

AWS Certified Security Specialty



Gagandeep Singh

Introduction

Name

Total Experience

Background – Development / Infrastructure / Database / Network

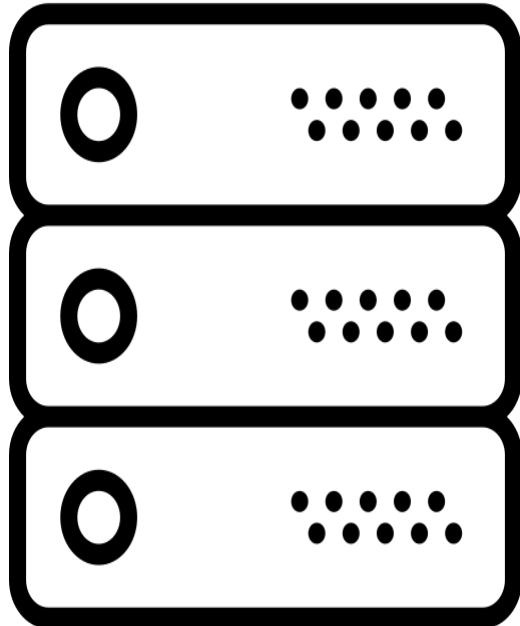
Experience on Cloud/AWS

Your expectations from this training

Cloud

What is Cloud?

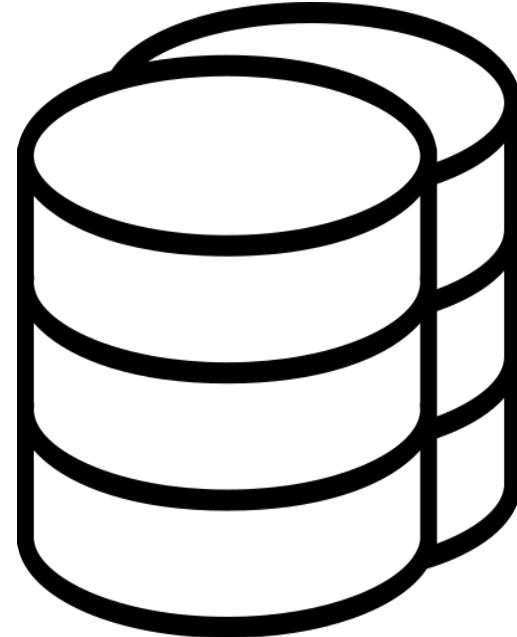
Traditional Datacenters



Servers

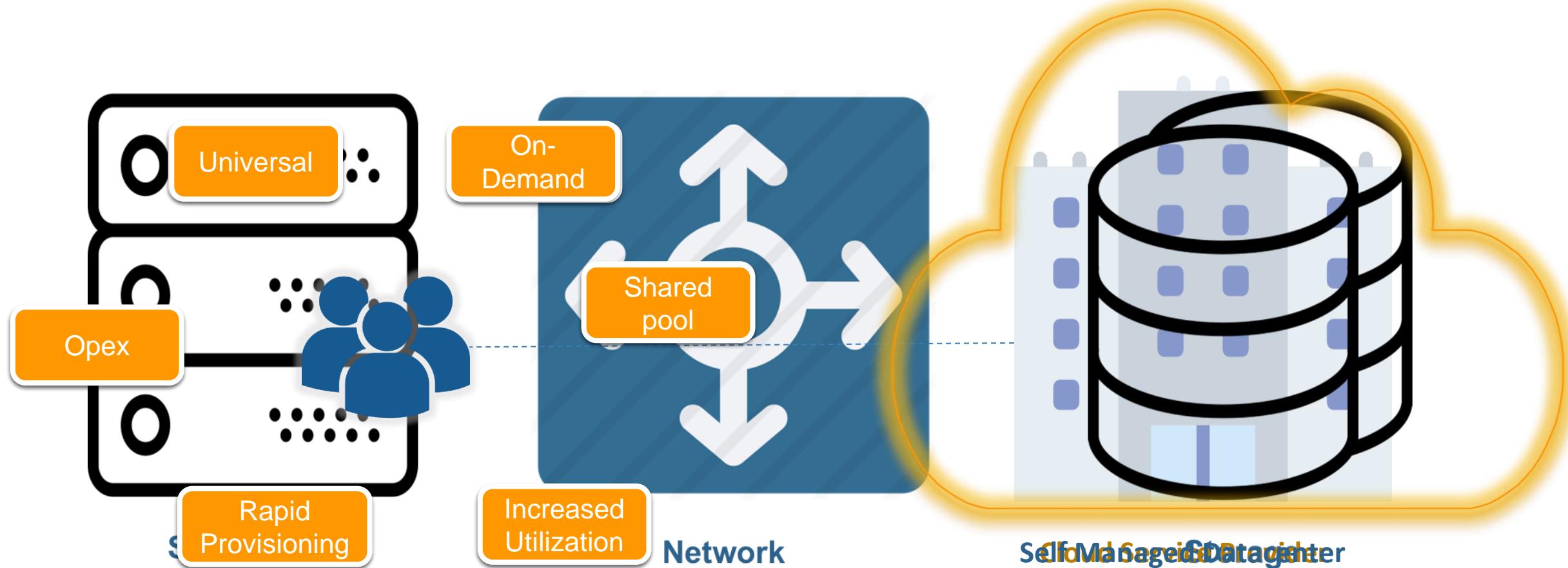


Network



Storage

Traditional Datacenters



Cloud Definition

CLOUD COMPUTING definition by NIST SP 800-145

“Cloud computing is a model for enabling **ubiquitous, convenient on-demand network access** to a **shared** pool of configurable **computing resources** (e.g., networks, server, storage, applications, and services) that can be **rapidly provisioned** and **released** with **minimal management effort** or service provider interaction.”

CLOUD COMPUTING definition by ISO/IEC 17788:2014(en)

“Cloud Computing: Paradigm for enabling **network access** to a **scalable** and **elastic** pool of **shareable physical or virtual resources** with **self-service provisioning** and **administration on-demand**.”

* ISO/IEC 17788:2014 talks about “Information technology — Cloud computing — Overview and vocabulary”

Characteristics of Cloud



Cloud Deployment Types



Service Models - CARaaS



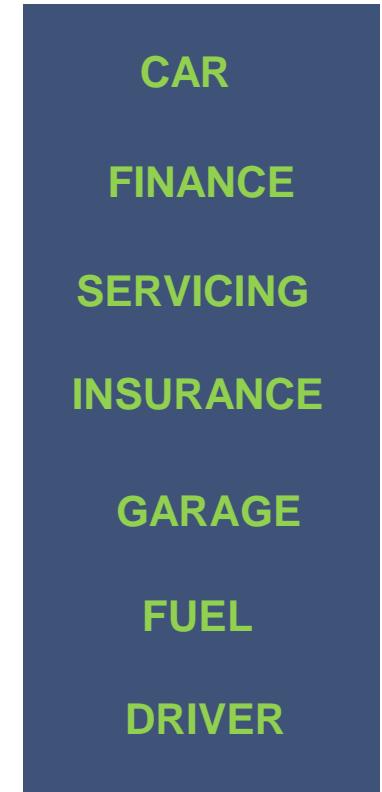
Car Owned



Car Leased



Car Hired



Taxi



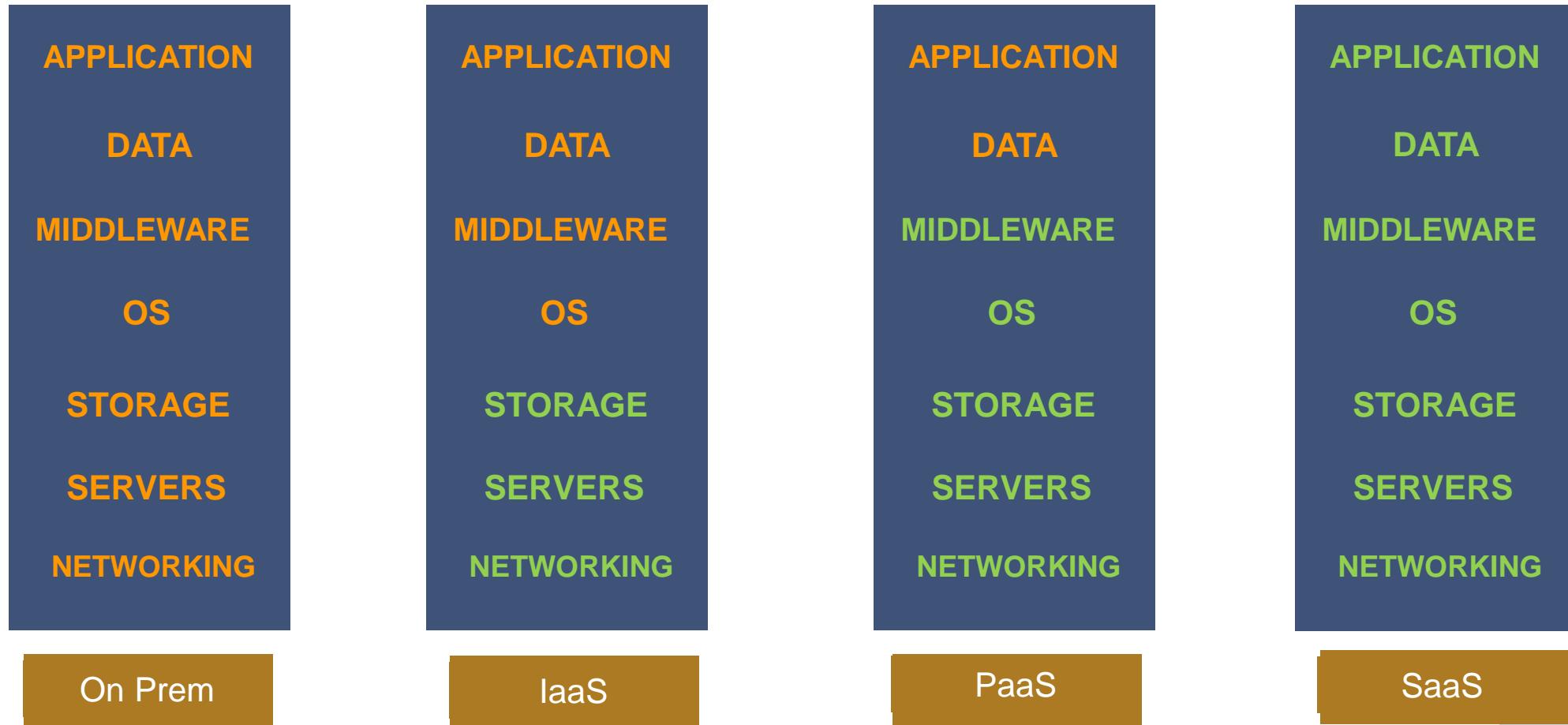
Managed By User



Managed By Service Provider

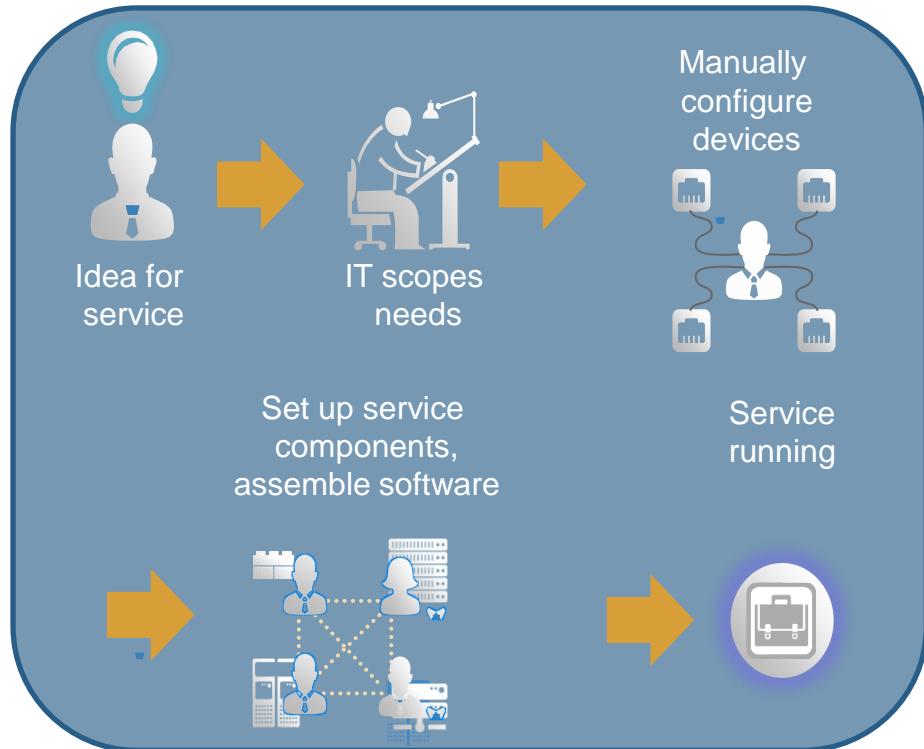
Copyrights © TechLanders Solutions 2020-21

Service Models



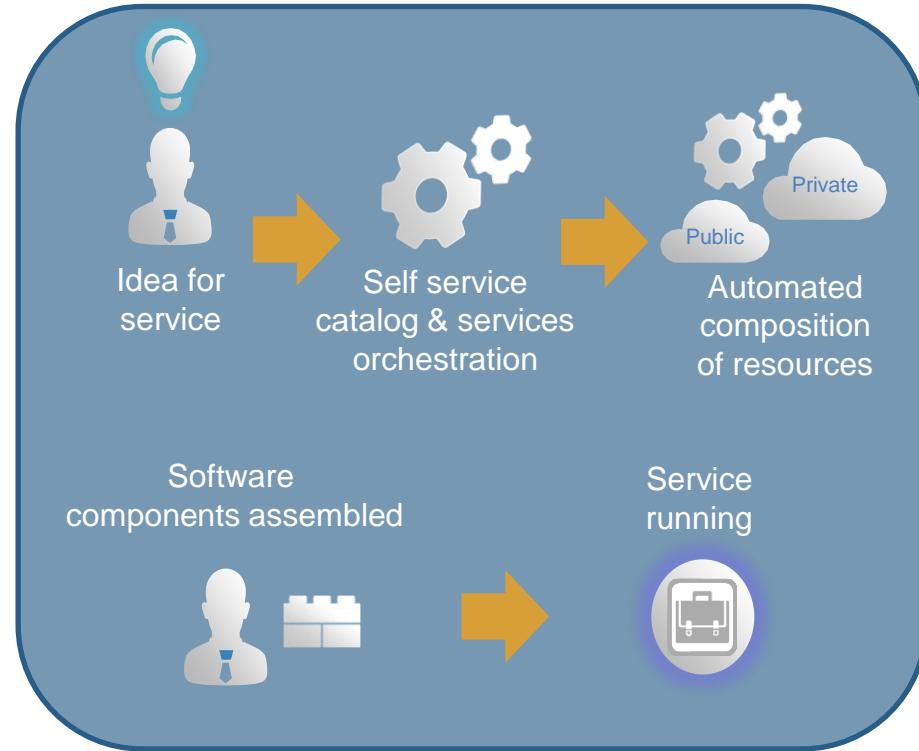
Business Impact

Traditional Datacenter



Time to Provision New Service: Months

Cloud Infrastructure



Time to Provision New Service: Minutes

Cloud Benefits

Six advantages

Stop
guessing capacity

Focus on
business
differentiators

Global in
minutes

Variable vs.
capital expense

Economies
of scale

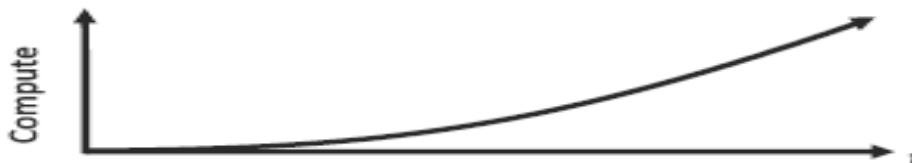
Increase
speed and agility

Cloud Major Use Cases



On and Off

On and off workloads (e.g. batch job)
Over provisioned capacity is wasted
Time to market can be cumbersome



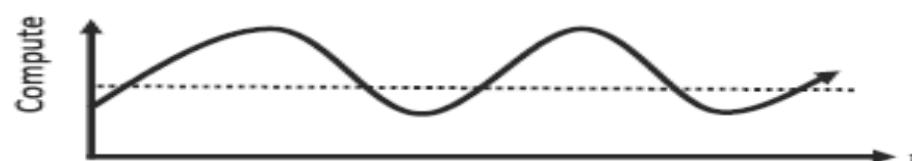
Growing Fast

Successful services needs to grow/scale
Keeping up with growth is a big IT challenge
Cannot provision hardware fast enough



Unpredictable Bursting

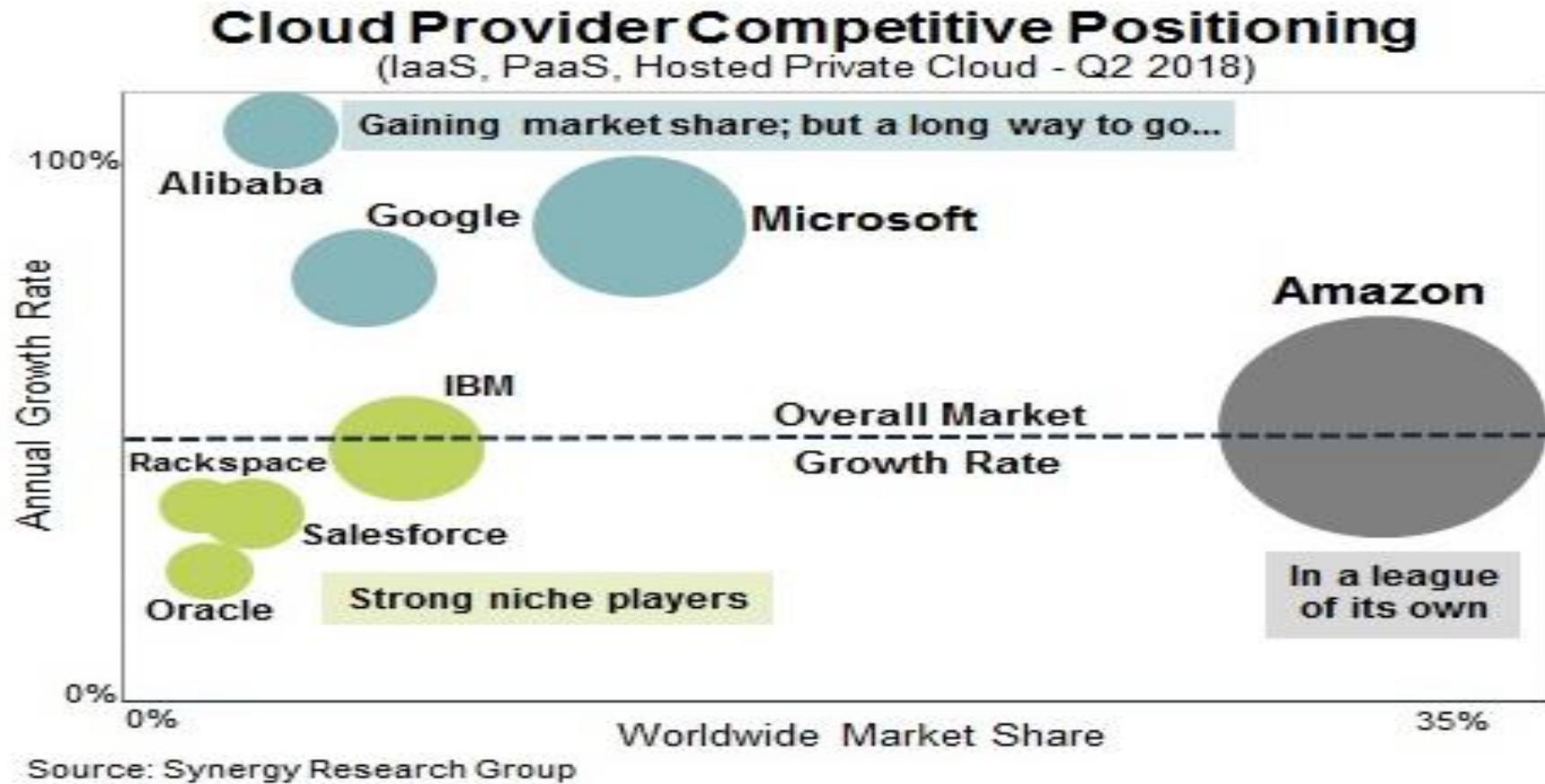
Unexpected/unplanned peak in demand
Sudden spike impacts performance
Cannot over provision for extreme cases



Predictable Bursting

Services with micro seasonality trends
Peaks due to periodic increased demand
IT complexity and wasted capacity

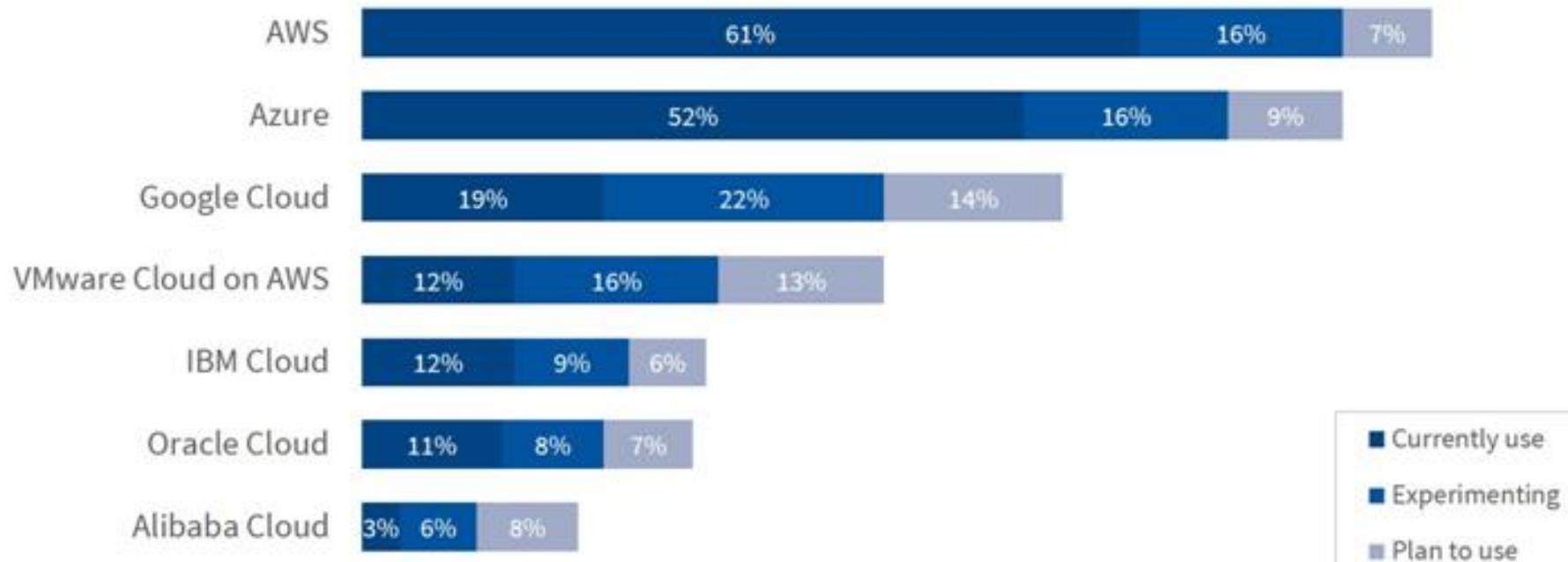
Cloud Players



Cloud Players

Public Cloud Adoption

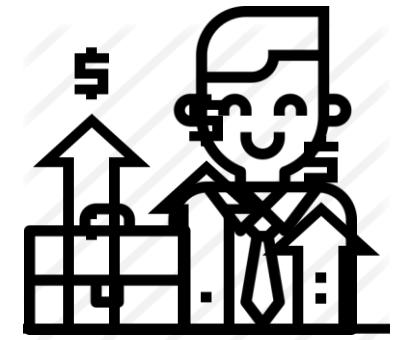
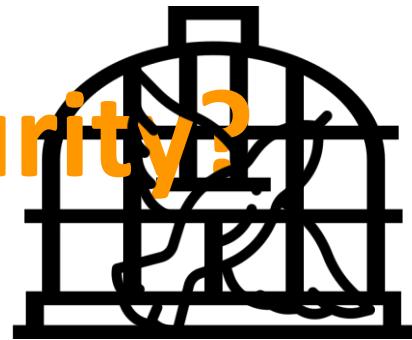
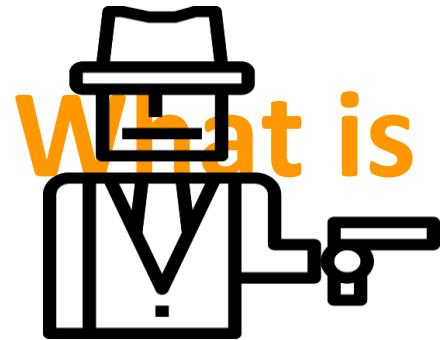
% of All Respondents



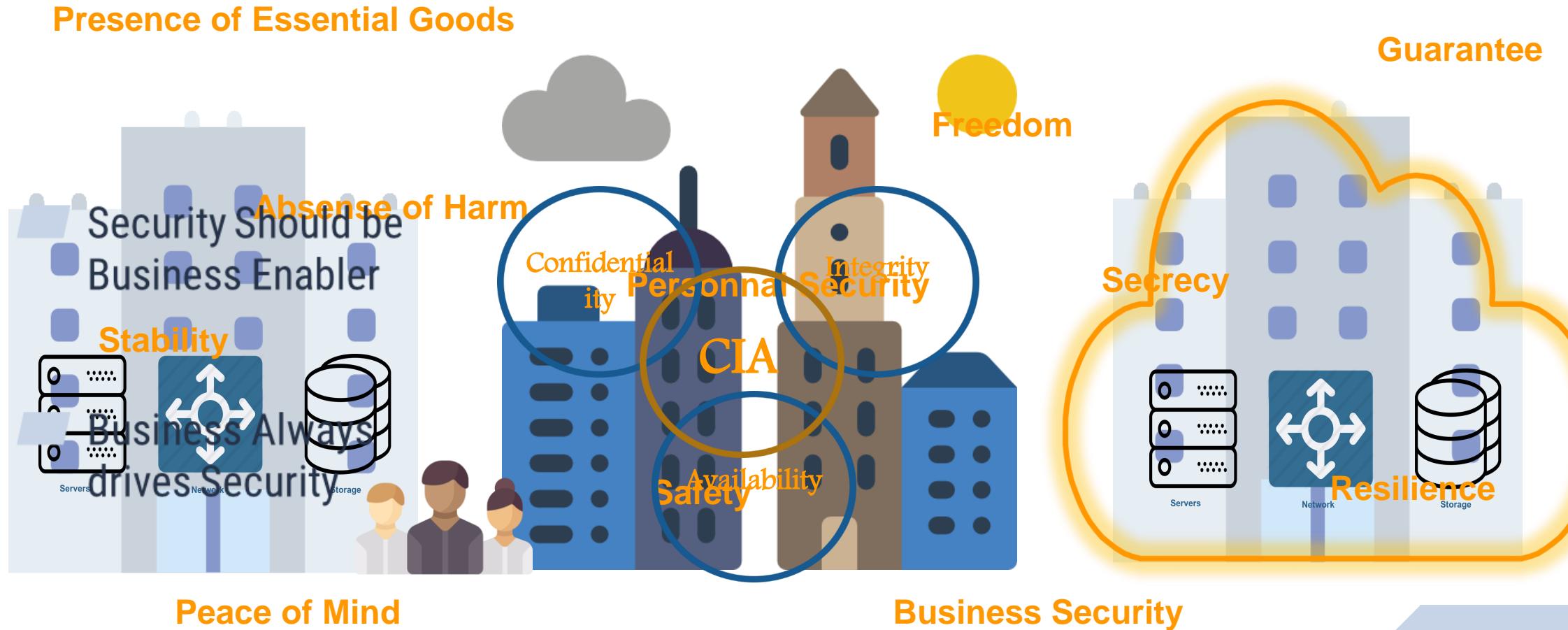
Source: RightScale 2019 State of the Cloud Report from Flexera

Security

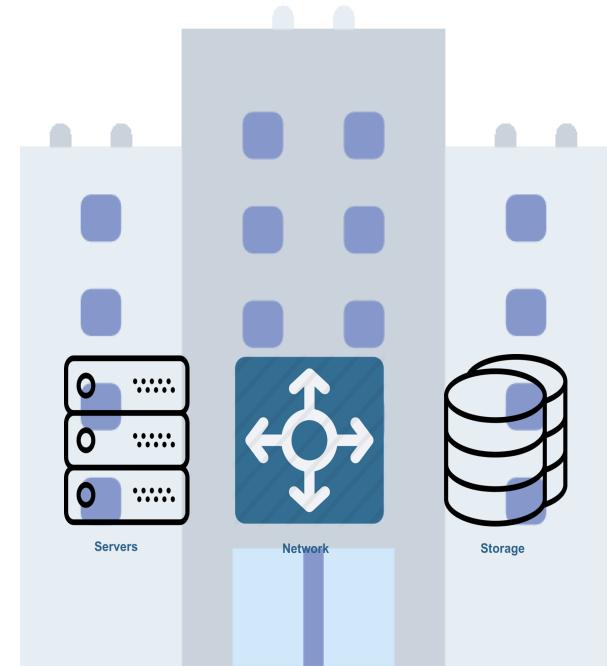
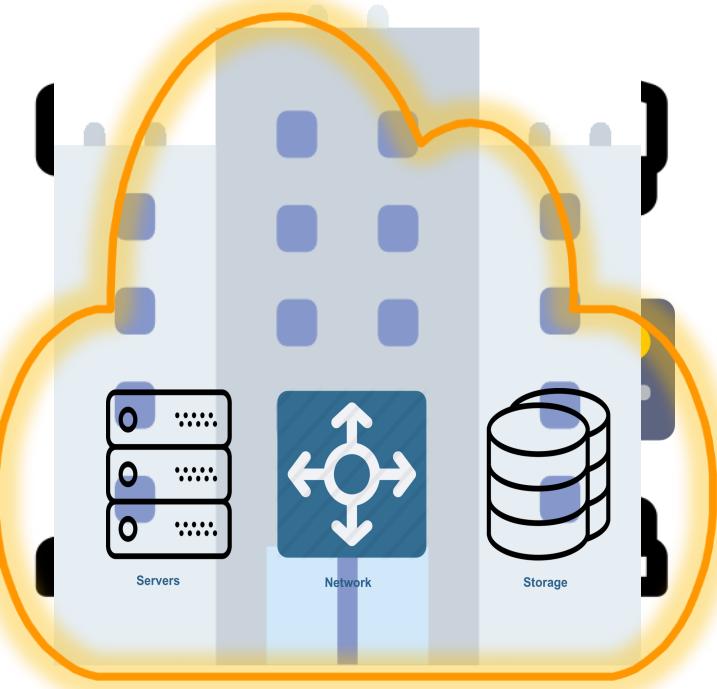
Security



Cybersecurity



CloSedSecurity



Why Security?



