Reaching Essential Eight maturity on AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Reaching Essential Eight maturity on AWS

Table of Contents

Reaching Essential Eight maturity on AWS: Security and compliance for Australian organizations

Amazon Web Services (contributors)

November 2024 (document history)

The Australian Signals Directorate (ASD) has created and prioritised strategies to help organizations mitigate the risks of cybersecurity threats. Eight of these strategies were chosen to form the *Essential Eight framework*. Many public and private sector organisations in Australia are required to reach maturity under the Essential Eight framework.

The Australian Cyber Security Centre (ACSC) created the Essential Eight framework to help protect Microsoft-based internet-connected networks. However, many organizations are required to reach Essential Eight maturity for all of their environments, both on-premises and in the cloud.

The Essential Eight framework also includes a maturity model designed to help organizations implement the framework through progressive iteration. The model outlines maturity levels zero through three. Maturity level three represents resilience against advanced cybersecurity tactics and highly targeted attacks. This guide provides specific, opinionated guidance to help you achieve Essential Eight maturity level three on AWS.

Security and compliance for Australian organizations

Many organizations in Australia use the AWS Cloud to store confidential data, process sensitive transactions, and build critical services.

Although this guide discusses how to adapt the Essential Eight framework for the cloud, AWS also provides the following certifications and models to help you meet your organization's security and compliance requirements:

- Information Security Registered Assessors Program
- Hosting Certification Framework
- AWS shared responsibility model
- AWS Well-Architected Framework

Information Security Registered Assessors Program

AWS services have been assessed under the Australian Cyber Security Centre (ACSC) Information Security Registered Assessors Program (IRAP) at the PROTECTED level. An independent Australian Signals Directorate (ASD) certified IRAP assessor completed the IRAP assessment of AWS. This assessment provides assurance that, with respect to AWS products and services, applicable controls are implemented for PROTECTED level workloads.

The AWS IRAP PROTECTED package is available through AWS Artifact. The IRAP report was developed using the ACSC Cloud security guidance (ACSC website). For a complete list of AWS services that are in scope, see AWS services in scope: IRAP.

Hosting Certification Framework

The Australian Hosting Certification Framework was developed to support the secure management of government systems and data. This framework is intended to help organizations mitigate supply chain and data centre ownership risks. AWS was granted certification at the Certified Strategic level. This helps government agencies continue to innovate at a rapid pace, knowing that AWS meets government requirements.

AWS shared responsibility model

The AWS shared responsibility model defines how you share responsibility with AWS for security and compliance in the cloud. AWS secures the infrastructure that runs all of the services offered in the AWS Cloud, and you are responsible for securing your use of those services, such as your data and applications.

This shared model can help relieve your compliance and operational burden because AWS operates, manages, and controls many components, from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. You assume responsibility for managing the guest operating system (including updates and security patches) and other associated application software. You also assume responsibility for configuring the security group firewall that AWS provides.

It is critical that you understand the AWS shared responsibility model when you approach Essential Eight maturity on AWS. Your responsibilities vary depending on the services used, the integration of those services into your IT environment, and applicable laws and regulations.

AWS Well-Architected Framework

AWS Well-Architected helps cloud architects build secure, high-performing, resilient, and efficient infrastructure for a variety of applications and workloads. The AWS Well-Architected Framework provides architectural best practices that help you design, build, and operate systems on AWS. This framework is built around six pillars: operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability.

AWS also provides a service for reviewing your workloads. The AWS Well-Architected Tool helps you review and assess your architecture by using the AWS Well-Architected Framework. It provides recommendations for making your workloads more reliable, secure, efficient, and cost-effective.

Reinterpreting the Essential Eight strategies for the cloud

The following are the original Essential Eight mitigation strategies that were designed for Microsoft-based internet-connected networks:

- Application control
- Patch applications
- Configure Microsoft Office macro settings
- User application hardening
- Restrict administrative privileges
- Patch operating systems
- Multi-factor authentication
- Regular backups

It is important to reiterate that the Essential Eight framework is not designed for cloud environments. However, the underlying principles are applicable, and there is overlap between the Essential Eight strategies and AWS Well-Architected Framework best practices.

Various cloud-native approaches can improve security and dramatically reduce your compliance burden. In on-premises environments, you are responsible for all aspects of security, and there are no inherited controls. When running workloads in the cloud, AWS is responsible for protecting the infrastructure that runs our services. You can also reduce your compliance burden by using automation and managed services. *Managed services*, also known as *abstracted services*, are AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. For more information, see the Theme 1: Use managed services section in this guide.

Therefore, some reinterpretation is required to make the Essential Eight strategies appropriate for workloads on AWS. This guide converts the Essential Eight strategies into AWS *themes*.

Using the themes

This guide is divided into eight themes. Each Essential Eight strategy is mapped to one or more of the following themes, and each theme is mapped to one or more best practices in the AWS Well-Architected Framework:

- Theme 1: Use managed services
- Theme 2: Manage immutable infrastructure through secure pipelines
- Theme 3: Manage mutable infrastructure with automation
- Theme 4: Manage identities
- Theme 5: Establish a data perimeter
- Theme 6: Automate backups
- Theme 7: Centralise logging and monitoring
- Theme 8: Implement mechanisms for manual processes

Each theme includes an overview of the topic, related AWS Well-Architected Framework best practices, and instructions for how to achieve Essential Eight maturity and monitor compliance. The instructions provide manual steps or help you configure automations by using AWS Config rules. Manual steps require mechanisms to make sure that findings are addressed. For more information, see Theme 8: Implement mechanisms for manual processes. AWS Config rules require similar oversight or automation in order to remediate noncompliant resources. By following the guidance aligned with these themes, you can reach Essential Eight maturity with an approach that also maximises cloud benefits.

Reinterpreting the Essential Eight strategies for the cloud

Because the Essential Eight framework is not designed for cloud environments, it is essential to take a cloud-native approach when addressing the underlying principles of each Essential Eight strategy. The approach varies depending on two key questions.

Which services are you using?

The AWS shared responsibility model can help relieve your compliance and operational burdens. Managed services shift more responsibility to AWS for maintaining the availability, performance, and security optimisation of the deployed service. Managed services also remove the operational and administrative burden of maintaining a service, providing more time to focus on innovation.

Managed services include serverless services, such as Amazon API Gateway, AWS Lambda, and DynamoDB. A database on Amazon Relational Database Service (Amazon RDS) requires less operational responsibility than a database on Amazon Elastic Compute Cloud (Amazon EC2).

For example, if you're adapting the *Patch operating systems* Essential Eight strategy for the cloud, you need to consider which services you are using and whether you're responsible for patching those resources. AWS is responsible for patching fully managed services, such as Lambda and DynamoDB. For other services, such as Amazon RDS or Amazon Redshift, you might need to manage patches during maintenance windows.

What deployment model are you using?

Is your organization using a mutable or immutable infrastructure approach?

The *mutable infrastructure* model updates and modifies the existing infrastructure for production workloads. This was the standard deployment method before the cloud, when replacing server infrastructure was so costly and time-consuming that the most practical approach was to apply changes to servers already in production. An example of a mutable approach in the cloud is deploying application changes directly onto running EC2 instances, either manually or by using a software deployment service, such as AWS Systems Manager Run Command or AWS CodeDeploy.

The *immutable infrastructure* model deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. An example of an immutable approach is defining an application stack in AWS CloudFormation or AWS Cloud Development Kit (AWS CDK). You can use these services to deploy an application stack through continuous integration and continuous delivery (CI/CD) pipelines. This approach uses deployment methods such as *rolling* or *blue/green*. For more information about this approach, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

For example, if you're adapting the *Patch operating systems* Essential Eight strategy for the cloud, you need to consider how patching applies to the deployment model. For mutable infrastructure, you can manually patch resources or could improve operational efficiency through automation. If you're using immutable infrastructure, then you'd use a CI/CD pipeline to deploy new infrastructure with the latest version of the operating system. In fact, the term *patching* is a misnomer under this model because the infrastructure would be replaced rather than patched.

Theme 1: Use managed services

Essential Eight strategies covered

Patch applications, restrict administrative privileges, patch operating systems

Managed services help you reduce your compliance obligations by allowing AWS to manage some security tasks, such as patching and vulnerability management.

As discussed in the AWS shared responsibility model section, you share responsibility with AWS for cloud security and compliance. This can reduce your operational burden because AWS operates, manages, and controls components, from the host operating system and virtualisation layer to the physical security of the facilities in which the service operates.

Your responsibilities might include managing maintenance windows for managed services, such as Amazon Relational Database Service (Amazon RDS) or Amazon Redshift, and scanning for vulnerabilities in AWS Lambda code or container images. As with all themes in this guide, you also retain responsibility for monitoring and compliance reporting. You can use Amazon Inspector to report vulnerabilities across all of your AWS accounts. You can use rules in AWS Config to make sure that services, such as Amazon RDS and Amazon Redshift, have minor updates and maintenance windows enabled.

