# AWS Security Best Practices

# Course Introduction

AWS Security Best Practices

# Course agenda

| Module | Topic and activity |
|---|---|
| Module 0 | - Course Introduction |
| Module 1 | - Security Overview |
| Module 2 | - IAM |

# Course agenda

| Module | Topic and activity |
|---|---|
| Module 3 | • Securing the Network |
| Module 4 | • Amazon EC2 Security |
| Module 5 | • Monitoring and Alerting |

# Course agenda

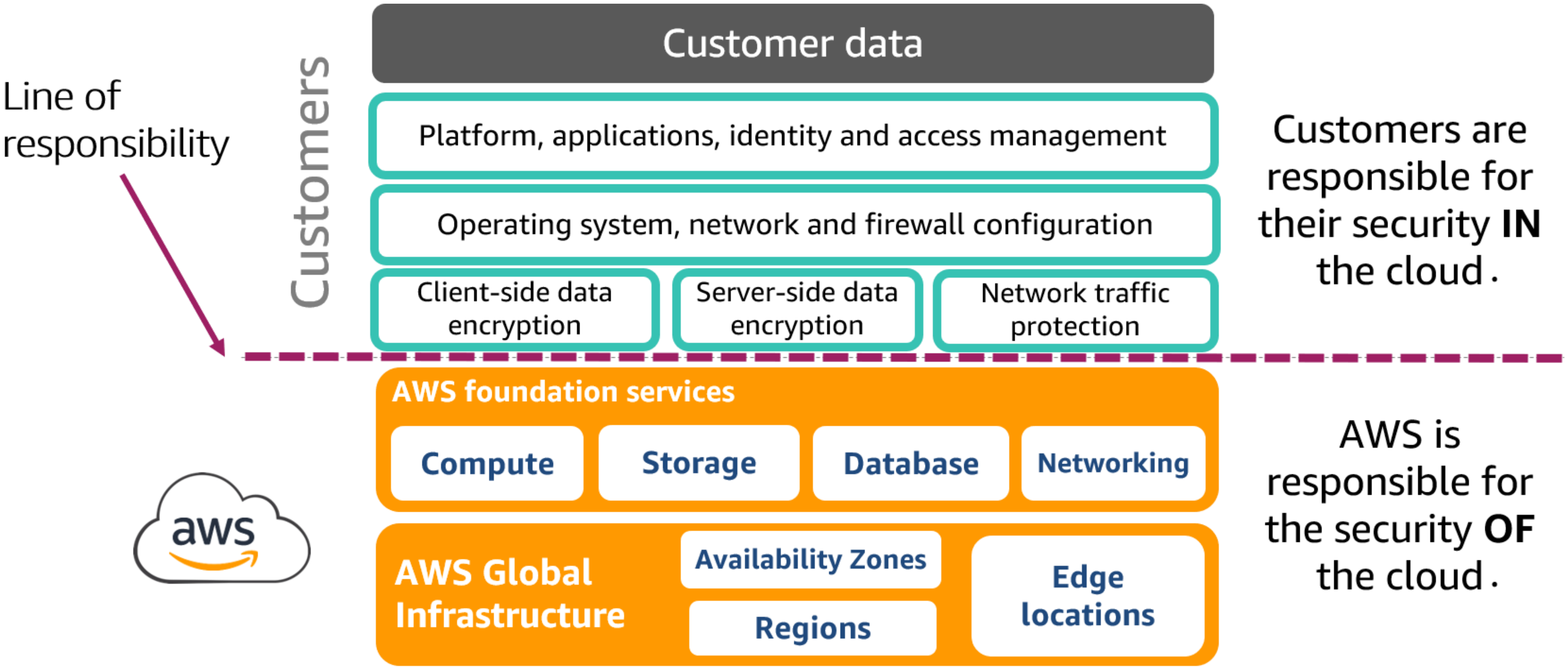
| Module | Topic and activity |
|---|---|
| Module 6 | • Account Provisioning w/Control Tower |
| | • Course Conclusion |

# Security Overview

## AWS Security Best Practices

# Shared responsibility model

# Shared responsibility model review



Line of responsibility

Customers

Customer data

Platform, applications, identity and access management

Operating system, network and firewall configuration

Client-side data encryption

Server-side data encryption

Network traffic protection

Customers are responsible for their security **IN** the cloud.

AWS foundation services

Compute

Storage

Database

Networking

AWS Global Infrastructure

Availability Zones

Regions

Edge locations

AWS is responsible for the security **OF** the cloud.

# Compliance in AWS

# Customer responsibilities

- Understanding what workloads must be regulated by which applicable standards
- Discovering applicable controls or checklist items that apply to workloads
- Mitigating risk and applying applicable controls
- Verifying that the applied controls are deployed and functionally tested against the workload

# AWS compliance programs

The IT standards that AWS complies with are broken out by:

- Certifications and attestations
- Laws, regulations, and privacy
- Alignments and frameworks

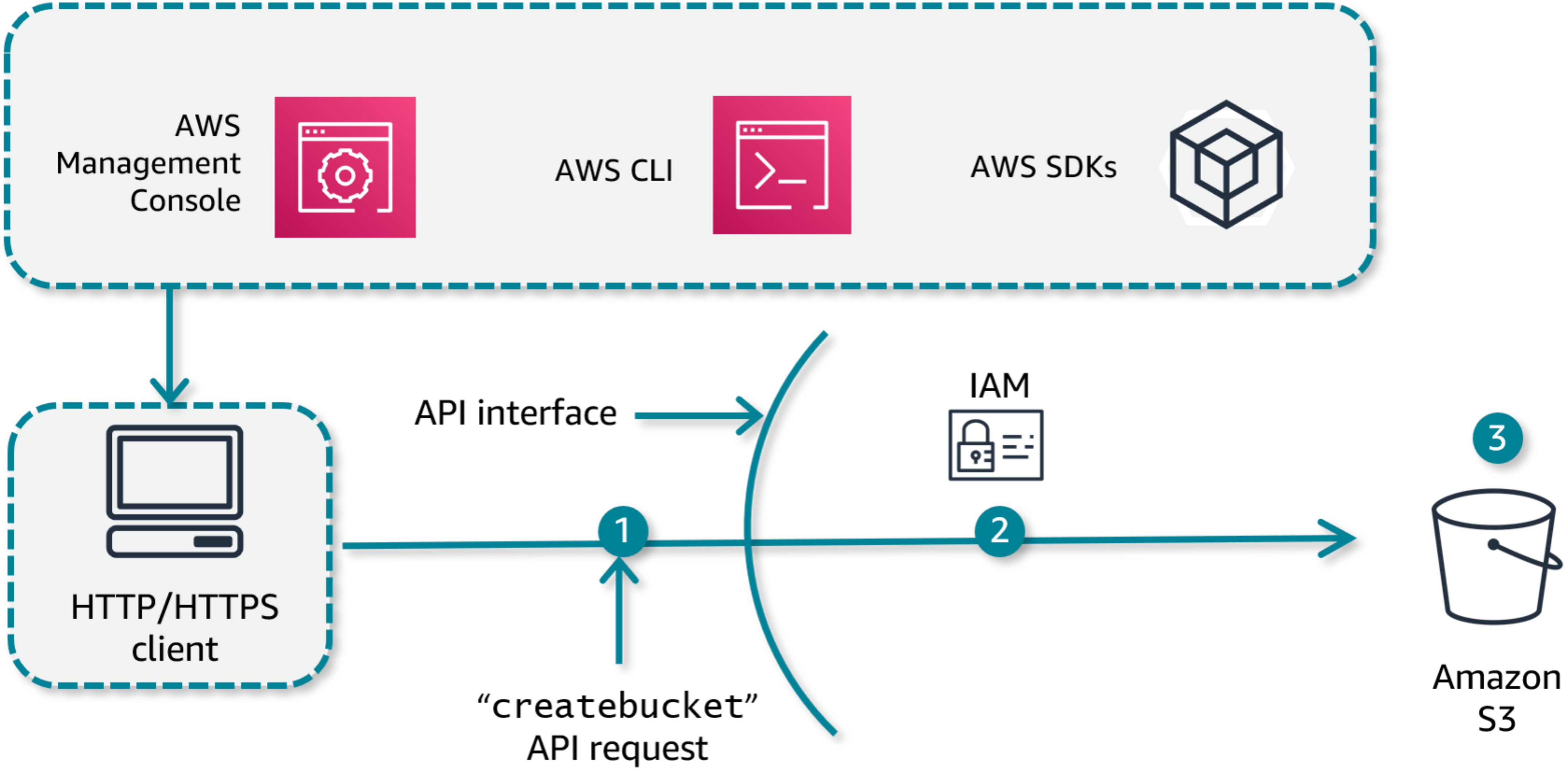
# AWS Artifact

- Reports on demand
- Global availability
- Straightforward identification
- Quick assessments
- Continuous monitoring
- Enhanced transparency

# IAM (Identity and Access Management)

AWS Security Best Practices

# Accessing the cloud



# AWS service spotlight: IAM

AWS Identity and Access Management (IAM) helps you to securely control access to your AWS resources:

- Assign granular permissions to **users**, **groups**, or **roles**.
- Share temporary access to your AWS account.
- Federate users in your corporate network or with an internet identity provider.

# IAM identity details

## IAM user

- Allow operators to sign in to the AWS Management Console or make programmatic requests to AWS.
- Grant permissions by making users members of a group or by directly assigning permissions.

## IAM group

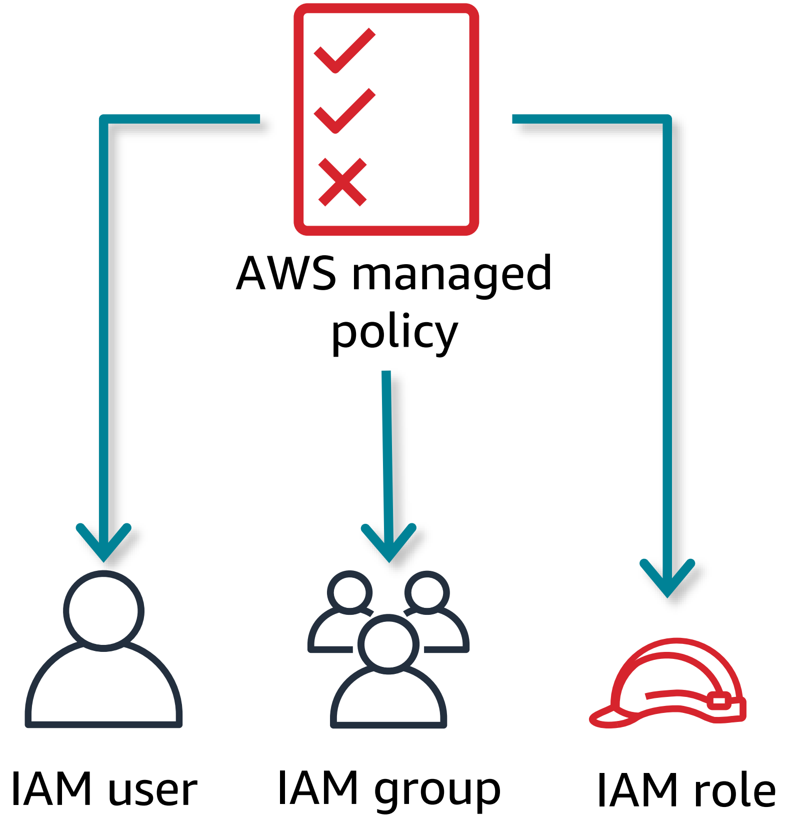
- Collection of IAM users with the same permissions.
- Groups can contain many users, and a user can belong to multiple groups.
- Simplifies users and permission management.