Microsoft



YAHOO!

SMBs would either go down for at least a day f



completely, or be forced to pay up to \$1,000 each. - VIPRE, October 2017



Financial Loss

Brand Loss

Legal Issues

Cyber-Attacks Costed Global Firms in 2018. - Online Trust Alliance
Jul 2019



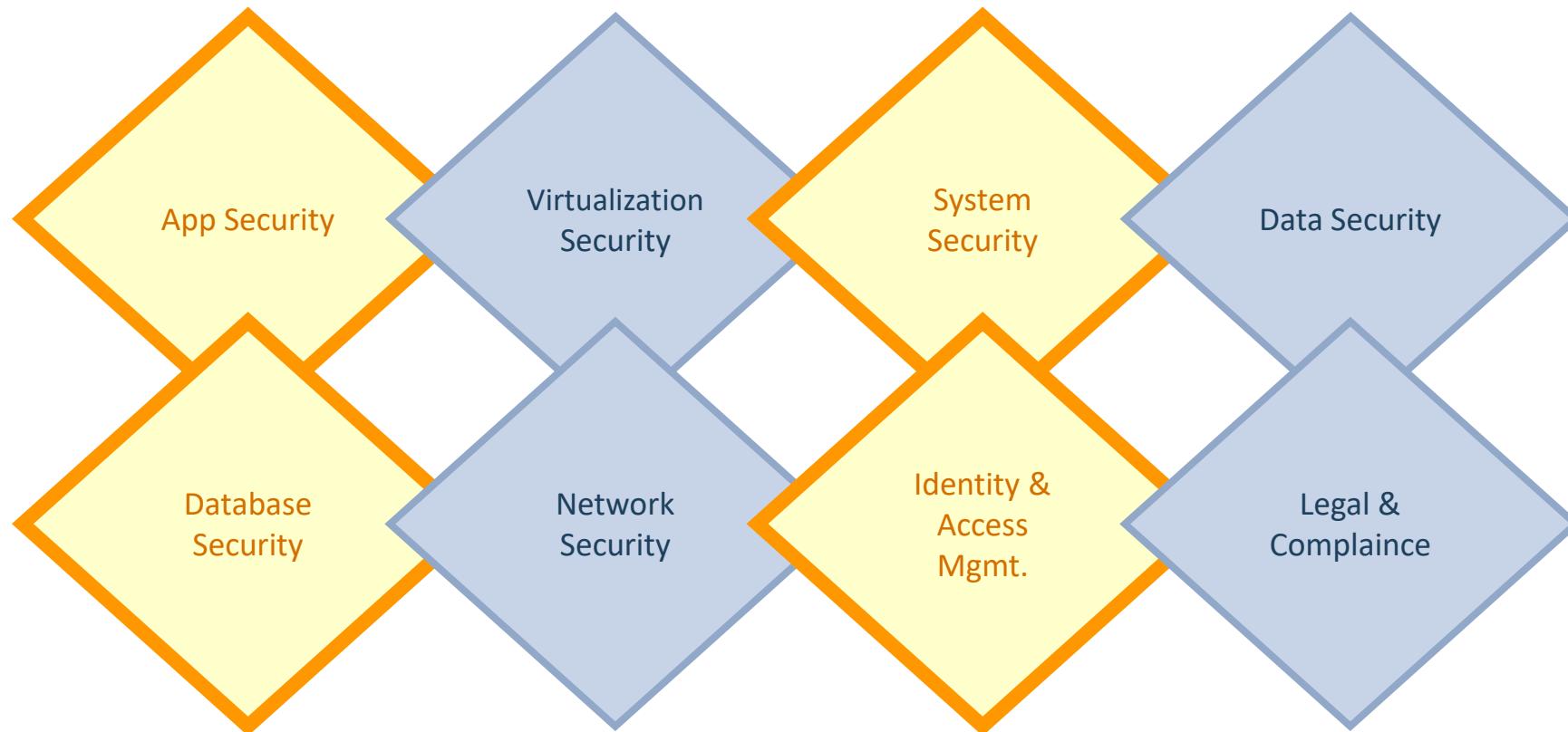
iCloud



s © Techlanders Solutions 2020-21

shutterstock.com • 282012710

Security Aspects



Cloud Cross-Cutting aspects

(ISO/IEC 17789)

- Auditability (clause 8.5.2);
- Availability (clause 8.5.3);
- Governance (clause 8.5.4);
- Interoperability (clause 8.5.5);
- Maintenance and Versioning (clause 8.5.6);
- Performance (clause 8.5.7);
- Portability (clause 8.5.8);
- Protection of Personally Identifiable Information (clause 8.5.9);
- Regulatory;
- Resiliency (clause 8.5.10);
- Reversibility (clause 8.5.11);
- Security (clause 8.5.12);
- Service Levels and Service Level Agreement (clause 8.5.13).

Security Challenges in Cloud

- **Virtualization** comes with its own challenges, where risk is always there for breach like vulnerabilities in different layers.
- **Shared Infra** with intruders
- **Auditing restrictions**
- **Multi-tenancy**
- **Compliance** and **Regulation** issues due to data logical placement, breaches etc.
- Conflict of Responsibility
- Vendor **Lock-in** and **Vendor-Lockout**
- Internal and External threats
- Much more.. will be discussing in further modules in details

Few Security Concepts in Cloud

- **Encryption & Key Management:** The ever green and most trusted method of security in cloud computing. Encryption can be at Rest or at Motion.
- **Access Control** through Identity and Access Management.
- **Data & Media Sanitization** – Data erasure, Formatting, Disk deletion are not the full-proof method of sanitization. Physical deletion is not possible in cloud. So the alternate left is **cryptographic eraser**.
- **Network Security:** With the help of VPC, Subnetting, NACL & Security groups & Egress monitoring.
- **Application Security:** With WAF & Physical Firewalls.
- **Virtualization security and hardening** at different layers.
- **OS based** security hardening
- **Data Security:** Masking, Obfuscation, Anonymization, Tokenization
- **Auditing:** Logging and Monitoring

AWS

Amazon Web Services

AWS (Amazon Web Services) is a group of web services (also known as cloud services) being provided by Amazon since 2006.

AWS provides huge list of services starting from basic IT infrastructure like CPU, Storage as a service, to advance services like Database as a service, Serverless applications, IOT, Machine Learning services etc..

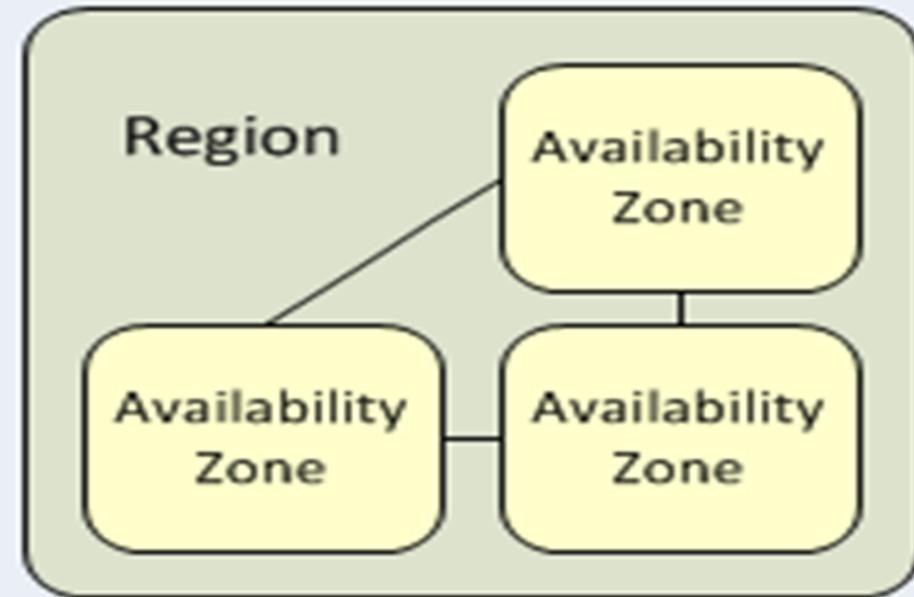
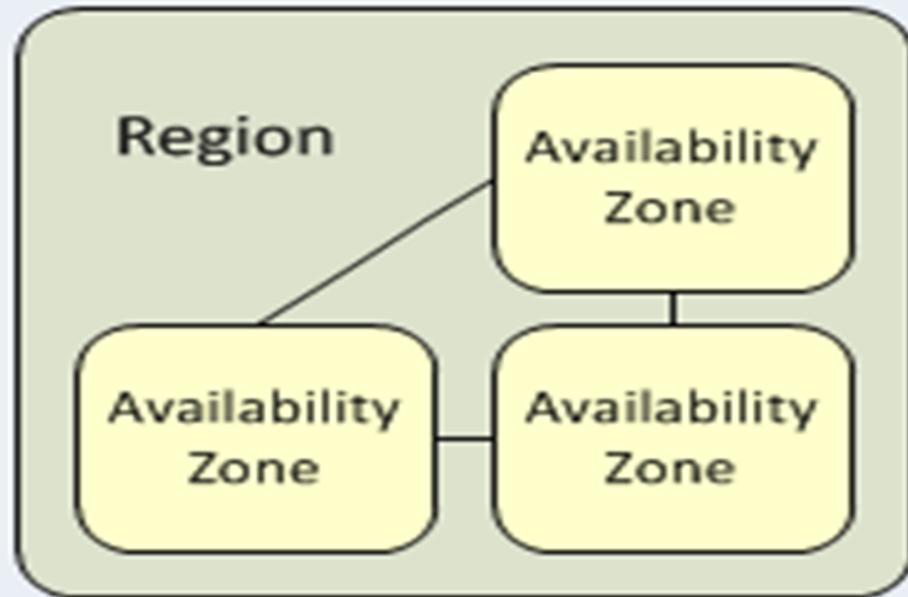
Hundreds of instances can be build and use in few minutes as and when required, which saves ample amount of hardware cost for any organizations and make them efficient to focus on their core business areas.

Currently AWS is present and providing cloud services in more than 190 countries.

Well-known for IaaS, but now growing fast in PaaS and SaaS.

Amazon Web Services

Amazon Web Services



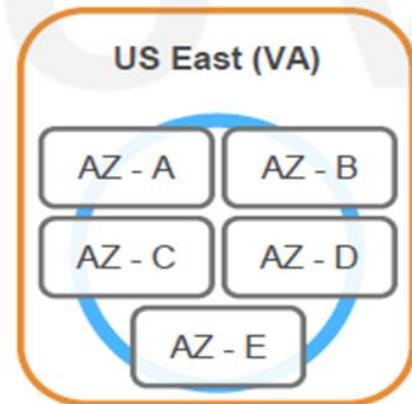
Amazon Web Services

At least 2 AZs per region.

Examples:

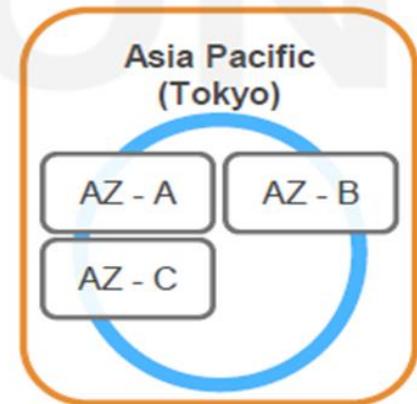
➤ US East (N. Virginia)

- us-east-1a
- us-east-1b
- us-east-1c
- us-east-1d
- us-east-1e



➤ Asia Pacific (Tokyo)

- ap-northeast-1a
- ap-northeast-1b
- ap-northeast-1c



Note: Conceptual drawing only. The number of Availability Zones (AZ) may vary.

Amazon Web Services

AWS Regions:

- Geographic Locations
- Consists of at least two Availability Zones(AZs)
- All of the regions are completely independent of each other with separate Power Sources, Cooling and Internet connectivity.
- This achieves the greatest possible fault tolerance and stability.
- There is a charge for data transfer between Regions.
- When you view your resources, you'll only see the resources tied to the Region you've specified.
- An AWS account provides multiple Regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.
- Resources aren't replicated across regions unless you do so specifically.

Amazon Web Services

AWS Availability Zones

- AZ is a distinct location within a region
- Each Availability Zone is isolated, but the Availability Zones in a Region are connected through low-latency links.
- Each Region has minimum two AZ's
- Most of the services/resources are replicated across AZs for HA/DR purpose.
- While launching instance you should specify an Availability Zone if your new instances must be close to, or separated from, your running instances.

Amazon Web Services

Current:

22 AWS Regions

69 AZs

Upcoming:

4 Regions

13 AZs

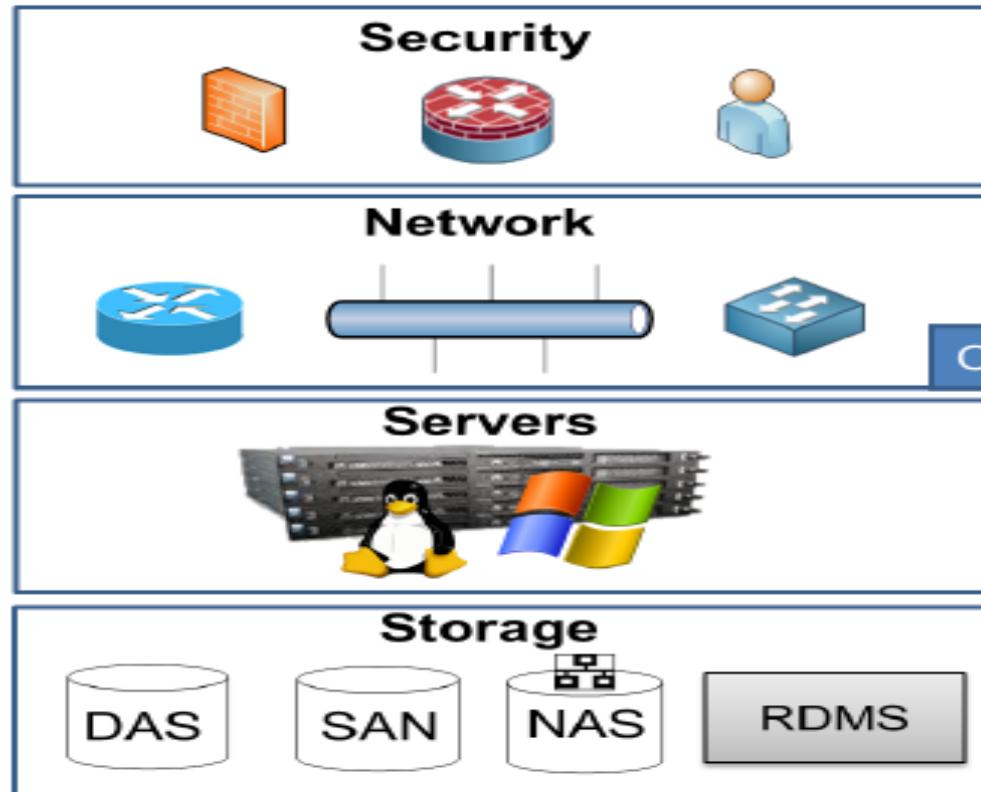


Amazon Web Services

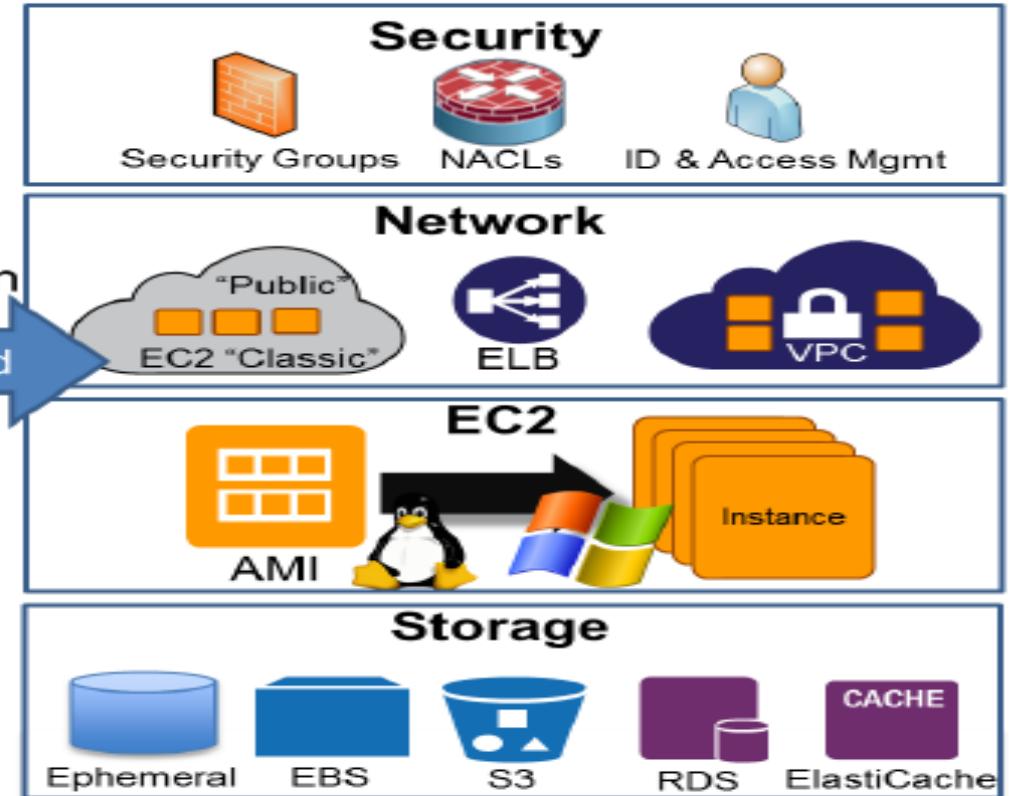


AWS

Enterprise Infrastructure



Amazon Web Services



Provision

On-Demand

Expand

Knowledge Checks

- To Achieve HA in AWS, my Servers should be in different (Racks, Datacenters, Availability Zones, Regions)?

- To Achieve DR/High Durability, where should I have by Server backup (Different AZ or Different Regions)?

Security Responsibilities

Security Concepts

Encryption & Key Management: The ever green and most trusted method of security in cloud computing. Encryption can be at Rest or at Motion.

Access Control through Identity and Access Management.

Data & Media Sanitization – Data erasure, Formatting, Disk deletion are not the full-proof method of sanitization. Physical deletion is not possible in cloud. So the alternate left is **cryptographic eraser**.

Network Security: With the help of VPC, Subnetting, NACL & Security groups & Egress monitoring.

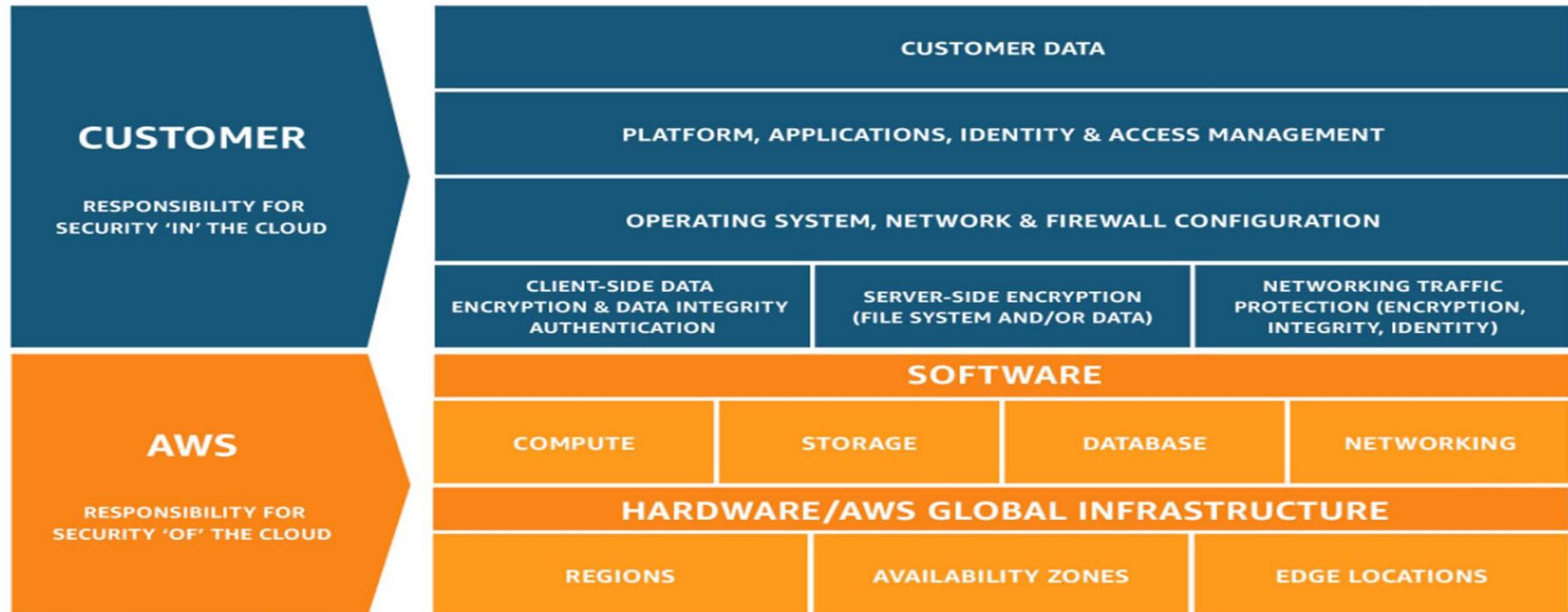
Application Security: With WAF & Physical Firewalls.

Virtualization security and hardening at different layers.

OS based security hardening

Data Security: Masking, Obfuscation, Anonymization, Tokenization

Shared Responsibility Model



Security Concepts

Inherited Controls – Controls which a customer fully inherits from AWS. I.e. Physical and Environmental controls.

Shared Controls – Controls which apply to both the infrastructure layer and customer layers, but in completely separate contexts or perspectives. In a shared control, AWS provides the requirements for the infrastructure and the customer must provide their own control implementation within their use of AWS services. Examples include:

Patch Management – AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest OS and applications.

Configuration Management – AWS maintains the configuration of its infrastructure devices, but a customer is responsible for configuring their own guest operating systems, databases, and applications.

Awareness & Training - AWS trains AWS employees, but a customer must train their own employees.

Customer Specific – Controls which are solely the responsibility of the customer based on the application they are deploying within AWS services. Examples include: Service and Communications Protection or Zone Security which may require a customer to route or zone data within specific security environments.

Security Responsibility

Responsibility	On-prem	IaaS	PaaS	SaaS
Data governance & rights management	Customer	Customer	Customer	Customer
Client endpoints	Customer	Customer	Customer	Customer
Account & access management	Customer	Customer	Customer	Customer
Identity & directory infrastructure	Customer	Customer	Microsoft	Microsoft
Application	Customer	Customer	Microsoft	Microsoft
Network controls	Customer	Customer	Microsoft	Microsoft
Operating system	Customer	Customer	Microsoft	Microsoft
Physical hosts	Customer	Microsoft	Microsoft	Microsoft
Physical network	Customer	Microsoft	Microsoft	Microsoft
Physical datacenter	Customer	Microsoft	Microsoft	Microsoft
█ Microsoft █ Customer				

Regardless of the deployment type, Client always retain responsibility for the following:

- Data
- Endpoints
- Accounts
- Access management

Regardless of the deployment type, Azure always retain responsibility for the following:

- Physical Host
- Physical Network
- Physical dataCenter

Security Considerations

Security In Implementation – Most Important (Everyone's responsibility)

Security Configuration Checks (Governance) – TL/Managers/PM/InfoSec Team

Security Incident Monitoring – Infosec Team/NOC Team

Legal, Compliance and Policies – Infosec and Legal Team

Security in IaaS Services

AWS Elastic Compute Cloud

- Amazon EC2 stands for Elastic Compute Cloud, and is the Primary AWS web service.
- Provides Resizable compute capacity
- Reduces the time required to obtain and boot new server instances to minutes
- There are two key concepts to Launch instances in AWS:
 - Instance Type
 - AMI
- EC2 Facts:
 - Scale capacity as your computing requirements change
 - Pay only for capacity that you actually use
 - Choose Linux or Windows OS as per need. You have to Manage the OS and Security of same.
 - Deploy across AWS Regions and Availability Zones for reliability/HA

AWS EC2

General purpose



Compute optimized



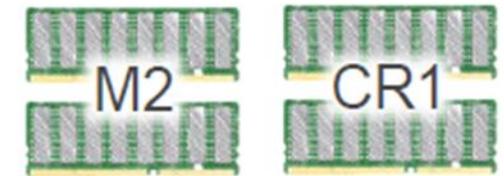
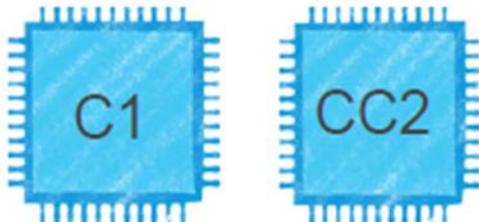
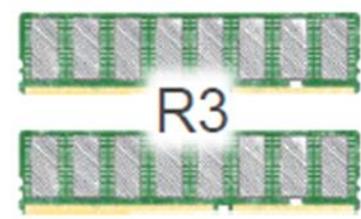
Storage and IO optimized



GPU enabled



Memory optimized



EC2 Security Group

Security Group is a Virtual Firewall Protection.

AWS allows you to control traffic in and out of your instances through virtual firewalls called security groups.

Security groups allow you to control traffic based on port, protocol, and source(inbound)/destination(outbound).

Security groups are associated with instances when they are launched. Every instance must have at least one security group. Though they can have more.

A security group is default deny.

AWS EC2 Pricing

On-Demand Instances

Pay by the hour.

Reserved Instances

Purchase at significant discount.
Instances are always available.

1-year to 3-year terms.

Scheduled Instances

Purchase a 1-year RI for a recurring period of time.

Spot Instances

Highest bidder uses instance at a significant discount.
Spot blocks supported.

Dedicated Hosts

Physical host is fully dedicated to run your instances. Bring your per-socket, per-core, or per-VM software licenses to reduce cost.