For example, if you run an Amazon EC2 instance, your responsibilities include the following:

- Application control
- Patching applications
- Restricting administrative privileges to the Amazon EC2 control plane and the operating system (OS)
- Patching the OS
- Enforcing multi-factor authentication (MFA) to access the AWS control plane and the OS
- Backing up the data and configuration

Whereas if you run a Lambda function, then your responsibilities are reduced and include the following:

- Application control
- Confirming that libraries are up-to-date
- Restricting administrative privileges to the Lambda control plane
- Enforcing MFA to access the AWS control plane
- Backing up the Lambda function code and configuration

Related best practices in the AWS Well-Architected Framework

- SEC01-BP05 Reduce security management scope

Implementing this theme

Enable patching

- Apply Amazon RDS updates
- Enable managed updates in AWS Elastic Beanstalk
- Be aware of Amazon Redshift cluster maintenance windows

Scan for vulnerabilities

- Scan Amazon Elastic Container Registry (Amazon ECR) container images with Amazon Inspector
- Scan Lambda functions with Amazon Inspector

Monitoring this theme

Implement governance checks

- Enable the Operational Best Practices for ACSC Essential 8 conformance pack in AWS Config

Monitor Amazon Inspector

- Assess account-level coverage
- Manage multiple accounts

Implement the following AWS Config rules

- RDS_AUTOMATIC_MINOR_VERSION_UPGRADE_ENABLED
- ELASTIC_BEANSTALK_MANAGED_UPDATES_ENABLED
- REDSHIFT_CLUSTER_MAINTENANCESETTINGS_CHECK
- EC2_MANAGEDINSTANCE_PATCH_COMPLIANCE_STATUS_CHECK
- EKS_CLUSTER_SUPPORTED_VERSION

Theme 2: Manage immutable infrastructure through secure pipelines

Essential Eight strategies covered

Application control, patch applications, patch operating systems

For immutable infrastructure, you must secure deployment pipelines for system changes. AWS Distinguished Engineer, Colm MacCárthaigh, explained this principle in the Zero-Privilege Operations: Running Services Without Access to Data (YouTube video) presentation at the 2022 AWS re:Invent conference.

By restricting direct access to configure AWS resources, you can require that all resources are deployed or changed through approved, secured, and automated pipelines. Usually, you create AWS Identity and Access Management (IAM) policies that allow users to access only the account that hosts the deployment pipeline. You also configure IAM policies that allow break-glass access for a limited number of users. To prevent manual changes, you can use security groups to block SSH and Windows remote desktop protocol (RDP) access to servers. Session Manager, a capability of AWS Systems Manager, can provide access to instances without the need to open inbound ports or maintain bastion hosts.

Amazon Machine Images (AMIs) and container images must be built securely and repeatably. For Amazon EC2 instances, you can use EC2 Image Builder to build AMIs that have built-in security features, such as instance discovery, application control, and logging. For more information about application control, see Implementing Application Control on the ACSC website. You can also use Image Builder to build container images, and you can use Amazon Elastic Container Registry (Amazon ECR) to share those images across accounts. A central security team can approve the automated process to build these AMIs and container images so that any resulting AMI or container image is approved for use by the application teams.

Applications must be defined in infrastructure as code (IaC), by using services such as AWS CloudFormation or AWS Cloud Development Kit (AWS CDK). Code analysis tools, such as AWS CloudFormation Guard, cfn-nag, or cdk-nag, can automatically test code against security best practices in your approved pipeline.

As with Theme 1: Use managed services, Amazon Inspector can report vulnerabilities across your AWS accounts. Centralised cloud and security teams can use this information to verify that the application team is meeting security and compliance requirements.

To monitor and report on compliance, perform ongoing reviews of IAM resources and logs. Use AWS Config rules to make sure that only approved AMIs are used, and make sure that Amazon Inspector is configured to scan Amazon ECR resources for vulnerabilities.

Related best practices in the AWS Well-Architected Framework

- OPS05-BP04 Use build and deployment management systems
- REL08-BP04 Deploy using immutable infrastructure
- SEC06-BP03 Reduce manual management and interactive access

Implementing this theme

Implement AMI and container build pipelines

- Use EC2 Image Builder and build the following into your AMIs:
  - AWS Systems Manager Agent (SSM Agent), which is used for instance discovery and management
  - Security tools for application control, such as Security Enhanced Linux (SELinux) (GitHub), File Access Policy Daemon (fapolicyd) (GitHub), or OpenSCAP
  - Amazon CloudWatch Agent, which is used for logging
- For all EC2 instances, include the `CloudWatchAgentServerPolicy` and `AmazonSSMManagedInstanceCore` policies in the instance profile or IAM role that Systems Manager uses to access your instance
- Share AMIs with the entire organization
- Share EC2 Image Builder resources
- Make sure that application teams are referencing the latest AMIs
- Use your AMI pipeline for patch management
- Implement container build pipelines:
  - Create a container image pipeline using the EC2 Image Builder console wizard

  - Build a continuous delivery pipeline for your container images by using Amazon ECR as a source (AWS blog post)
- Share ECR container images across your organization through multi-account and multi-Region architectures

Implement secure application build pipelines

- Implement build pipelines for IaC, such as by using EC2 Image Builder and AWS CodePipeline (AWS blog post)
- Use code analysis tools, such as AWS CloudFormation Guard, cfn-nag (GitHub), or cdk-nag (GitHub), in CI/CD pipelines to help detect violations of best practices, such as:
  - IAM policies that are too permissive, such as those that use wildcards
  - Security group rules that are too permissive, such as those that use wildcards or allow SSH access
  - Access logs that are not enabled
  - Encryption that is not enabled
  - Password literals
- Implement scanning tools in pipelines (AWS blog post)
- Use AWS Identity and Access Management Access Analyzer in pipelines (AWS blog post) to validate IAM policies that are defined in CloudFormation templates
- Configure IAM policies and service control policies for least-privilege access to use the pipeline or make any modifications to it

Implement vulnerability scanning

- Enable Amazon Inspector in all accounts in your organization
- Use Amazon Inspector to scan AMIs in your AMI build pipeline:
  - Manage the lifecycle of AMIs in EC2 Image Builder (GitHub)
- Configure enhanced scanning for Amazon ECR repositories by using Amazon Inspector
- Build a vulnerability management program to triage and remediate security findings

Monitoring this theme

Monitor IAM and logs on an ongoing basis

- Periodically review your IAM policies to make sure that:
  - Only deployment pipelines have direct access to resources
  - Only approved services have direct access to data
  - Users don't have direct access to resources or data
- Monitor AWS CloudTrail logs to confirm that users are modifying resources through pipelines and aren't directly modifying resources or accessing data
- Periodically review IAM Access Analyzer findings
- Set up an alert to notify you if the root user credentials for an AWS account are used

Implement the following AWS Config rules

- `APPROVED_AMIS_BY_ID`
- `APPROVED_AMIS_BY_TAG`
- `ECR_PRIVATE_IMAGE_SCANNING_ENABLED`

Theme 3: Manage mutable infrastructure with automation

Essential Eight strategies covered

Application control, patch applications, patch operating systems

Similar to immutable infrastructure, you manage mutable infrastructure as IaC, and you modify or update this infrastructure through automated processes. Many of the implementation steps for immutable infrastructure also apply to mutable infrastructure. However, for mutable infrastructure, you must also implement manual controls to make sure that modified workloads still follow best practices.

For mutable infrastructure, you can automate patch management by using Patch Manager, a capability of AWS Systems Manager. Enable Patch Manager in all accounts in your AWS organization.

Prevent direct SSH and RDP access and require users to use Session Manager or Run Command, which are also capabilities of Systems Manager. Unlike SSH and RDP, these capabilities can log system access and changes.

To monitor and report on compliance, you must perform ongoing reviews of patch compliance. You can use AWS Config rules to make sure that all Amazon EC2 instances are managed by Systems Manager, have the required permissions and installed applications, and are in patch compliance.