## IAM role

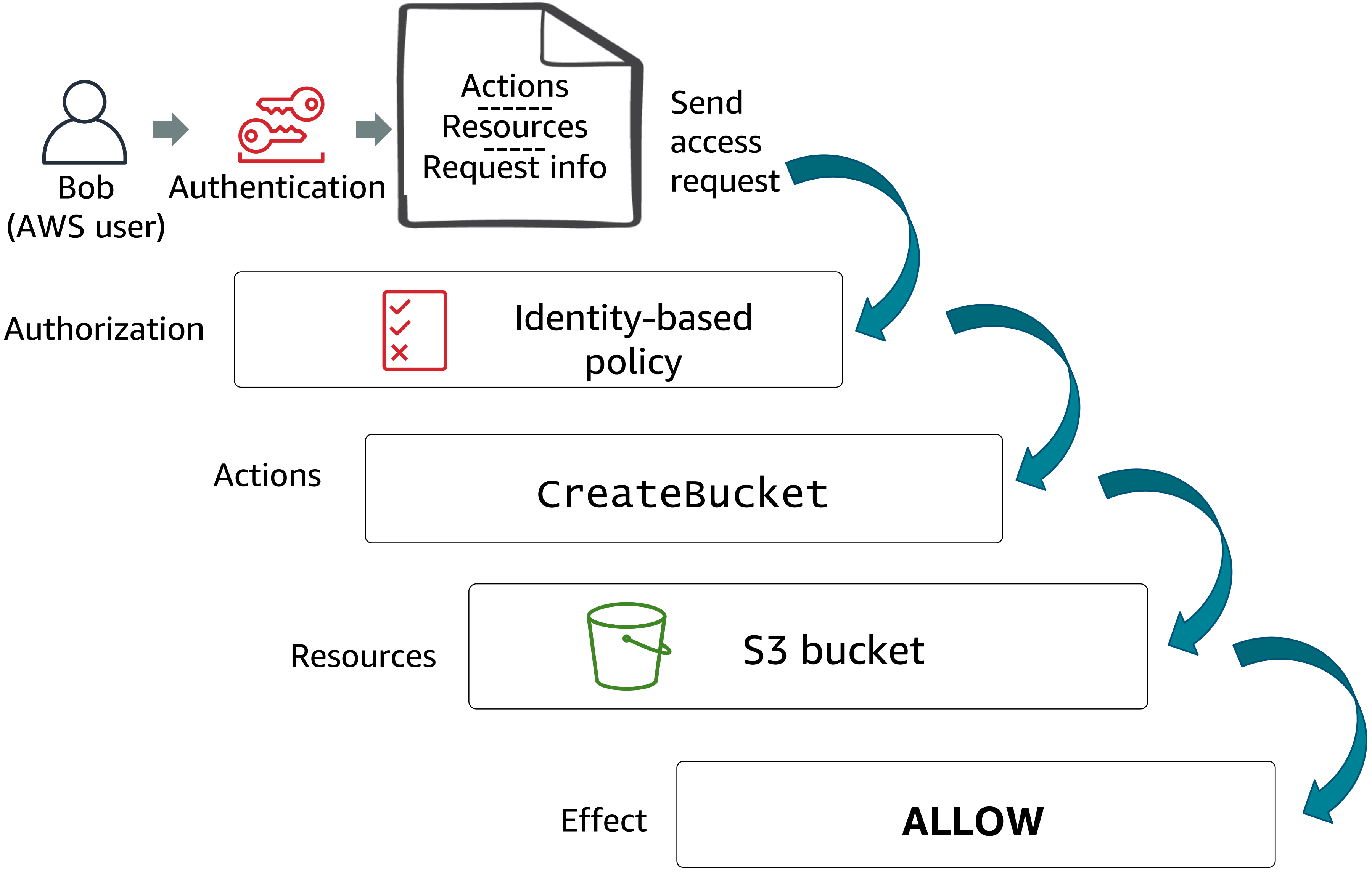
- Provide temporary AWS credentials to trusted users, applications, or services.
- This is an alternative to embedding keys within an app.
- Grant access to users who already have identities defined outside of AWS.

# Access through identity-based policies

- Defines permissions for an IAM identity
- The types of policies include the following:
  - AWS managed
  - Customer managed
  - Inline



# Granting access



Bob (AWS user)

Authentication

Actions
-------
Resources
-----
Request info

Send access request

Authorization

Identity-based policy

Actions

`CreateBucket`

Resources

S3 bucket

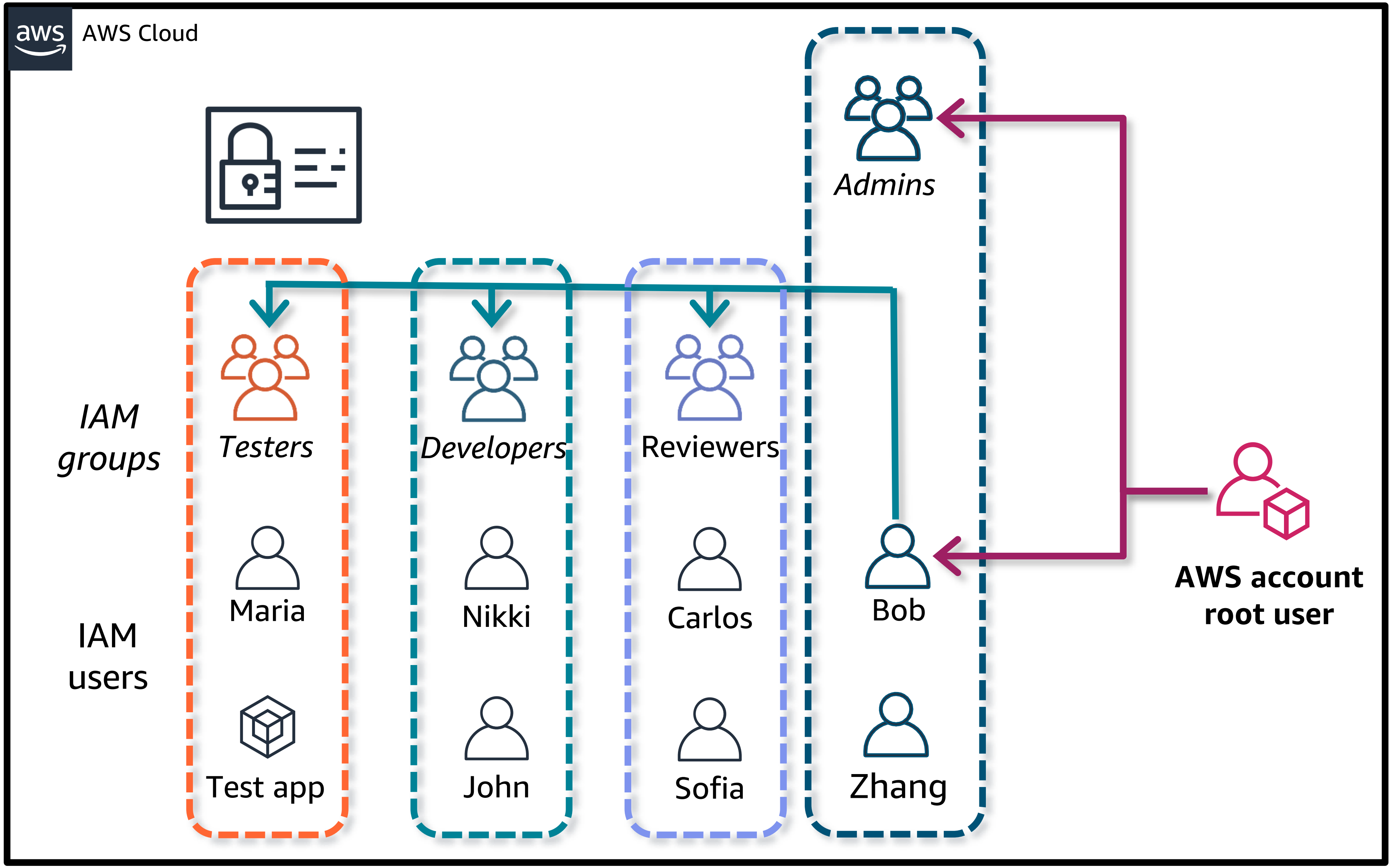
Effect

**ALLOW**

# AWS account

- AWS account root user:
  - Manage your AWS security credentials.
  - Modify account settings.
  - Restore IAM user permissions.
  - Close an account.
- Having multiple accounts can help with cost and resource allocation.

# AWS account root user



# Protection by resource-based policies

# AWS service spotlight: Amazon S3

- Protect your data with encryption and the right level of data access.
- Amazon S3 data is fully private to the account, and public access is blocked by default.
- You can encrypt your data and have AWS manage the keys.
- Detailed server access and audit logs are available.

# Protection through resource-based access

## Amazon S3 Bucket policy

- Policy documents grant permission to specified principals for specific actions on that bucket.
- You can also define under what conditions this applies.
- Only inline policies are supported.
- Amazon S3 bucket and object ACLs are **no longer recommended.**

# Identity-based and resource-based permissions

## Identity-based permissions for Jane

| Jane | Get | Put | List |
|---|---|---|---|
| bucket X | ALLOW | ALLOW | ALLOW |
| bucket Y | | | ALLOW |

## Resource-based permissions for bucket X and bucket Y

| Bucket X | Get | Put | List |
|---|---|---|---|
| Jane | ALLOW | DENY | ALLOW |

| Bucket Y | Get | Put | List |
|---|---|---|---|
| Jane | ALLOW | | ALLOW |

Can Jane GET, PUT, and LIST bucket X?

Can Jane GET and LIST bucket Y?

# Securing the Network

AWS Security Best Practices

# Flexible and secure

# Starting with the Virtual Private Cloud (VPC)

*Network architecture is your foundation.*

A sound strategy for designing, building, and maintaining the network architecture provides the best foundation for scaling and security.

- A good design builds in security.
- Customers have full control over their VPC.
- Stakeholder input helps develop the strategy.

# Security inside your VPC

## Best practices

- Use subnets to isolate the tiers of your application (for example, web, application, and database) within a single VPC.
- Avoid opening Secure Shell (SSH) or Remote Desktop Protocol (RDP) between or within instances of the production environment whenever possible.

# Designing a network

Monitor at boundaries

Subnet to create isolation

Connect externally
through protective devices

# Network segmentation

Advantages of using subnets for network segmentation include the following:

- Limiting the spread and damage of potential attacks by creating smaller impact areas
- Improving visibility and control over traffic movement, device access, and external access
- Reducing the scope when auditing for specific requirements

# DNS operations and security

# Amazon Route 53 using DNSSEC

- Domain Name Security Extensions (DNSSEC) helps prevent DNS attacks like DNS cache poisoning and DNS spoofing.

Store private keys in AWS KMS

Sign public hosted zones or use DNSSEC validation

Use a single key across multiple public hosted zones

# Route 53 Resolver DNS Firewall

- Define domain name filtering rules to control access to sites and block DNS-level threats
- Customize the responses for blocked DNS queries
- Filter on a domain name (not an IP address)
- Filters User Datagram Protocol DNS traffic (not HTTPS, TLS, SSH or, other protocols)
- Centralize management with AWS Firewall Manager

# Security inside the VPC

# Overall network security guidance

## Best practices

- Layer security groups and network ACLs together.
- Use multiple Availability Zone deployments and Elastic Load Balancing (ELB) for high availability.
- Use out-of-band management whenever possible.
- Use Amazon CloudWatch to monitor your VPC components *(covered in module 4).*
- Use flow logs to capture information about traffic in your VPC *(covered in module 4).*
- Always use Identity and Access Management (IAM) to limit access to your resources, including the VPC and related components.