EC2 Security

- AMI - CIS Benchling, Quickstart/Marketplace images/Company Hardened Images
- Instance type – Dedicated/Shared?
- Public IP?
- Subnet - Public/Private?
- IAM Role?
- Enable termination protection
- Enable encryption at storage
- Security Group Rules

Lab: EC2 Security

- 1) Create an EC2 instance with CentOS 7 (must be an official image)
- 2) Select Instance type t2.micro at step2 and observe other models with diff resources
- 3) Observe Placement group and Dedicated hosts
- 4) Enable Termination protection
- 5) Enable Encryption at Data disks and don't encrypt the OS disks.
- 6) Create your own security group – {yourname-sg1}
- 7) Enable login via ssh
- 8) Login to server and then check outgoing connection to google.com (142.250.194.78)
- 9) Disable outgoing ping requests to 142.250.194.78 //** Check how same can be done
- 10)Create and Attach another firewall (yourname-2) to your server.
- 11)Enable ping for your machine via second firewall

AWS KMS (Key Management Services)

Encryption in flight

Data is encrypted before sending and decrypted after receiving

TLS certificates help with encryption (HTTPS)

Encryption in flight ensures no **man in the middle attack** can happen



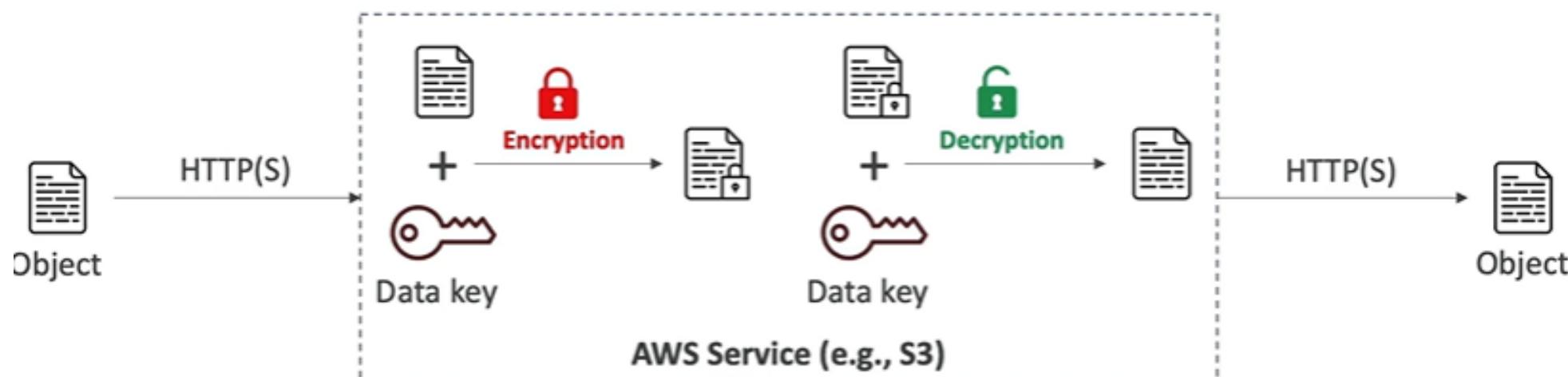
Server-side encryption at rest

Data is encrypted after being received by the server

Data is decrypted before being sent

It is stored in an encrypted form (using data keys)

The encryption/decryption keys must be managed somewhere and server must have access to it



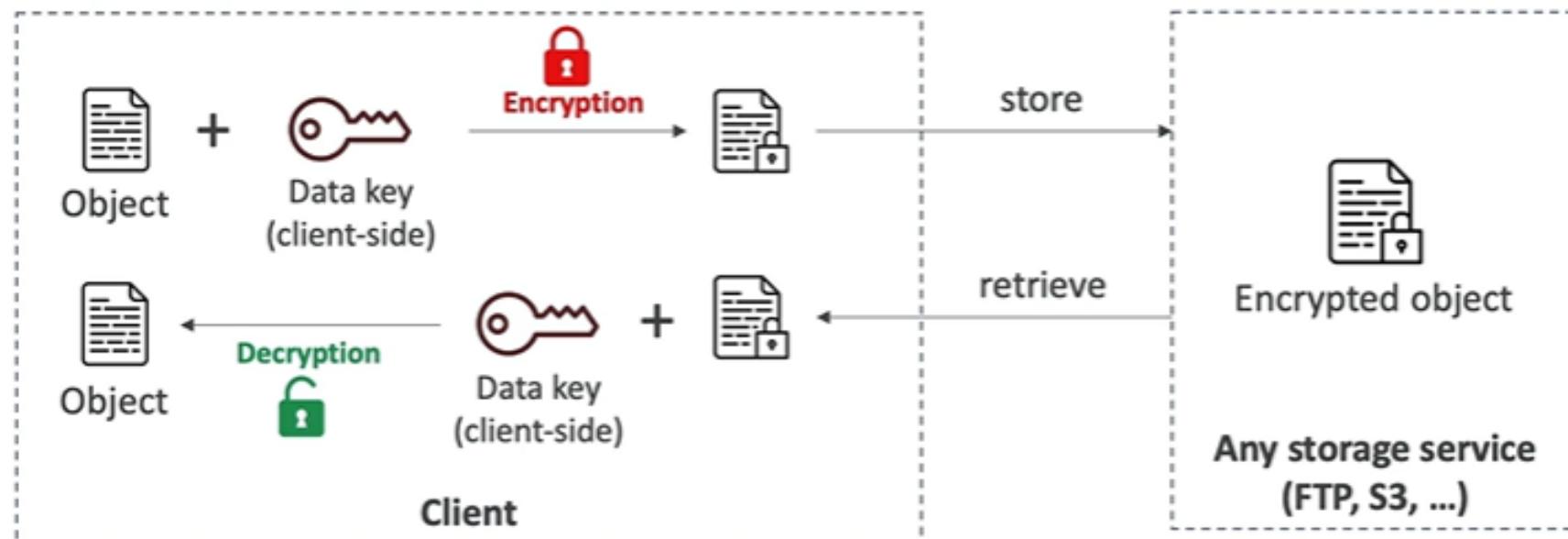
Client-side encryption

Data is encrypted by the client and never decrypted by the server

Data will be decrypted by a receiving client

The server should not be able to decrypt the data

Could leverage Envelope Encryption



KMS

AWS Key Management Service (KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data.

AWS Key Management Service is integrated with other AWS services including Amazon EBS, Amazon S3, Amazon Redshift, and others to make it simple to encrypt your data with encryption keys that you manage.

by default AWS KMS generates the key material for that CMK. But you can create a CMK without key material and then import your own key material into that CMK, a feature often known as "bring your own key" (BYOK).

You can Enable, Disable the key and even do schedule deletion (no Immediate deletion of keys permitted).

KMS Benefits

Fully managed

Centralized key management

Integrated with AWS services

Encryption for all your applications (using SDKs)

Built-in auditing

No commitment for usage

Secure with FIPS 140-2 validated hardware security modules

KMS uses highly durable storage and a resilient architecture to ensure that your keys are always available and are never lost.

You can set usage policies that determine which users can use your keys and what actions they can perform. All requests to use these keys are logged in AWS CloudTrail so that you can track who used which key, how and when.

KMS

Earlier AWS was supporting only Symmetric Keys But since 2019 Nov, its started providing support for Asymmetric keys too.

Symmetric (default): A 256-bit single encryption key which is used for both encryption and decryption.

Asymmetric: A public and private key pair that can be used for encrypt/decrypt or sign/verify operations

If your use case requires encryption outside of AWS by users who cannot call AWS KMS, asymmetric CMKs are a good choice. However, if you are creating a CMK to encrypt the data that you store or manage in an AWS service, use a symmetric CMK.

Compare the keys services and operations at:

<https://docs.aws.amazon.com/kms/latest/developerguide/symm-asymm-compare.html>

Symmetric Keys

A 256-bit encryption key. Symmetric keys are used in symmetric encryption, where the same key is used for encryption and decryption. AWS services that are integrated with AWS KMS use symmetric CMKs to encrypt your data.

You can use a **symmetric CMK** in AWS KMS to **encrypt**, **decrypt**, and **re-encrypt** data, generate data keys and data key pairs, and generate random byte strings.

You can import your own key material into a symmetric CMK and create symmetric CMKs in custom key stores.

The symmetric encryption algorithm that AWS KMS uses is fast, efficient, and assures the confidentiality and authenticity of data. It supports authenticated encryption with additional authenticated data (AAD), defined as an encryption context. This type of CMK requires both the sender and recipient of encrypted data to have valid AWS credentials to call AWS KMS.

Imported key material is supported only for symmetric CMKs in AWS KMS key stores.

Asymmetric Keys

A public and private key pair that can be used for **encrypt/decrypt** or **sign/verify** operations.

An asymmetric CMK represents a mathematically related public key and private key pair. You can give the public key to anyone, even if they're not trusted, but the private key must be kept secret.

In an asymmetric CMK, the private key is created in AWS KMS and never leaves AWS KMS unencrypted.

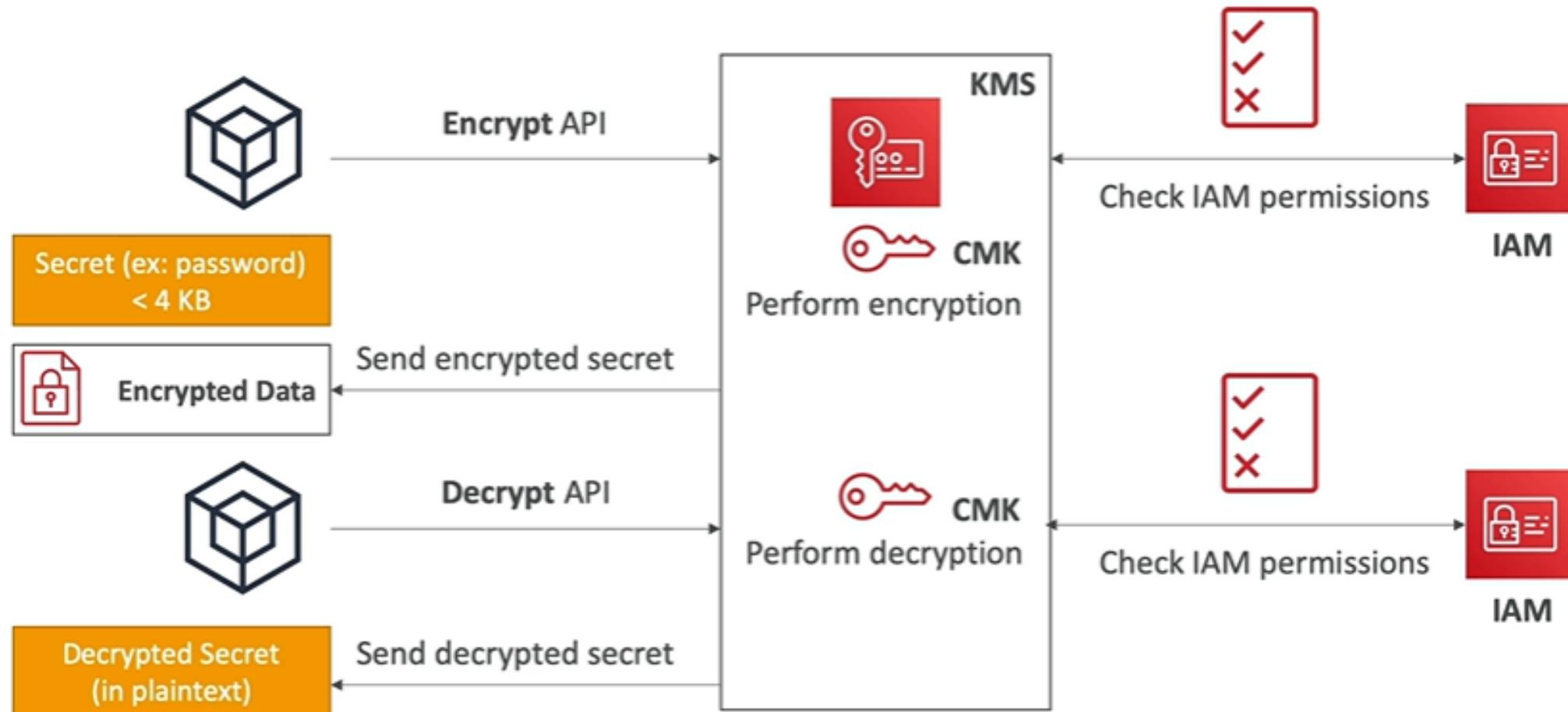
AWS KMS supports two types of asymmetric CMKs.

RSA CMKs: A CMK with an RSA key pair for encryption and decryption or signing and verification (but not both). KMS supports several key lengths for different security requirements.

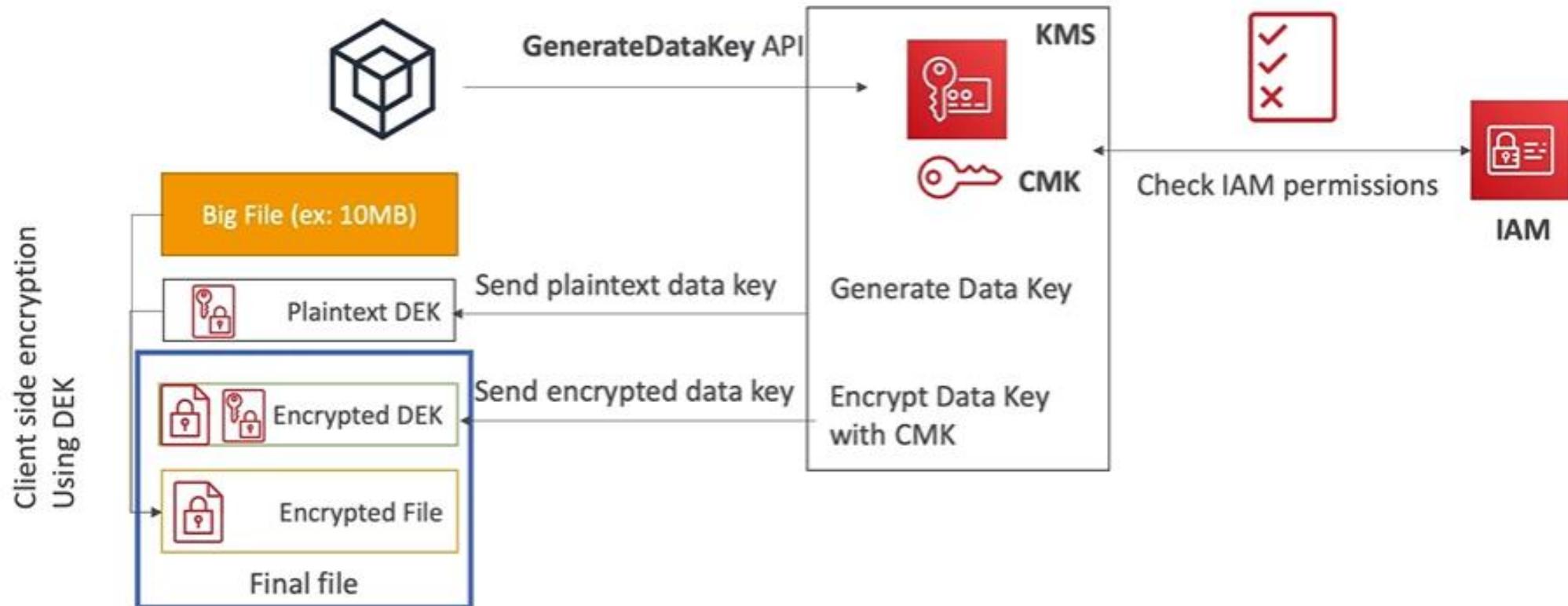
Elliptic Curve (ECC) CMKs: A CMK with an elliptic curve key pair for signing and verification. KMS supports several commonly-used curves.

To sign messages and verify signatures, you must use an asymmetric CMK.

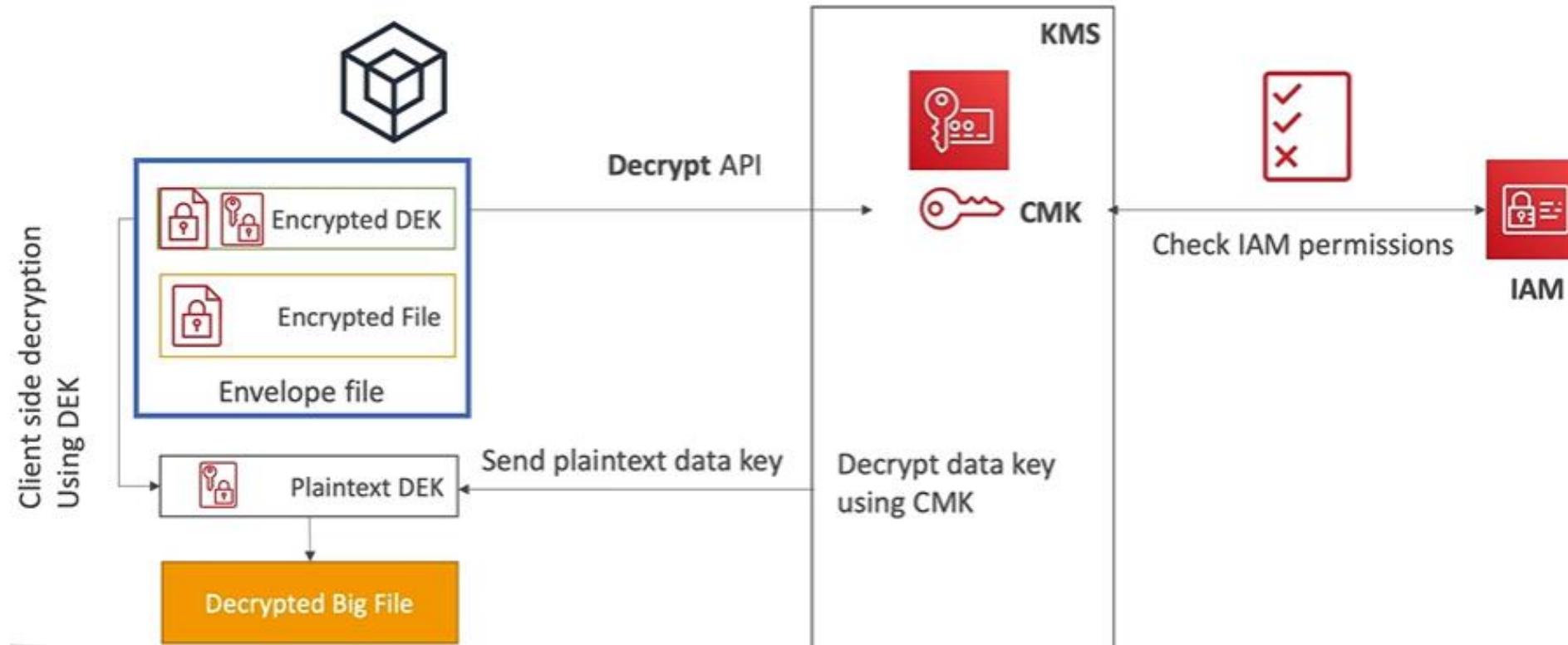
How KMS Works



Envelope Encryption



Envelope Decryption



Keys

Valid key usage for CMK types		
CMK type	Encrypt and decrypt	Sign and verify
Symmetric CMKs	✓	✗
Asymmetric CMKs with RSA key pairs	✓	✓
Asymmetric CMKs with ECC key pairs	✗	✓

KMS

Scheduling a deletion will disable the Key.

Disabled keys can't be used for any new usage. Existing encryptions service will go on temporarily. Any minor change(i.e. reboot or unmount)/over-the-period, encryption will be stopped and service using same will be hung.

You can cancel the deletion of key, anytime before the schedule deletion waiting period is over.

KMS is a regional Service. Key created in one Region, will not be available in another regions.

Note: -Deleting a key makes all data encrypted under that key unrecoverable.

LAB: KMS

Managing key under KMS

Go to **IAM → encryption keys**

1. Click on create key
2. Provide name and click on advance tab to select the key origin. For now, keep it KMS only.
3. Select Admins, who can manage the Key
4. Select users, who can use this key to encrypt & decrypt their data
5. Preview and launch the key creation
6. Try to create a new volume and check in Volume encryption, whether new key is popping up or not.
7. Try to enable & disable this key.
8. Do schedule key deletion and set a period of 7 days to get the key deleted.

S3 Bucket Key for SSE-KMS encryption

- Decrease the number of API calls made to KMS from S3 by 99 percent
- Decrease of cost of overall KMS encryption with Amazon S3 by 99 percent
- This leveraged data keys , a S3 bucket key is generated
- That key is used to encrypt KMS objects with new data keys
- You will see less KMS CloudTrail events in CloudTrail



AWS Cloud HSM

It's a Managed hardware security module (HSM) in the AWS Cloud.

AWS CloudHSM provides cloud-based hardware security modules (HSMs) for generating and using your own encryption keys in the AWS Cloud.

With CloudHSM, you can manage your own encryption keys using **FIPS 140-2 Level 3** validated HSMs and integrate with your applications using industry-standard APIs.

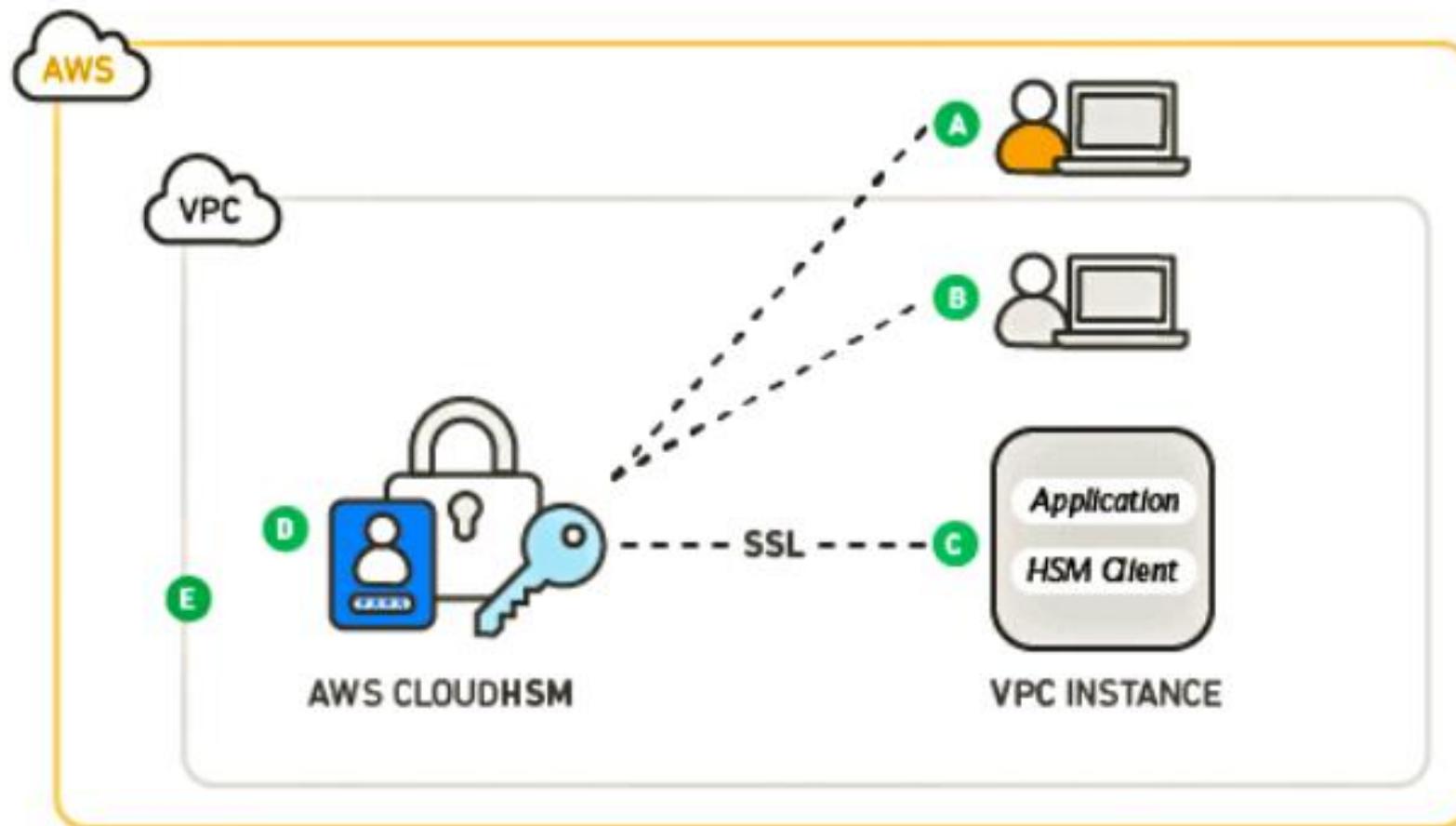
AWS CloudHSM runs in your own Amazon Virtual Private Cloud (VPC), so that you can easily use your HSMs with applications that run on your Amazon EC2 instances.

With CloudHSM, you can use standard VPC security controls to manage access to your HSMs. Your applications connect securely and with better performance.

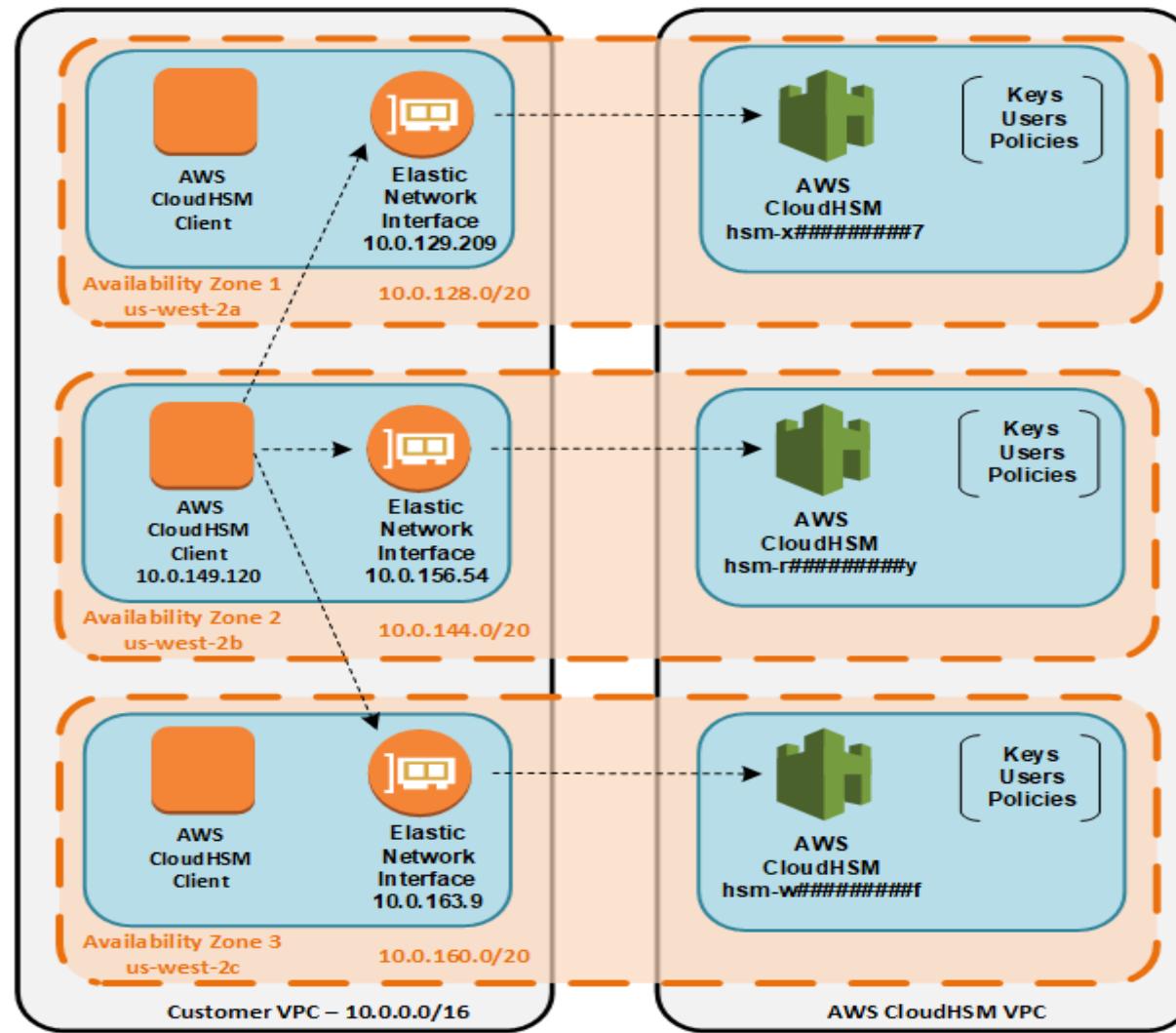
Tamper-resistant HSM instances in your own Amazon Virtual Private Cloud (VPC).

* **FIPS: The Federal Information Processing Standard Publication**

AWS Cloud HSM



AWS Cloud HSM



AWS Cloud HSM

Benefits:

- Generate and use encryption keys on FIPS 140-2 level 3 validated HSMs
- Integrate with custom applications using industry-standard APIs, such as **PKCS#11, Java Cryptography Extensions (JCE), and Microsoft CryptoNG (CNG) libraries**. You can also transfer your keys to other commercial HSM solutions.
- AWS CloudHSM automatically handles/provides HA, Patching, load balances requests and securely duplicates keys stored in any HSM to all HSMs in the cluster.
- Security, Compliance with high reliability and low latency
- AWS has no visibility or access to your encryption keys.