Related best practices in the AWS Well-Architected Framework

- SEC06-BP03 Reduce manual management and interactive access
- SEC06-BP05 Automate compute protection

Implementing this theme

Automate patching

- Implement the steps in Enable Patch Manager in all accounts in your AWS organization

- For all EC2 instances, include the `CloudWatchAgentServerPolicy` and `AmazonSSMManagedInstanceCore` in the instance profile or IAM role that Systems Manager uses to access your instance

Use automation rather than manual processes

- Implement the guidance in Implement AMI and container build pipelines in Theme 2: Manage immutable infrastructure through secure pipelines
- Use Session Manager or Run Command instead of direct SSH or RDP access

Use automation to install the following on EC2 instances

- AWS Systems Manager Agent (SSM Agent), which is used for instance discovery and management
- Security tools for application control, such as Security Enhanced Linux (SELinux) (GitHub), File Access Policy Daemon (fapolicyd) (GitHub), or OpenSCAP
- Amazon CloudWatch Agent, which is used for logging

Use peer review before any release to ensure that changes are meeting best practices

- IAM policies that are too permissive, such as those that use wildcards
- Security group rules that are too permissive, such as those that use wildcards or allow SSH access
- Access logs that aren't enabled
- Encryption that isn't enabled
- Password literals
- Secure IAM policies

Use identity-level controls

- To require that users modify resources through automated processes and prevent manual configuration, allow read-only permissions for roles that users can assume
- Grant permissions to modify resources only to service roles, such as the role used by Systems Manager

Implement vulnerability scanning

- Implement the guidance in Implement vulnerability scanning in Theme 2: Manage immutable infrastructure through secure pipelines
- Scan your EC2 instances by using Amazon Inspector

Monitoring this theme

Monitor patch compliance on an ongoing basis

- Report on patch compliance by using automation and dashboards
- Implement a mechanism to review dashboards for patch compliance

Monitor IAM and logs on an ongoing basis

- Periodically review your IAM policies to make sure that:
  - Only deployment pipelines have direct access to resources
  - Only approved services have direct access to data
  - Users don't have direct access to resources or data
- Monitor AWS CloudTrail logs to make sure that users are modifying resources through pipelines and aren't directly modifying resources or accessing data
- Periodically review AWS Identity and Access Management Access Analyzer findings
- Set up an alert to notify you if the root user credentials for an AWS account are used

Implement the following AWS Config rules

- `EC2_MANAGEDINSTANCE_PATCH_COMPLIANCE_STATUS_CHECK`
- `EC2_INSTANCE_MANAGED_BY_SSM`
- `EC2_MANAGEDINSTANCE_APPLICATIONS_REQUIRED - SELinux/fapolicyd/OpenSCAP, CW Agent`
- `EC2_MANAGEDINSTANCE_APPLICATIONS_BLACKLISTED - any unsupported apps`
- `IAM_ROLE_MANAGED_POLICY_CHECK - CW Logs, SSM`
- `EC2_MANAGEDINSTANCE_ASSOCIATION_COMPLIANCE_STATUS_CHECK`

- REQUIRED_TAGS
- RESTRICTED_INCOMING_TRAFFIC - 22, 3389

Theme 4: Manage identities

Essential Eight strategies covered

Restrict administrative privileges, multi-factor authentication

Robust management of identity and permissions is a critical aspect of managing security in the cloud. Strong identity practices balance necessary access and least privilege. This helps development teams move quickly without compromising security.

Use identity federation to centralise management of identities. This makes it easier to manage access across multiple applications and services because you are managing access from a single location. This also helps you implement temporary permissions and multi-factor authentication (MFA).

Grant users only the permissions that they require to perform their tasks. AWS Identity and Access Management Access Analyzer can validate policies and verify public and cross-account access. Features such as AWS Organizations service control policies (SCPs), IAM policy conditions, IAM permissions boundaries, and AWS IAM Identity Center permission sets can help you configure fine-grained access control (FGAC).

When doing any type of authentication, it is best to use temporary credentials to reduce or eliminate risks—such as credentials being inadvertently disclosed, shared, or stolen. Use IAM roles instead of IAM users.

Use strong sign-in mechanisms, such as MFA, to mitigate the risk where sign-in credentials have been inadvertently disclosed or are easily guessed. Require MFA for the root user, and you can also require it at a federation level. If use of IAM users is unavoidable, enforce MFA.

To monitor and report on compliance, you must continually work to reduce permissions, monitor findings from IAM Access Analyzer, and remove unused IAM resources. Use AWS Config rules to make sure that strong sign-in mechanisms are enforced, credentials are short-lived, and IAM resources are in use.

Related best practices in the AWS Well-Architected Framework

- SEC02-BP01 Use strong sign-in mechanisms

- SEC02-BP02 Use temporary credentials
- SEC02-BP03 Store and use secrets securely
- SEC02-BP04 Rely on a centralized identity provider
- SEC02-BP05 Audit and rotate credentials periodically
- SEC02-BP06 Employ user groups and attributes
- SEC03-BP01 Define access requirements
- SEC03-BP02 Grant least privilege access
- SEC03-BP03 Establish emergency access process
- SEC03-BP04 Reduce permissions continuously
- SEC03-BP05 Define permission guardrails for your organization
- SEC03-BP06 Manage access based on lifecycle
- SEC03-BP07 Analyze public and cross-account access
- SEC03-BP08 Share resources securely within your organization

Implementing this theme

Implement identity federation

- Require human users to federate with an identity provider to access AWS by using temporary credentials
- Implement temporary elevated access to your AWS environments

Apply least privilege permissions

- Safeguard your root user credentials and don't use them for everyday tasks
- Use IAM Access Analyzer to generate least-privilege policies based on access activity
- Verify public and cross-account access to resources with IAM Access Analyzer
- Use IAM Access Analyzer to validate your IAM policies for secure and functional permissions
- Establish permissions guardrails across multiple accounts
- Use permissions boundaries to set the maximum permissions that an identity-based policy can grant
- Use conditions in IAM policies to further restrict access

- Regularly review and remove unused users, roles, permissions, policies, and credentials
- Get started with AWS managed policies and move toward least-privilege permissions
- Use the permission sets feature in IAM Identity Center

Rotate credentials

- Require workloads to use IAM roles to access AWS
- Automate deletion of unused IAM roles
- Rotate access keys regularly for use cases that require long-term credentials

Enforce MFA

- Require MFA for the root user
- Require MFA through IAM Identity Center
- Consider requiring MFA to service-specific API actions

Monitoring this theme

Monitor least privilege access

- Send IAM Access Analyzer findings to AWS Security Hub CSPM
- Consider setting up notifications for critical IAM Identity Center findings
- Regularly review credential reports for your AWS accounts

Implement the following AWS Config rules

- `ACCESS_KEYS_ROTATED`
- `IAM_ROOT_ACCESS_KEY_CHECK`
- `IAM_USER_MFA_ENABLED`
- `IAM_USER_UNUSED_CREDENTIALS_CHECK`
- `IAM_PASSWORD_POLICY`
- `ROOT_ACCOUNT_HARDWARE_MFA_ENABLED`

Theme 5: Establish a data perimeter

Essential Eight strategies covered

Restrict administrative privileges

A *data perimeter* is a set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. These guardrails serve as always-on boundaries that help protect your data across a broad set of AWS accounts and resources. These organisation-wide guardrails do not replace your existing fine-grained access controls. Instead, they help improve your security strategy by making sure that all AWS Identity and Access Management (IAM) users, roles, and resources adhere to a set of defined security standards.

You can establish a data perimeter by using policies that prevent access from outside of an organisation boundary, typically created in AWS Organizations. The three primary perimeter authorization conditions used to establish a data perimeter are:

- **Trusted identities** – Principals (IAM roles or users) within your AWS accounts, or AWS services acting on your behalf.
- **Trusted resources** – Resources that are in your AWS accounts or are managed by AWS services acting on your behalf.
- **Expected networks –** Your on-premises data centres and virtual private clouds (VPCs), or the networks of AWS services acting on your behalf.

Consider implementing data perimeters between environments of different data classifications, such as `OFFICIAL:SENSITIVE` or `PROTECTED`, or different risk levels, such as development, test, or production. For more information, see Building a data perimeter on AWS (AWS whitepaper) and Establishing a data perimeter on AWS: Overview (AWS blog post).

Related best practices in the AWS Well-Architected Framework

- SEC03-BP05 Define permission guardrails for your organization
- SEC07-BP02 Apply data protection controls based on data sensitivity

Implementing this theme

Implement identity controls

- **Allow only trusted identities to access your resources** – Use resource-based policies with the condition keys `aws:PrincipalOrgID` and `aws:PrincipalIsAWSService`. This allows only principals from your AWS organization and from AWS to access your resources.
- **Allow trusted identities only from your network –** Use VPC endpoint policies with the condition keys `aws:PrincipalOrgID` and `aws:PrincipalIsAWSService`. This allows only principals from your AWS organization and from AWS to access services through VPC endpoints.

Implement resource controls

- **Allow your identities to access only trusted resources –** Use service control policies (SCPs) with the condition key `aws:ResourceOrgID`. This allows your identities to access only resources in your AWS organization.
- **Allow access to trusted resources only from your network** – Use VPC endpoint policies with the condition key `aws:ResourceOrgID`. This allows your identities to access services only through VPC endpoints that are part of your AWS organization.

