS Batch is configured with AWS Cloudwatch Logs.	☐	☐

Recommendation		Set Correctly	
		Yes	No
10.1	Ensure Managed Platform updates is configured	☐	☐
10.2	Ensure Persistent logs is setup and configured to S3	☐	☐
10.3	Ensure access logs are enabled.	☐	☐
12.1	Ensure AWS Config is Enabled for Lambda and Serverless	☐	☐
12.2	Ensure Cloudwatch Lambda insights is enabled	☐	☐
12.3	Ensure AWS Secrets manager is configured and being used by Lambda for databases	☐	☐
12.4	Ensure least privilege is used with Lambda function access	☐	☐
12.5	Ensure every Lambda function has its own IAM Role	☐	☐
12.6	Ensure Lambda functions are not exposed to everyone.	☐	☐
12.7	Ensure Lambda functions are referencing active execution roles.	☐	☐
12.8	Ensure that Code Signing is enabled for Lambda functions.	☐	☐
12.11	Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates.	☐	☐

Appendix: CIS Controls v8 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
2.1.1	Ensure Consistent Naming Convention is used for Organizational AMI	☐	☐
2.1.2	Ensure Amazon Machine Images (AMIs) are encrypted	☐	☐
2.1.3	Ensure Only Approved Amazon Machine Images (AMIs) are Used	☐	☐
2.1.4	Ensure Images (AMI) are not older than 90 days	☐	☐
2.1.5	Ensure Images are not Publicly Available	☐	☐
2.2.1	Ensure EBS volume encryption is enabled	☐	☐
2.2.2	Ensure Public Access to EBS Snapshots is Disabled	☐	☐
2.2.3	Ensure EBS volume snapshots are encrypted	☐	☐
2.2.4	Ensure unused EBS volumes are removed	☐	☐
2.3	Ensure Tag Policies are Enabled	☐	☐
2.4	Ensure an Organizational EC2 Tag Policy has been Created	☐	☐
2.5	Ensure no AWS EC2 Instances are Older than 180 days	☐	☐
2.6	Ensure detailed monitoring is enable for production EC2 Instances	☐	☐
2.7	Ensure Default EC2 Security groups are not being used.	☐	☐
2.8	Ensure the Use of IMDSv2 is Enforced on All Existing Instances	☐	☐
2.9	Ensure use of AWS Systems Manager to manage EC2 instances	☐	☐
2.10	Ensure unused ENIs are removed	☐	☐
2.11	Ensure instances stopped for over 90 days are removed	☐	☐
2.12	Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination	☐	☐
2.13	Ensure Secrets and Sensitive Data are not stored directly in EC2 User Data	☐	☐

Recommendation		Set Correctly	
		Yes	No
2.14	Ensure EC2 Auto Scaling Groups Propagate Tags to EC2 Instances that it launches	☐	☐
3.1	Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host	☐	☐
3.2	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services	☐	☐
3.3	Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host'	☐	☐
3.4	Ensure Amazon ECS task definitions do not have 'privileged' set to 'true'	☐	☐
3.5	Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions	☐	☐
3.6	Ensure secrets are not passed as container environment variables in Amazon ECS task definitions	☐	☐
3.7	Ensure logging is configured for Amazon ECS task definitions	☐	☐
3.8	Ensure Amazon ECS Fargate services are using the latest Fargate platform version	☐	☐
3.9	Ensure monitoring is enabled for Amazon ECS clusters	☐	☐
3.10	Ensure Amazon ECS services are tagged	☐	☐
3.11	Ensure Amazon ECS clusters are tagged	☐	☐
3.12	Ensure Amazon ECS task definitions are tagged	☐	☐
3.13	Ensure only trusted images are used with Amazon ECS	☐	☐
3.14	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets	☐	☐
5.1	Apply updates to any apps running in Lightsail	☐	☐
5.3	Disable SSH and RDP ports for Lightsail instances when not needed.	☐	☐
5.4	Ensure SSH is restricted to only IP address that should have this access.	☐	☐
5.5	Ensure RDP is restricted to only IP address that should have this access.	☐	☐
5.6	Disable IPv6 Networking if not in use within your organization.	☐	☐