Price:

Pay by the hour with no long term commitments or upfront payments

AWS Cloud HSM

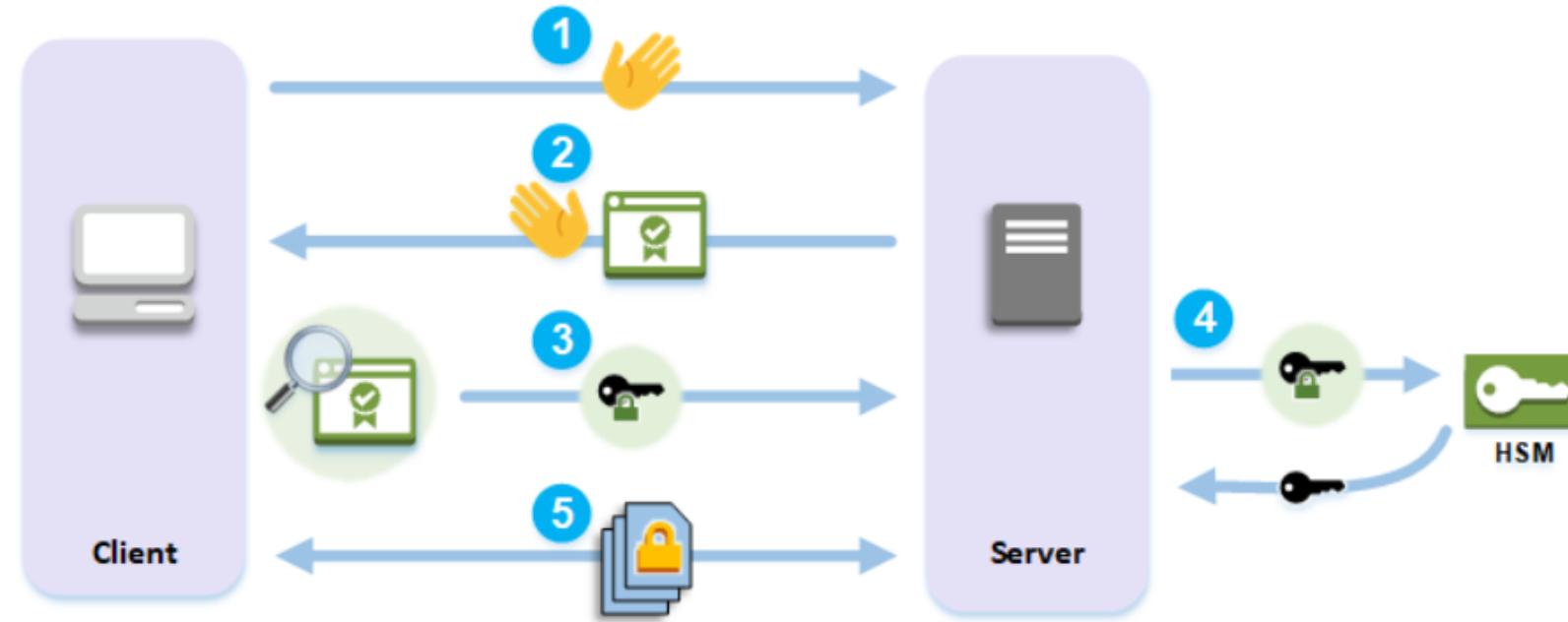
Use Cases:

- Achieve Regulatory Compliance
- Encrypt and Decrypt data
- Sign and verify document with private and public keys
- Authenticate messages using Cipher Message Authentication Codes (CMACs) and Hash-based Message Authentication Codes (HMACs)
- Leverage the benefits of AWS CloudHSM and AWS Key Management Service
- Offload SSL/TLS processing for web servers
- Enable Transparent Data Encryption (TDE)
- Manage the private keys of an issuing certificate authority (CA)
- Generate random numbers

AWS Cloud HSM

Use Cases:

- Offload SSL/TLS processing for web servers

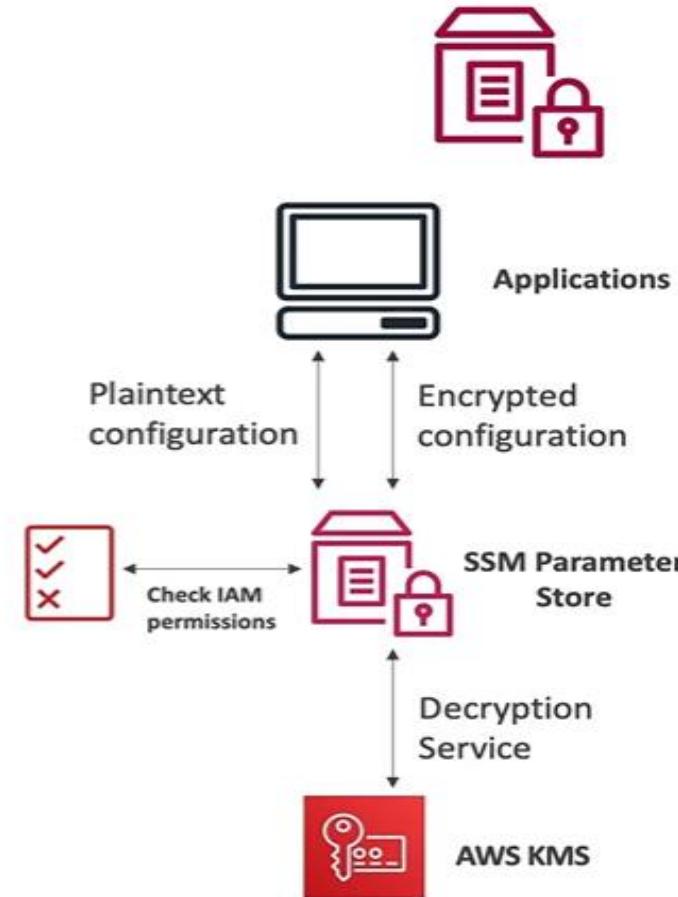


Cloud HSM vs KMS

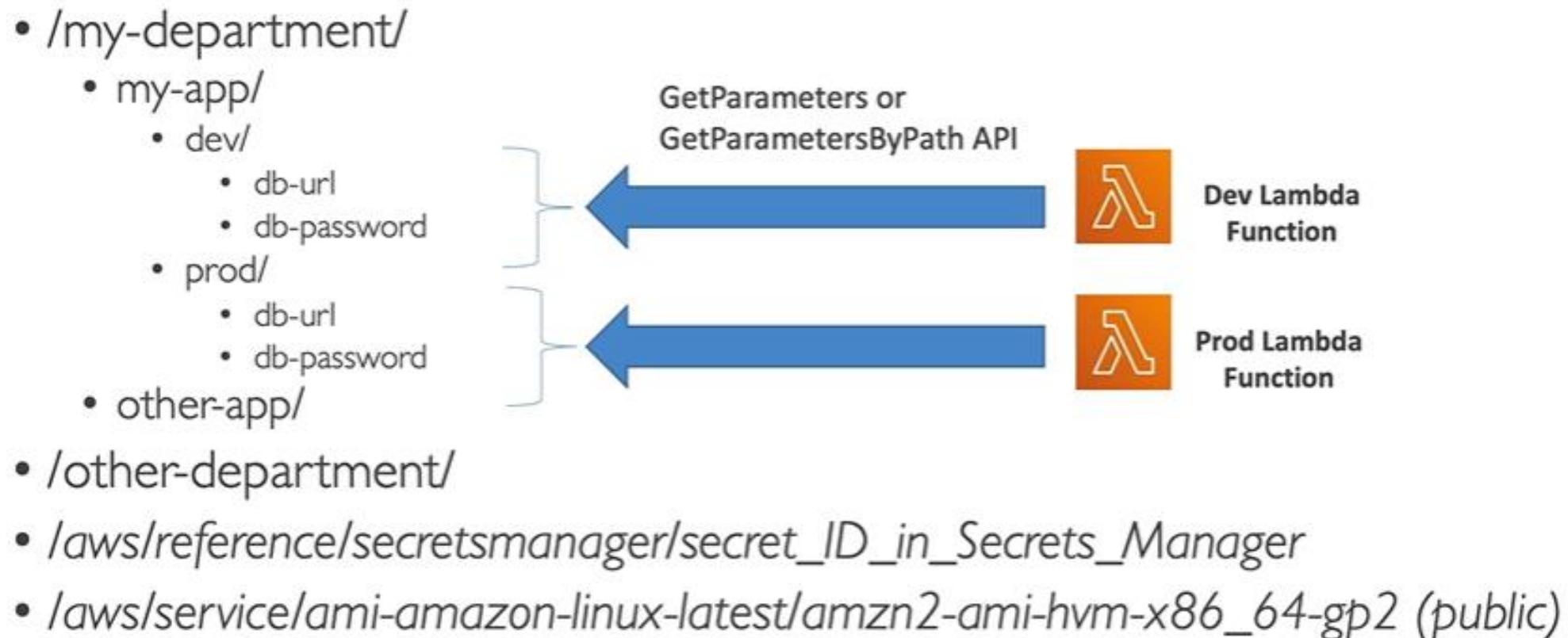
Feature	AWS KMS	AWS CloudHSM
Tenancy	Multi-Tenant	Single-Tenant
Standard	FIPS 140-2 Level 3	FIPS 140-2 Level 3
Master Keys	<ul style="list-style-type: none">AWS Owned CMKAWS Managed CMKCustomer Managed CMK	Customer Managed CMK
Key Types	<ul style="list-style-type: none">SymmetricAsymmetricDigital Signing	<ul style="list-style-type: none">SymmetricAsymmetricDigital Signing & Hashing
Key Accessibility	Accessible in multiple AWS regions (can't access keys outside the region it's created in)	<ul style="list-style-type: none">Deployed and managed in a VPCCan be shared across VPCs (VPC Peering)
Cryptographic Acceleration	None	<ul style="list-style-type: none">SSL/TLS AccelerationOracle TDE Acceleration
Access & Authentication	AWS IAM	You create users and manage their permissions

SSM Parameter Store

- Secure storage for configuration and secrets
- Optional Seamless Encryption using KMS
- Serverless, scalable, durable, easy SDK
- Version tracking of configuration/secrets
- Security through IAM
- Notifications with Amazon EventBridge
- Integration with CloudFormation



SSM Parameter Hierarchy



Standard and Advanced parameter

Types

	Standard	Advanced
Total number of parameters allowed (per AWS account and Region)	10,000	100,000
Maximum size of a parameter value	4 KB	8 KB
Parameter policies available	No	Yes
Cost	No additional charge	Charges apply
Storage Pricing	Free	\$0.05 per advanced parameter per month

Standard and Advanced parameter tiers

- Allow to assign a TTL to a parameter (expiration date) to force updating or deleting sensitive data such as passwords
- Can assign multiple policies at a time

Expiration (to delete a parameter)

```
{  
  "Type": "Expiration",  
  "Version": "1.0",  
  "Attributes": {  
    "Timestamp": "2020-12-02T21:34:33.000Z"  
  }  
}
```

ExpirationNotification (EventBridge)

```
{  
  "Type": "ExpirationNotification",  
  "Version": "1.0",  
  "Attributes": {  
    "Before": "15",  
    "Unit": "Days"  
  }  
}
```

NoChangeNotification (EventBridge)

```
{  
  "Type": "NoChangeNotification",  
  "Version": "1.0",  
  "Attributes": {  
    "After": "20",  
    "Unit": "Days"  
  }  
}
```

AWS Secret Manager

- It is used for storing secrets
- Capability to force rotation of secrets every X days
- Automate generation of secrets on rotation using Lambda
- Integration with Amazon RDS (MySQL, PostgreSQL, Aurora)
- Secrets are encrypted using KMS
- Mostly meant for RDS integration

AWS Secret Manager

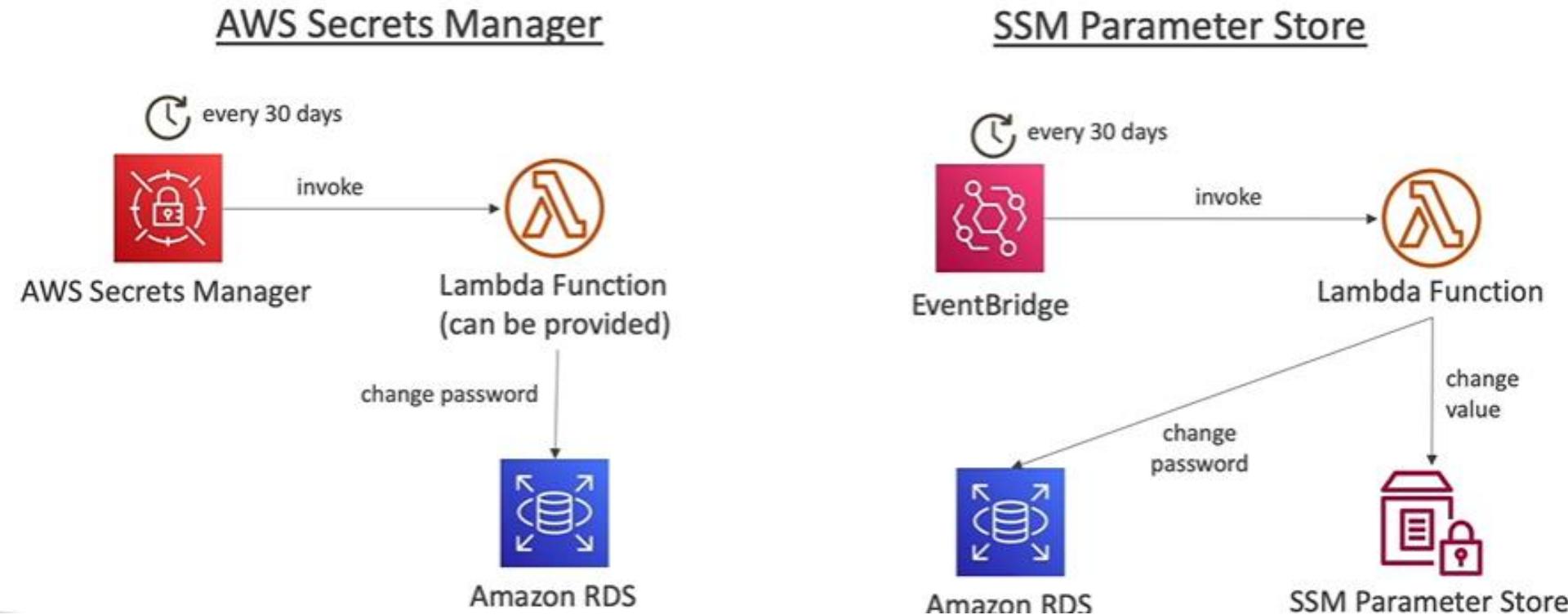
- Replicate secrets across multiple AWS regions
- Secret Manager keeps read replicas in sync with the primary secret
- Ability to promote a read replica Secret to a standalone Secret
- Use cases: multi-region apps, disaster recovery strategies, multi-region DB



Parameter Store vs Secret Manager

- Secrets Manager (\$\$\$):
 - Automatic rotation of secrets with AWS Lambda
 - Lambda function is provided for RDS, Redshift, DocumentDB
 - KMS encryption is mandatory
 - Can integration with CloudFormation
- SSM Parameter Store (\$):
 - Simple API
 - No secret rotation (can enable rotation using Lambda triggered by EventBridge)
 - KMS encryption is optional
 - Can integration with CloudFormation
 - Can pull a Secrets Manager secret using the SSM Parameter Store API

Parameter Store vs Secret Manager



Cloud Data Security

Data – The New Gold

Data - Probably the most sensitive and critical asset for any company. But first we need to know- how to get the sensitivity of same, before we move on to cloud.

Business Requirements Analysis:

- **An Inventory of Assets** (tangible & Intangible).
- **A Valuation of each Asset** - BIA- Business Impact Analysis, Data classification to be done
- **A Determination of critical paths, Processes and Assets.** Criticality of Tangible Assets (i.e. CMP for CSP), Intangible Asset (i.e. Copyright of music/software), process (i.e. SLA), Data Paths (i.e. vendor co-ordination), Personnel (i.e. Network-Admin/Doctor). *Identification of SPOF*
- **A clear understanding of Risk Appetite** – Acceptable Risk within Limit (i.e. stray car parking)

Data Privacy Terms

In cloud Infra, Data keeps on moving and processed /stored /owned by different personnel. Below are the roles assigned:

- **Data Owner/ Data controller** : Organization (usually Dept. Head) who collected or Created data. **Cloud Customer** is usually the Data Owner /controller.
- **Data Custodian/Data Processor**: Organization/person who manipulates, stores or moves data on behalf of Data Owner. A **cloud Service provider** is usually known as Data Custodian/Processor.
- **Data subject**: an identifiable subject who can be identified by reference to an id number, or one or more factors specific to the his physical, physiological, mental, economic, cultural, or social identity (Telephone number, SSN, IP address, etc.)
- **Personal data (PII)**: information relating to an identified or identifiable natural person—biometrics, health data, etc.
- **Processing**: Operations performed on personal data—collection, recording, organization, storage, etc.

Data Privacy Terms

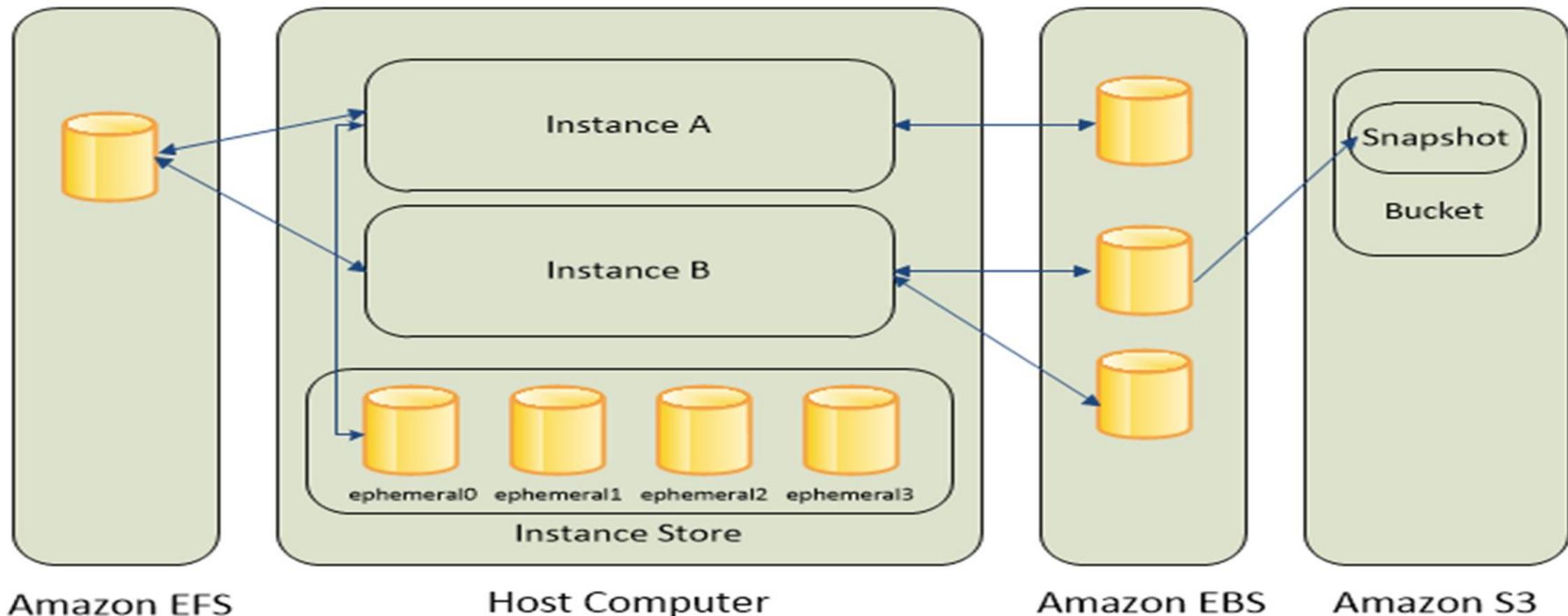
Note:

- Data Custodian can be third party in supply chain.
- **Data Owner is ultimately legally responsible for data and its breaches; even though the breach is done by Data Custodian/Cloud-Service-Provider.**
- Because of varying laws and regulations, customers should always know where their physical data is stored and is stored in compliance with their needs.

Cloud Storage Architecture

- **Volume Storage:**
 - File/Content based Storage (NFS Architecture- IaaS/SaaS)
 - Block Storage (Volume based Architecture – IaaS)
 - Ephemeral Disk (Volume based Architecture – IaaS)
- **Object-Based Storage** (Binary Objects – IaaS/PaaS)
- **Databases** (Relational & Non-Relational - PaaS)
- **Content Delivery Network** (Cache - PaaS)
- **Long Term Archival** (Backup Storage – IaaS/PaaS)

Storage Services



S3 Storage Services

Able to store an unlimited number of objects in a bucket.

Objects up to 5 TB; no bucket size limit.

Designed for 99.99999999% durability and 99.99% availability of objects over a given year.

HTTP/S endpoint to store and retrieve any amount of data, at any time, from anywhere on the web.

Highly scalable, reliable, fast, and inexpensive.

Optional server-side encryption using AWS or customer- managed provided client-side encryption.

Amazon S3 objects are automatically replicated on multiple devices in multiple facilities within a region.

S3 Use Cases

- Backup and archive for on-premises or cloud data
- Content, media, and software storage and distribution
- Big data analytics
- Static website hosting
- Software Delivery
- Store AMIs and Snapshots
- Cloud-native mobile and Internet application hosting
- Disaster Recovery

S3 Concepts

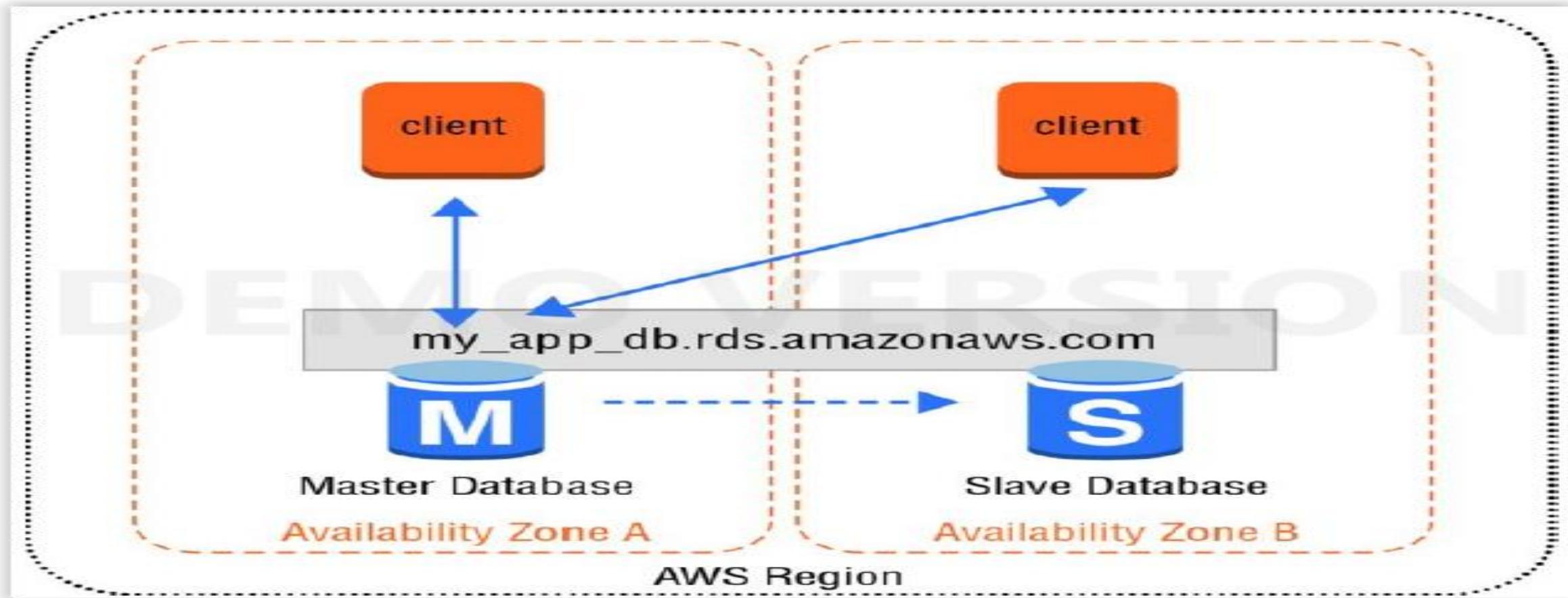
Buckets: A bucket is a container (web folder) for objects (files) stored in Amazon S3. Every Amazon S3 object is contained in a bucket. Buckets form the top-level namespace for Amazon S3 and bucket names are global.

AWS Regions: Amazon S3 bucket is created in a specific region that you choose. You control the location of your data; data in an Amazon S3 bucket is stored in that region unless you explicitly copy it to another bucket located in a different region.

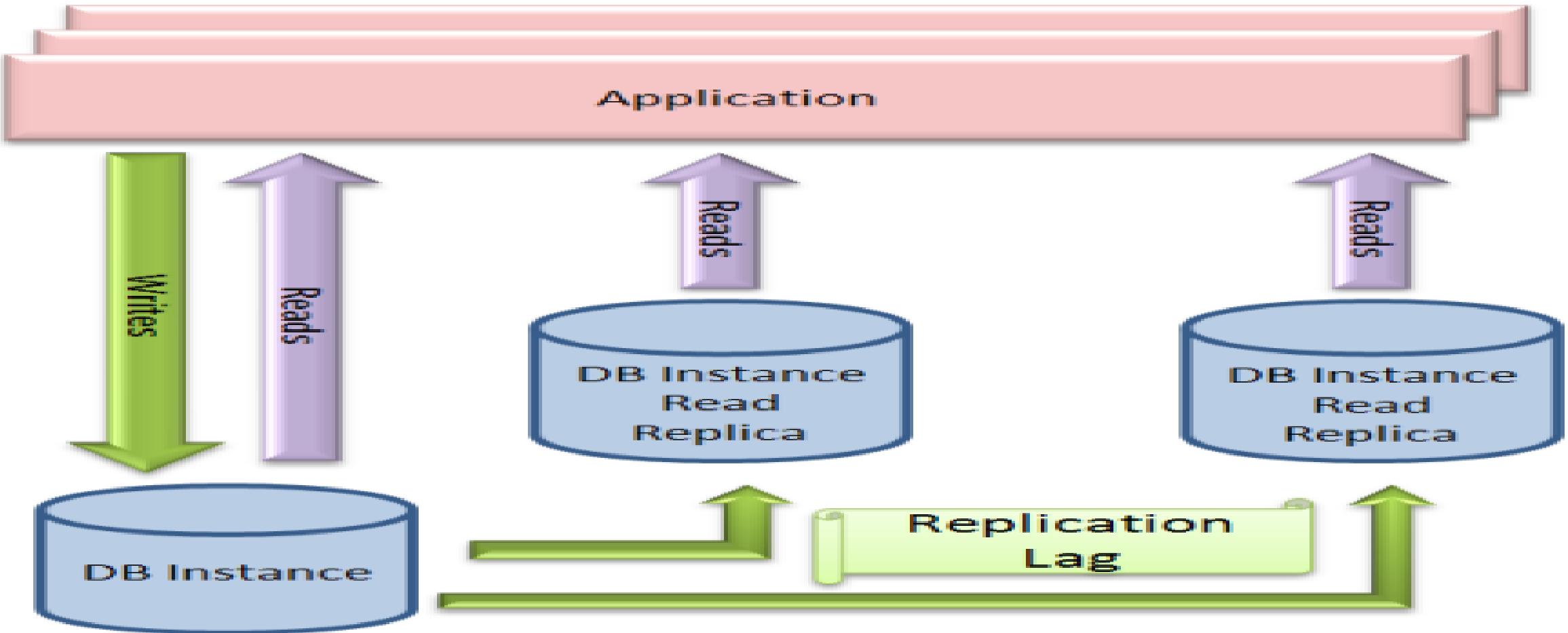
Objects: Objects are the entities or files stored in Amazon S3 buckets. Objects can range in size from 0 bytes up to 5TB, and a single bucket can store an unlimited number of objects.

Security in PaaS Services

PaaS Example



PaaS Example



AWS CLI

Resource Management

- **AWS Management Console**
- AWS Console Mobile App (View resources)
- **AWS Command line interface**
- AWS Toolkit for PowerShell
- **AWS SDK (Software Development Kit's)**
- REST API/ SOAP / Query
- AWS-Shell (In preview)

AWS CLI

AWS CLI is a command-based utility to manage AWS resources

The primary distribution method for the AWS CLI on Linux, Windows, and macOS is pip, a package manager for Python that provides an easy way to install, upgrade, and remove Python packages and their dependencies

- <http://docs.aws.amazon.com/cli/latest/userguide/installing.html>
- **Requirements**
 - Python 2 version 2.6.5+ or Python 3 version 3.3+
 - Windows, Linux, macOS, or Unix
 - Pip package should be present (else install python-pip)
- Install AWSCLI: **pip install awscli --upgrade --user**
- For Windows, directly download the Windows installer from CLI webpage

AWS CLI

- Lets install an AWSCLI
 - <https://aws.amazon.com/cli>
- **aws --version**
- aws help
- aws ec2 help / aws s3 help / aws <any-subcommand> help
- Configure your default keys and region:

```
root@ip-172-31-28-145:~# aws configure
AWS Access Key ID [None]: #####
AWS Secret Access Key [None]: #####
Default region name [None]: us-west-2
Default output format [None]:
root@ip-172-31-28-145:~#
```

LAB: AWS CLI

Check the details for all running instances using CLI

- aws ec2 describe-instances

Creation of an AWS Instance using CLI:

- aws ec2 run-instance help
- aws ec2 run-instances --image-id ami-76d6f519 --instance-type t2.micro --key-name test
- aws ec2 describe-instances
- aws ec2 stop-instances --instance-ids i-02e6b6c6c4dd3bbe0
- aws ec2 terminate-instances --instance-ids i-0297acea9e1b39a56

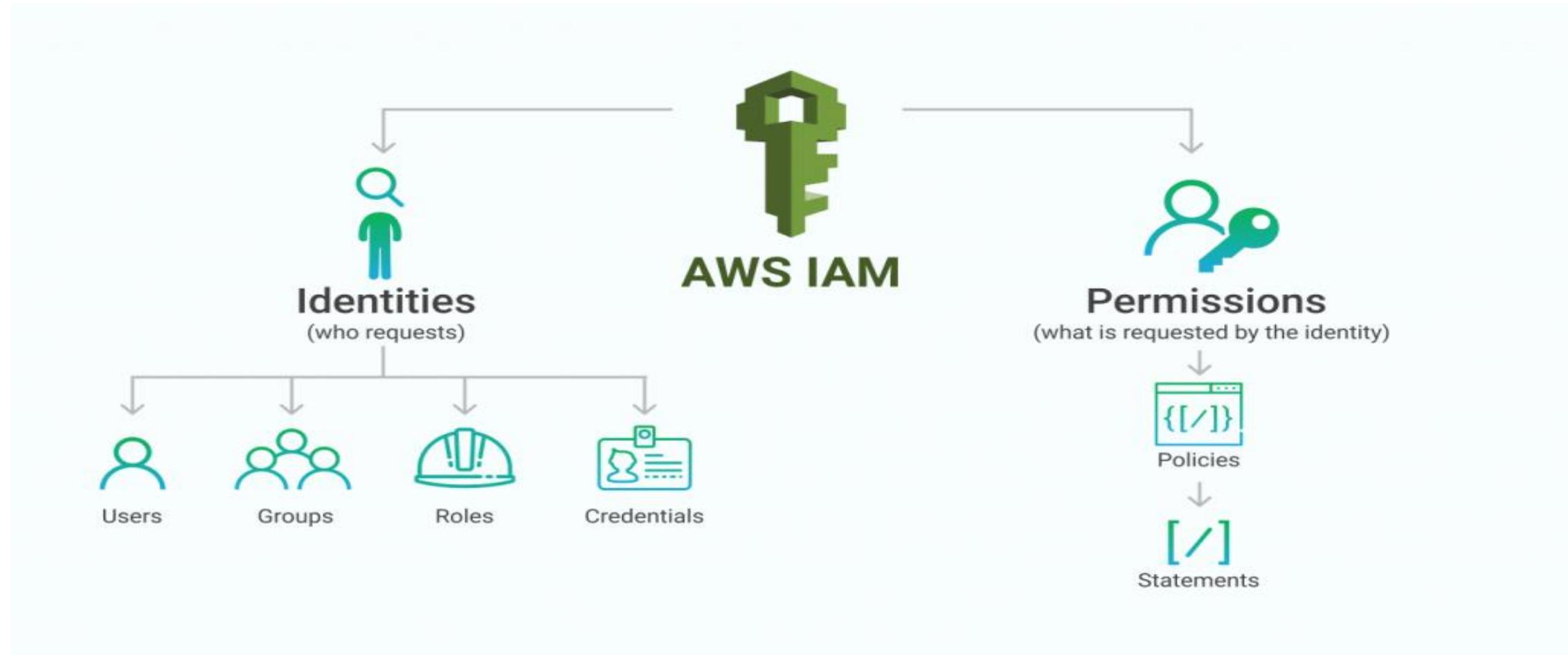
AWS IAM

IAM

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users. You use IAM to control who can use your AWS resources (authentication) and what resources they can use and in what ways (authorization).