Implement network controls

- **Allow identities to access resources only from expected networks** – Use SCPs with the condition keys `aws:SourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, and `aws:ViaAWSService`. This allows your identities to access resources only from expected IP addresses, VPCs, and VPC endpoints, and through AWS services.
- **Allow access to your resources only from expected networks** – Use resource-based policies with the condition keys `aws:SourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:ViaAWSService`, and `aws:PrincipalIsAWSService`. This allows access to your resources only from expected IPs, from expected VPCs, from expected VPC endpoints, through AWS services, or when the calling identity is an AWS service.

Monitoring this theme

Monitor policies

- Implement mechanisms to review SCPs, IAM policies, and VPC endpoint policies

Implement the following AWS Config rules

- `SERVICE_VPC_ENDPOINT_ENABLED`

Theme 6: Automate backups

Essential Eight strategies covered

Regular backups

"Failures are a given and everything will eventually fail over time: from routers to hard disks, from operating systems to memory units corrupting TCP packets, from transient errors to permanent failures. This is a given, whether you are using the highest-quality hardware or lowest cost components." —Werner Vogels, CTO, Amazon, All Things Distributed

Data backup and recovery is a critical part of the reliability of a system. AWS is designed to make it easier to create backups, maintain durability of backed-up data, and make sure that backed-up data remains recoverable.

AWS Backup is a fully managed service that centralises and automates the backup of data across AWS services. It supports multiple AWS resource types and helps you implement and maintain a backup strategy for workloads that use multiple AWS resources that must be backed up collectively. AWS Backup also helps you to collectively monitor a backup and restore operation of multiple AWS resources.

AWS Backup Vault Lock is an optional feature of a backup vault, and it can provide additional security and control. When a lock is active in Compliance mode and the grace time is over, the vault configuration cannot be altered or deleted by a user, account or data owner, or AWS. Each vault can have one vault lock in place. This provides *write-once, read-many (WORM)* configuration and enforcement of retention periods.

If you follow the current configuration guidance, AWS Backup can provide 99.999999999% annual durability, also known as *11 nines*. It uses the AWS global infrastructure to replicate your backups across multiple Availability Zones. For more information, see Resilience in AWS Backup.

AWS Backup helps you automate the recovery and testing of backed-up data to verify backup integrity and processes.

Related best practices in the AWS Well-Architected Framework

- SEC09-BP01 Implement secure key and certificate management

- SEC09-BP02 Enforce encryption in transit
- SEC09-BP03 Authenticate network communications

Implementing this theme

Automate data backup and recovery

- Implement data backup on AWS
- Automate data backup at scale (AWS blog post)
- Automate data recovery validation with AWS Backup (AWS blog post)

Implement governance across your AWS Backup outcomes

- Top 10 security best practices for securing backups in AWS (AWS blog post)
- Use AWS Backup Vault Lock to improve the security of your backup vaults
- Use AWS Backup Audit Manager to audit the compliance of your AWS Backup policies

Monitoring this theme

Implement the following AWS Config rules

- `RDS_IN_BACKUP_PLAN`
- `RDS_LAST_BACKUP_RECOVERY_POINT_CREATED`
- `RDS_RESOURCES_PROTECTED_BY_BACKUP_PLAN`
- `REDSHIFT_BACKUP_ENABLED`
- `AURORA_LAST_BACKUP_RECOVERY_POINT_CREATED`
- `AURORA_RESOURCES_PROTECTED_BY_BACKUP_PLAN`
- `BACKUP_PLAN_MIN_FREQUENCY_AND_MIN_RETENTION_CHECK`
- `BACKUP_RECOVERY_POINT_ENCRYPTED`
- `BACKUP_RECOVERY_POINT_MANUAL_DELETION_DISABLED`
- `BACKUP_RECOVERY_POINT_MINIMUM_RETENTION_CHECK`
- `DB_INSTANCE_BACKUP_ENABLED`

- DYNAMODB_IN_BACKUP_PLAN
- DYNAMODB_LAST_BACKUP_RECOVERY_POINT_CREATED
- DYNAMODB_RESOURCES_PROTECTED_BY_BACKUP_PLAN
- EBS_IN_BACKUP_PLAN
- EBS_LAST_BACKUP_RECOVERY_POINT_CREATED
- EBS_RESOURCES_PROTECTED_BY_BACKUP_PLAN
- EC2_LAST_BACKUP_RECOVERY_POINT_CREATED
- S3_LAST_BACKUP_RECOVERY_POINT_CREATED
- S3_RESOURCES_PROTECTED_BY_BACKUP_PLAN
- STORAGEGATEWAY_LAST_BACKUP_RECOVERY_POINT_CREATED
- STORAGEGATEWAY_RESOURCES_PROTECTED_BY_BACKUP_PLAN
- VIRTUALMACHINE_LAST_BACKUP_RECOVERY_POINT_CREATED
- VIRTUALMACHINE_RESOURCES_PROTECTED_BY_BACKUP_PLAN

Theme 7: Centralise logging and monitoring

Essential Eight strategies covered

Application control, patch applications, restrict administrative privileges, multi-factor authentication

AWS provides tools and features that enable you to see what's happening in your AWS environment. These include:

- AWS CloudTrail helps you monitor your AWS deployments by creating a historical trail of AWS API calls for your account, including API calls made through the AWS Management Console, AWS SDKs, and command line tools. For services that support CloudTrail, you can also identify which users and accounts called the service's API, the source IP address the calls were made from, and when the calls occurred.
- Amazon CloudWatch helps you monitor the metrics of your AWS resources and the applications you run on AWS in real time.
- Amazon CloudWatch Logs helps you centralize the logs from all your systems, applications, and AWS services so you can monitor them and archive them securely.
- Amazon GuardDuty is a continuous security monitoring service that analyses and processes logs to identify unexpected and potentially unauthorized activity in your AWS environment. GuardDuty integrates with Amazon EventBridge in order to start an automated response or notify a human.
- AWS Security Hub CSPM provides a comprehensive view of your security state in AWS. It also helps you check your AWSenvironment against security industry standards and best practices.

These tools and features are designed to increase visibility and help you address issues before they negatively affect your environment. This helps you improve your organization's security posture in the cloud and reduces the risk profile of your environment.

Related best practices in the AWS Well-Architected Framework

- SEC04-BP01 Configure service and application logging
- SEC04-BP02 Capture logs, findings, and metrics in standardized locations

Implementing this theme

Enable logging

- Use the CloudWatch agent to publish system-level logs to CloudWatch Logs
- Set up alerts for GuardDuty findings
- Create an organization trail in CloudTrail

Implement logging security best practices

- Implement CloudTrail security best practices
- Use SCPs to prevent users from disabling security services (AWS blog post)
- Encrypt log data in CloudWatch Logs by using AWS Key Management Service

Centralise logs

- Receive CloudTrail logs from multiple accounts
- Send logs to a log archive account
- Centralise CloudWatch Logs in an account for auditing and analysis (AWS blog post)
- Centralize management of Amazon Inspector
- Create an organisation-wide aggregator in AWS Config (AWS blog post)
- Centralise management of Security Hub CSPM
- Centralise management of GuardDuty
- Consider using Amazon Security Lake

Monitoring this theme

Implement mechanisms

- Establish a mechanism to review log findings
- Establish a mechanism to review Security Hub CSPM findings
- Establish a mechanism to respond to GuardDuty findings

Implement the following AWS Config rules

- CLOUDTRAIL_SECURITY_TRAIL_ENABLED
- GUARDDUTY_ENABLED_CENTRALIZED
- SECURITYHUB_ENABLED
- ACCOUNT_PART_OF_ORGANIZATIONS

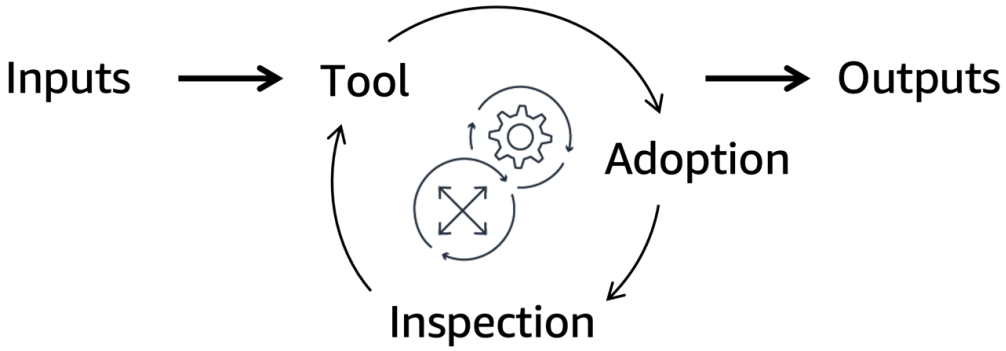
Theme 8: Implement mechanisms for manual processes

Essential Eight strategies covered

Application control, patch applications

At Amazon, we have a saying: Good intentions don't work—mechanisms do (AWS blog post). This means that you must replace best efforts with automated, repeatable, scalable processes and tools in order to achieve the desired outcomes.