# Network filtering methods

## Stateless

- Focus on the content of individual packets
- Generally use information from headers (IP source or destination, protocol, and so on) for filtering
- Generally fast and has no issue with heavy traffic loads
- Includes network access control lists

## Stateful

- Track and filter all traffic that is part of a stateful associated (for example in the same TCP session)
- Can identify TCP connection stages, packet state, and other key statuses
- Includes security groups and firewalls

# Using Network ACLs in your VPC

## Best practices

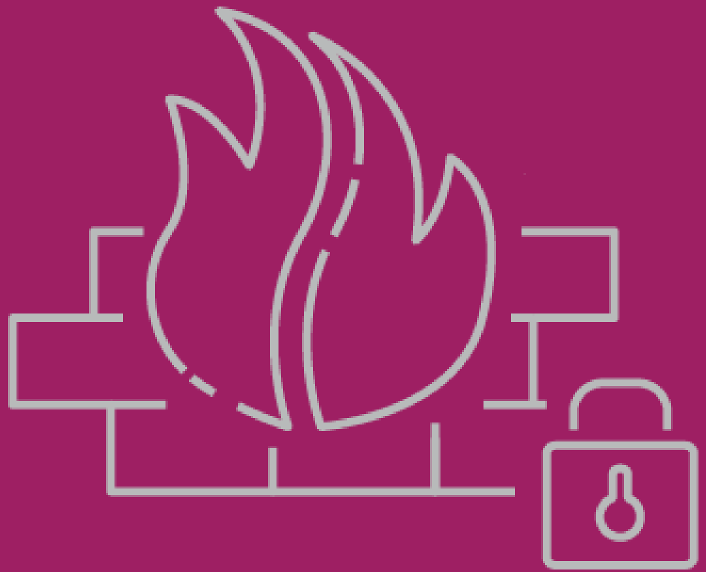
- Remember the default network ACL.
- Monitor and audit network ACLs for ineffective “deny” rules.
- Consider limitations.
- Do not ignore outbound rules on network ACLs.

# Using security groups in your VPC

## Best practices

- Never keep unattached security groups.
- Track rate of change in production environments.
- Ensure that security groups do not have a large range of ports open.
- Use elastic load balancers with security groups to restrict access to the internet.
- Limit modifications to only certain IAM roles.
- Do not ignore outbound rules of security groups.

# Service highlight: AWS Network Firewall



AWS Network Firewall is a managed network protection service that provides the following:

- Web filtering
- Intrusion protection
- Central management and visibility
- Rule management and customization
- Partner integrations

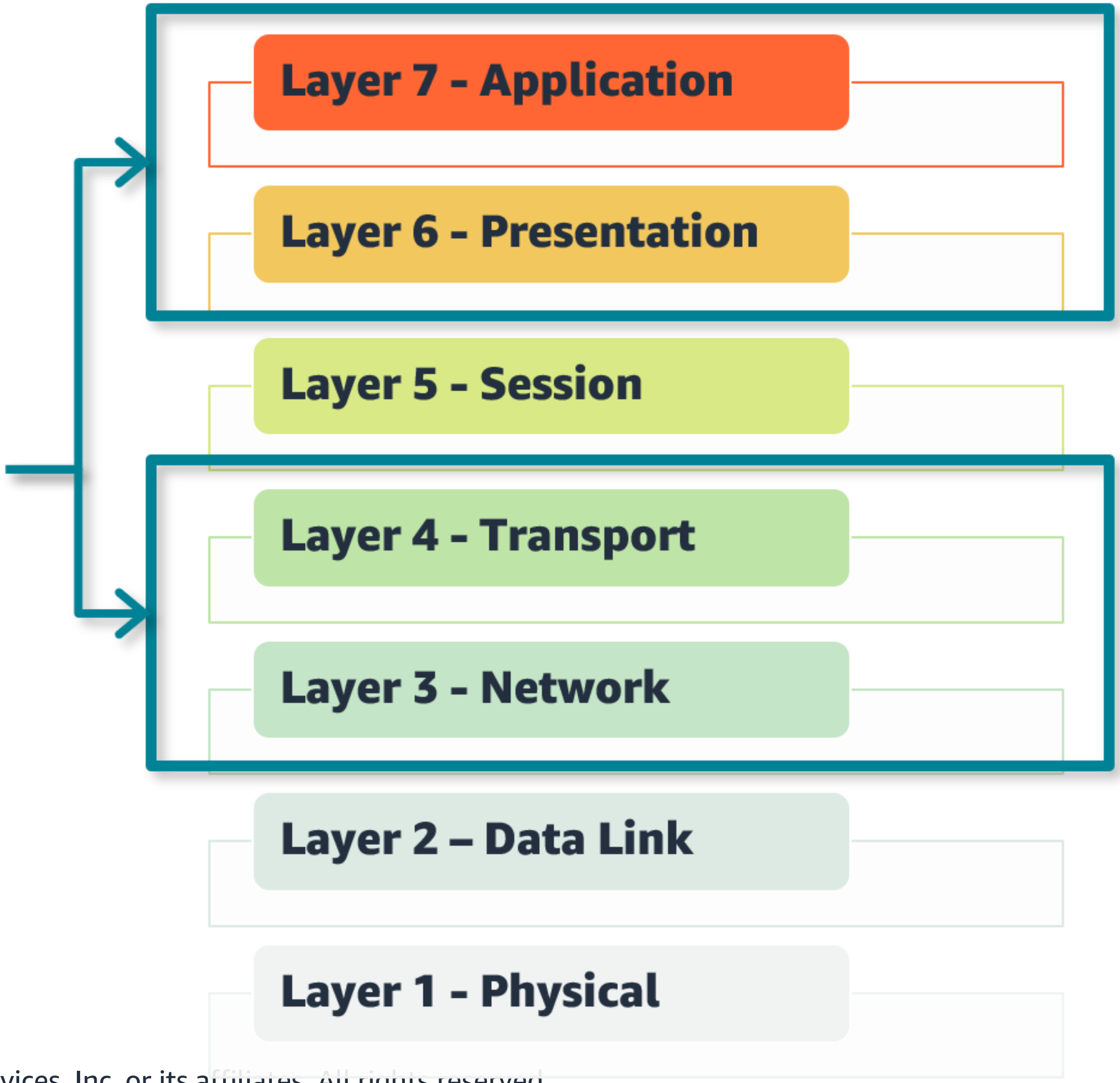
# Network Firewall and other AWS security services

| | Network Firewall | VPC security group | Network ACL | AWS WAF |
|---|---|---|---|---|
| **Where is the protection applied?** | Route level, based on VPC routes | Amazon EC2-instance level | Subnet level | Endpoint level (API Gateway, ALB, CloudFront) |
| **Stateful or stateless** | Both | Stateful | Stateless | Stateless |
| **Which flows are protected?** | All ingress/egress flows at perimeter of VPC (e.g., IGW, VGW, DX, VPN, VPC–VPC) | All ingress/egress flows at instance level (EC2–EC2, EC2–IGW, EC2–DX, etc.) | All ingress/egress flows at subnet level (subnet–subnet, subnet–IGW, subnet–DX, etc.) | Ingress only from internet to API Gateway, ALB, CloudFront |
| **Which OSI layer?** | L3-7 | L4 | L3 | L7 |
| **Features** | Stateless/ACL L3 rules, stateful/L4 rules, IPS–IDS/L7 rules, FQDN filtering, protocol detection, deep packet inspection, large IP block/allow lists | IP \| port \| protocol filtering | IP \| port \| protocol filtering | Deep application layer filtering, managed rules |
| **Default behavior** | Allow | Deny | Allow | Customer chooses |

# Security services

# Threat highlight: Distributed Denial of Service attack

DDoS are most common at the following Open Systems Interconnection (OSI) model layers:



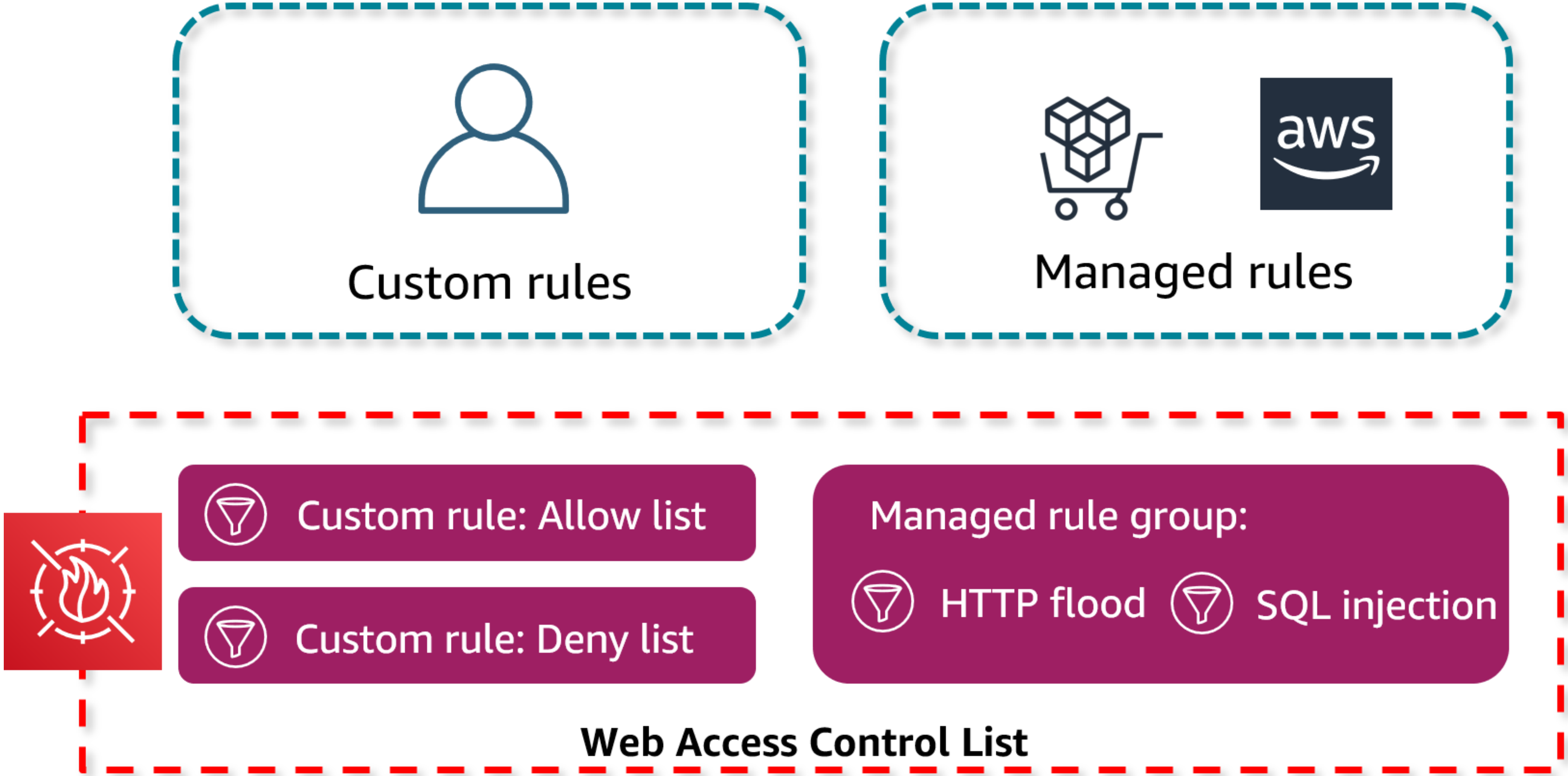
# AWS Web Application Firewall (WAF)

AWS WAF filters traffic for your web applications based on the following criteria:

- IP address origin of the request
- Country of origin of the request
- String match or regular expression (regex) match in a part of the request
- Size of a particular part of the request
- Malicious SQL code or scripting

This service is provided to customers using AWS Shield Advanced for no additional cost and adds additional DDoS protection

# AWS WAF rules and rule groups



# AWS Shield

## Standard Protection

- Available to all AWS Customers at **no additional cost**
- Automatic detection and mitigation
- Protection from most common DDoS attacks (SYN/UDP Floods, Reflection Attacks, etc.)