- ✓ Shared access to your AWS account
- ✓ Granular permissions
- ✓ Secure access to AWS resources for applications that run on Amazon EC2
- ✓ Integrated with many AWS services
- ✓ Free to use

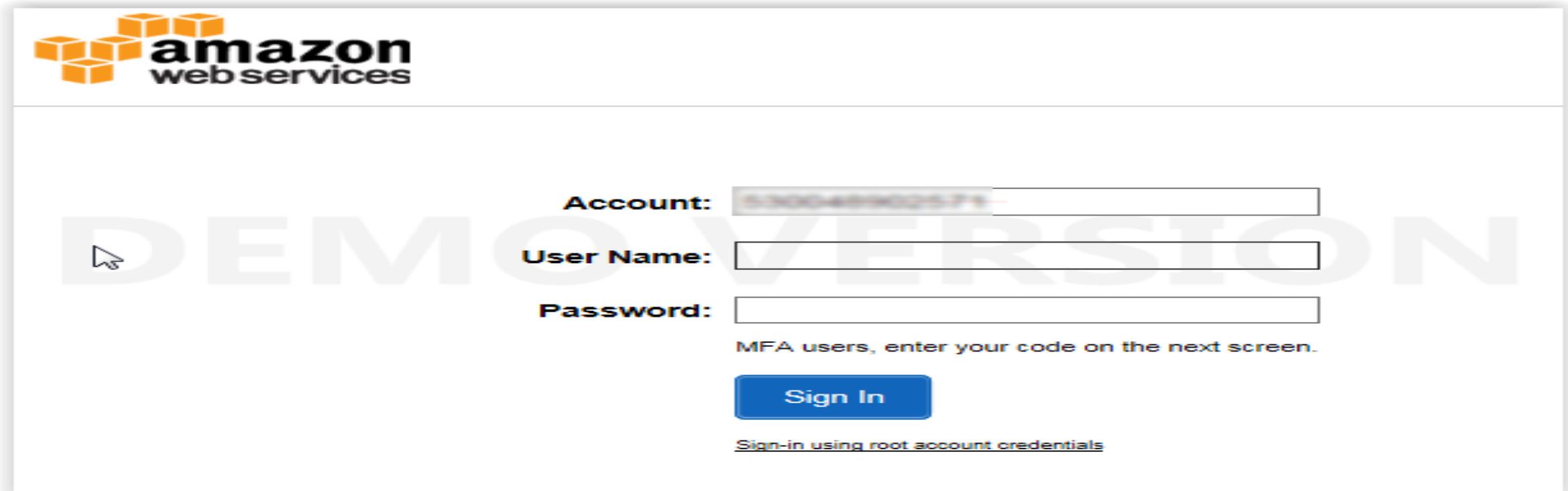
IAM



IAM Users

You can create IAM users to distribute your work among team/users.

Different permissions can be given to IAM users



IAM Groups

Users are part of IAM groups

Common policies can be applied on groups

Summary

Group ARN:	arn:aws:iam::210050880000:group/read_group	
Users (in this group):	2	
Path:	/	
Creation Time:	2018-05-02 22:21 UTC+0530	

[Users](#)[Permissions](#)[Access Advisor](#)

This view shows all users in this group: **2 Users**

User	Actions
gagandeep	Remove User from Group
Tejas	Remove User from Group

IAM Roles

IAM roles are a secure way to grant permissions to entities that you trust.

- ✓ IAM user in another account
- ✓ Application code running on an EC2 instance that needs to perform actions on AWS resources
- ✓ An AWS service that needs to act on resources in your account to provide its features
- ✓ Users from a corporate directory who use identity federation with SAML

Identity Policies

Policy	
AWS Managed Policy	<ul style="list-style-type: none">An <i>AWS managed policy</i> is a standalone policy that is created and administered by AWS. <i>Standalone policy</i> means that the policy has its own Amazon Resource Name (ARN) that includes the policy name.You cannot change the permissions defined in AWS managed policies. AWS occasionally updates the permissions defined in an AWS managed policy.
Customer Managed Policy	<ul style="list-style-type: none">You can create standalone policies in your own AWS account that you can attach to principal entities (users, groups, and roles).You create these <i>customer managed policies</i> for your specific use cases, and you can change and update them as often as you like.
Inline Policy	<ul style="list-style-type: none">An inline policy is a policy created for a single IAM identity (a user, group, or role).Inline policies maintain a strict one-to-one relationship between a policy and an identity. They are deleted when you delete the identity.

Identity Policies

Policies are the way to define permissions/authorization. Policies can be assigned to Users, Groups & Roles.

Contains 4 parts:

- ✓ Service
- ✓ Action
- ✓ Resources
- ✓ Request Conditions

A policy is a document (written in the Access Policy Language) that acts as a container for one or more statements.

Example Policies

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "service-prefix:action-name",  
      "Resource": "*",  
      "Condition": {  
        "DateGreaterThanOrEqual": {"aws:CurrentTime": "2020-04-01T00:00:00Z"},  
        "DateLessThan": {"aws:CurrentTime": "2020-06-30T23:59:59Z"}  
      } }]
```

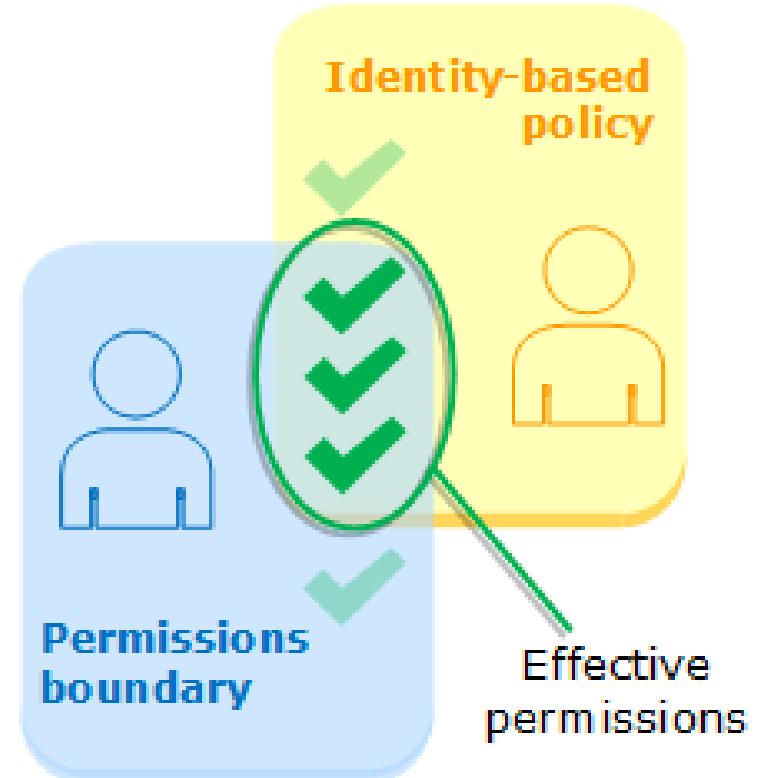
IAM Permission Boundaries

AWS supports permissions boundaries for IAM entities (users or roles). A permissions boundary is an advanced feature for using a managed policy to set the maximum permissions that an identity-based policy can grant to an IAM entity.

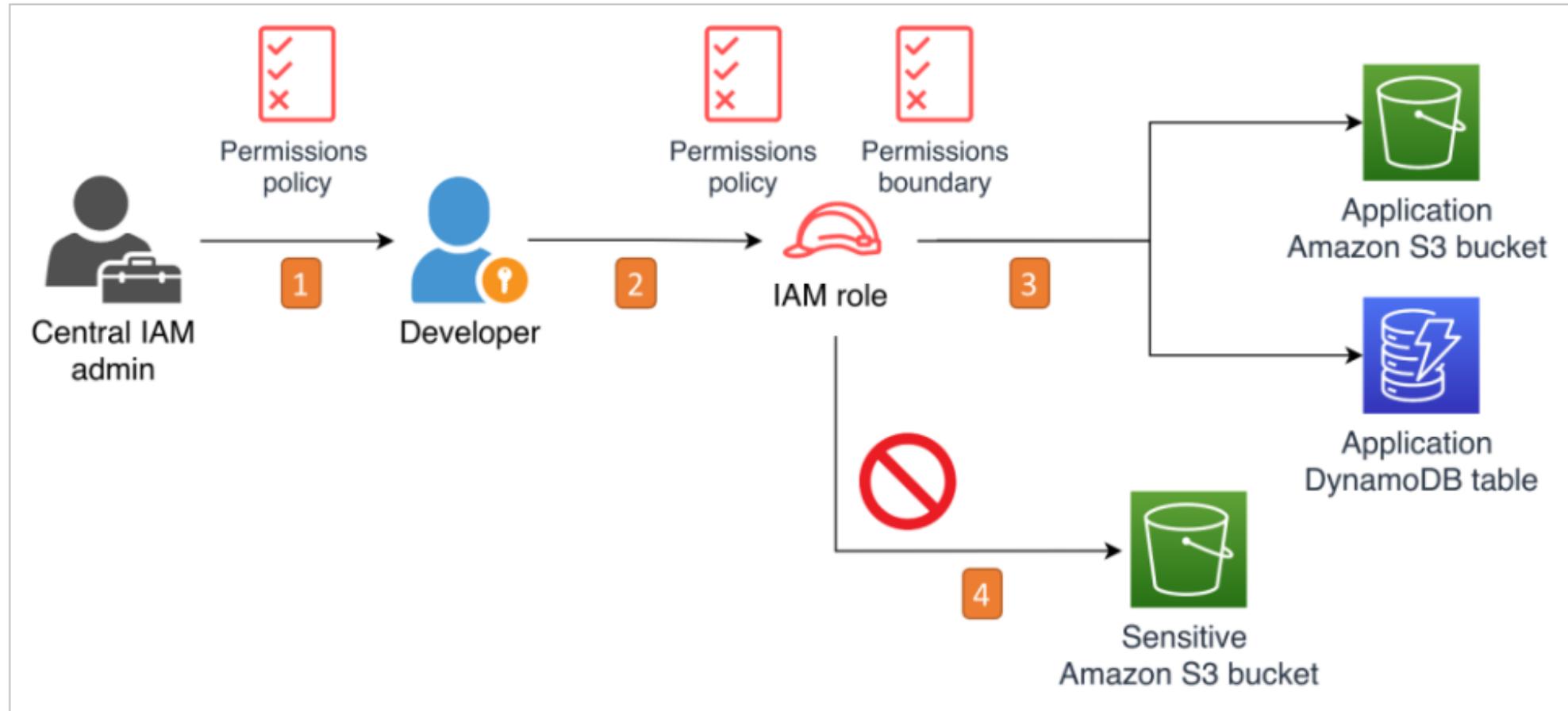
An entity's permissions boundary allows it to perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.

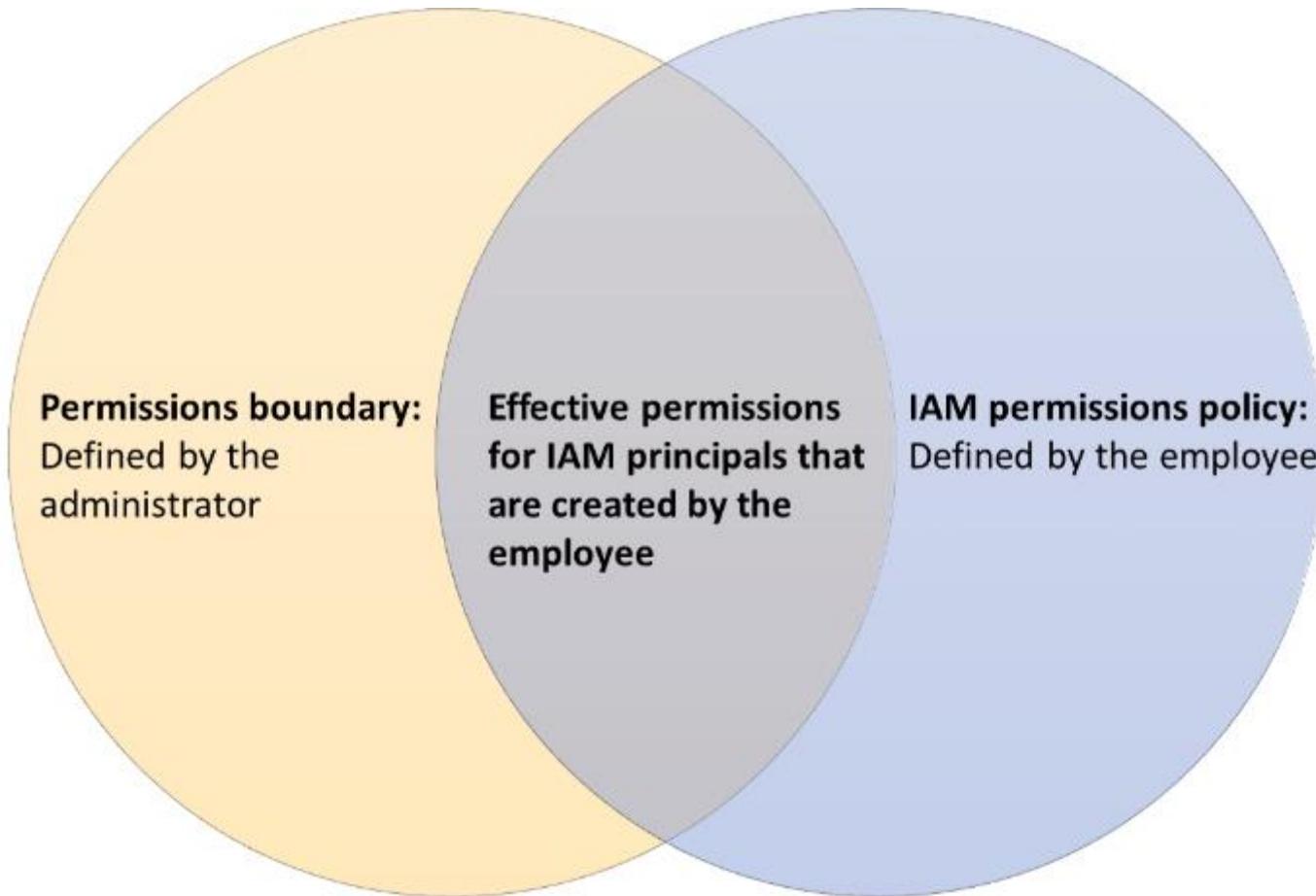
You can set allow or deny rule in Permission boundaries.

Permission boundaries are to set the limit, not to give the permission. Any permission added in permission boundaries, which is not there in IAM grant policies, then it'll not work.



IAM Permission Boundaries





IAM Permission Boundaries

- Developers often need to create new IAM roles and policies for their applications because these applications need permissions to interact with AWS resources.
- Organization grows and your centralized team receives an increasing number of requests to create and manage new downstream roles and policies.
- By setting up permissions boundaries, you allow your developers to focus on tasks that add value to your business, while simultaneously freeing your centralized security and IAM teams to work on other critical tasks, such as governance and support.
- A permissions boundary alone doesn't grant access to anything. Rather, it enforces a boundary that can't be exceeded.
- Permissions boundaries can only be up to 6,144 characters long. You can have up to 10 managed policies and 1 permissions boundary attached to an IAM role

IAM Permission Boundaries: Demo

- Create a group and add permission (EC2FullAccess)
- Create a user and add permission (EC2ReadOnlyAccess)
- Access the console using the new user and try to create/stop/restart an instance
- Add the user to the above group and try to create/stop/restart an instance
- Now add IAM Permission boundary (EC2ReadOnlyAccess)
- Now try to create/stop/restart an instance

IAM Duties

IAM resource	Purpose	Owner/maintainer	Applies to
Federated roles and policies	Grant permissions to federated users for experimentation in lower environments	Central team	People represented by users in the enterprise identity provider
IAM workload roles and policies	Grant permissions to resources used by applications, services	Developer	IAM roles representing specific tasks performed by applications
Permissions boundaries	Limit permissions available to workload roles and policies	Central team	Workload roles and policies created by developers
IAM users and policies	Allowed only by exception when there is no alternative that satisfies the use case	Central team plus senior leadership approval	Break-glass access; legacy workloads unable to use IAM roles

Resource based Policies

Resource-based policies are attached to a resource. For example, you can attach resource-based policies to Amazon S3 buckets, Amazon SQS queues, and AWS Key Management Service encryption keys.

With resource-based policies, you can specify who has access to the resource and what actions they can perform on it.

Resource-based policies differ from resource-level permissions. You can attach resource-based policies directly to a resource. Resource-level permissions refer to the ability to use ARNs to specify individual resources in a policy.

Resource-based policies are supported only by some AWS services.

Identity-based policies and **resource-based policies** are both permissions policies and are evaluated together. For a request to which only permissions policies apply, AWS first checks all policies for a Deny. If one exists, then the request is denied. Then AWS checks for each Allow. If at least one policy statement allows the action in the request, the request is allowed. It doesn't matter whether the Allow is in the identity-based policy or the resource-based policy.

Account ID: 123456789012

Identity-based policies

John Smith

Can List, Read
On Resource X

Carlos Salazar

Can List, Read
On Resource Y,Z

MaryMajor

Can List, Read, Write
On Resource X,Y,Z

ZhangWei

No policy

Resource-based policies

Resource X

JohnSmith: Can List, Read
MaryMajor: Can List, Read

Resource Y

CarlosSalazar: Can List, Write
ZhangWei: Can List, Read

Resource Z

CarlosSalazar: Denied access
ZhangWei: Allowed full access

Policy Generator

You can use AWS policy Generator to generate policies for your accounts:

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

<https://awspolicygen.s3.amazonaws.com/policygen.html>

Password Policy

Set password policy

A password policy is a set of rules that define complexity requirements and mandatory rotation periods for your IAM users' passwords. [Learn more](#)

Select your account password policy requirements:

Enforce minimum password length

6 characters

- Require at least one uppercase letter from Latin alphabet (A-Z)
- Require at least one lowercase letter from Latin alphabet (a-z)
- Require at least one number
- Require at least one non-alphanumeric character (!@#\$%^&*()_+-=[{}])
- Enable password expiration
- Password expiration requires administrator reset
- Allow users to change their own password
- Prevent password reuse

AWS Organizations

AWS Organizations is an account management service that lets you consolidate multiple AWS accounts (like test/dev/prod or Projects or compliance-based diff accounts for PCI/Hippa) into an organization or in further organizational Units, that you create and centrally manage.

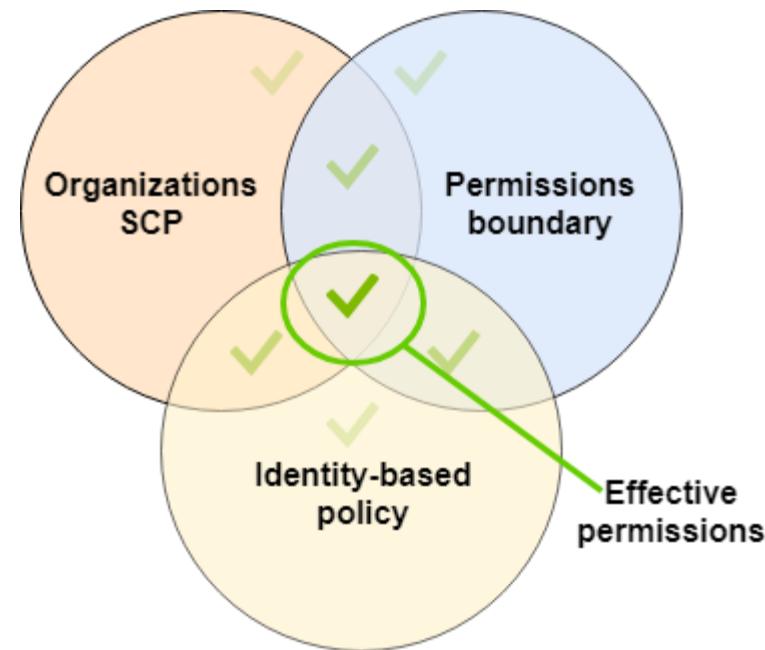
AWS Organizations helps you centrally govern your environment as you grow and scale your workloads on AWS.

Organizations helps you to centrally manage billing; control access, compliance, and security; and share resources across your AWS accounts.

- Centrally manage policies across multiple AWS accounts
- Automate AWS account creation and management through Organization API
- Consolidate billing across multiple AWS accounts
- Govern access to AWS services, resources, and regions i.e. Service Control Policies
- Configure AWS services across multiple accounts i.e. SSO.

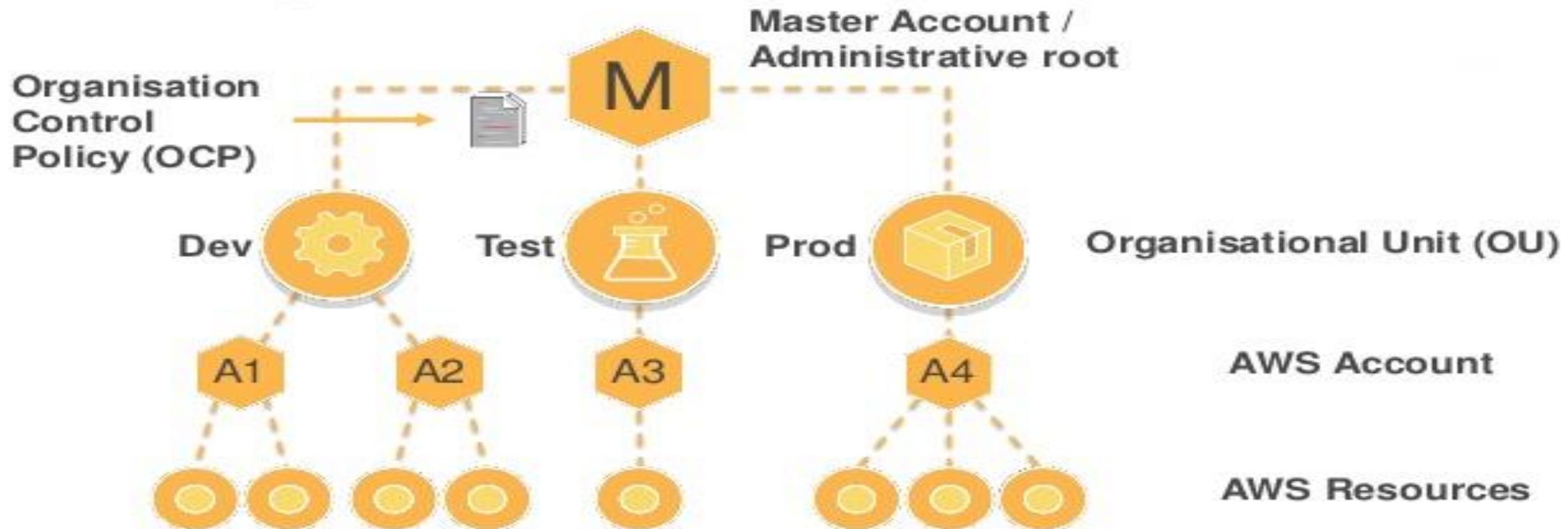
Organization Policies

SCPs are applied to an entire AWS account. They limit permissions for every request made by a principal within the account.



AWS Organizations

AWS Organisations



Organizations Level Policies

Service control policies (SCPs) offer central control over the maximum available permissions for all accounts in your organization, allowing you to ensure your accounts stay within your organization's access control guidelines.

Tag policies help you standardize tags on all tagged resources across your organization. You can use tag policies to define tag keys and their values.

Artificial Intelligence (AI) services opt-out policies enable you to control whether AWS AI services can store and use your content.

Backup policies enable you to deploy organization-wide backup plans to help ensure compliance across your organization's accounts.

EC2 Metadata

AWS Instance Metadata –

Instance metadata is data/information about your instance.

An HTTP call to **<http://169.254.169.254/latest/meta-data/>** will return the top node of the instance metadata tree and will return the information about your instance e.g. Public IP, DNS name, OS type etc..

For Linux systems:

`curl http://169.254.169.254/latest/meta-data/`

Dynamic Instance Identity metadata:

<http://169.254.169.254/latest/dynamic/instance-identity/>

IAM Best Practices

Delete AWS account (root) access keys.

Create individual IAM users.

Use groups to assign permissions to IAM users.

Grant least privilege.

Configure a strong password policy.

Enable MFA for privileged users.

Keep your passwords and AK/SK safe. Make sure to not to write AK/SK in code and take care while uploading such codes to some public repositories.

Use Roles for applications and Services, wherever possible.

Your AWS credentials are the actual threat. So keep your credentials as safe as you want to safeguard your MC production servers.

AWS IAM Advanced

CLI Credentials Priority

- The CLI will look for credentials in this order
 1. Command line options – --region, --output, and --profile
 2. Environment variables – AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, and AWS_SESSION_TOKEN
 3. CLI credentials file –aws configure
~/.aws/credentials on Linux / Mac & C:\Users\user\.aws\credentials on Windows
 4. CLI configuration file – aws configure
~/.aws/config on Linux / macOS & C:\Users\USERNAME\.aws\config on Windows
 5. Container credentials – for ECS tasks
 6. Instance profile credentials – for EC2 Instance Profiles
- Note: CLI actually uses Python BOTO3 SDK in backend.

SDK Credentials Priority

1. Environment variables –

AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY

2. Java system properties – aws.accessKeyId and aws.secretKey

3. The default credential profiles file – ex at: `~/.aws/credentials`, shared by many SDK

4. Amazon ECS container credentials – for ECS containers

5. Instance profile credentials – used on EC2 instances

Credentials Best practice

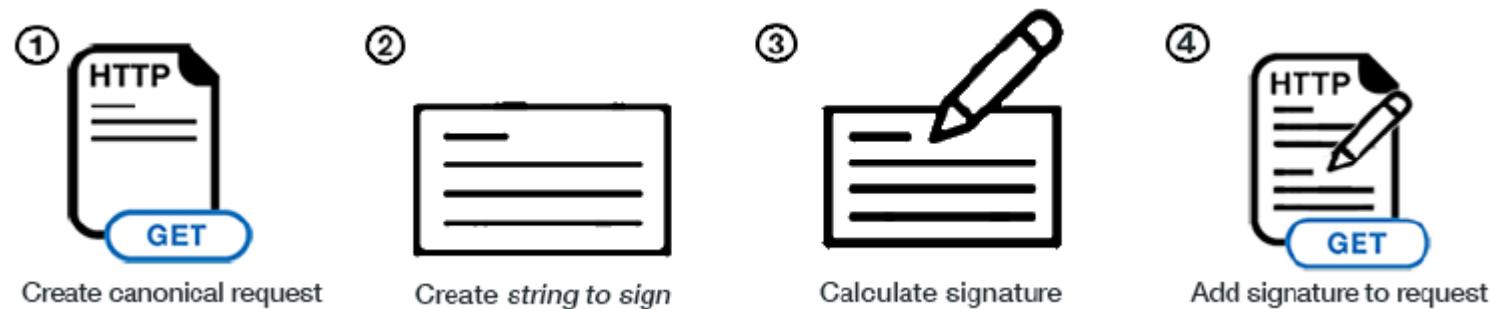
Overall, NEVER EVER STORE AWS CREDENTIALS IN YOUR CODE

- Best practice is for credentials to be inherited from the credentials chain
- If working within AWS, use IAM Roles
 - => EC2 Instances Roles for EC2 Instances
 - => ECS Roles for ECS tasks
 - => Lambda Roles for Lambda functions
- If working outside of AWS, use environment variables

Signing AWS Request

When you call the AWS HTTP API, you sign the request so that AWS can identify you, using your AWS credentials (access key & secret key)

- If you use the SDK or CLI, the HTTP requests are signed for you
- You should sign an AWS HTTP request using Signature v4 (SigV4)



Sig-v4 requests

Signature Version 4 is the process to add authentication information to AWS requests sent by HTTP.

When you send HTTP requests to AWS, you sign the requests so that AWS can identify who sent them. You sign requests with your AWS access key, which consists of an access key ID and secret access key. Some requests do not need to be signed, such as anonymous requests to Amazon Simple Storage Service (Amazon S3) and some API operations in **AWS Security Token Service (AWS STS)** such as **AssumeRoleWithWebIdentity**.

When you write custom code to send HTTP requests to AWS, you need to include code to sign the requests. You might do this for the following reasons:

- You are working with a programming language for which there is no AWS SDK.
- You want complete control over how a request is sent to AWS.

You don't need to sign a request when you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs. These tools manage the connection details, such as calculating signatures, handling request retries, and error handling. In most cases, they also contain sample code, tutorials, and other resources to help you get started writing applications that interact with AWS.

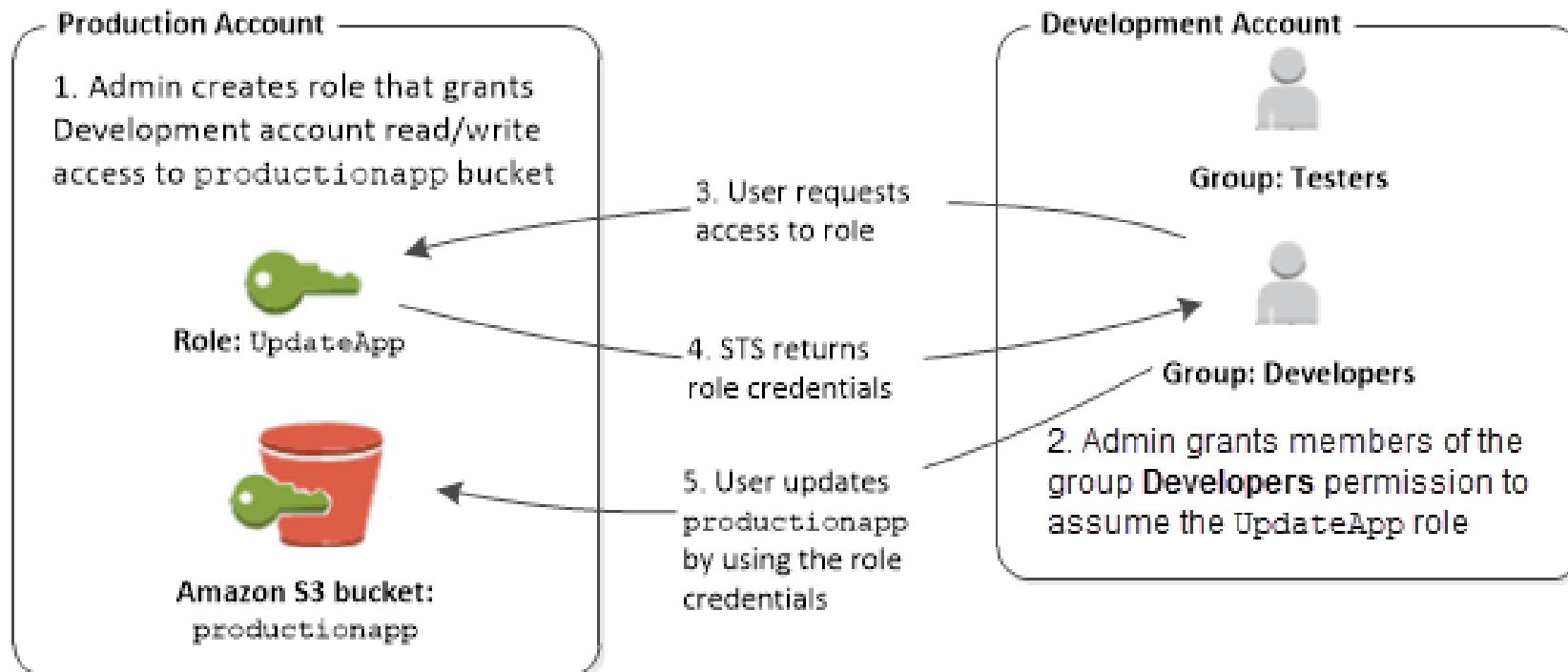
STS

AWS Security Token Service (AWS STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users).

Allows to grant limited and temporary access to AWS resources (up to 1 hour).

- **AssumeRole**: Assume roles within your account or cross account
- **AssumeRoleWithSAML**: return credentials for users logged with SAML
- **AssumeRoleWithWebIdentity**
- **GetSessionToken**: for MFA, from a user or AWS account root user
- **GetFederationToken**: obtain temporary creds for a federated user
- **GetCallerIdentity**: return details about the IAM user or role used in the API call
- **DecodeAuthorizationMessage**: decode error message when an AWS API is denied

Cross-Account Permission



IAM Pass role

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iam:PassRole",  
            "Resource": "arn:aws:iam::123456789012:role/S3Access"  
        }  
    ]  
}
```