As shown in the following diagram, a *mechanism* is a complete process where you create a tool, drive adoption of the tool, and then inspect the results in order to adjustments. It is a cycle that reinforces and improves itself as it operates. It takes controllable inputs and transforms them into ongoing outputs to address a recurring business challenge. For more information, see Building mechanisms in the AWS Well-Architected Framework.



Related best practices in the AWS Well-Architected Framework

- OPS02-BP01 Resources have identified owners
- OPS02-BP02 Processes and procedures have identified owners
- OPS02-BP03 Operations activities have identified owners responsible for their performance
- OPS02-BP04 Mechanisms exist to manage responsibilities and ownership
- OPS03-BP01 Provide executive sponsorship
- OPS03-BP03 Escalation is encouraged

Implementing this theme

- Establish mechanisms to review and address compliance gaps
- Establish mechanisms to update security policies
- Remove applications that are unsupported and then add them to the AWS Config rule deny list
- Validate access policies with AWS Identity and Access Management Access Analyzer
- Enable Amazon Inspector, which automatically keeps vulnerability registers up-to-date
- At a minimum, review application control rule sets annually
- Consider implementing automation, such as AWS Config rules, to reduce the burden of manual processes
- Consider using AWS Systems Manager Inventory to gain visibility into which instances are running software required by your software policy

Monitoring this theme

- Establish oversight for executive sponsors to that can track progress toward goals—including compliance, inspection of gaps, and evaluation of mechanisms.

Indicative case study for reaching Essential Eight maturity on AWS

This chapter presents an indicative case study for a government agency targeting Essential Eight maturity on AWS.

Sections in this chapter:

Scenario and architecture overview

The government agency has three workloads in the AWS Cloud:

- A serverless data lake that uses Amazon Simple Storage Service (Amazon S3) for storage and AWS Lambda for extract, transform, and load (ETL) operations
- A containerised web service that runs on Amazon Elastic Container Service (Amazon ECS) and uses a database in Amazon Relational Database Service (Amazon RDS)
- A commercial off-the-shelf (COTS) software running on Amazon EC2

A *cloud team* provides a centralised platform for the organisation, running core services for the AWS environment. A cloud team provides core services for the AWS environment. Each workload is owned by a distinct *application team*, also known as a *developer team* or *delivery team*.

Core architecture

The cloud team has already established the following capabilities in the AWS Cloud:

- Identity federation links AWS IAM Identity Center to their Microsoft Entra ID (formerly *Azure Active Directory*) instance. The federation enforces MFA, automatic expiry of user accounts, and the use of short-lived credentials through AWS Identity and Access Management (IAM) roles.
- A centralised AMI pipeline is used to patch OSs and core applications with EC2 Image Builder.

- Amazon Inspector is enabled to identify vulnerabilities, and all security findings are sent to Amazon GuardDuty for centralised management.
- Established mechanisms are used to update application control rules, respond to cyber security events, and review compliance gaps.
- AWS CloudTrail is used for logging and monitoring.
- Security events, such as login of the root user, initiate alerts.
- SCPs and VPC endpoint policies establish data perimeters for your AWS environments.
- SCPs prevent application teams from disabling security and logging services, such as CloudTrail and AWS Config.
- AWS Config findings are aggregated from across the whole AWS organization into a single AWS account for security.
- The AWS Config ACSC Essential 8 conformance pack is enabled across all AWS accounts in your organization.

Workload example: Serverless data lake

This workload is an example of Theme 1: Use managed services.

The data lake uses Amazon S3 for storage and AWS Lambda for ETL. These resources are defined in an AWS Cloud Development Kit (AWS CDK) app. Changes to the system are deployed through AWS CodePipeline. This pipeline is restricted to the application team. When the application team makes a pull request for the code repository, the two-person rule is used.

For this workload, the application team takes the following actions to address the Essential Eight strategies.

Application control

- The application team enables Lambda Protection in GuardDuty and Lambda scanning in Amazon Inspector.
- The application team implements mechanisms to inspect and manage Amazon Inspector findings.

Patch applications

- The application team enables Lambda scanning in Amazon Inspector and configures alerts for deprecated or vulnerable libraries.
- The application team enable AWS Config to track AWS resources for asset discovery.

Restrict administrative privileges

- As described in the Core architecture section, the application team already restricts access to production deployments through an approval rule on their deployment pipeline.
- The application team relies on the centralised identity federation and centralised logging solutions that are described in the Core architecture section.
- The application team creates an AWS CloudTrail trail and Amazon CloudWatch filters.
- The application team sets up Amazon Simple Notification Service (Amazon SNS) alerts for CodePipeline deployments and AWS CloudFormation stack deletions.

Patch operating systems

- The application team enables Lambda scanning in Amazon Inspector and configures alerts for deprecated or vulnerable libraries.

Multi-factor authentication

- The application team relies on the centralised identity federation solution described in the Core architecture section. This solution enforces MFA, logs authentications, and alerts on or automatically responds to suspicious MFA events.

Regular backups

- The application team stores code, such as AWS CDK apps and Lambda functions and configurations, in a code repository.
- The application team enables versioning and Amazon S3 Object Lock to help prevent objects from deletion or modification.
- The application team relies on built-in Amazon S3 durability rather than replicating their entire dataset to another AWS Region.
- The application team runs a copy of the workload in another AWS Region that meets their data sovereignty requirements. They use Amazon DynamoDB global tables and Amazon S3 Cross-

Region Replication to replicate data automatically from the primary Region to the secondary Region.

Workload example: Containerised web service

This workload is an example of Theme 2: Manage immutable infrastructure through secure pipelines.

The web service runs on Amazon ECS and uses a database in Amazon RDS. The application team defines these resources in an CloudFormation template. Containers are created with EC2 Image Builder and stored in Amazon ECR. The application team deploys changes to the system through AWS CodePipeline. This pipeline is restricted to the application team. When the application team makes a pull request for the code repository, the two-person rule is used.

For this workload, the application team takes the following actions to address the Essential Eight strategies.

Application control

- The application team enables scanning for Amazon ECR container images in Amazon Inspector.
- The application team build the File Access Policy Daemon (fapolicyd) security tool into the EC2 Image Builder pipeline. For more information, see Implementing Application Control on the ACSC website.
- The application team configures the Amazon ECS task definition to log output to Amazon CloudWatch Logs.
- The application team implements mechanisms to inspect and manage Amazon Inspector findings.

Patch applications

- The application team enables scanning for Amazon ECR container images in Amazon Inspector and configures alerts for deprecated or vulnerable libraries.
- The application team automates their responses to Amazon Inspector findings. New findings initiate their deployment pipeline through an Amazon EventBridge trigger, and CodePipeline is the target.
- The application team enables AWS Config to track AWS resources for asset discovery.

Restrict administrative privileges

- The application team is already restricting access to production deployments through an approval rule on their deployment pipeline.
- The application team relies on the centralised cloud team's identity federation for rotation of credentials and centralised logging.
- The application team creates a CloudTrail trail and CloudWatch filters.
- The application team sets up Amazon SNS alerts for CodePipeline deployments and CloudFormation stack deletions.

Patch operating systems

- The application team enables scanning for Amazon ECR container images in Amazon Inspector and configures alerts for OS patch updates.
- The application team automates their response to Amazon Inspector findings. New findings initiate their deployment pipeline through an EventBridge trigger, and CodePipeline is the target.
- The application team subscribes to Amazon RDS event notifications so that they are informed about updates. They make a risk-based decision with their business owner about whether to apply these updates manually or let Amazon RDS apply them automatically.
- The application team configures the Amazon RDS instance to be a multi-Availability Zone cluster in order to reduce the impact of maintenance events.

Multi-factor authentication

- The application team relies on the centralised identity federation solution described in the Core architecture section. This solution enforces MFA, logs authentications, and alerts on or automatically responds to suspicious MFA events.

Regular backups

- The application team configures AWS Backup to automate backup of the data their Amazon RDS cluster.
- The application team stores CloudFormation templates in a code repository.
- The application team develops an automated pipeline to create a copy of their workload in another Region and run automated tests (AWS blog post). After the automated tests run, the

pipeline destroys the stack. This pipeline automatically runs once a month and validates the effectiveness of the recovery procedures.

Workload example: COTS software on Amazon EC2

This workload is an example of Theme 3: Manage mutable infrastructure with automation.

The workload running on Amazon EC2 was created manually by using the AWS Management Console. Developers manually update the system by logging into the EC2 instances and updating the software.

For this workload, the cloud and application teams take the following actions to address the Essential Eight strategies.