## Advanced Protection

- Paid service that provides additional protection, features, and benefits.
- Includes Shield Response Team (SRT), AWS WAF for layer 7 DDoS attack mitigation, and AWS Firewall Manager

# Shield Response Team (SRT)

- Shield Advanced includes the option to receive proactive support from the Shield Response Team (SRT).
- During a DDoS attack, the SRT will provide resolution support if necessary.

# Amazon EC2 Security

AWS Security Best Practices

# Compute hardening

# Common vulnerabilities

Some examples of common vulnerabilities include the following:

- Unintentionally exposing Amazon Elastic Cloud Compute (EC2) instances to the public
- Sensitive information in metadata
- Unused or unneeded services or software
- Outdated or nonpatched OS or installed software
- Application configuration weaknesses (such as startup and configuration scripts containing sensitive information)
- Overly permissive identity and access management policies

# Hardening your systems

## Examples of hardening:

- Changing default passwords
- Removing or disabling unnecessary software or services
- Removal of unnecessary user names or logins
- Installing anti-malware and host intrusion detection and prevention systems (HIDS/HIPS)
- Using AWS Systems Manager Agent (**SSM** Agent) for remote access

## AWS Services that can help

- AWS Systems Manager
- Amazon Inspector
- AWS Config

# Hardening with benchmarks

## Best practice

- Create an Amazon Machine Image (AMI) from your instance to save the configuration as a template for launching future instances.

-or-

- Use EC2 Image Builder to create and maintain images.
- Use benchmarks (from CIS and others) to harden common vulnerabilities and help minimize the attack surface.

# CIS Benchmarks purpose

Benchmarks something to compare to and can help with the following.

- Using industry best practices
- Removing the guesswork in hardening
- Consistently evaluating against a known baseline
- Reducing complexity in risk management and auditing for critical, audited, and regulated systems

# CIS Benchmarks alignment

CIS Benchmarks align closely with, or map to, regulatory frameworks including the following:

- National Institute of Standards and Technology Cybersecurity Framework (NIST CSF)
- Payment Card Industry Data Security Standard (PCI DSS)
- Health Insurance Portability and Accountability Act (HIPAA)
- International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC 2700)
- European Union's General Data Protection Regulation (GDPR)

# Amazon EBS encryption

# Amazon EBS backed instances

## Best practices

- Use separate Amazon EBS volumes for the OS and your data.
- Encrypt EBS volumes and snapshots.
- Understand the implications of the root device type for data persistence, backup, and recovery.

# Encryption by default

Encryption by default is a best practice to ensure security of data at rest.

- Encryption by default is a Region-specific setting.
- You can launch an instance only if the instance type supports Amazon EBS encryption.
- Do not use encryption by default while using automated migration services.

# AWS KMS

AWS KMS supports many of the security best practices discussed in this course, providing centralized and secure management of cryptographic keys. This demo provides a quick look at some important features within AWS KMS.

- Prerequisites:
  - IAM user with appropriate AWS KMS permissions

# Secure management and maintenance

# Management and maintenance

## Best practices

- Limit access and authorization for connecting to instances *(Session Manager)*
- Securely manage instances at scale (using Run Command).
- Regularly patch and update with defined maintenance windows *(using Patch Manager).*
- Automate monitoring and remediate of configuration drift *(using State Manager).*
- Secure, monitor, and rotate secrets *(using Secrets Manager or Parameter Store).*

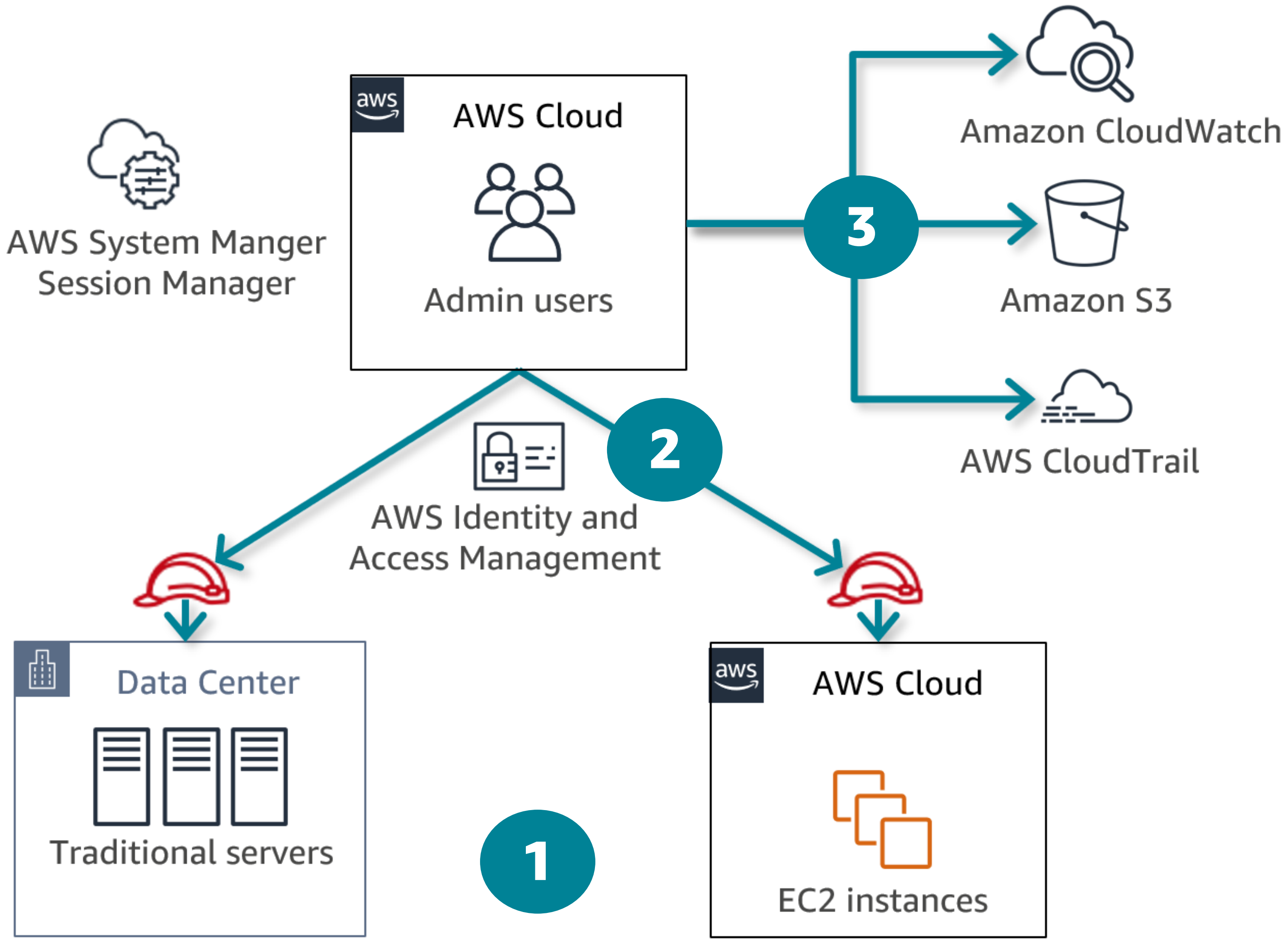
# AWS Systems Manager

## Node Management Highlights

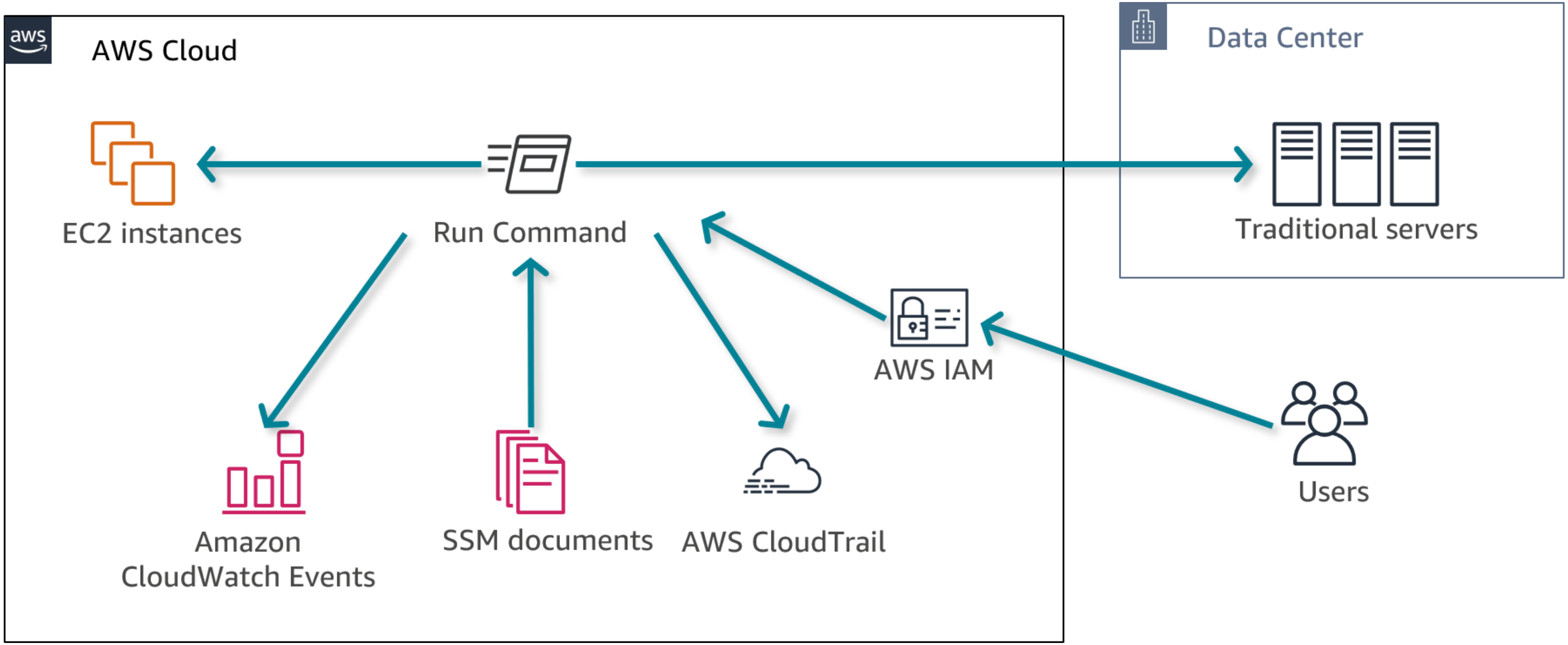
- Session Manager
- Run Command
- State Manager
- Patch Manager
- Parameter Store (compared to AWS Secrets Manager)