Recommendation		Set Correctly	
		Yes	No
5.7	Ensure you are using an IAM policy to manage access to buckets in Lightsail.	☐	☐
5.8	Ensure Lightsail instances are attached to the buckets	☐	☐
5.9	Ensure that your Lightsail buckets are not publicly accessible	☐	☐
5.10	Enable storage bucket access logging	☐	☐
5.11	Ensure your Windows Server based lightsail instances are updated with the latest security patches.	☐	☐
6.1	Ensure you are using VPC Endpoints for source code access	☐	☐
8.1	Ensure AWS Batch is configured with AWS Cloudwatch Logs.	☐	☐
10.1	Ensure Managed Platform updates is configured	☐	☐
10.2	Ensure Persistent logs is setup and configured to S3	☐	☐
10.3	Ensure access logs are enabled.	☐	☐
10.4	Ensure that HTTPS is enabled on load balancer	☐	☐
11.1	Ensure customer-managed keys are used to encrypt AWS Fargate ephemeral storage data for Amazon ECS	☐	☐
12.1	Ensure AWS Config is Enabled for Lambda and Serverless	☐	☐
12.2	Ensure Cloudwatch Lambda insights is enabled	☐	☐
12.3	Ensure AWS Secrets manager is configured and being used by Lambda for databases	☐	☐
12.4	Ensure least privilege is used with Lambda function access	☐	☐
12.5	Ensure every Lambda function has its own IAM Role	☐	☐
12.6	Ensure Lambda functions are not exposed to everyone.	☐	☐
12.7	Ensure Lambda functions are referencing active execution roles.	☐	☐
12.8	Ensure that Code Signing is enabled for Lambda functions.	☐	☐
12.11	Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates.	☐	☐
12.12	Ensure encryption in transit is enabled for Lambda environment variables	☐	☐

Recommendation		Set Correctly	
		Yes	No
16.1	Ensure communications between your applications and clients is encrypted.	☐	☐

Appendix: CIS Controls v8 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
2.1.1	Ensure Consistent Naming Convention is used for Organizational AMI	☐	☐
2.1.2	Ensure Amazon Machine Images (AMIs) are encrypted	☐	☐
2.1.3	Ensure Only Approved Amazon Machine Images (AMIs) are Used	☐	☐
2.1.4	Ensure Images (AMI) are not older than 90 days	☐	☐
2.1.5	Ensure Images are not Publicly Available	☐	☐
2.2.1	Ensure EBS volume encryption is enabled	☐	☐
2.2.2	Ensure Public Access to EBS Snapshots is Disabled	☐	☐
2.2.3	Ensure EBS volume snapshots are encrypted	☐	☐
2.2.4	Ensure unused EBS volumes are removed	☐	☐
2.3	Ensure Tag Policies are Enabled	☐	☐
2.4	Ensure an Organizational EC2 Tag Policy has been Created	☐	☐
2.5	Ensure no AWS EC2 Instances are Older than 180 days	☐	☐
2.6	Ensure detailed monitoring is enable for production EC2 Instances	☐	☐
2.7	Ensure Default EC2 Security groups are not being used.	☐	☐
2.8	Ensure the Use of IMDSv2 is Enforced on All Existing Instances	☐	☐
2.9	Ensure use of AWS Systems Manager to manage EC2 instances	☐	☐
2.10	Ensure unused ENIs are removed	☐	☐
2.11	Ensure instances stopped for over 90 days are removed	☐	☐
2.12	Ensure EBS volumes attached to an EC2 instance is marked for deletion upon instance termination	☐	☐
2.13	Ensure Secrets and Sensitive Data are not stored directly in EC2 User Data	☐	☐

Recommendation		Set Correctly	
		Yes	No
2.14	Ensure EC2 Auto Scaling Groups Propagate Tags to EC2 Instances that it launches	☐	☐
3.1	Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host	☐	☐
3.2	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services	☐	☐
3.3	Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host'	☐	☐
3.4	Ensure Amazon ECS task definitions do not have 'privileged' set to 'true'	☐	☐
3.5	Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions	☐	☐
3.6	Ensure secrets are not passed as container environment variables in Amazon ECS task definitions	☐	☐
3.7	Ensure logging is configured for Amazon ECS task definitions	☐	☐
3.8	Ensure Amazon ECS Fargate services are using the latest Fargate platform version	☐	☐
3.9	Ensure monitoring is enabled for Amazon ECS clusters	☐	☐
3.10	Ensure Amazon ECS services are tagged	☐	☐
3.11	Ensure Amazon ECS clusters are tagged	☐	☐
3.12	Ensure Amazon ECS task definitions are tagged	☐	☐
3.13	Ensure only trusted images are used with Amazon ECS	☐	☐
3.14	Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets	☐	☐
5.1	Apply updates to any apps running in Lightsail	☐	☐
5.3	Disable SSH and RDP ports for Lightsail instances when not needed.	☐	☐
5.4	Ensure SSH is restricted to only IP address that should have this access.	☐	☐
5.5	Ensure RDP is restricted to only IP address that should have this access.	☐	☐
5.6	Disable IPv6 Networking if not in use within your organization.	☐	☐