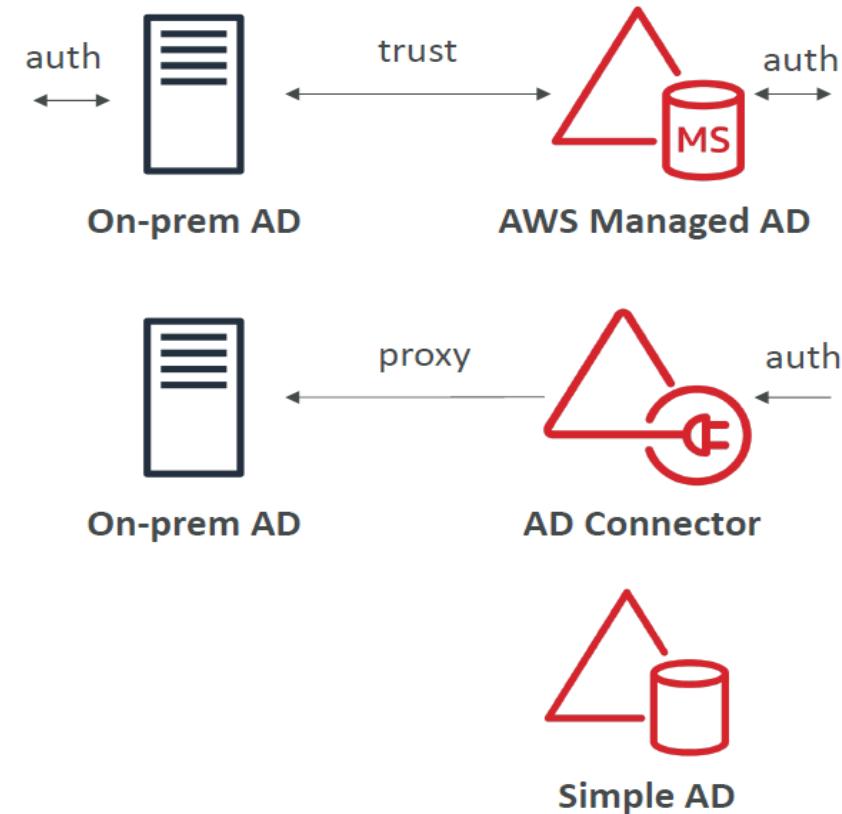
IAM Pass role : Demo

- Create a **role** to access S3 bucket (**S3accessForEC2**)
- Create a **user demo** who has access full access to EC2 only and also attach a policy of **iam:PassRole** to it.
- Create an instance using the demo user and try to attach any random role.
- Try to attach the **S3accessForEC2** role

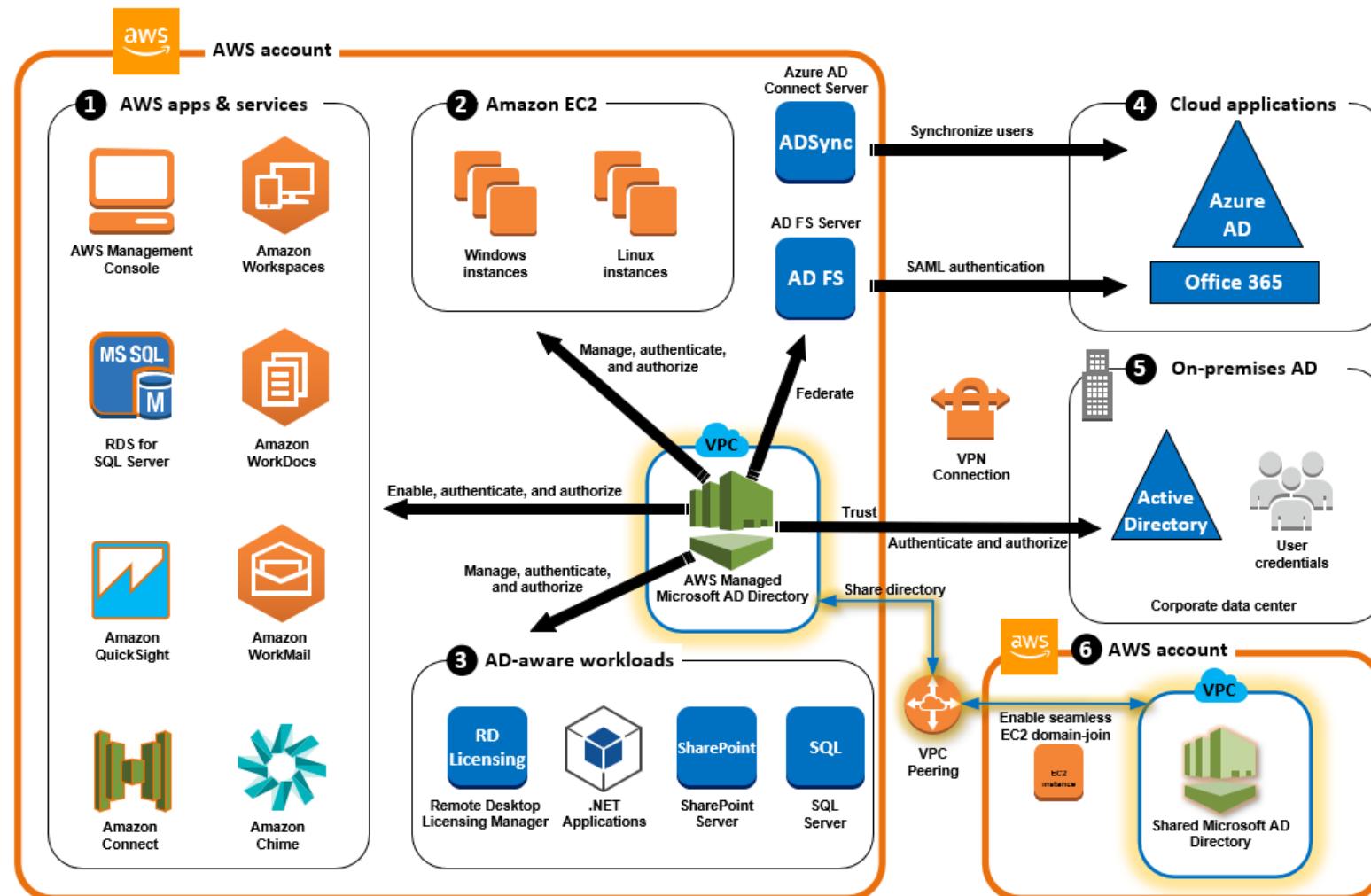
```
{  
  "Effect": "Allow",  
  "Action": "iam:PassRole",  
  "Resource":  
    "arn:aws:iam::685421549691:role/S3accessForEC2"  
,  
  {  
    "Effect": "Allow",  
    "Action": "iam>ListInstanceProfiles",  
    "Resource": "*"  
}
```

Active Directory on AWS

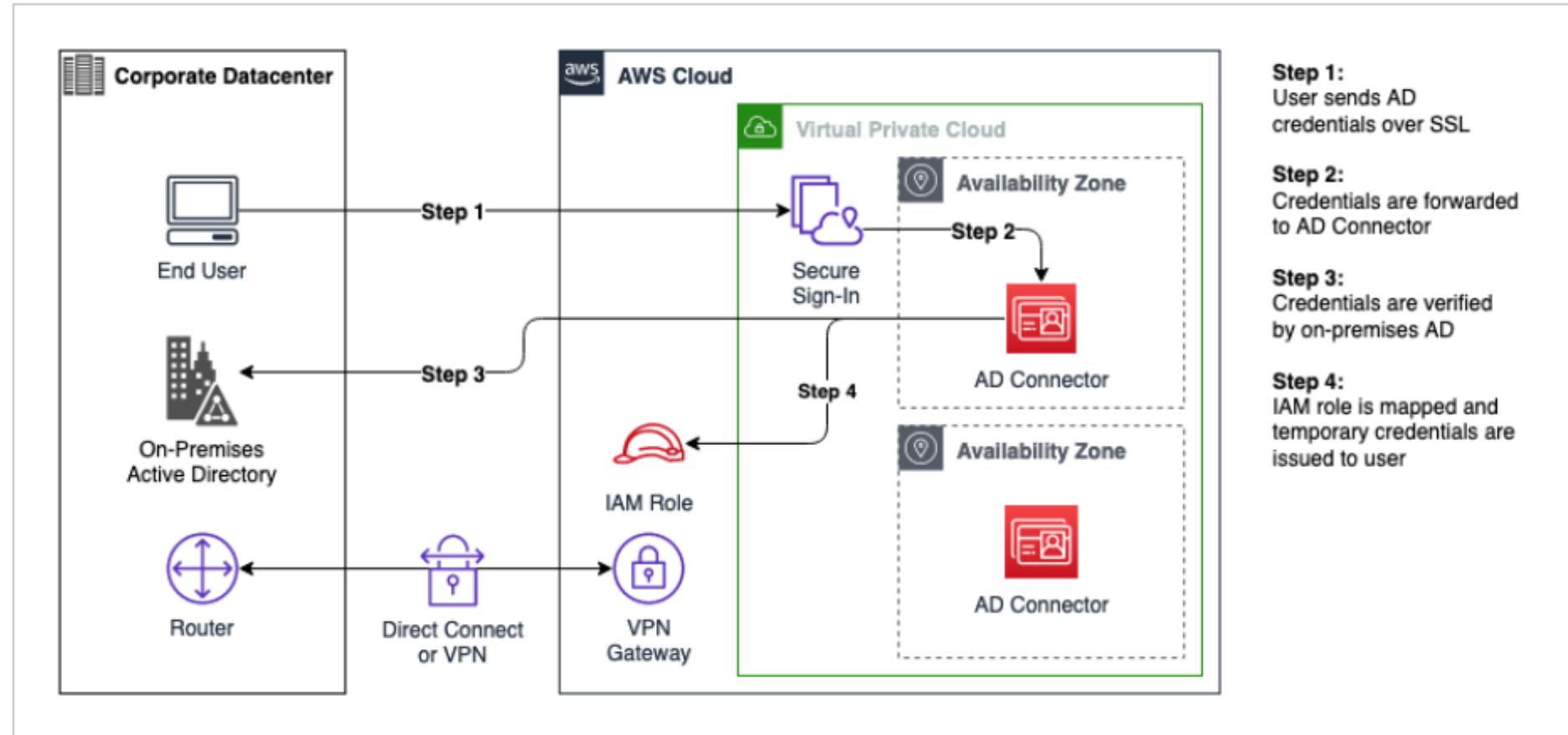
- AWS Managed Microsoft AD
 - Create your own AD in AWS, manage users locally, supports MFA
 - Establish “trust” connections with your on-premise AD
- AD Connector
 - Directory Gateway (proxy) to redirect to on-premise AD
 - Users are managed on the on-premise AD
- Simple AD
 - AD-compatible managed directory on AWS
 - Cannot be joined with on-premise AD



AWS Managed Microsoft AD



AD Connector



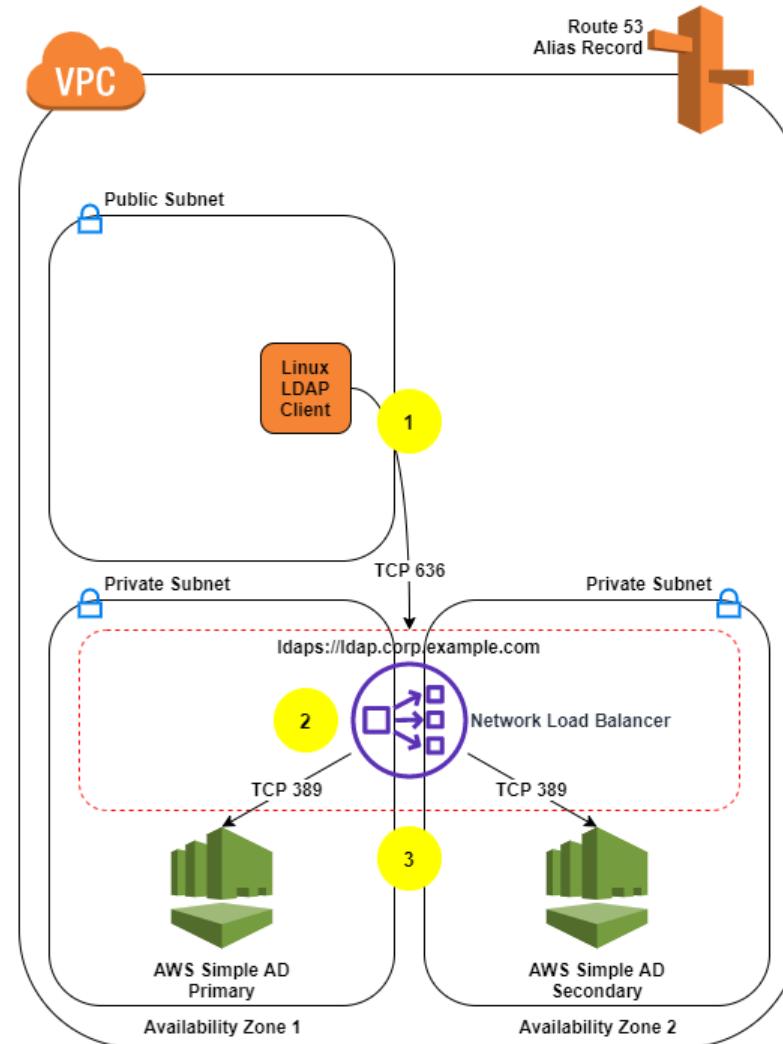
AD Connector

- AD connector is a directory gateway with which you can redirect directory requests to your on-premises Microsoft Active Directory without caching any information in the cloud.
- AD connector comes in two sizes: **Small** and **Large**

Benefits:

- Your end users and IT administrators can use their existing corporate credentials to log on to AWS applications such as Amazon Workspaces, Amazon WorkDocs or Amazon WorkMail.
- You can manage AWS resources like Amazon EC2 instance or Amazon S3 buckets through IAM role based access to the AWS management console
- You can consistently enforce existing security policies

Simple AD



Simple AD

- PAAS offering from AWS
- Based on LDAP as a managed directory service that is AD compatible
- Requires a VPC with two subnets that can reach the internet
- Two Sizes
 - Small: 500 users
 - Large: 5000 users
- Support for Linux and Windows instances
- Admin from Limited PowerShell API or Windows Server via RDP
- Support AWS Console Federation

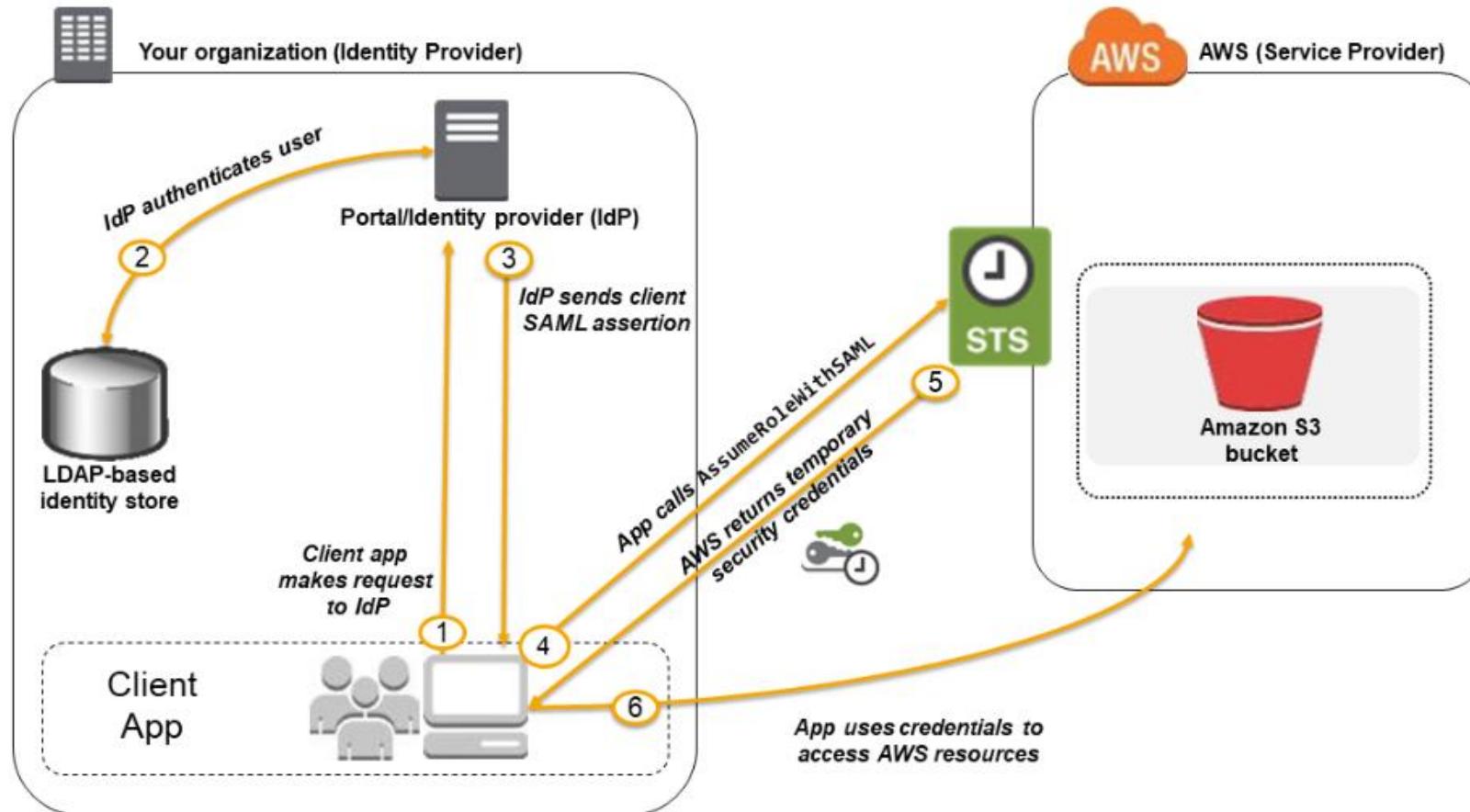
Identity Federation

- If you want users already have a way to be authenticated such as authentication through your corporate network. You can federate that user identity to AWS
- Federation is useful when user has the identity in a corporate directory and your corporate directory is compatible with “Security Assertion Markup Language 2.0”
- You can configure your corporate directory to provide Single Sign On (SSO) access to the AWS console
- If your directory is Microsoft Active Directory, you can use AWS directory service to establish trust between your corporate directory and your AWS account.

Security Assertion Markup Language

- SAML is an XML based open standard data format for exchanging authentication and authorization data
- Leverages central directory services
- User provides user id and password then redirected to requesting site
- Granted a level of access controlled by requesting site

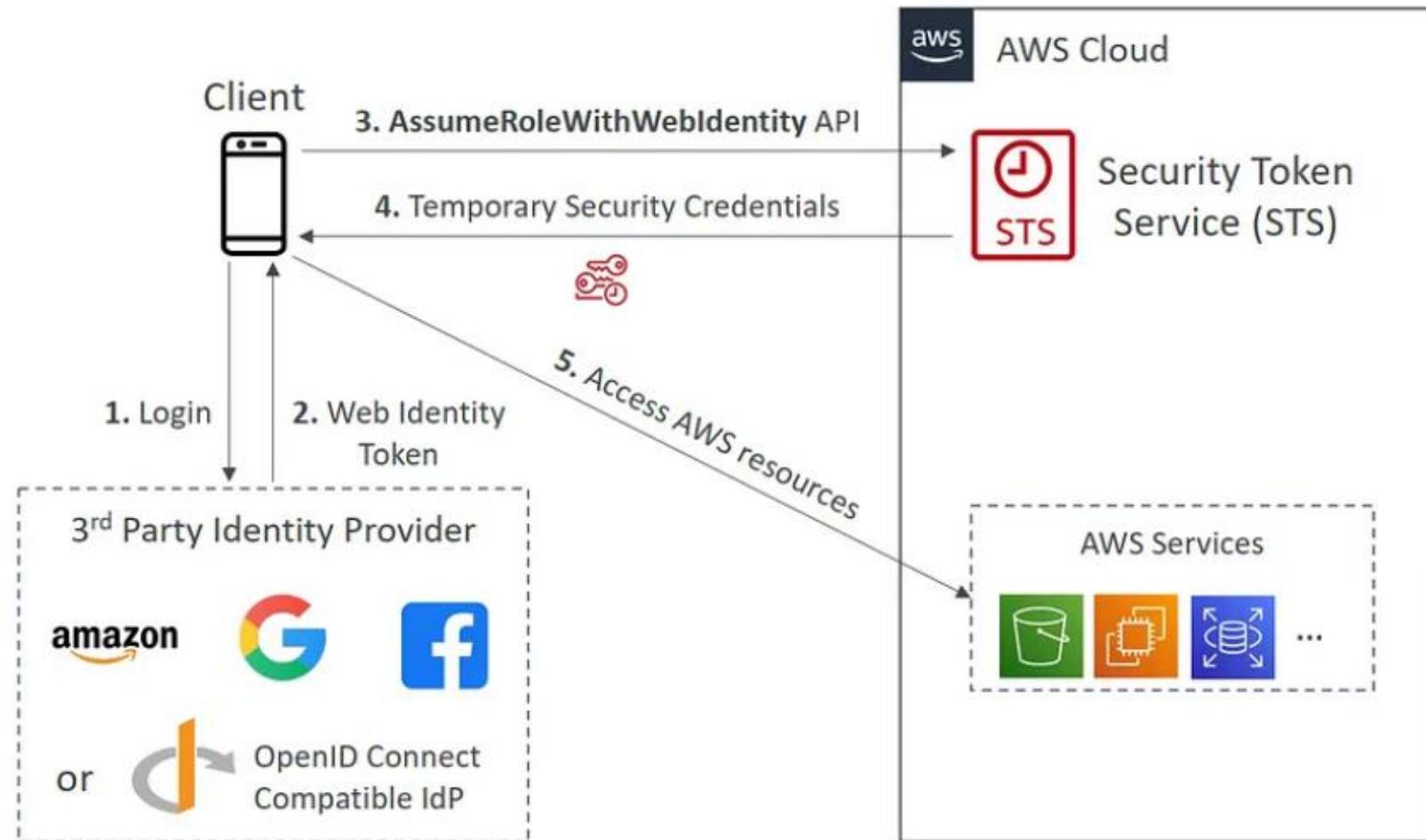
Identity Federation



Web Federation or OpenID Authentication

- Grants access to AWS resources from common directories
- Common Scenario: Developers off loads account management
- Web Federation & OpenID solves this account management
- User authenticate to common directory
- The app uses the **AssumeRoleWithWebIdentity** to create temporary credentials
- This does not grant access to the AWS console