# Session manager

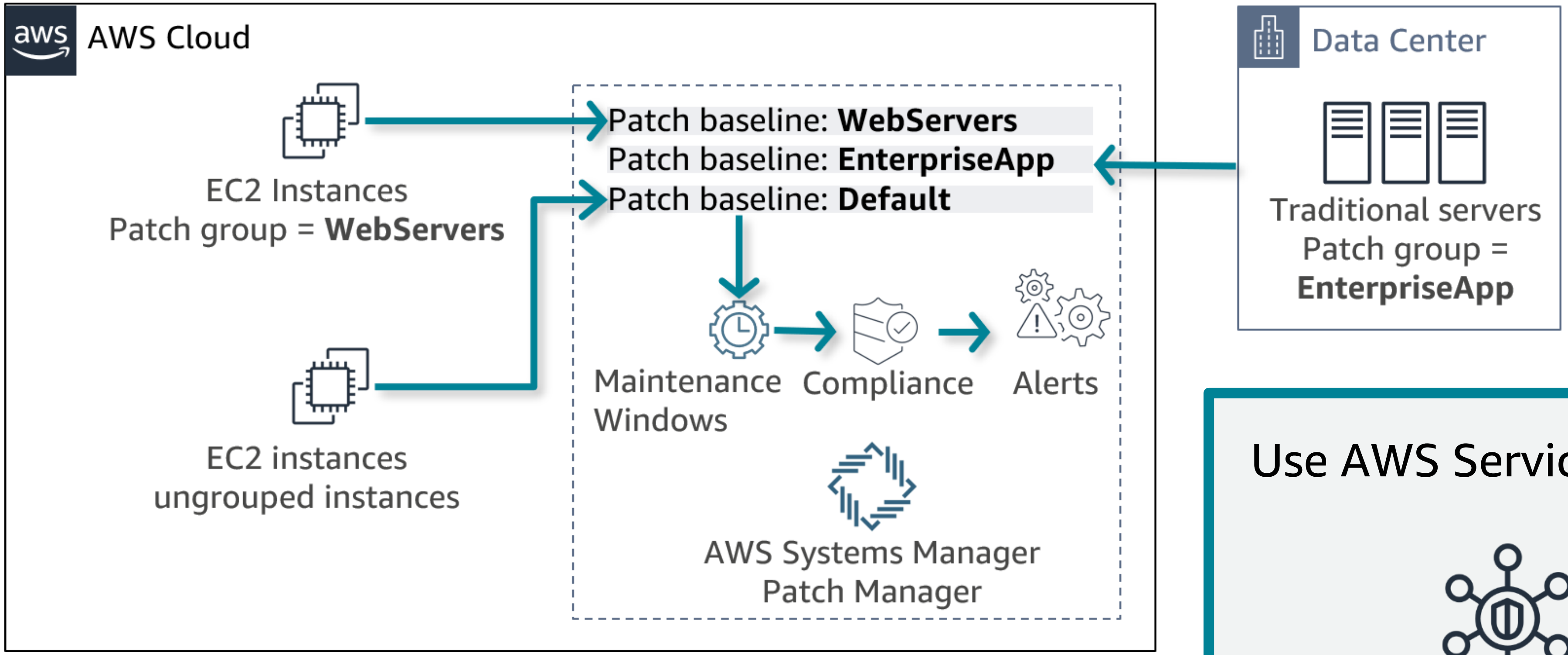
- Centralized access control to managed nodes using IAM policies
- No open inbound ports and no need to manage bastion hosts or SSH keys
- Logging and auditing session activity



# Run command


AWS Cloud
Data Center
EC2 instances
Run Command
Traditional servers
AWS IAM
Users
Amazon CloudWatch Events
SSM documents
AWS CloudTrail

# Patching best practices


AWS Cloud
EC2 Instances
Patch group = **WebServers**
EC2 instances
ungrouped instances
Patch baseline: **WebServers**
Patch baseline: **EnterpriseApp**
Patch baseline: **Default**
Maintenance Windows
Compliance
Alerts
AWS Systems Manager Patch Manager
Data Center
Traditional servers
Patch group = **EnterpriseApp**

- Deploy patches at scale.
- Schedule dedicated maintenance periods.
- Test patches in a nonproduction environment.

Use AWS Service Integrations

AWS Security Hub AWS Config

# State Manager

## Usage

- Maintain visibility over system states.
- Apply configurations based on policies.
- Create and push alerts when configuration drifts are detected.
- Query statuses for on-demand visibility into compliance status.

## Best practices

- Update SSM Agent using the preconfigured AWS-UpdateSSMAgent document.
- Use tags to create groups then target nodes using the targets parameter.
- Use a centralized configuration repository for your SSM documents, and share it across your organization.

# Parameter Store and Secrets Manager

## Parameter Store

- Can notify you of expiring secrets but cannot rotate them for you
- Can be referenced from AWS CloudFormation templates
- Supports storing values under a name or key, encryption of secrets, and versioning

## Secrets Manager

In addition to the capabilities of Parameter Store:

- Provides full key rotation integration with Amazon RDS
- Randomly generates passwords in CloudFormation and stores the password in Secrets Manager
- Shares secrets across different AWS accounts
- Can exceed storage capacity of Parameter Store, but has costs associated to storage of secrets and API calls

# Exploring AWS Systems Manager

You explored just a few of the node management capabilities of AWS Systems Manager. There are many other features available that can help to operate and maintain your environment securely:

- Distributor
- Fleet Manager
- Parameter Store
- Many more…

# Detecting vulnerabilities

# Amazon Inspector

- Amazon Inspector continuously scans your resources to help you do the following:
  - Prioritize patch remediation.
  - Meet compliance requirements.
  - Identify zero-day vulnerabilities sooner.
- Amazon Inspector integrates with AWS Organizations, AWS Security Hub, and Amazon EventBridge.

# AWS Config benefits

- Automatically discover resources.
- Record the current state of a resource.
- Track changes; collect a historical record of the changes .
- Evaluate configuration changes against compliance policies.
- Automate remediation activities.
- Create real-time alerts using Amazon SNS and EventBridge.

# Monitoring and Alerting

## AWS Security Best Practices

# Logging network traffic

# VPC Flow Logs

## What they are

*VPC Flow Log capture packet metadata like the source IP address, destination IP address, ports, protocol, packet size and other metadata.*

- Flow Logs cannot monitor packet contents (payload or application layer data).
- They are not real-time, they use aggregation interval for capture.
- Some types of traffic traversing your network are **NOT** captured by Flow Logs.
- They have no affect on network throughput or latency.

## Best practices

- **VPC flow logging should be enabled for packet rejects for all VPCs.**
- Flow logging is instrumental to network traffic investigations.
- AWS Config has a rule to check if a VPC has flow logging enabled.

# Traffic Mirroring

**Using traffic mirroring provides a detective control that allows you to send your traffic to out-of-band security appliances for the following:**

- Content inspection
- Threat monitoring
- Troubleshooting

# Reasons for Traffic Mirroring

- Detect network and security anomalies
  - You can extract traffic of interest from any workload in a VPC and route it to the detection tools of your choice. You can detect and respond to attacks more quickly than is possible with traditional log-based tools.
- Implement compliance and security controls
  - You can meet regulatory and compliance requirements that mandate monitoring, logging, and so forth.

# Logging user and API traffic

# AWS CloudTrail functions

- Simplify compliance audits by automatically recording and storing activity logs for an AWS account.
- Increase visibility into user and resource activity.
- Discover and troubleshoot security and operational issues by capturing a comprehensive history of changes that occurred in an AWS account.

**AWS CloudTrail** tracks the who, what, where, and when of activity that occurs in your AWS environment and records this activity in audit logs.

# AWS CloudTrail best practices

# Amazon S3 log storage

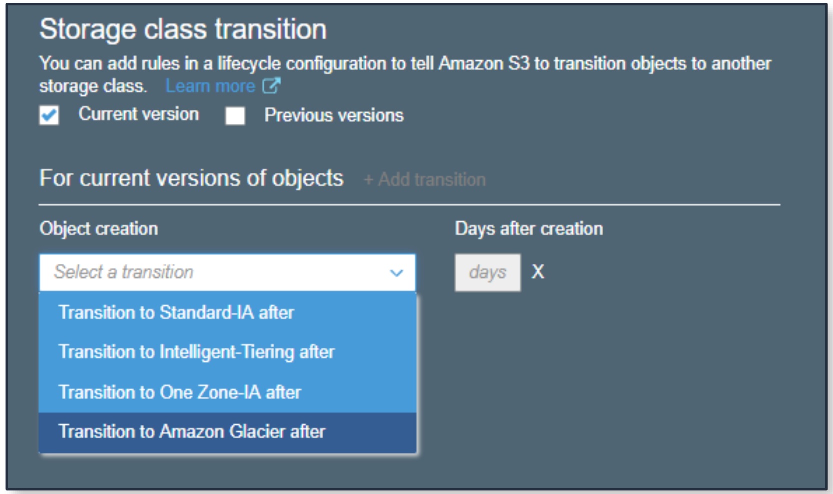
## Best practice

- Use a dedicated S3 bucket for CloudTrail logs.
- Implement least-privilege access to buckets where you store log files.
- Enable multi-factor authentication (MFA) Delete on the log storage bucket.
- Limit access to the “AWSCloudTrail_FullAccess” policy.

# CloudTrail: Lifecycle management

## Best practices

- Configured through Amazon S3
- Available actions:
  - Transition to different storage tier
  - Expire (delete) object
  - Transition and expire



### Storage class transition

You can add rules in a lifecycle configuration to tell Amazon S3 to transition objects to another storage class. Learn more

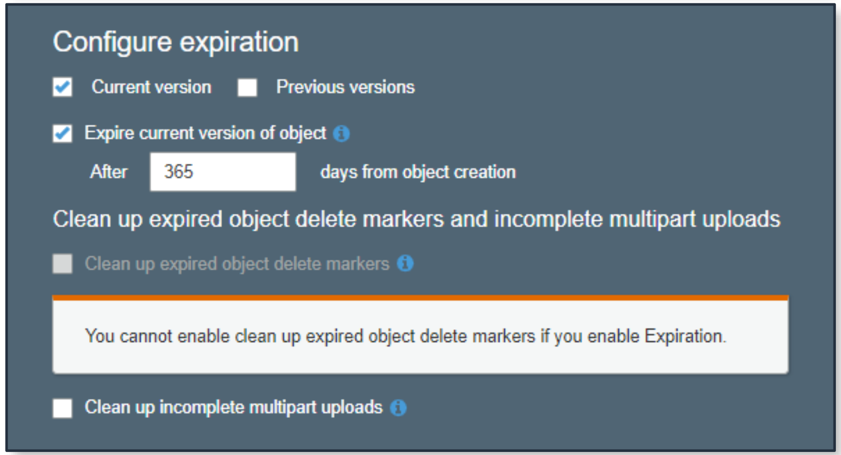
☑ Current version □ Previous versions

For current versions of objects + Add transition

Object creation: Select a transition

Days after creation: days X

Transition to Standard-IA after
Transition to Intelligent-Tiering after
Transition to One Zone-IA after
Transition to Amazon Glacier after