Recommendation		Set Correctly	
		Yes	No
5.7	Ensure you are using an IAM policy to manage access to buckets in Lightsail.	☐	☐
5.8	Ensure Lightsail instances are attached to the buckets	☐	☐
5.9	Ensure that your Lightsail buckets are not publicly accessible	☐	☐
5.10	Enable storage bucket access logging	☐	☐
5.11	Ensure your Windows Server based lightsail instances are updated with the latest security patches.	☐	☐
6.1	Ensure you are using VPC Endpoints for source code access	☐	☐
8.1	Ensure AWS Batch is configured with AWS Cloudwatch Logs.	☐	☐
8.2	Ensure Batch roles are configured for cross-service confused deputy prevention	☐	☐
10.1	Ensure Managed Platform updates is configured	☐	☐
10.2	Ensure Persistent logs is setup and configured to S3	☐	☐
10.3	Ensure access logs are enabled.	☐	☐
10.4	Ensure that HTTPS is enabled on load balancer	☐	☐
11.1	Ensure customer-managed keys are used to encrypt AWS Fargate ephemeral storage data for Amazon ECS	☐	☐
12.1	Ensure AWS Config is Enabled for Lambda and Serverless	☐	☐
12.2	Ensure Cloudwatch Lambda insights is enabled	☐	☐
12.3	Ensure AWS Secrets manager is configured and being used by Lambda for databases	☐	☐
12.4	Ensure least privilege is used with Lambda function access	☐	☐
12.5	Ensure every Lambda function has its own IAM Role	☐	☐
12.6	Ensure Lambda functions are not exposed to everyone.	☐	☐
12.7	Ensure Lambda functions are referencing active execution roles.	☐	☐
12.8	Ensure that Code Signing is enabled for Lambda functions.	☐	☐
12.9	Ensure there are no Lambda functions with admin privileges within your AWS account	☐	☐

Recommendation		Set Correctly	
		Yes	No
12.10	Ensure Lambda functions do not allow unknown cross account access via permission policies.	☐	☐
12.11	Ensure that the runtime environment versions used for your Lambda functions do not have end of support dates.	☐	☐
12.12	Ensure encryption in transit is enabled for Lambda environment variables	☐	☐
16.1	Ensure communications between your applications and clients is encrypted.	☐	☐

Appendix: CIS Controls v8 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
5.2	Change default Administrator login names and passwords for applications	☐	☐
5.12	Change the auto-generated password for Windows based instances.	☐	☐

Appendix: Change History

Date	Version	Changes for this version
Jan 17, 2025	1.1.0	UPDATE - Ensure Images (AMI's) are encrypted (Ticket 17807)
Jan 17, 2025	1.1.0	UPDATE - Ensure Images (AMI) are not older than 90 days (Ticket 17809)
Jan 17, 2025	1.1.0	UPDATE - Audit, Remediation and Assessment Status Changes. (Ticket 17793)
Jan 17, 2025	1.1.0	UPDATE - Ensure EBS volume snapshots are encrypted (Ticket 17804)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure Amazon ECS services are tagged" (Ticket 23423)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure Amazon ECS clusters are tagged" (Ticket 23424)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure Amazon ECS task definitions are tagged" (Ticket 23428)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure only trusted images are used with Amazon ECS" (Ticket 23439)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS task sets" (Ticket 23443)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure customer-managed keys are used to encrypt AWS Fargate ephemeral storage data for Amazon ECS" (Ticket 23438)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure monitoring is enabled for Amazon ECS clusters" (Ticket 23422)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure Amazon ECS Fargate services are using the latest Fargate platform version" (Ticket 23411)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure logging is configured for Amazon ECS task definitions" (Ticket 23410)

Date	Version	Changes for this version
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure secrets are not passed as container environment variables in Amazon ECS task definitions" (Ticket 23408)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure 'readonlyRootFilesystem' is set to 'true' for Amazon ECS task definitions" (Ticket 23407)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure Amazon ECS task definitions do not have 'privileged' set to 'true'" (Ticket 23406)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure Amazon ECS task definitions do not have 'pidMode' set to 'host'" (Ticket 23405)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure 'assignPublicIp' is set to 'DISABLED' for Amazon ECS services" (Ticket 23401)
Jan 29, 2025	1.1.0	ADD - Recommendation proposal for "Ensure Amazon ECS task definitions using 'host' network mode do not allow privileged or root user access to the host" (Ticket 23399)