IAM Federation

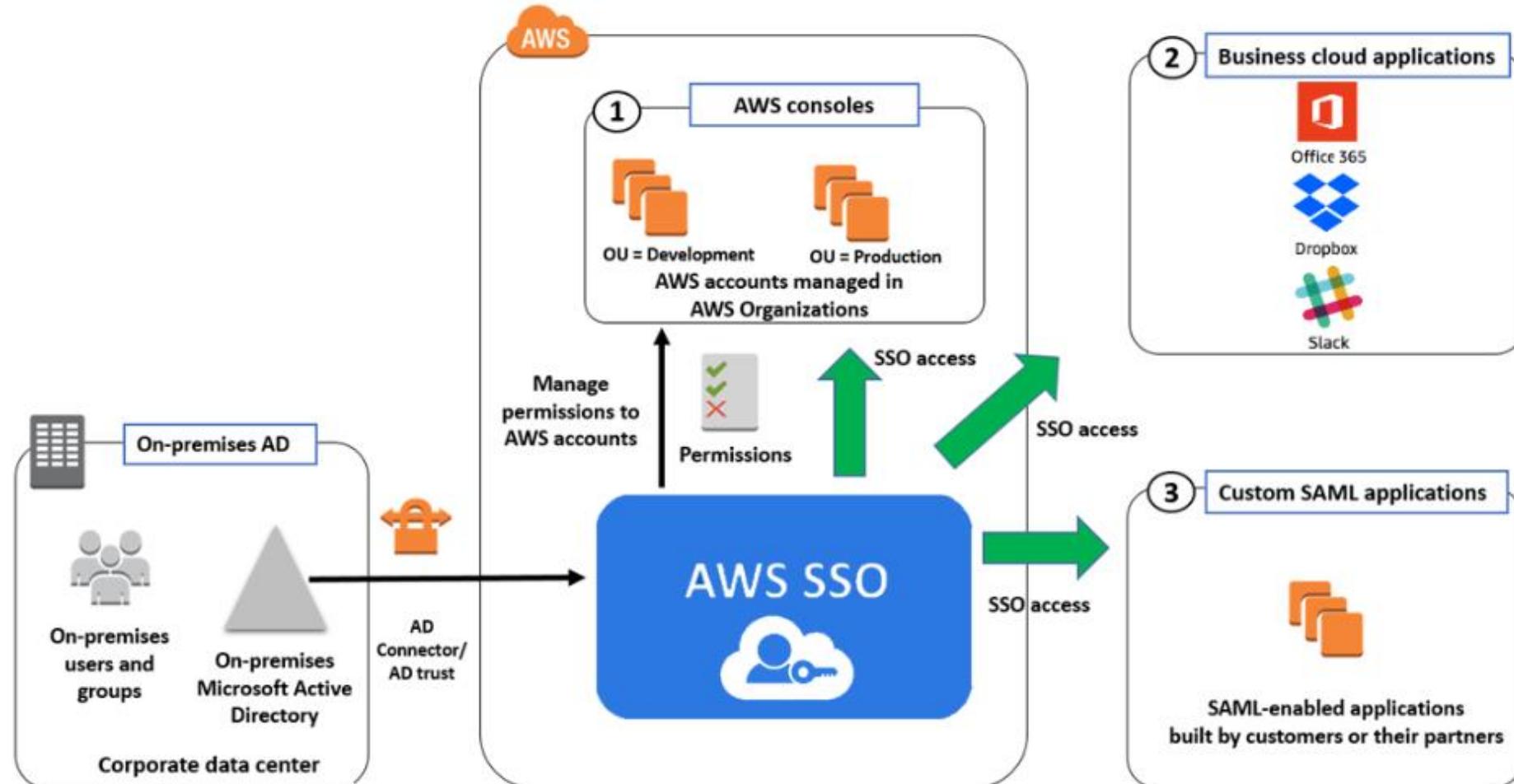


OpenID Authentication: Demo

GitHub OIDC connect with AWS

- <https://github.com/CloudSihmar/aws-oidc-new/tree/main>

AWS SSO



Load Balancers

Elastic Load Balancer

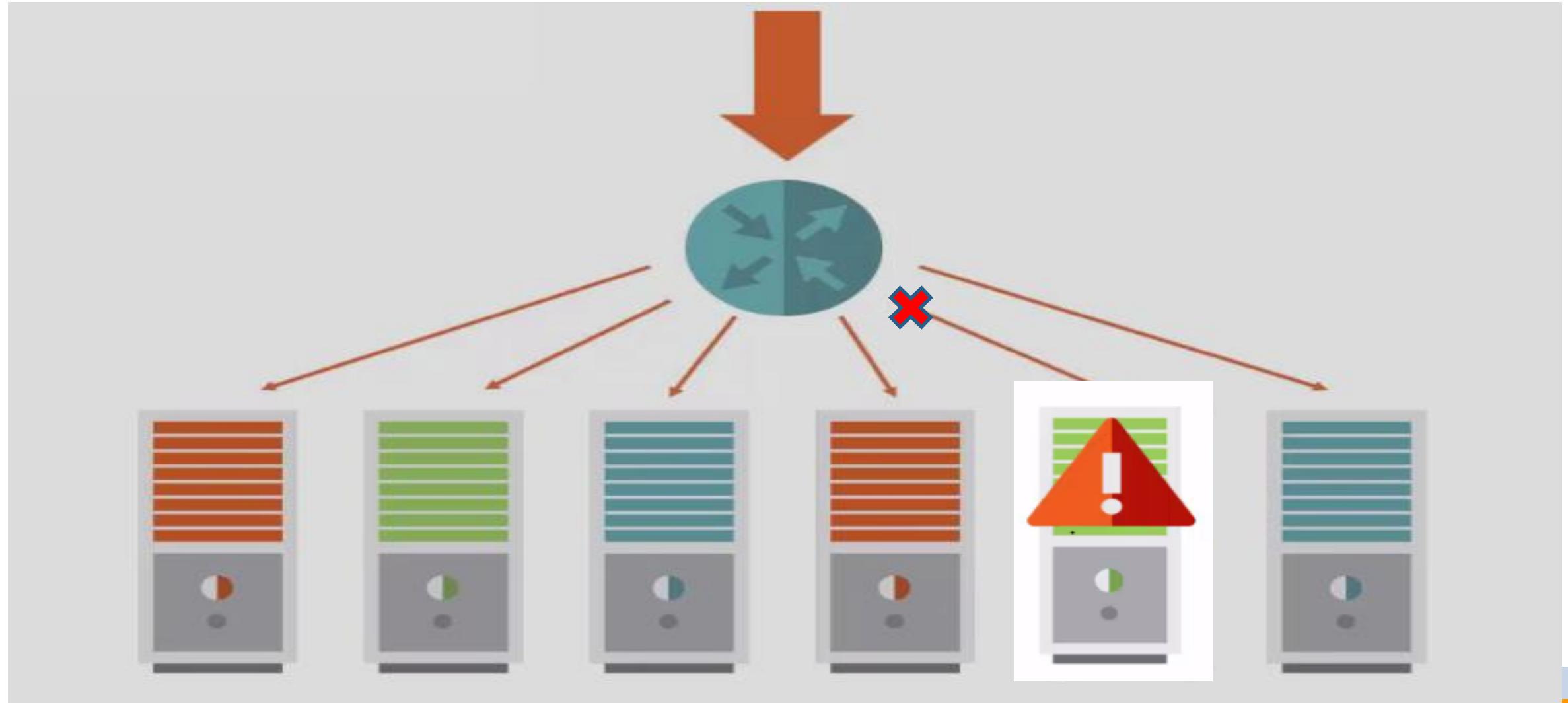
Distributes traffic across multiple instances

Supports **health checks** to detect unhealthy Amazon EC2 instances

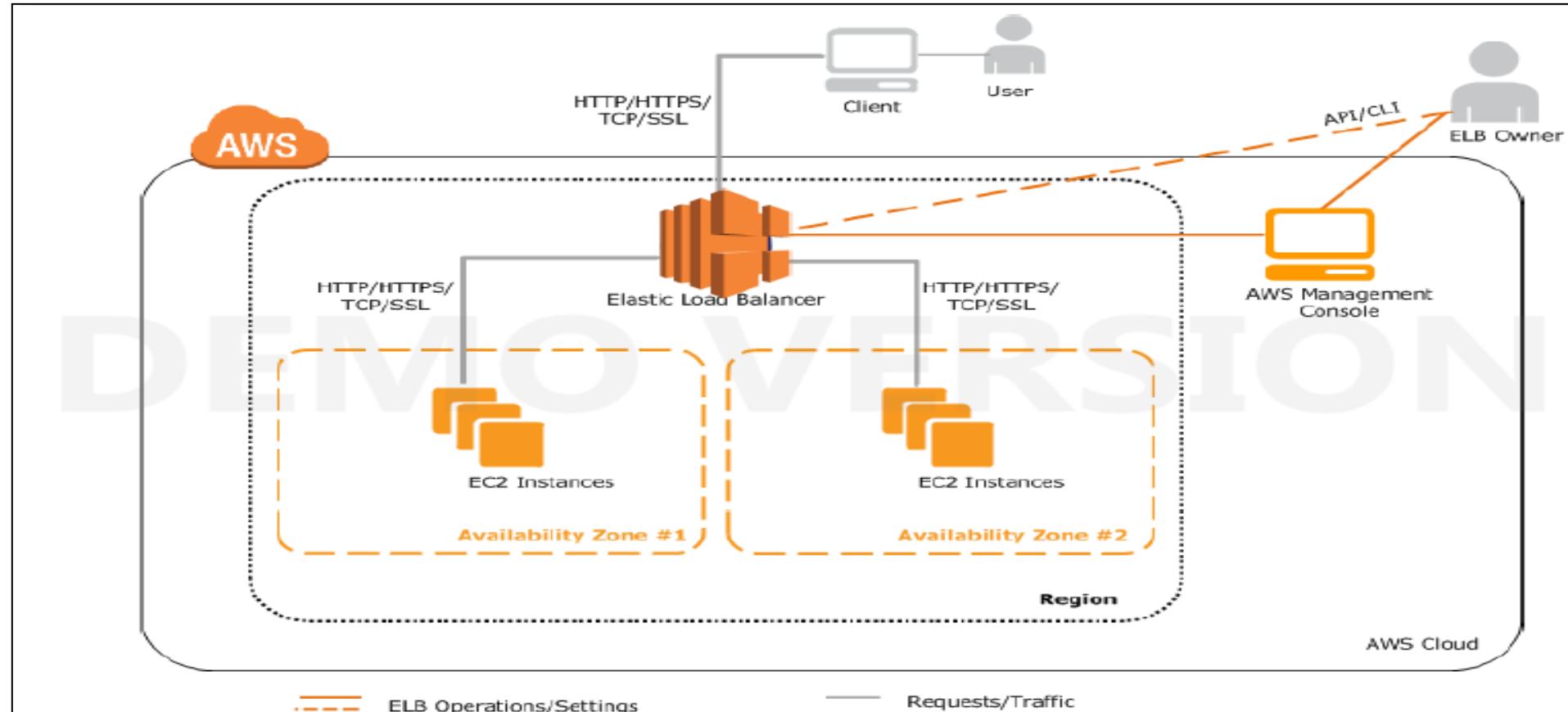
Supports the **routing and load balancing** of HTTP, HTTPS, and TCP traffic to Amazon EC2 instances

Works within Region and across Availability zones, which means it can divert traffic to servers in different AZs, but within same Region.

Elastic Load Balancer

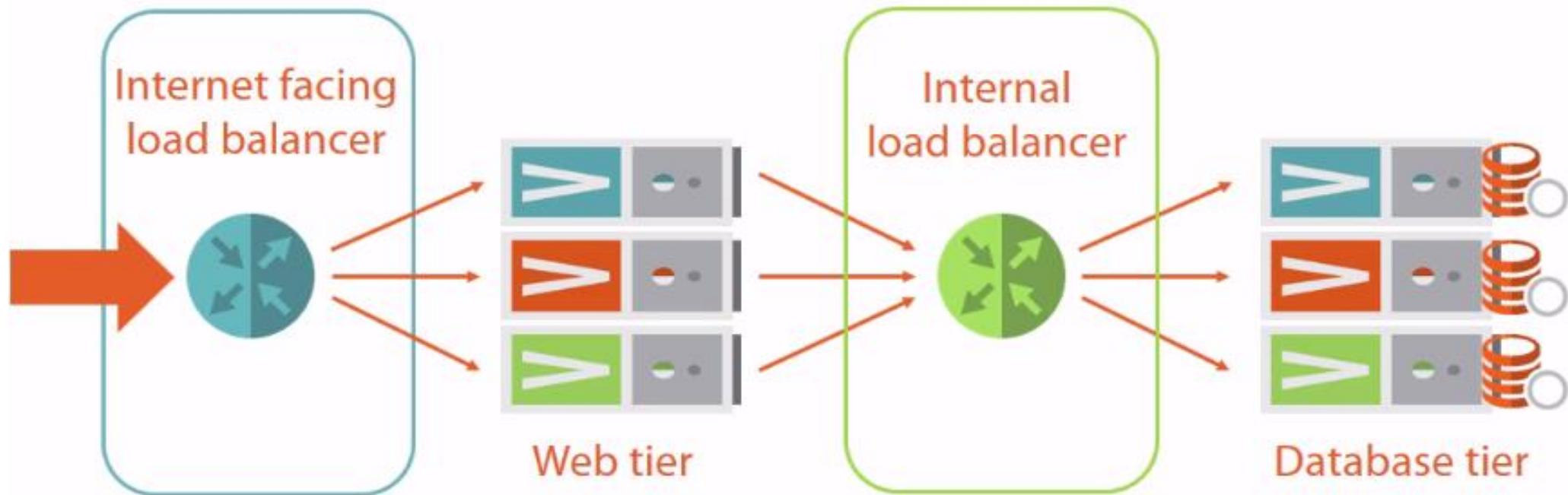


Elastic Load Balancer



Elastic Load Balancer

Tiered Applications & Load Balancers



Elastic Load Balancer

- **External/Internet Facing Load Balancer:** Receives Traffic from Open Internet. So carrying a public IP (or DNS name).
- **Internal Load Balancer:** Receives traffic from other VMs in same Network.
- **Frontend IP :** Public IP (Private IP in case of Internal LB) address for the incoming network traffic.
- **Target group :** Backend VM's Group which will receive actual traffic
- **Target:** Backend VM machines
- **Listener:** Port on ELB, which will accept request from end-client
- **Health checks:** Determines if a VM in the pool is healthy.

ELB Types

- **Application Load Balancer:**
 - An advance Load balancer which works on OSI Layer 7 (Application layer).
 - It contains flexible feature set (i.e. TLS termination, SSL handling etc.) for your web applications with HTTP and HTTPS traffic.
- **Network Load Balancer:**
 - Most commonly used Load balancer which works at any TCP port.
 - Works on Layer 4 (Transport layer).
 - LB with very high performance and capable of handling millions of requests per second while maintaining ultra-low latencies.
- **Gateway Load Balancer:**
 - AWS Gateway Load Balancer (GWLB), a service that makes it easy and cost-effective to deploy, scale and manage the availability of third-party virtual appliances such as firewalls, intrusion detection and prevention systems and deep packet inspection systems in the cloud.
- **Classic Load Balancer:**

Old type of Load-balancer service which works with EC2-Classic network. Not being used any more.

LAB 12 : Working with ELB (22)

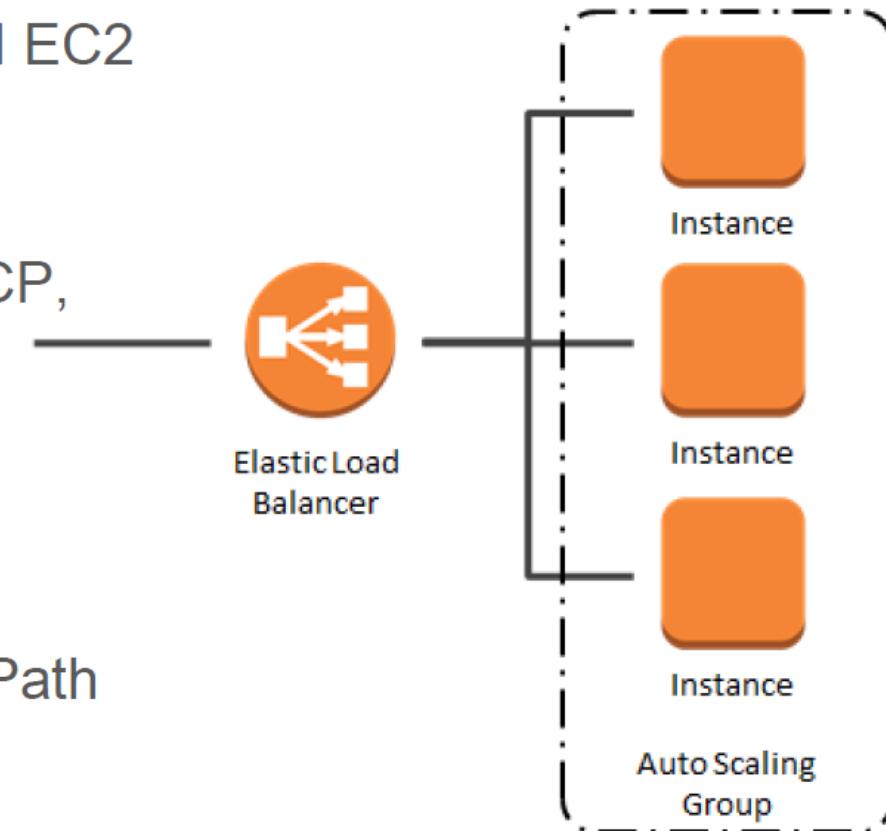
- Create two Ubuntu Linux servers in different-2 AZs
- Check that port 22 is accessible for both servers from public
- Go to EC2 Instance → LB → Load balancers → Create Load balancers
 - Select Network Load balancer
 - Provide any name and under listener select port 80
 - Select both AZs
 - Create a target group with port 22
 - Under advance health check settings, override the port to port 22
 - Register your servers under Register target
 - Review and Launch
- Cross-check the status of ELB and registered targets
- SSH the ELB listener IP on port 80 and observe the result
- Shutdown 1 server and then observe the traffic diversion
- Delete Load balancer

LAB 13 : Working with ELB (80)

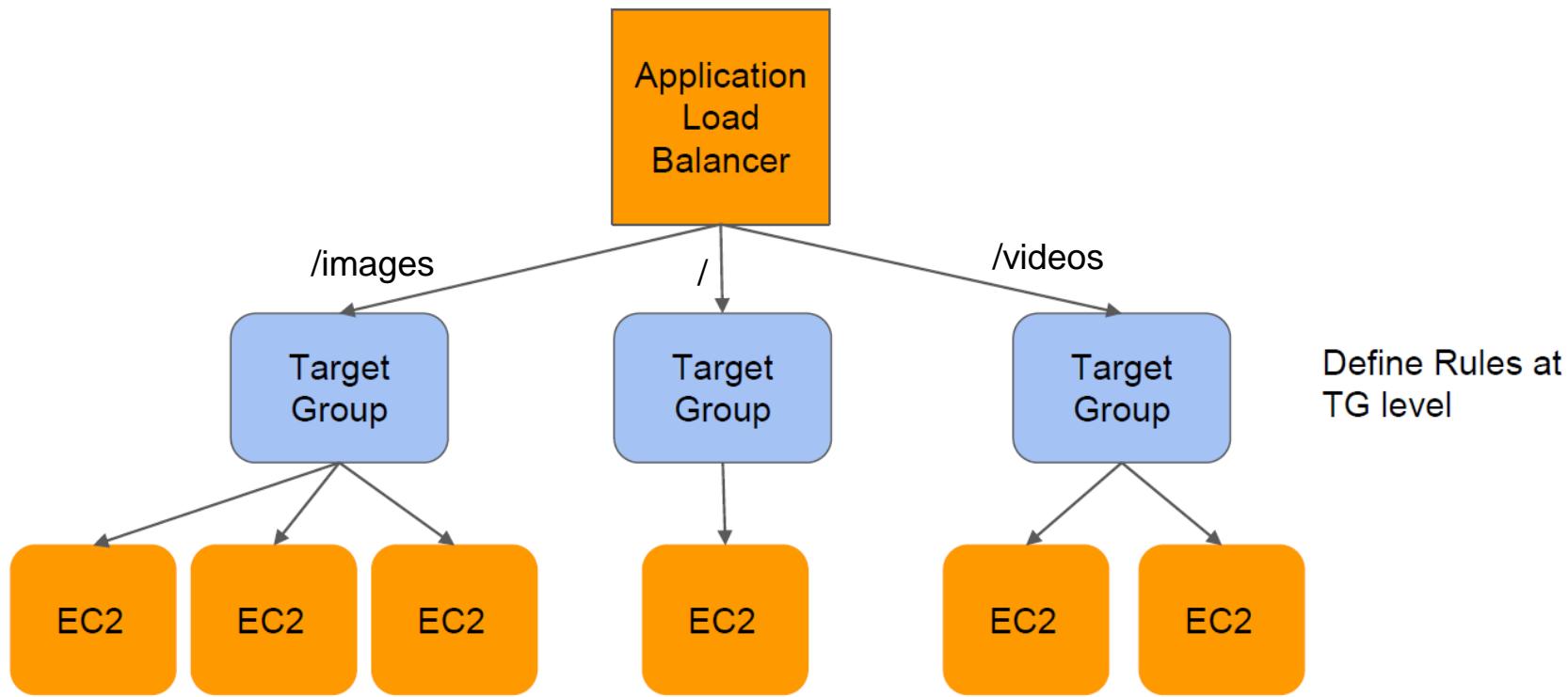
1. Login into your instances and configure Web service with the given commands in chat.
2. Run "curl localhost" in both servers to make sure that web service is running
3. Add security group rule to open traffic for your VPC range on port 80
4. Go to EC2 Instance -> LB -> Load balancers -> Create Load balancers
5. Select Network Load balancer
6. Provide your name-lb under name section and select type as Internet Load balancer.
7. Select your vpc and all azs subnets
8. Select next, next, to configure Routing tab
9. Add new target group with your name-tg
10. Target type instances
11. Select "next" and go to next step of register target
12. Select your instances (webservers) and click on "ADD TO REGISTERED" button
13. Click on Next- Review and Create

Application Load Balancer

- Distribute incoming traffic across attached EC2 instances across AZs
- Highly available and Scalable
- Supports protocols like HTTP, HTTPS, TCP, SSL (termination)
- Receives Public DNS Canonical name (CNAME) for endpoint
- Performs instance health checks
- Supports various routing - Round Robin, Path based, Host Based



Path based Forwarding



Cloud Application Security

Cloud Application Security

Software Development Life Cycle

- Analysis
- Design
- Implementation
- Testing

Mitigation Techniques

- Input Validation
- Error and exception handling
- XSS mitigation
- Testing
- Application Hardening

Error and Exception Handling

- Errors can expose useful information to attackers, and should be trapped and handled securely
 - Log Files
 - Do not display details errors to users (debug mode)
- Error handling:
 - Function based (evaluates the return)
 - Structured (try/catch) which is the preferred method
- Fail Safe
 - Data in transit
 - Ephemeral data
- Noise

Input Validation

- SQL Injection
- XSS
- Path Traversal
- Canonicalization attacks
 - Using various encoding to inject malicious data
 - See RFC 2279
- Consider All input to be hostile: trust no one
 - Sanitize all input
 - Verify encoding
 - Remove illegal characters
 - Whitelist acceptable values

Fuzz Testing

- **Fuzzing**
 - Systematically providing random data as input
 - Exception or crashes can be analyzed to find the flaw
- **Can be effective for detecting**
 - Memory leaks
 - Buffer Overflow

Cross Site Scripting Prevention

- Recall that XSS attacks can be
 - Non-persistent
 - Persistent
 - DOM-Based
- **XSS Attacks**
 - Theft of authentication info
 - Session hijacking
 - Deploy hostile content
 - Change settings
 - Phishing
- **Protect by**
 - Anti-XSS libraries
 - Limiting types and sizes of inputs and uploaded files
 - Whitelisting

Cross Site Request Forgery

- **Exploits a sit's trust with authenticated users**
 - Bank example from the text
- **Mitigation Techniques**
 - Limit authentication times
 - Cookie expiration
 - Header checking
 - Random tokens in form submissions

Application Hardening

- **Techniques:**

- Remove unnecessary components
- Ensure proper configuration
- Updates and patches

Application Configuration Baseline

- **Techniques:**

- Applications and security settings
- Access control
- Documentation

Application Patch Management

- Run the most secure and recent version of an application
- A formal process should be established to ensure application of patches

NoSQL vs SQL

- **Relational vs Non-relational Databases**
 - Newer technologies and frameworks represent new vectors or attack

Server-Side vs Client-Side Validation

- **Form Validation can occur client or server side**
 - **Typically, client side validation done using JavaScript**
 - **Typically, Server side validation done using server side scripting**
 - PHP
 - ASP.NET
 - Java
 - Ruby
 - Node.JS
 - Etc
- **Form validation on the client side can not be trusted**
- **All form validation should be done on the server side**

Cloud Application Security

Training and Awareness for Application Security

- Cloud Development Basics
- Common cloud vulnerabilities
- Common Pitfalls
 - Developers must also determine security requirements based on the selected deployment model
 - Cloud based applications may also have a higher reliance on operational metrics
 - Migration
 - Documentation
 - API challenges

Cloud Application Security

Describe the Secure Software Development Life Cycle (SDLC) Process

- Business Requirements
 - MSDL
 - NIST 800-64
 - ISO 27034-1 (ONF, ANF)
- Phases and Methodologies
- Shift left approach
- Devops and DevSecOps

Cloud Application Security

Apply the Secure Software Development Life Cycle

Avoid Common Vulnerabilities during Development

Cloud specific risks

Quality Assurance

Threat Modelling (During the design phase)

Software Configuration Management and Versioning (Chef, Ansible, Puppet)

Favourable Contract

OpenAPI (Help in migrating to another cloud)

Audit

Cloud Application Security

Apply Cloud Software Assurance and Validation

- Functional Testing
- Security Testing Methodologies
 - Black box Testing
 - White Box Testing
 - SAST
 - DAST

Cloud Application Security

Use verified Secure Software

- Approved application programming interfaces (API)
 - SOAP (heavy weight)
 - REST (Light Weight)
- Supply chain management (verify third party applications)
- Third Party Management
- Validated Open Source Software

Cloud Application Security

Comprehend the specifics of Cloud Application Architecture

- Supplemental Security Components
 - Web Application Firewall (WAF) (Mod Security)
 - Database activity Monitoring (DAM) (SQL injection attack, DB attack)
 - Extensible Markup Language (XML) firewalls (Parser Leveral attack)
 - Application Programming Interface (API) gateway
- Cryptography (SSL, TLS, Side channel attack – Cold Boot Attack)
- Sandboxing (Testing, research, updates, containerization)
- Application virtualization and Orchestration

Cloud Application Security

Design Appropriate Identity and Access Management (IAM) Solutions

- Federated Identity
- Identity Providers
- Single Sign-On (SSO)
- Multi-factor Authentication
- Cloud Access Security Broker (CASB)

Cryptography

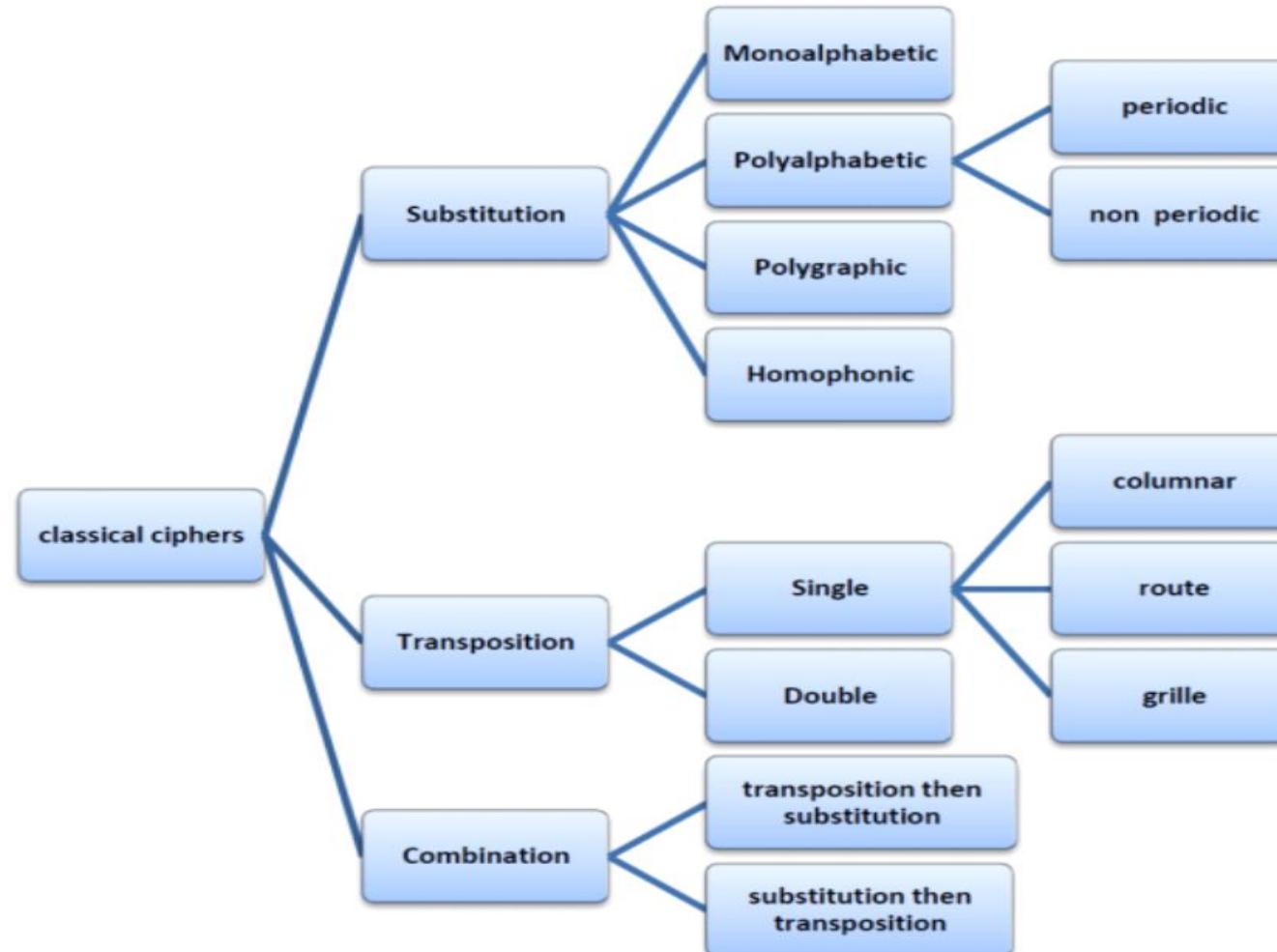
Cryptography is made of :

- Crypto - Secret
- Graphy – Writing

Types:

- Modern (Encryption)
- Classical (Encoding)

Cryptography



Cryptography

Transposition: Changing the position

For example:

Key = 1

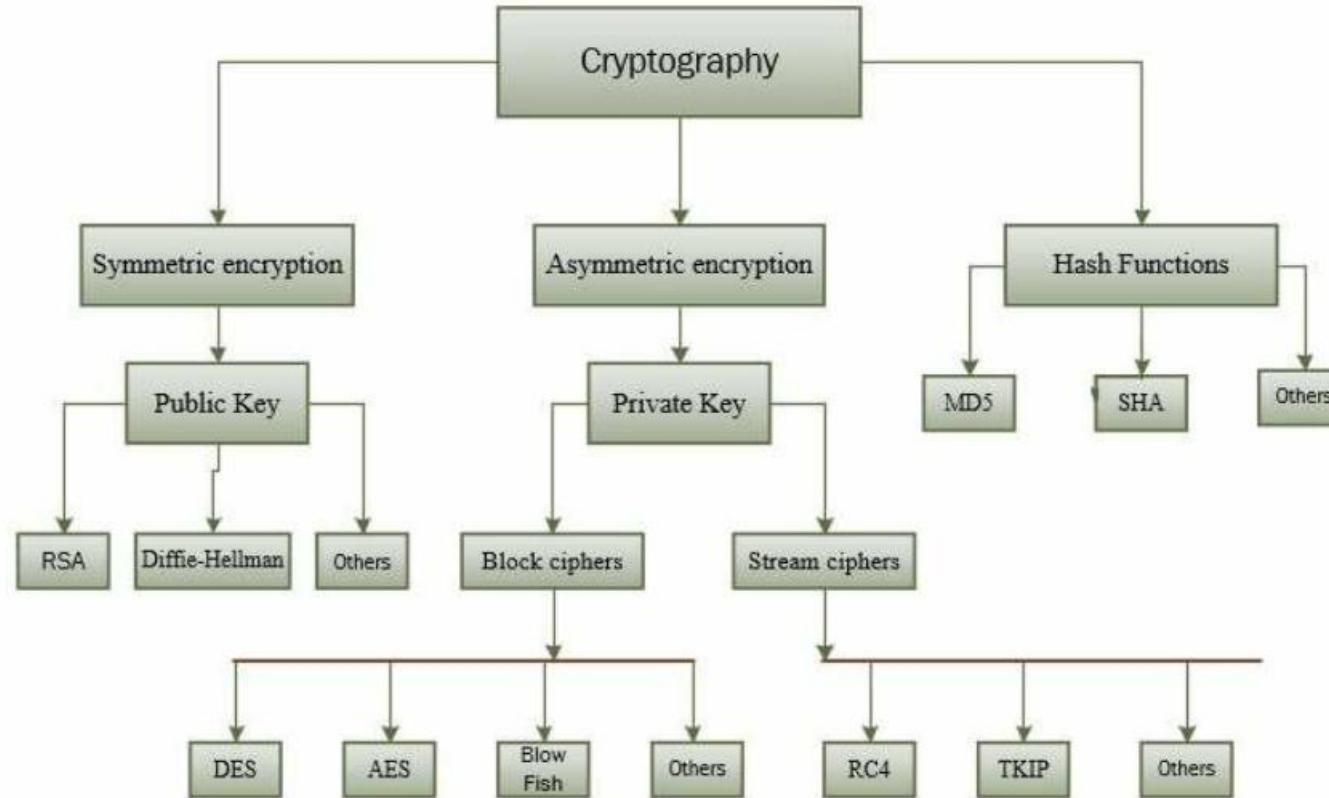
Algorithm= +1

Plain Text = ABCDE → Cipher Text = BCDEA

Substitution: Substitute the value

Plain Text = ABCDE → Cipher Text = BCDEF

Cryptography



Cryptography

Symmetric:

- Shared Key Cryptography
- Data Encryption

Asymmetric:

- Public Key Cryptography
- Key Exchange

Hybrid:

- It uses both Symmetric and Asymmetric methods

Cryptography

Goals:

Confidentiality: Unauthorized parties cannot access information

Authenticity: Validating the source of the message to ensure the sender is properly identified. To get the proof of origin

Integrity: Assurance that the message was not modified during transmission, accidentally or intentionally

Non-repudiation: A sender can not deny sending the message at a later date, It is achieved through digital signature.