Application control

- The cloud team configures their centralised AMI pipeline to install and configure AWS Systems Manager Agent (SSM Agent), CloudWatch agent, and SELinux. They share the resulting AMI across all accounts in the organization.
- The cloud team uses AWS Config rules to confirm that all running EC2 instances are managed by Systems Manager and have SSM Agent, CloudWatch agent, and SELinux installed.
- The cloud team sends Amazon CloudWatch Logs output to a centralised security information and event management (SIEM) solution that runs on Amazon OpenSearch Service.
- The application team implements mechanisms in order inspect and manage findings from AWS Config, GuardDuty, and Amazon Inspector. The cloud team implements their own mechanisms to catch any findings that the application team misses. For more guidance about creating a vulnerability management program to address findings, see Building a scalable vulnerability management program on AWS.

Patch applications

- The application team patches instances based on Amazon Inspector findings.
- The cloud team patches the base AMI, and the application team receives an alert when that AMI changes.
- The application team restricts direct access to their EC2 instances by configuring security group rules to allow traffic only on the ports that the workload requires.

- The application team uses Patch Manager to patch instances instead of logging in to individual instances.
- To run arbitrary commands on groups of EC2 instances, the application team uses Run Command.
- On the rare occasions when the application team needs direct access to an instance, they use Session Manager. This access approach uses federated identities and logs any session activity for audit purposes.

Restrict administrative privileges

- The application team configures security group rules to allow traffic only on the ports that the workload requires. This restricts direct access to Amazon EC2 instances and requires that users access EC2 instances through Session Manager.
- The application team relies on the centralised cloud team's identity federation for rotation of credentials and centralised logging.
- The application team creates a CloudTrail trail and CloudWatch filters.
- The application team sets up Amazon SNS alerts for CodePipeline deployments and CloudFormation stack deletions.

Patch operating systems

- The cloud team patches the base AMI, and the application team receives an alert when that AMI changes. The application team deploys new instances by using this AMI, and then they use State Manager, a capability of Systems Manager, to install required software.
- The application team uses Patch Manager to patch instances, instance of logging in to individual instances.
- To run arbitrary commands on groups of EC2 instances, the application team uses Run Command.
- On the rare occasions when the application team needs direct access, they use Session Manager.

Multi-factor authentication

- The application team relies on the centralised identity federation solution described in the Core architecture section. This solution enforces MFA, logs authentications, and alerts on or automatically responds to suspicious MFA events.

Regular backups

- The application team creates an AWS Backup plan for its EC2 instances and Amazon Elastic Block Store (Amazon EBS) volumes.
- The application team implements a mechanism to perform a backup restoration manually every month.

Resources

AWS documentation

- AWS Security Reference Architecture (AWS SRA)
- AWS security documentation
- Security pillar of the AWS Well-Architected Framework

Other AWS resources

- AWS Cloud Security
- AWS Cloud Adoption Framework (Security perspective)

Australian Cyber Security Centre resources

- Essential Eight Explained
- Essential Eight Maturity Model
- Essential Eight Assessment Process Guide

Contributors

Contributors to this document include:

- James Kingsmill, Senior Solutions Architect, AWS Solutions Architecture
- Chris Harding, Senior Solutions Architect, AWS Solutions Architecture
- Jess Modini, Advisory Solutions Architect, AWSSolutions Architecture
- Justin Bowden, Security Assurance Principal, AWS Security Assurance
- Rob Powell, Senior Solutions Architect, AWS Solutions Architecture
- Tony Mihaljevic, Senior Cloud Architect, AWS Professional Services
- Volker Rath, Principal Security Advisor, AWS Global Services Security

Appendix: Essential Eight controls matrices

The following tables link the Essential Eight strategies to AWS implementation guidance and relevant best practices in the AWS Well-Architected Framework. For Essential Eight controls that are not applicable in the AWS Cloud, the table includes a link to additional guidance from the Australian Cyber Security Centre (ACSC).

Control matrices:

- Application control
- Patch applications
- Configure Microsoft Office macro settings
- User application hardening
- Restrict administrative privileges
- Patch operating systems
- Multi-factor authentication
- Regular backups

Application control

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Application control is implemented on workstations and servers to restrict the execution of executables, software libraries, scripts, installers, compiled HTML, HTML applications, control panel applets and drivers to an	Theme 2: Manage immutable infrastructure through secure pipelines: Implement AMI and container build pipelines	Use EC2 Image Builder and build in: • AWS Systems Manager Agent (SSM Agent) • Security tools for application control, such as Security Enhanced Linux (SELinux) (GitHub), File Access Policy	SEC06-BP02 Provision compute from hardened images

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
organisation-approved set.		Daemon (fapolicyd) (GitHub), or OpenSCAP Amazon CloudWatch Agent Share AMIs with the entire organization Make sure that application teams are referencing the latest AMIs Use your AMI pipeline for patch management	
Microsoft's 'recommended block rules' are implemented.	See Implementing Application Control (ACSC website)	Not applicable	Not applicable
Microsoft's 'recommended driver block rules' are implemented.			
Application control rulesets are validated on an annual or more frequent basis.	Theme 8: Implement mechanisms for manual processes : Implement mechanism to update security policies	Not available	SEC01-BP08 Evaluate and implement new security services and features regularly

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Allowed and blocked executions on workstations and servers are centrally logged and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.	Theme 7: Centralise logging and monitoring: Enable logging	Use the CloudWatch agent to publish system-level logs to CloudWatch Logs Set up alerts for GuardDuty findings Create an organization trail in CloudTrail Protect data stored in Amazon S3 by using versioning and S3 Object Lock	SEC04-BP01 Configure service and application logging SEC04-BP02 Capture logs, findings, and metrics in standardized locations
	Theme 7: Centralise logging and monitoring: Implement logging security best practices	Implement CloudTrail security best practices Use SCPs to prevent users from disabling security services (AWS blog post) Encrypt log data in CloudWatch Logs by using AWS Key Management Service	SEC04-BP01 Configure service and application logging SEC04-BP02 Capture logs, findings, and metrics in standardized locations

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
	Theme 7: Centralise logging and monitoring: Centralise logs	Receive CloudTrail logs from multiple accounts Send logs to a log archive account Centralise CloudWatch Logs in an account for auditing and analysis (AWS blog post) Centralize management of Amazon Inspector Create an organisation-wide aggregator in AWS Config (AWS blog post) Centralise management of Security Hub CSPM Centralise management of GuardDuty Consider using Amazon Security Lake	SEC04-BP02 Capture logs, findings, and metrics in standardized locations

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
	Theme 8: Implement mechanisms for manual processes: Implement mechanisms to review and address compliance gaps	Consider implementing automation, such as AWS Config rules, to reduce the burden of manual processes	OPS02-BP02 Processes and procedures have identified owners OPS02-BP03 Operations activities have identified owners responsible for their performance OPS02-BP04 Mechanisms exist to manage responsibilities and ownership

Patch applications

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
An automated method of asset discovery is used at least fortnightly to support the detection of assets for subsequent vulnerability scanning activities.	Theme 1: Use managed services: Scan for vulnerabilities Theme 2: Manage immutable infrastructure through secure pipelines: Implement vulnerability scanning	Enable Amazon Inspector in all accounts in your organization Configure enhanced scanning for Amazon ECR repositories by using Amazon Inspector Build a vulnerability management	SEC06-BP01 Perform vulnerability management SEC06-BP05 Automate compute protection

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
	Theme 3: Manage mutable infrastructure with automation: Implement vulnerability scanning	program to triage and remediate security findings	

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
	Theme 7: Centralise logging and monitoring: Centralise logs	Receive CloudTrail logs from multiple accounts Send logs to a log archive account Centralise CloudWatch Logs in an account for auditing and analysis (AWS blog post) Centralize management of Amazon Inspector Create an organisation-wide aggregator in AWS Config (AWS blog post) Centralise management of Security Hub CSPM Centralise management of GuardDuty Consider using Security Lake	SEC04-BP02 Capture logs, findings, and metrics in standardized locations

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
A vulnerability scanner with an up-to-date vulnerability database is used for vulnerability scanning activities. A vulnerability scanner is used at least daily to identify missing patches or updates for security vulnerabilities in internet-facing services.	Theme 1: Use managed services: Scan for vulnerabilities Theme 2: Manage immutable infrastructure through secure pipelines: Implement vulnerability scanning Theme 3: Manage mutable infrastructure with automation: Implement vulnerability scanning	Enable Amazon Inspector in all accounts in your organization Configure enhanced scanning for Amazon ECR repositories by using Amazon Inspector Build a vulnerability management program to triage and remediate security findings	SEC06-BP01 Perform vulnerability management SEC06-BP05 Automate compute protection
A vulnerability scanner is used at least weekly to identify missing patches or updates for security vulnerabilities in office productivity suites, web browsers and their extensions, email clients, PDF software, and security products.	See Technical example: Patch applications (ACSC website)	Not applicable	Not applicable