### Configure expiration

☑ Current version □ Previous versions

☑ Expire current version of object

After 365 days from object creation

Clean up expired object delete markers and incomplete multipart uploads

□ Clean up expired object delete markers

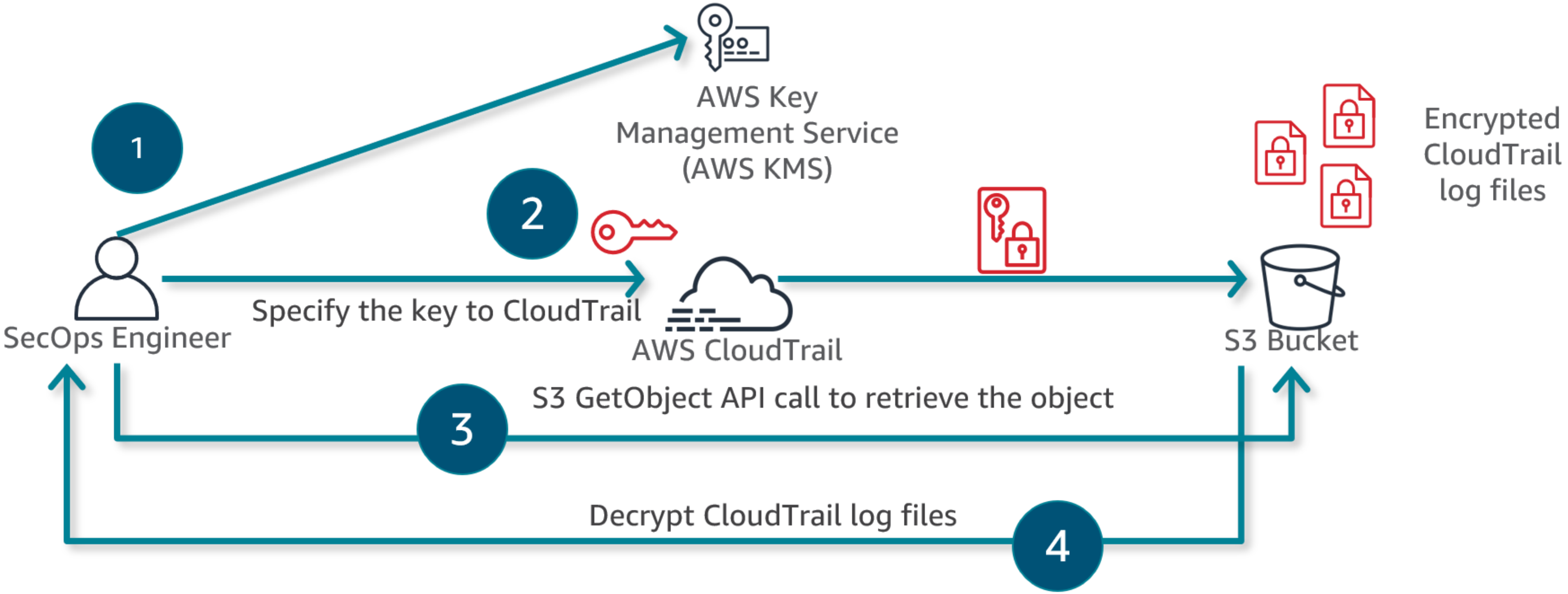
You cannot enable clean up expired object delete markers if you enable Expiration.

□ Clean up incomplete multipart uploads

# CloudTrail confidentiality: AWS KMS encryption

## Best practice

- Create or use an existing AWS Key Management Service (KMS) key and apply key policy to allow CloudTrail to encrypt and SecOps engineers to decrypt.


AWS Key Management Service (AWS KMS)
1
2
Encrypted CloudTrail log files
SecOps Engineer
Specify the key to CloudTrail
AWS CloudTrail
S3 Bucket
3
S3 GetObject API call to retrieve the object
Decrypt CloudTrail log files
4

# Enable log integrity validation

## Best practice

Once you turn on log file integrity validation, CloudTrail will start delivering digest files on an hourly basis to the same S3 bucket where you receive your CloudTrail log files, but with a different prefix.

- CloudTrail log files are delivered to: /optional_prefix/AWSLogs/AccountID/CloudTrail/*
- CloudTrail digest files are delivered to: /optional_prefix/AWSLogs/AccountID/CloudTrail-Digest/*

# Integrate with CloudWatch Logs

## Best practices

- Monitor and alert on specific events.
- Simple searching is provided.
- Use AWS Config to ensure CloudTrail is sending events to CloudWatch Logs.

# Visibility with Amazon CloudWatch

# Indicators of compromise

- Abnormal CPU utilization
- Significant or sudden increases in database reads
- HTML response sizes
- Mismatched port-application traffic
- Unusual DNS requests
- Unusual outbound network traffic
- Anomalies in privileged user account activity
- Geographical irregularities (source of traffic)
- Unusually high traffic at irregular hours
- Multiple, repeated, or irregular login attempts

# CloudWatch Alarms best practices

These are just a few examples of areas that should be monitored with CloudWatch Alarms:

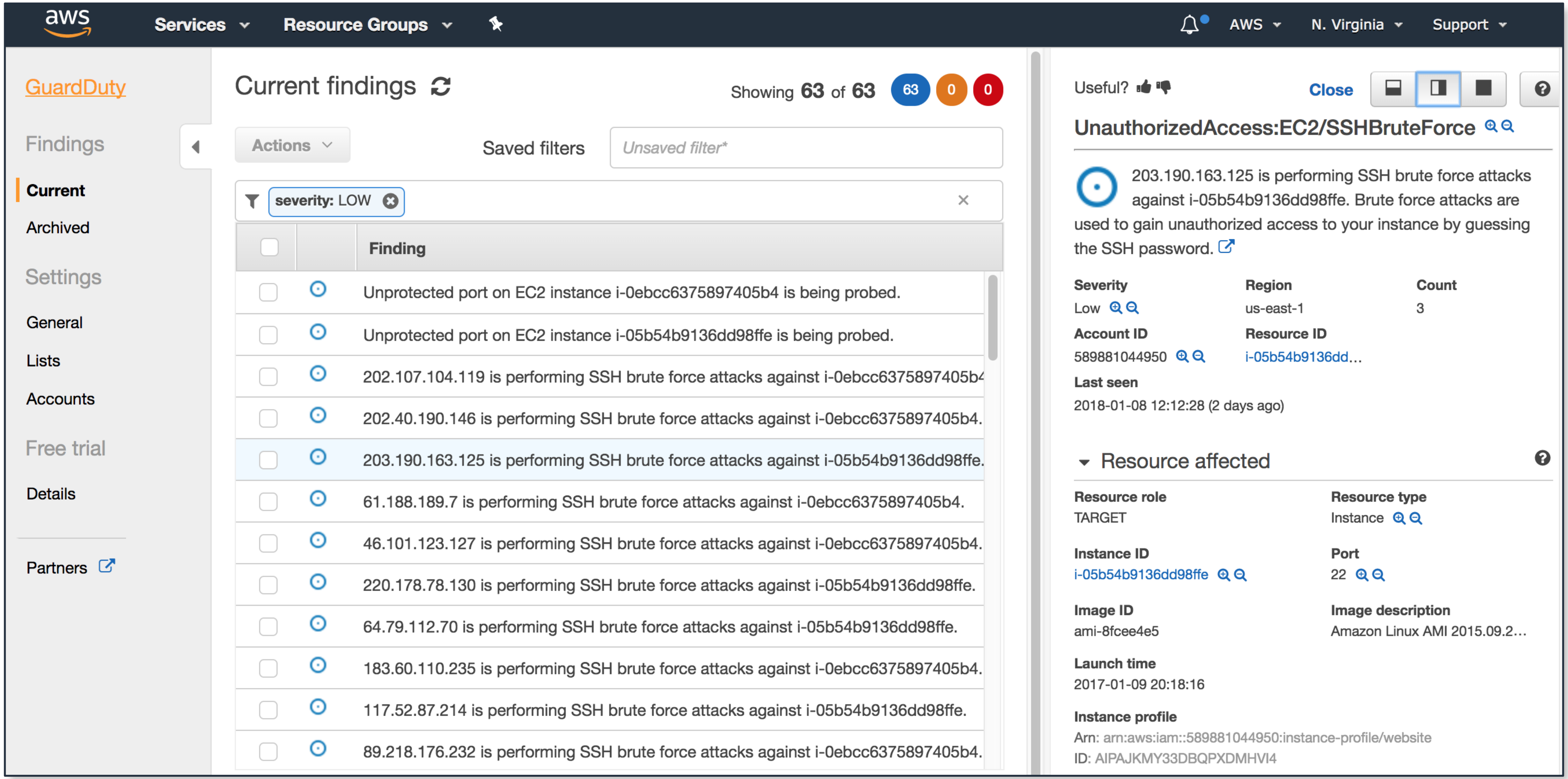
- AWS Console sign-In requests without MFA
- IAM policy configuration changes
- Root account usage
- Authorization failures; unauthorized API calls made within your AWS account
- AWS KMS key configuration changes
- AWS CloudTrail configuration changes
- AWS EC2 instance and S3 changes
- AWS VPC, Route table, Internet Gateway, ACLs or security group configuration changes

# Enhancing monitoring and alerting

# Detect with: Amazon GuardDuty

- One-click activation without architectural or performance impact
- Continuous monitoring of AWS accounts and resources
- Instant On provides findings in minutes
- No agents, no sensors, no network appliances
- Global coverage, regional results
- Built-in anomaly detection with machine learning
- Partner integrations for additional protections

# GuardDuty: Findings



Current findings

Showing 63 of 63 63 0 0

Saved filters Unsaved filter*

severity: LOW

| □ | | Finding |
|---|---|---|
| □ | ⊙ | Unprotected port on EC2 instance i-0ebcc6375897405b4 is being probed. |
| □ | ⊙ | Unprotected port on EC2 instance i-05b54b9136dd98ffe is being probed. |
| □ | ⊙ | 202.107.104.119 is performing SSH brute force attacks against i-0ebcc6375897405b4 |
| □ | ⊙ | 202.40.190.146 is performing SSH brute force attacks against i-0ebcc6375897405b4. |
| □ | ⊙ | 203.190.163.125 is performing SSH brute force attacks against i-05b54b9136dd98ffe. |
| □ | ⊙ | 61.188.189.7 is performing SSH brute force attacks against i-0ebcc6375897405b4. |
| □ | ⊙ | 46.101.123.127 is performing SSH brute force attacks against i-0ebcc6375897405b4. |
| □ | ⊙ | 220.178.78.130 is performing SSH brute force attacks against i-05b54b9136dd98ffe. |
| □ | ⊙ | 64.79.112.70 is performing SSH brute force attacks against i-05b54b9136dd98ffe. |
| □ | ⊙ | 183.60.110.235 is performing SSH brute force attacks against i-0ebcc6375897405b4. |
| □ | ⊙ | 117.52.87.214 is performing SSH brute force attacks against i-05b54b9136dd98ffe. |
| □ | ⊙ | 89.218.176.232 is performing SSH brute force attacks against i-0ebcc6375897405b4. |

UnauthorizedAccess:EC2/SSHBruteForce

203.190.163.125 is performing SSH brute force attacks against i-05b54b9136dd98ffe. Brute force attacks are used to gain unauthorized access to your instance by guessing the SSH password.

**Severity** Low

**Region** us-east-1

**Count** 3

**Account ID** 589881044950

**Resource ID** i-05b54b9136dd…

**Last seen** 2018-01-08 12:12:28 (2 days ago)

Resource affected

**Resource role** TARGET

**Resource type** Instance

**Instance ID** i-05b54b9136dd98ffe

**Port** 22

**Image ID** ami-8fcee4e5

**Image description** Amazon Linux AMI 2015.09.2…

**Launch time** 2017-01-09 20:18:16

**Instance profile**
Arn: arn:aws:iam::589881044950:instance-profile/website
ID: AIPAJKMY33DBQPXDMHVI4

# Manage and remediate with: AWS Security Hub

- Managed AWS service
- Consolidates and aggregates findings.
- Provides controls for the following standards:
  - Center for Internet Security (CIS) AWS Foundations
  - Payment Card Industry Data Security Standard (PCI DSS)
  - AWS Foundational Security Best Practices
- Integrates with ticketing, chat, incident management, investigation, GRC, SOAR, and SIEM tools.