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
A vulnerability scanner is used at least fortnightly to identify missing patches or updates for security vulnerabilities in other applications.	Theme 1: Use managed services: Scan for vulnerabilities Theme 2: Manage immutable infrastructure through secure pipelines: Implement vulnerability scanning Theme 3: Manage mutable infrastructure with automation: Implement vulnerability scanning	Enable Amazon Inspector in all accounts in your organization Configure enhanced scanning for Amazon ECR repositories by using Amazon Inspector Build a vulnerability management program to triage and remediate security findings	SEC06-BP01 Perform vulnerability management SEC06-BP05 Automate compute protection

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Patches, updates or vendor mitigations for security vulnerabilities in internet-facing services are applied within two weeks of release, or within 48 hours if an exploit exists.	Theme 1: Use managed services: Scan for vulnerabilities Theme 2: Manage immutable infrastructure through secure pipelines: Implement vulnerability scanning Theme 3: Manage mutable infrastructure with automation: Implement vulnerability scanning	Enable Amazon Inspector in all accounts in your organization Configure enhanced scanning for Amazon ECR repositories by using Amazon Inspector Build a vulnerability management program to triage and remediate security findings	SEC06-BP01 Perform vulnerability management
	Theme 3: Manage mutable infrastructure with automation: Automate patching	Enable Patch Manager in all accounts in your AWS organization	SEC06-BP01 Perform vulnerability management SEC06-BP05 Automate compute protection

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Patches, updates or vendor mitigations for security vulnerabilities in office productivity suites, web browsers and their extensions, email clients, PDF software, and security products are applied within two weeks of release, or within 48 hours if an exploit exists.	See Technical example: Patch applications (ACSC website)	Not applicable	Not applicable
Patches, updates or vendor mitigations for security vulnerabilities in other applications are applied within one month of release.	Theme 1: Use managed services: Scan for vulnerabilities Theme 2: Manage immutable infrastructure through secure pipelines: Implement vulnerability scanning Theme 3: Manage mutable infrastructure with automation: Implement vulnerability scanning	Enable Amazon Inspector in all accounts in your organization Configure enhanced scanning for Amazon ECR repositories by using Amazon Inspector Build a vulnerability management program to triage and remediate security findings	SEC06-BP01 Perform vulnerability management

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
	Theme 3: Manage mutable infrastructure with automation: Automate patching	Enable Patch Manager in all accounts in your AWS organization	SEC06-BP01 Perform vulnerability management SEC06-BP05 Automate compute protection
Applications that are no longer supported by vendors are removed.	Theme 8: Implement mechanisms for manual processes: Implement mechanisms to review and address compliance gaps	Consider using AWS Systems Manager Inventory to gain visibility into which instances are running software required by your software policy	SEC06-BP02 Provision compute from hardened images

Configure Microsoft Office macro settings

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Microsoft Office macros are disabled for users that do not have a demonstrated business requirement.	See Technical example: Configure macro settings (ACSC website)	Not applicable	Not applicable
Only Microsoft Office macros running from within a sandboxed environment, a Trusted Location or that are digitally			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
signed by a trusted publisher are allowed to execute.			
Only privileged users responsible for validating that Microsoft Office macros are free of malicious code can write to and modify content within Trusted Locations.			
Microsoft Office macros digitally signed by an untrusted publisher cannot be enabled via the Message Bar or Backstage View.			
Microsoft Office's list of trusted publishers is validated on an annual or more frequent basis.			
Microsoft Office macros in files originating from the internet are blocked.			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Microsoft Office macro antivirus scanning is enabled.			
Microsoft Office macros are blocked from making Win32 API calls.			
Microsoft Office macro security settings cannot be changed by users.			
Allowed and blocked Microsoft Office macro executions are centrally logged and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.			

User application hardening

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Web browsers do not process Java from the internet.	See Technical example: User application hardening (ACSC website)	Not applicable	Not applicable
Web browsers do not process web advertisements from the internet.			
Internet Explorer 11 is disabled or removed.			
Microsoft Office is blocked from creating child processes.			
Microsoft Office is blocked from creating executable content.			
Microsoft Office is blocked from injecting code into other processes.			
Microsoft Office is configured to prevent activation of OLE packages.			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
PDF software is blocked from creating child processes.			
ACSC or vendor hardening guidance for web browsers, Microsoft Office and PDF software is implemented.			
Web browser, Microsoft Office and PDF software security settings cannot be changed by users.			
.NET Framework 3.5 (includes .NET 2.0 and 3.0) is disabled or removed.			
Windows PowerShell 2.0 is disabled or removed.			
PowerShell is configured to use Constrained Language Mode.			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Blocked PowerShell script executions are centrally logged and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.			

Restrict administrative privileges

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Requests for privileged access to systems and applications are validated when first requested.	Theme 4: Manage identities: Implement identity federation	Require human users to federate with an identity provider to access AWS by using temporary credentials	SEC02-BP04 Rely on a centralized identity provider SEC03-BP01 Define access requirements
Privileged access to systems and applications is automatically disabled after 12 months unless revalidated.	Theme 4: Manage identities: Implement identity federation	Require human users to federate with an identity provider to access AWS by using temporary credentials	SEC02-BP04 Rely on a centralized identity provider

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
	Theme 4: Manage identities: Rotate credentials	Require workloads to use IAM roles to access AWS Automate deletion of unused IAM roles Rotate access keys regularly for use cases that require long-term credentials AWS Summit ANZ 2023: Your journey to temporary credentials in the cloud (YouTube video)	SEC02-BP05 Audit and rotate credentials periodically

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Privileged access to systems and applications is automatically disabled after 45 days of inactivity.	Theme 4: Manage identities: Implement identity federation Theme 4: Manage identities: Rotate credentials	Require human users to federate with an identity provider to access AWS by using temporary credentials Require workloads to use IAM roles to access AWS Automate deletion of unused IAM roles Rotate access keys regularly for use cases that require long-term credentials AWS Summit ANZ 2023: Your journey to temporary credentials in the cloud (YouTube video)	SEC02-BP04 Rely on a centralized identity provider SEC02-BP05 Audit and rotate credentials periodically

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Privileged access to systems and applications is limited to only what is required for users and services to undertake their duties.	Theme 4: Manage identities: Apply least privilege permissions	Safeguard your root user credentials and don't use them for everyday tasks Use IAM Access Analyzer to generate least-privilege policies based on access activity Verify public and cross-account access to resources with IAM Access Analyzer Use IAM Access Analyzer to validate your IAM policies for secure and functional permissions Establish permissions guardrails across multiple accounts Use permissions boundaries to set the maximum permissions that an identity-based policy can grant	SEC01-BP02 Secure account root user and properties SEC03-BP02 Grant least privilege access

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
		Use conditions in IAM policies to further restrict access Regularly review and remove unused users, roles, permissions, policies, and credentials Get started with AWS managed policies and move toward least-privilege permissions Use the permission sets feature in IAM Identity Center	
Privileged accounts are prevented from accessing the internet, email and web services.	See Technical example: Restrict administrative privileges (ACSC website)	Consider implementing an SCP that prevents any VPC that doesn't already have internet access from getting it	Not applicable

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Privileged users use separate privileged and unprivileged operating environments.	Theme 5: Establish a data perimeter	Establish a data perimeter. Consider implementing data perimeters between environments of different data classifications, such as `OFFICIAL: SENSITIVE` or `PROTECTED`, or different risk levels, such as development, test, or production.	SEC06-BP03 Reduce manual management and interactive access
Privileged operating environments are not virtualised within unprivileged operating environments.			
Unprivileged accounts cannot logon to privileged operating environments.			
Privileged accounts (excluding local administrator accounts) cannot logon to unprivileged operating environments.			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Just-in-time administration is used for administering systems and applications.	Theme 4: Manage identities: Implement identity federation	Require human users to federate with an identity provider to access AWS by using temporary credentials Implement temporary elevated access to your AWS environments (AWS blog post)	SEC02-BP04 Rely on a centralized identity provider
Administrative activities are conducted through jump servers.	Theme 1: Use managed services Theme 3: Manage mutable infrastructure with automation: Use automation rather than manual processes	Use Session Manager or Run Command instead of direct SSH or RDP access	SEC01-BP05 Reduce security management scope SEC06-BP03 Reduce manual management and interactive access
Credentials for local administrator accounts and service accounts are unique, unpredictable and managed.	See Technical example: Restrict administrative privileges (ACSC website)	Not applicable	Not applicable
Windows Defender Credential Guard and Windows Defender Remote Credential Guard are enabled.			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Use of privileged access is centrally logged and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.	Theme 7: Centralise logging and monitoring: Enable logging Theme 7: Centralise logging and monitoring: Centralise logs	Use CloudWatch Agent to publish OS-level logs to CloudWatch Logs Enable CloudTrail for your organization Centralise CloudWatch Logs in an account for auditing and analysis (AWS blog post) Centralize management of Amazon Inspector Centralise management of Security Hub CSPM Create an organisation-wide aggregator in AWS Config (AWS blog post) Centralise management of GuardDuty Consider using Amazon Security Lake	SEC04-BP01 Configure service and application logging SEC04-BP02 Capture logs, findings, and metrics in standardized locations
Changes to privileged accounts and groups are centrally logged and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
		Receive CloudTrail logs from multiple accounts Send logs to a log archive account	