# Remediation with Security Hub

## Manual remediation

- This is best for anything that has the potential to impact business objectives. This type of intervention is slower, but notifications can help expedite response.
- This option should also be used to test newly created automatic remediations before they are put into a production environment.

## Automatic remediation

- This is best when there is a low risk of a negative impact to the workloads in the account.
- For example, you would not use an automatic remediation that stops an EC2 instance responsible for a business-critical function.

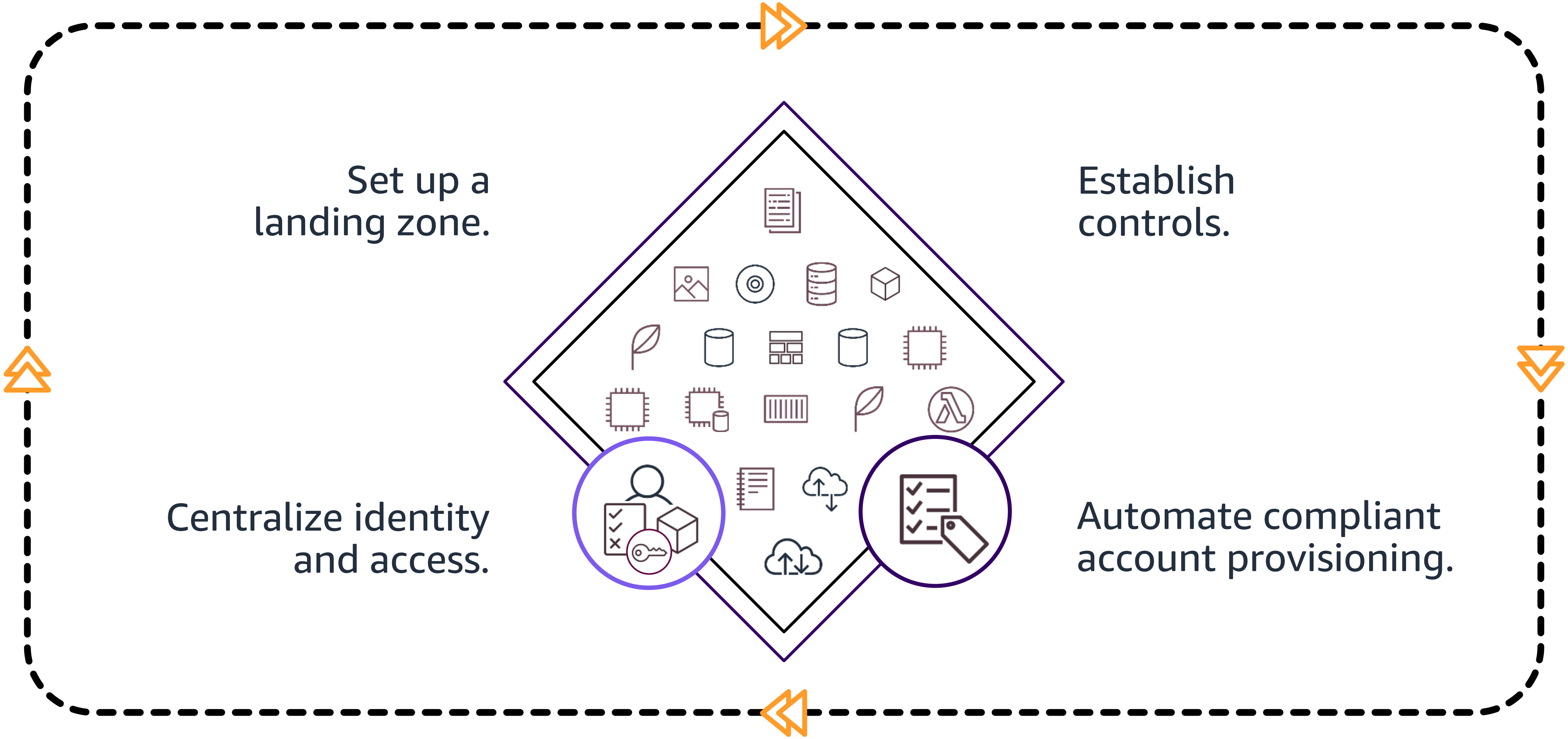
# Account Provisioning w/Control Tower

# AWS Control Tower

- Use AWS Control Tower to set up a multi-account AWS solution
- Solutions based on best practices that have been identified after working with thousands of enterprises.
- Built on trusted and reliable AWS services, to include AWS Service Catalog, AWS Single Sign-On, and AWS Organizations.

# Implement with Governance at Scale

## Manage governance at scale



Set up a landing zone.

Establish controls.

Centralize identity and access.

Automate compliant account provisioning.

# Set up an AWS Control Tower landing zone

Section Topic

# Governance with a landing zone

- AWS Control Tower provides the easiest way to set up and govern a secure, multi-account AWS environment called a landing zone.
- AWS Control Tower creates your landing zone using AWS Organizations.
  - Supports ongoing account management.
  - Implements best practices.

# General setup considerations

**AWS account**

Create an AWS account and lock away the root credentials

**Management account**

Create a management account and a user with administrator permissions

**MFA**

Enable multi-factor authentication on all management account users

**Shared accounts**

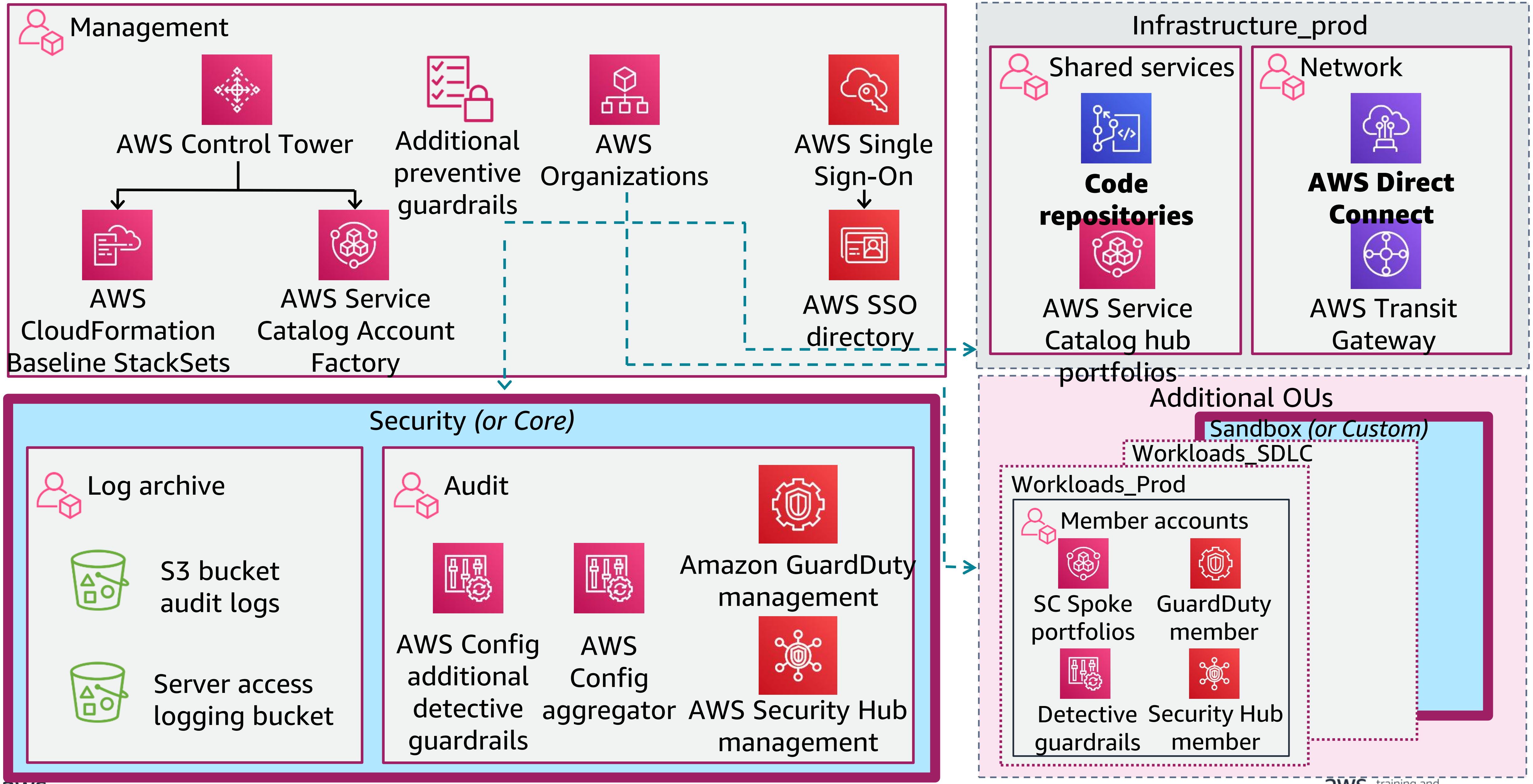
Create unique email addresses to provision audit and log accounts

# Existing organization considerations

After setting up the landing zone, an existing AWS organization has an environment that:

- Promotes unified billing
- Centralizes account management
- Manages guardrails for new and existing accounts