Patch operating systems

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Patches, updates or vendor mitigations for security vulnerabilities in operating systems of internet-facing services are applied within two weeks of release, or within 48 hours if an exploit exists.	Theme 2: Manage immutable infrastructure through secure pipelines: Implement AMI and container build pipelines	Use EC2 Image Builder and build in: • AWS Systems Manager Agent (SSM Agent) • Security tools for application control, such as Security Enhanced Linux (SELinux) (GitHub), File Access Policy Daemon (fapolicyd) (GitHub), or OpenSCAP • Amazon CloudWatch Agent Share AMIs with the entire organization	SEC01-BP05 Reduce security management scope SEC06-BP01 Perform vulnerability management SEC06-BP03 Reduce manual management and interactive access

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
		Make sure that application teams are referencing the latest AMIs Use your AMI pipeline for patch management	
	Theme 1: Use managed services: Enable patching Theme 3: Manage mutable infrastructure with automation: Automate patching	Enable Patch Manager in all accounts in your AWS organization	SEC06-BP01 Perform vulnerability management SEC06-BP05 Automate compute protection

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Patches, updates or vendor mitigations for security vulnerabilities in operating systems of workstations, servers and network devices are applied within two weeks of release, or within 48 hours if an exploit exists.	Theme 2: Manage immutable infrastructure through secure pipelines: Implement AMI and container build pipelines	Use EC2 Image Builder and build in: • AWS Systems Manager Agent (SSM Agent) • Security tools for application control, such as Security Enhanced Linux (SELinux) (GitHub), File Access Policy Daemon (fapolicyd) (GitHub), or OpenSCAP • Amazon CloudWatch Agent Share AMIs with the entire organization Make sure that application teams are referencing the latest AMIs Use your AMI pipeline for patch management	SEC01-BP05 Reduce security management scope SEC06-BP01 Perform vulnerability management SEC06-BP02 Provision compute from hardened images

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
	Theme 1: Use managed services: Enable patching Theme 3: Manage mutable infrastructure with automation: Automate patching	Enable Patch Manager in all accounts in your AWS organization	SEC06-BP01 Perform vulnerability management SEC06-BP05 Automate compute protection
A vulnerability scanner is used at least daily to identify missing patches or updates for security vulnerabilities in operating systems of internet-facing services. A vulnerability scanner is used at least weekly to identify missing patches or updates for security vulnerabilities in operating systems of workstations, servers and network devices.	Theme 1: Use managed services: Scan for vulnerabilities Theme 2: Manage immutable infrastructure through secure pipelines: Implement vulnerability scanning Theme 3: Manage mutable infrastructure with automation: Implement vulnerability scanning	Enable Amazon Inspector in all accounts in your organization Configure enhanced scanning for Amazon ECR repositories by using Amazon Inspector Build a vulnerability management program to triage and remediate security findings	SEC01-BP05 Reduce security management scope SEC06-BP01 Perform vulnerability management SEC06-BP02 Provision compute from hardened images

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
The latest release, or the previous release, of operating systems are used for workstations, servers and network devices. Operating systems that are no longer supported by vendors are replaced.	Theme 2: Manage immutable infrastructure through secure pipelines: Implement vulnerability scanning	Use EC2 Image Builder and build in: • AWS Systems Manager Agent (SSM Agent) • Security tools for application control, such as Security Enhanced Linux (SELinux) (GitHub), File Access Policy Daemon (fapolicyd) (GitHub), or OpenSCAP • Amazon CloudWatch Agent Share AMIs with the entire organization Make sure that application teams are referencing the latest AMIs Use your AMI pipeline for patch management	SEC01-BP05 Reduce security management scope SEC06-BP01 Perform vulnerability management SEC06-BP02 Provision compute from hardened images

Multi-factor authentication

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Multi-factor authentication is used by an organisation's users if they authenticate to their organisation's internet-facing services.	Theme 4: Manage identities: Implement identity federation	Require human users to federate with an identity provider to access AWS by using temporary credentials Implement temporary elevated access to your AWS environments	SEC02-BP04 Rely on a centralized identity provider
	Theme 4: Manage identities: Enforce MFA	Require MFA for the root user Require MFA through AWS IAM Identity Center Consider requiring MFA to service-specific API actions	SEC02-BP01 Use strong sign-in mechanisms
Multi-factor authentication is used by an organisation's users if they authenticate to third-party internet-facing services that process, store or communica	See Implementing Multi-Factor Authentication (ACSC website)	Not applicable	Not applicable

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
te their organisation's sensitive data.			
Multi-factor authentication (where available) is used by an organisation's users if they authenticate to third-party internet-facing services that process, store or communicate their organisation's non-sensitive data.			
Multi-factor authentication is enabled by default for non-organisational users (but users can choose to opt out) if they authenticate to an organisation's internet-facing services.			

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Multi-factor authentication is used to authenticate privileged users of systems.	Theme 4: Manage identities: Implement identity federation	Require human users to federate with an identity provider to access AWS by using temporary credentials Implement temporary elevated access to your AWS environments	SEC02-BP04 Rely on a centralized identity provider
	Theme 4: Manage identities: Enforce MFA	Require MFA for the root user Require MFA through IAM Identity Center Consider requiring MFA to service-specific API actions	SEC02-BP01 Use strong sign-in mechanisms
Multi-factor authentication is used to authenticate users accessing important data repositories.	Theme 4: Manage identities: Enforce MFA	Consider requiring MFA to service-specific API actions	SEC02-BP01 Use strong sign-in mechanisms

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Multi-factor authentication is verifier impersonation resistant and uses either: something users have and something users know, or something users have that is unlocked by something users know or are.	See Implementing Multi-Factor Authentication (ACSC website)	Not applicable	Not applicable

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Successful and unsuccessful multi-factor authentications are centrally logged and protected from unauthorised modification and deletion, monitored for signs of compromise, and actioned when cyber security events are detected.	Theme 7: Centralise logging and monitoring: Enable logging Theme 7: Centralise logging and monitoring: Centralise logs	Centralise CloudWatch Logs in an account for auditing and analysis (AWS blog post) Centralize management of Amazon Inspector Centralise management of Security Hub CSPM Create an organisation-wide aggregator in AWS Config (AWS blog post) Centralise management of GuardDuty Consider using Security Lake Receive CloudTrail logs from multiple accounts Send logs to a log archive account	SEC04-BP01 Configure service and application logging SEC04-BP02 Capture logs, findings, and metrics in standardized locations

Regular backups

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Backups of important data, software and configuration settings are performed and retained in a coordinated and resilient manner in accordance with business continuity requirements.	Theme 6: Automate backups: Automate data backup and recovery	Implement data backup on AWS Automate data backup at scale (AWS blog post)	REL09-BP01 Identify and back up all data that needs to be backed up, or reproduce the data from sources REL09-BP02 Secure and encrypt backups REL09-BP03 Perform data backup automatically
Restoration of systems, software and important data from backups is tested in a coordinated manner as part of disaster recovery exercises.	Theme 6: Automate backups: Automate data backup and recovery Theme 6: Automate backups: Implement governance across your AWS Backup outcomes	Automate data recovery validation with AWS Backup (AWS blog post) Use AWS Backup Audit Manager to audit the compliance of your AWS Backup policies	REL09-BP04 Perform periodic recovery of the data to verify backup integrity and processes
Unprivileged accounts, and privileged accounts (excluding backup administrators), cannot access backups.	Theme 6: Automate backups: Implement governance across your AWS Backup outcomes	Top 10 security best practices for securing backups in AWS (AWS blog post) Use AWS Backup Vault Lock to improve	SEC08-BP04 Enforce access control

Essential Eight control	Implementation guidance	AWS resources	AWS Well-Architected guidance
Unprivileged accounts, and privileged accounts (excluding backup break glass accounts), are prevented from modifying or deleting backups.		the security of your backup vaults Use AWS Backup Audit Manager to audit the compliance of your AWS Backup policies	

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Best practices updates	We updated this guide to reflect the latest best practices in the security pillar of the AWS Well-Architected Framework.	November 6, 2024
Initial publication	—	October 20, 2023