# Landing zone reference architecture



Management

AWS Control Tower

AWS CloudFormation Baseline StackSets

AWS Service Catalog Account Factory

Additional preventive guardrails

AWS Organizations

AWS Single Sign-On

AWS SSO directory

Infrastructure_prod

Shared services

**Code repositories**

AWS Service Catalog hub portfolios

Network

**AWS Direct Connect**

AWS Transit Gateway

Security *(or Core)*

Log archive

S3 bucket audit logs

Server access logging bucket

Audit

AWS Config additional detective guardrails

AWS Config aggregator

Amazon GuardDuty management

AWS Security Hub management

Additional OUs

Sandbox *(or Custom)*

Workloads_SDLC

Workloads_Prod

Member accounts

SC Spoke portfolios

GuardDuty member

Detective guardrails

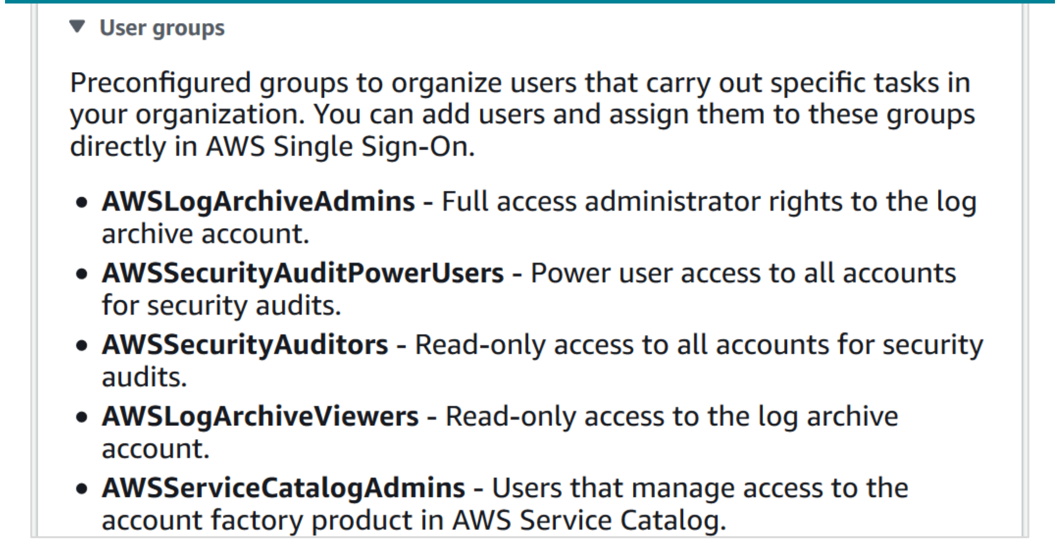
Security Hub member

# Centralize identity and access management

Section Topic

# Default AWS IAM Identity Center

## Default user groups example

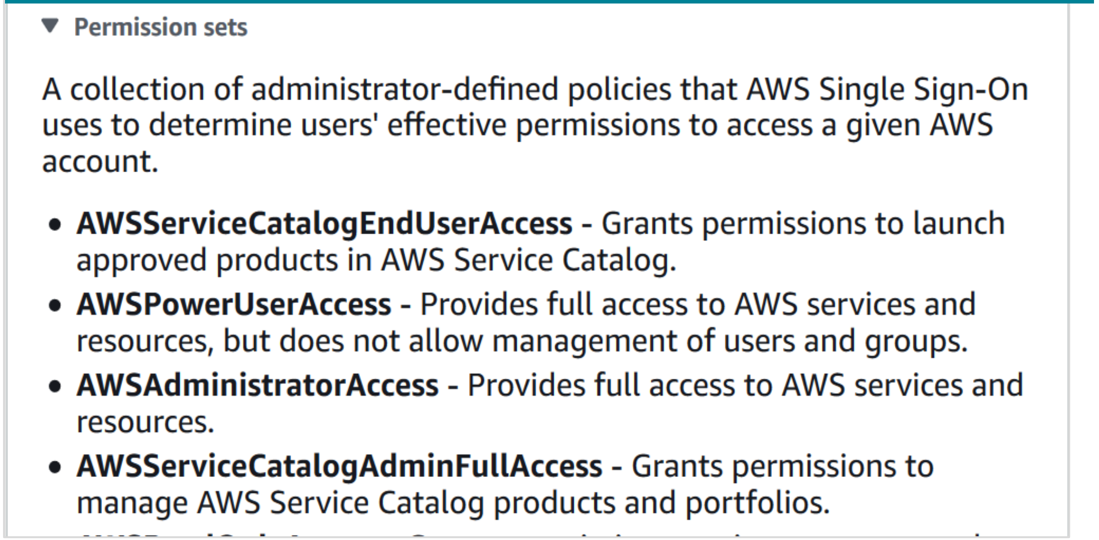


▼ User groups

Preconfigured groups to organize users that carry out specific tasks in your organization. You can add users and assign them to these groups directly in AWS Single Sign-On.

- **AWSLogArchiveAdmins** - Full access administrator rights to the log archive account.
- **AWSSecurityAuditPowerUsers** - Power user access to all accounts for security audits.
- **AWSSecurityAuditors** - Read-only access to all accounts for security audits.
- **AWSLogArchiveViewers** - Read-only access to the log archive account.
- **AWSServiceCatalogAdmins** - Users that manage access to the account factory product in AWS Service Catalog.

## Preconfigured permission sets example



▼ Permission sets

A collection of administrator-defined policies that AWS Single Sign-On uses to determine users' effective permissions to access a given AWS account.

- **AWSServiceCatalogEndUserAccess** - Grants permissions to launch approved products in AWS Service Catalog.
- **AWSPowerUserAccess** - Provides full access to AWS services and resources, but does not allow management of users and groups.
- **AWSAdministratorAccess** - Provides full access to AWS services and resources.
- **AWSServiceCatalogAdminFullAccess** - Grants permissions to manage AWS Service Catalog products and portfolios.

# AWS IAM Identity Center

## User group provisioning

- Create users and groups directly in IAM Identity Center.
- Before IAM Identity Center can be used to assign user-access and group-access permissions in an AWS account, it must first be aware of the users and groups.

## Federated access

- IAM Identity Center supports identity federation with SAML (Security Assertion Markup Language).
- IAM Identity Center uses this information to provide federated single sign-on access for users authorized to use applications within AWS access portal.

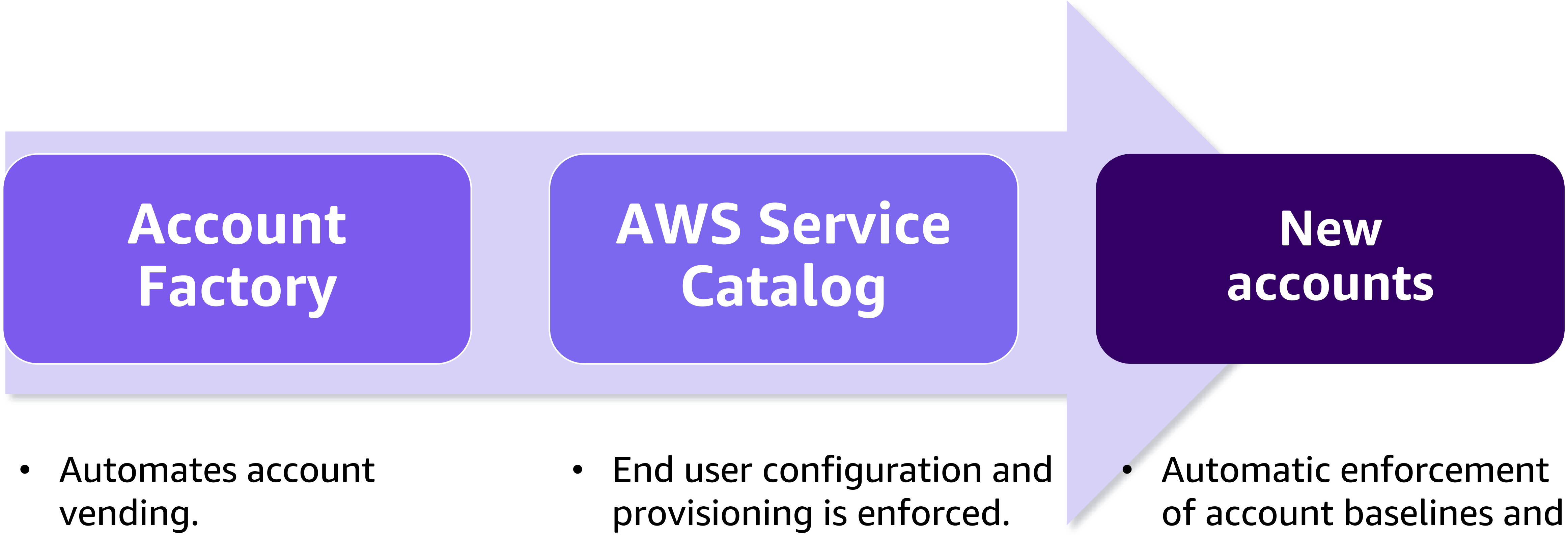
# External Identity Providers

AWS Control Tower uses AWS IAM Identity Center for federated access.

- SAML doesn’t allow querying IdPs.
- IAM Identity Center must be made aware of users and groups.

- Automatically provision when the provider supports System for Cross-domain Identity Management (SCIM).
- Manually provision when the IdP doesn’t support SCIM. With this, both the group and user must be manually managed in the AWS IAM Identity Center console.

# Account Factory and provisioning



Account Factory

AWS Service Catalog

New accounts

- Automates account vending.
- Prevents account sprawl.

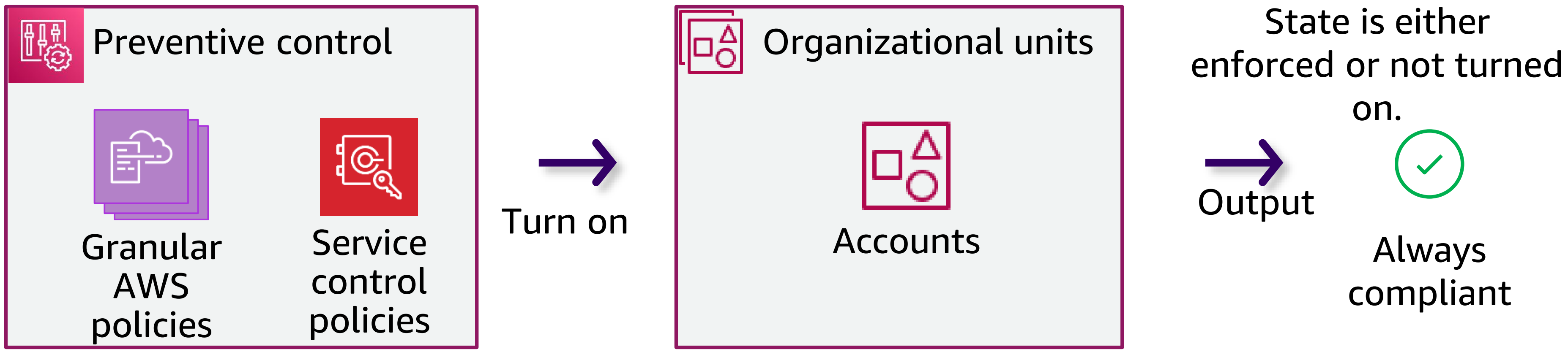
- End user configuration and provisioning is enforced.

- Automatic enforcement of account baselines and controls.

# Control types

## Preventive

- Preventive controls make sure accounts maintain compliance.



Preventive control

Granular AWS policies

Service control policies

Turn on

Organizational units

Accounts

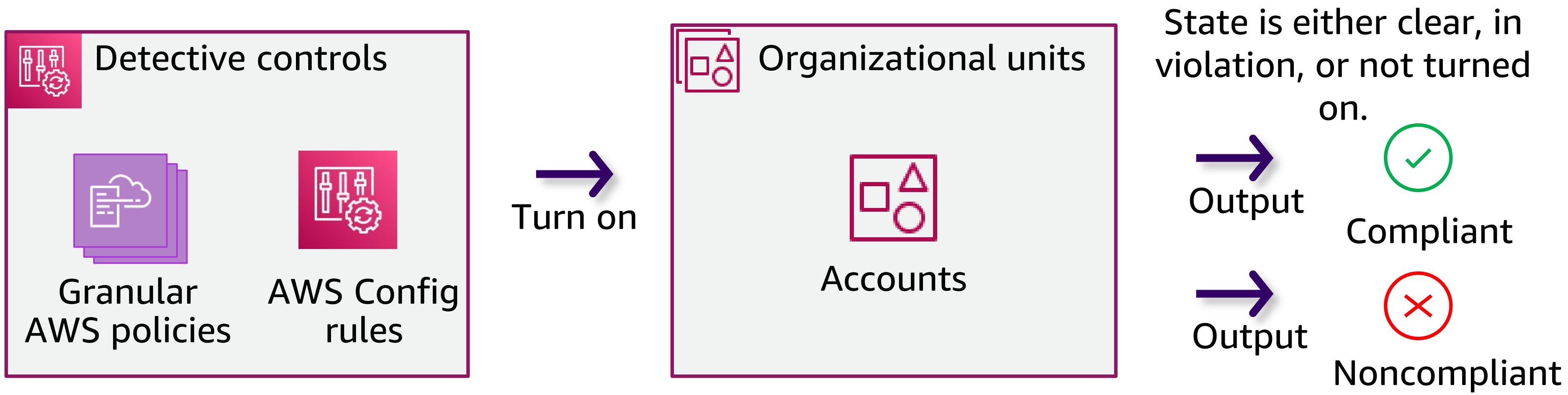
State is either enforced or not turned on.

Output

Always compliant

## Detective

- Detective controls detect noncompliance of resources in accounts.



Detective controls

Granular AWS policies

AWS Config rules

Turn on

Organizational units

Accounts

State is either clear, in violation, or not turned on.

Output

Compliant

Output

Noncompliant

# Course conclusion

AWS Security Best Practices

# Thank you

Corrections, feedback, or other questions? Contact us at https://support.aws.amazon.com/#/contacts/aws-training. All trademarks are the property of their owners.