API Gateway

API Gateway



API Gateway

- AWS Lambda + API Gateway: No infrastructure to manage
- Support for the WebSocket Protocol
- Handle API versioning (v1, v2...)
- Handle different environments (dev, test, prod...)
- Handle security (Authentication and Authorization)
- Create API keys, handle request throttling
- Swagger / Open API import to quickly define APIs
- Transform and validate requests and responses
- Generate SDK and API specifications
- Cache API responses

API Gateway Integrations

Lambda Function

- Invoke Lambda function
- Easy way to expose REST API backed by AWS Lambda

HTTP

- Expose HTTP endpoints in the backend
- Example: internal HTTP API on premise, Application Load Balancer...
- Why? Add rate limiting, caching, user authentications, API keys, etc...

AWS Service

- Expose any AWS API through the API Gateway?
- Example: start an AWS Step Function workflow, post a message to SQS
- Why? Add authentication, deploy publicly, rate control...

API Gateway Integrations

Edge-Optimized (default): For global clients

- Requests are routed through the CloudFront Edge locations (improves latency)
- The API Gateway still lives in only one region

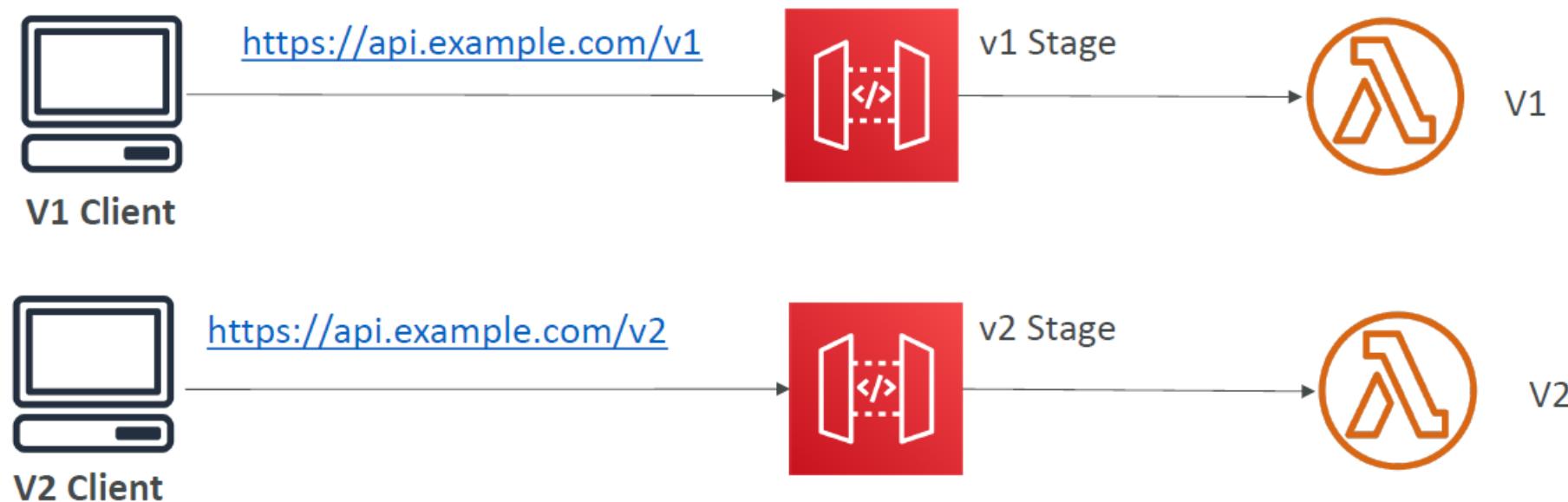
Regional:

- For clients within the same region
- Could manually combine with CloudFront (more control over the caching strategies and the distribution)

Private:

- Can only be accessed from your VPC using an interface VPC endpoint (ENI)
- Use a resource policy to define access

API Gateway Integrations



API Gateway Stage Variables

Stage variables are like environment variables for API Gateway

Use them to change often changing configuration values

They can be used in:

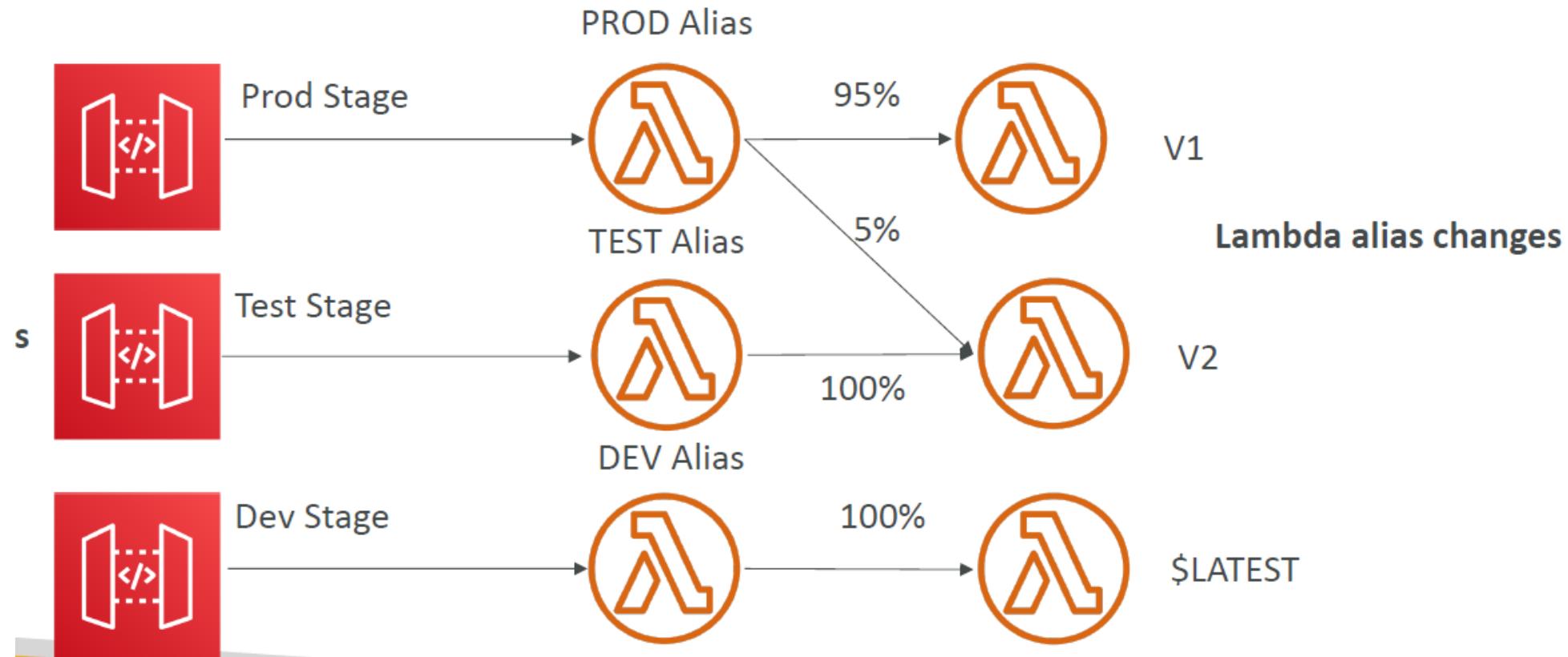
- Lambda function ARN
- HTTP Endpoint
- Parameter mapping templates

Use cases:

- Configure HTTP endpoints your stages talk to (dev, test, prod...)
- Pass configuration parameters to AWS Lambda through mapping templates

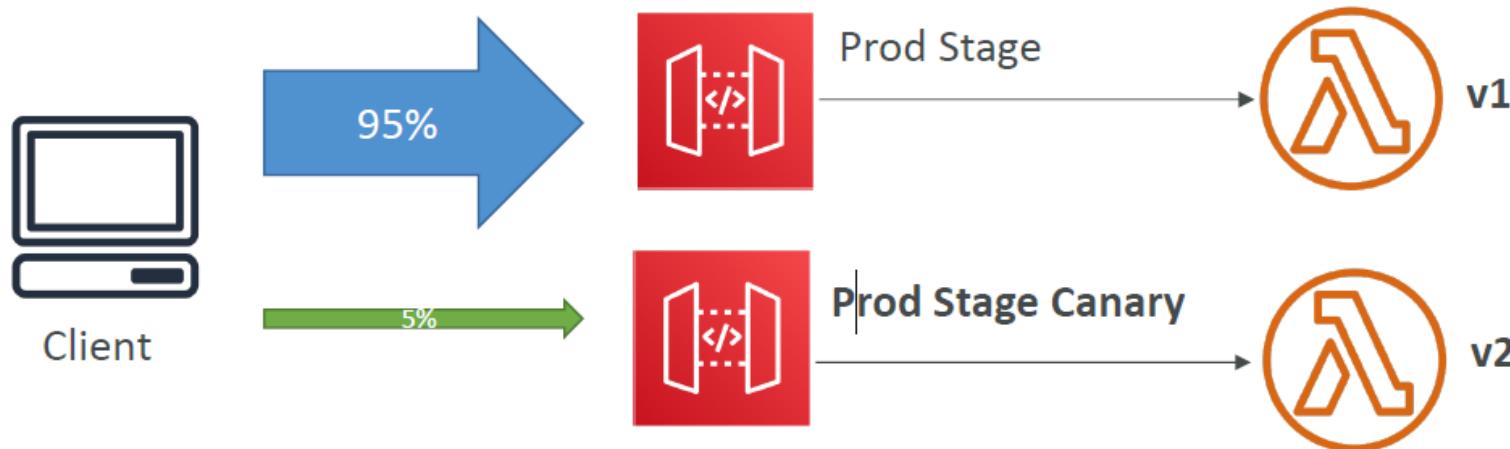
Stage variables are passed to the "context" object in AWS Lambda

Using stages and Alias



Canary Deployment

- Possibility to enable canary deployments for any stage (usually prod)
- Choose the % of traffic the canary channel receives



- Metrics & Logs are separate (for better monitoring)
- Possibility to override stage variables for canary
- This is blue / green deployment with AWS Lambda & API Gateway

Mapping Templates

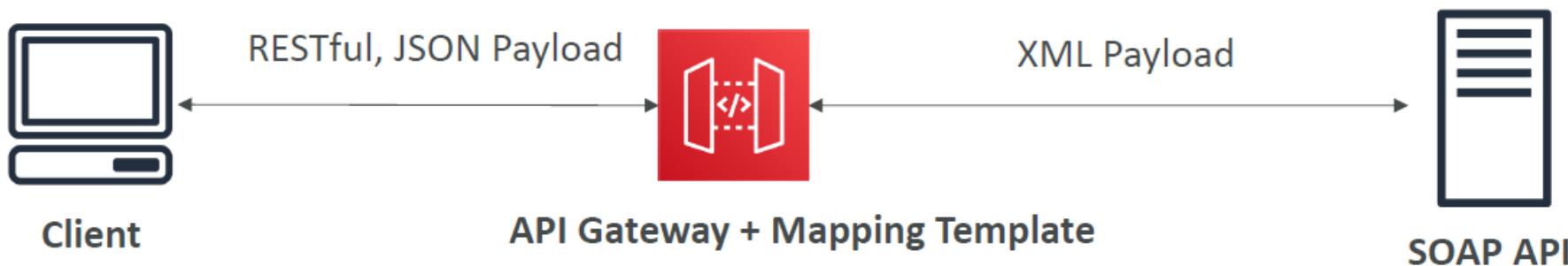
- Mapping templates can be used to modify request / responses
- Rename / Modify query string parameters
- Modify body content
- Add headers
- Uses Velocity Template Language (VTL): for loop, if etc...
- Filter output results (remove unnecessary data)

Mapping Example: JSON to XML with SOAP

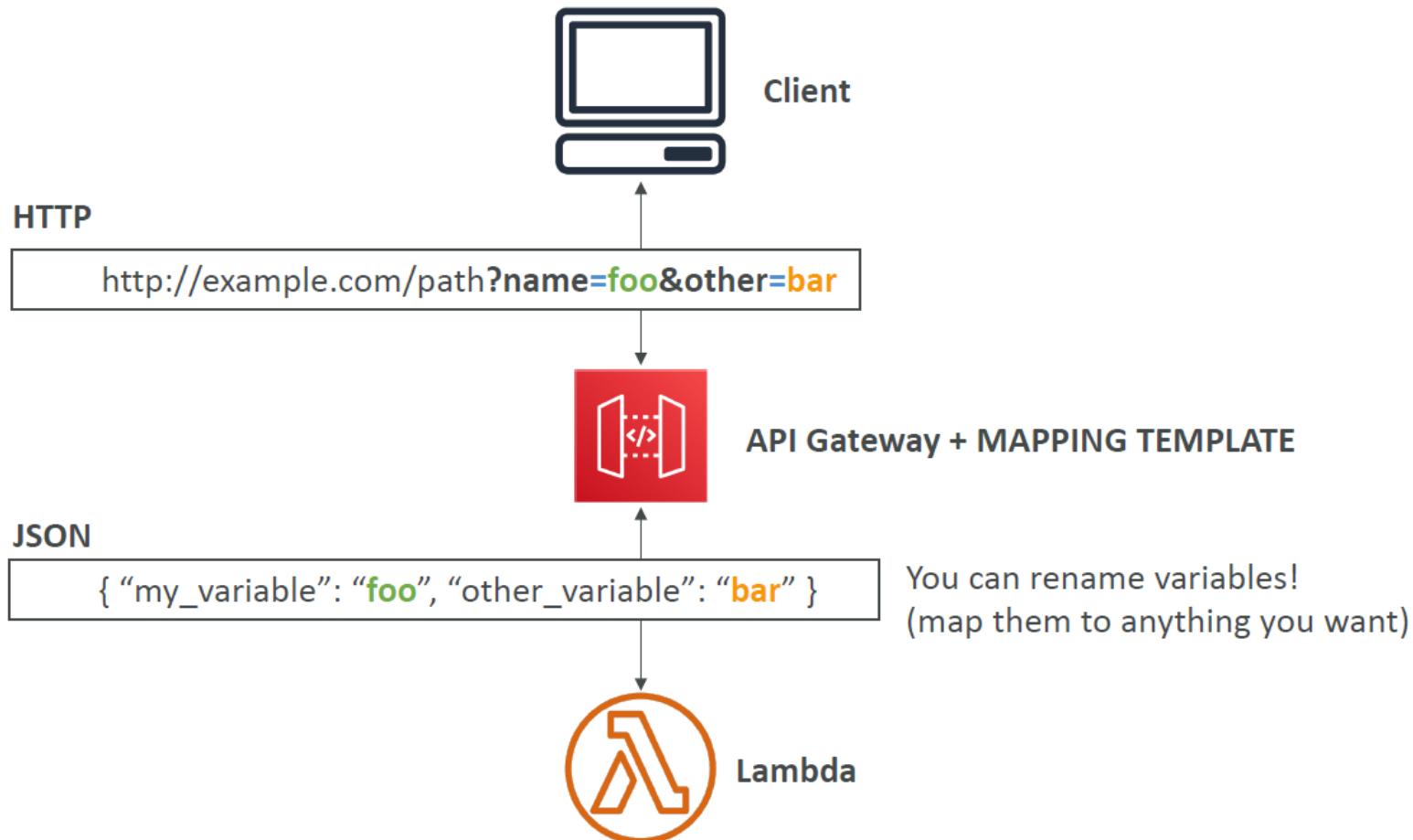
SOAP API are XML based, whereas REST API are JSON based

In this case, API Gateway should:

- Extract data from the request: either path, payload or header
- Build SOAP message based on request data (mapping template)
- Call SOAP service and receive XML response
- Transform XML response to desired format (like JSON), and respond to the user



Mapping Example: JSON to XML with SOAP



Mapping Example: Demo

Demo

1. Create a lambda function

```
import json
def lambda_handler(event, context):
    # TODO: Implement your logic here
    return {
        'hello': "world"
    }
```

2. Create an API Gateway and create a template under integration response

```
{
    "title" : "new-data",
    "new-key" : $input.json('$.hello')
}
```

AWS API Gateway Swagger / Open API spec

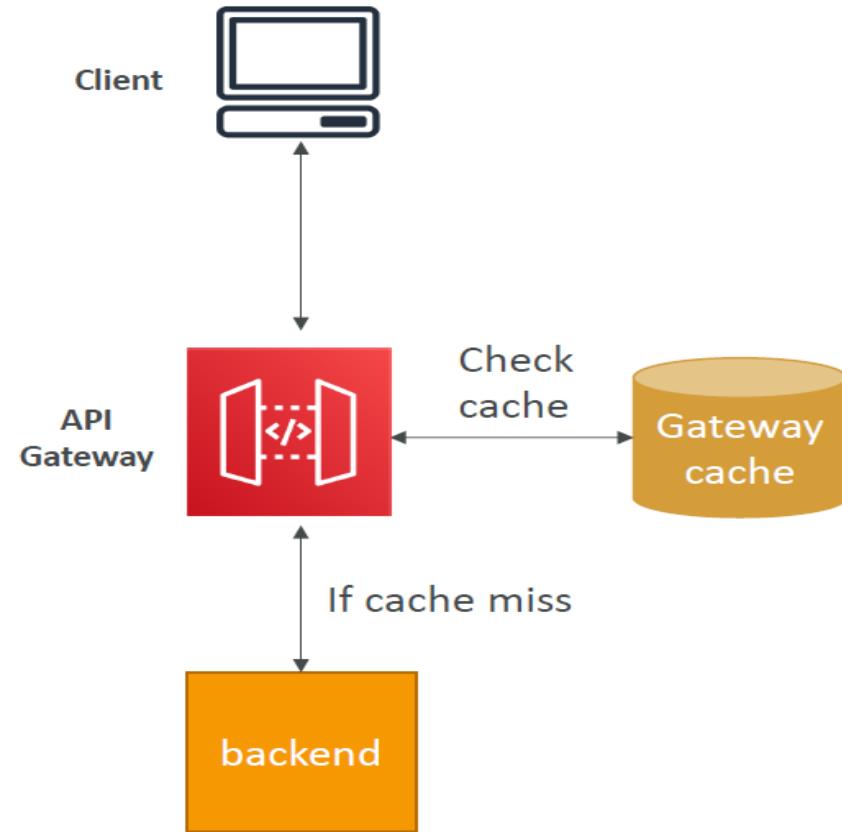
Common way of defining REST APIs, using API definition as code

- Import existing Swagger / OpenAPI 3.0 spec to API Gateway
 - Method
 - Method Request
 - Integration Request
 - Method Response
 - + AWS extensions for API gateway and setup every single option
- Can export current API as Swagger / OpenAPI spec
- Swagger can be written in YAML or JSON
- Using Swagger we can generate SDK for our applications

AWS API Gateway Caching

Caching reduces the number of calls made to the backend

- Default TTL (time to live) is 300 seconds
(min: 0s, max: 3600s)
- Caches are defined per stage
- Possible to override cache settings per method
- Cache encryption option
- Cache capacity between 0.5GB to 237GB
- Cache is expensive, makes sense in production, may not make sense in dev / test



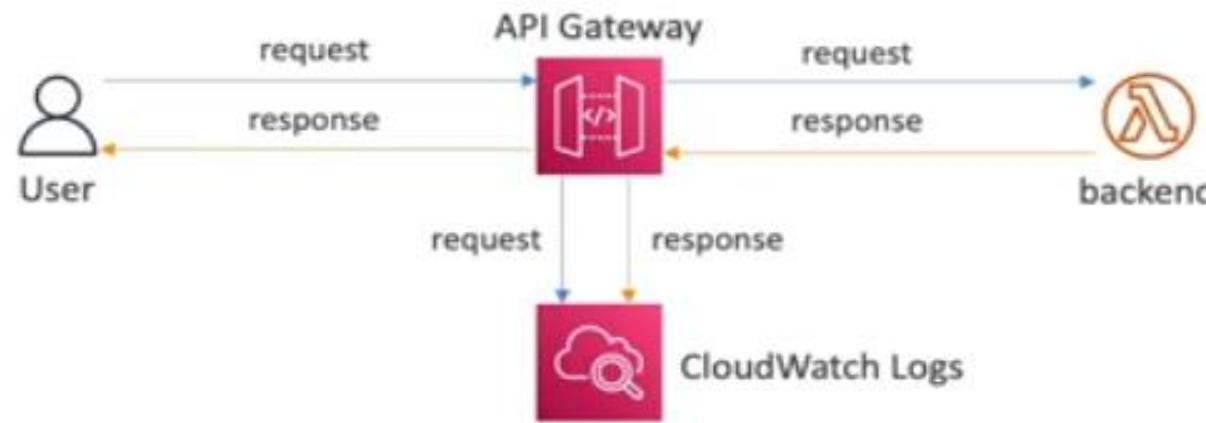
API Gateway – Usage Plans & API Keys

If you want to make an API available as an offering (\$) to your customers

- Usage Plan:
 - who can access one or more deployed API stages and methods
 - how much and how fast they can access them
 - uses API keys to identify API clients and meter access
 - configure throttling limits and quota limits that are enforced on individual client
- API Keys:
 - alphanumeric string values to distribute to your customers
 - Ex: WBjHxNtoAb4WPKBC7cGm64CBiblb24b4jt8jJHo9
 - Can use with usage plans to control access
 - Throttling limits are applied to the API keys
 - Quotas limits is the overall number of maximum requests

API Gateway – Logging and Tracing

- CloudWatch Logs
 - Log contains information about request/response body
 - Enable CloudWatch logging at the Stage level (with Log Level - ERROR, DEBUG, INFO)
 - Can override settings on a per API basis



- X-Ray
 - Enable tracing to get extra information about requests in API Gateway
 - X-Ray API Gateway + AWS Lambda gives you the full picture

API Gateway – Metrics

- Metrics are by stage, Possibility to enable detailed metrics
- CacheHitCount & CacheMissCount: efficiency of the cache
- Count: The total number API requests in a given period.
- IntegrationLatency: The time between when API Gateway relays a request to the backend and when it receives a response from the backend.
- Latency: The time between when API Gateway receives a request from a client and when it returns a response to the client. The latency includes the integration latency and other API Gateway overhead.
- 4XXError (client-side) & 5XXError (server-side)

OWASP API Top 10

OWASP API Top 10 (2019)		OWASP API Top 10 (2023)	
API1	Broken Object Level Authorization	API1	Broken Object Level Authorization
API2	Broken User Authentication	API2	Broken Authentication
API3	Excessive Data Exposure	API3	Broken Object Property Level Authorization
API4	Lack of Resources and Rate Limiting	API4	Unrestricted Resource Consumption
API5	Broken Function Level Authorization	API5	Broken Function Level Authorization
API6	Mass Assignment	API6	Server-Side Request Forgery
API7	Security Misconfiguration	API7	Security Misconfiguration
API8	Injection	API8	Lack of Protection from Automated Threats
API9	Improper Access Management	API9	Improper Asset Management
API10	Insufficient Logging and Monitoring	API10	Unsafe Consumption of APIs

■ Included in 2019 version

■ Removed

■ Newly introduced

API Gateway – Security

IAM:

- Great for users / roles already within your AWS account, + resource policy for cross account
- Handle authentication + authorization
- Leverages Signature v4

Custom Authorizer:

- Great for 3rd party tokens
- Very flexible in terms of what IAM policy is returned
- Handle Authentication verification + Authorization in the Lambda function
- Pay per Lambda invocation, results are cached

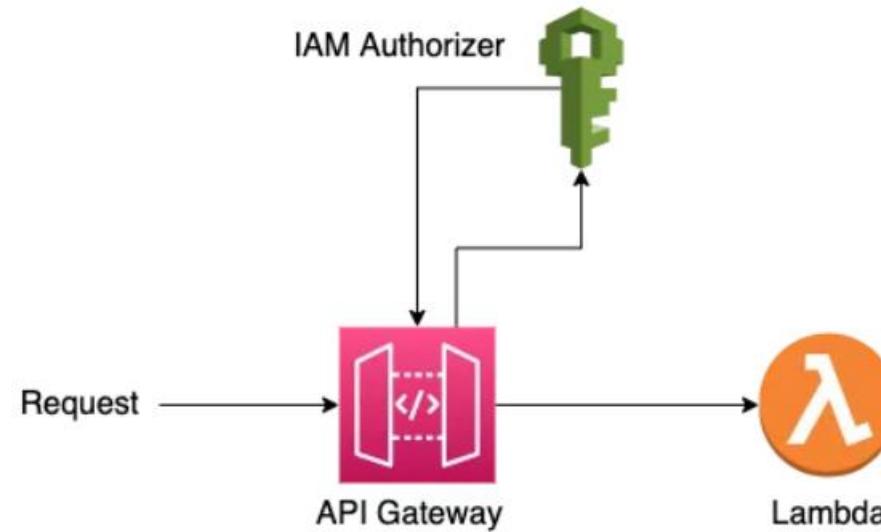
Cognito User Pool:

- You manage your own user pool (can be backed by Facebook, Google login etc...)
- No need to write any custom code
- Must implement authorization in the backend

API Gateway – Security

IAM:

- Great for users / roles already within your AWS account + **resource policy** for cross account
- Handle Authentication = IAM Authorization = IAM Policy
- Leverages Signature v4



API Gateway – Security

IAM:

- Create a group (Demo-group)
- Create a user (Demo-user)
- Create a policy for **ExecuteAPI → Invoke**
- Attach the policy to the user
- Enable IAM authentication in Api Gateway method

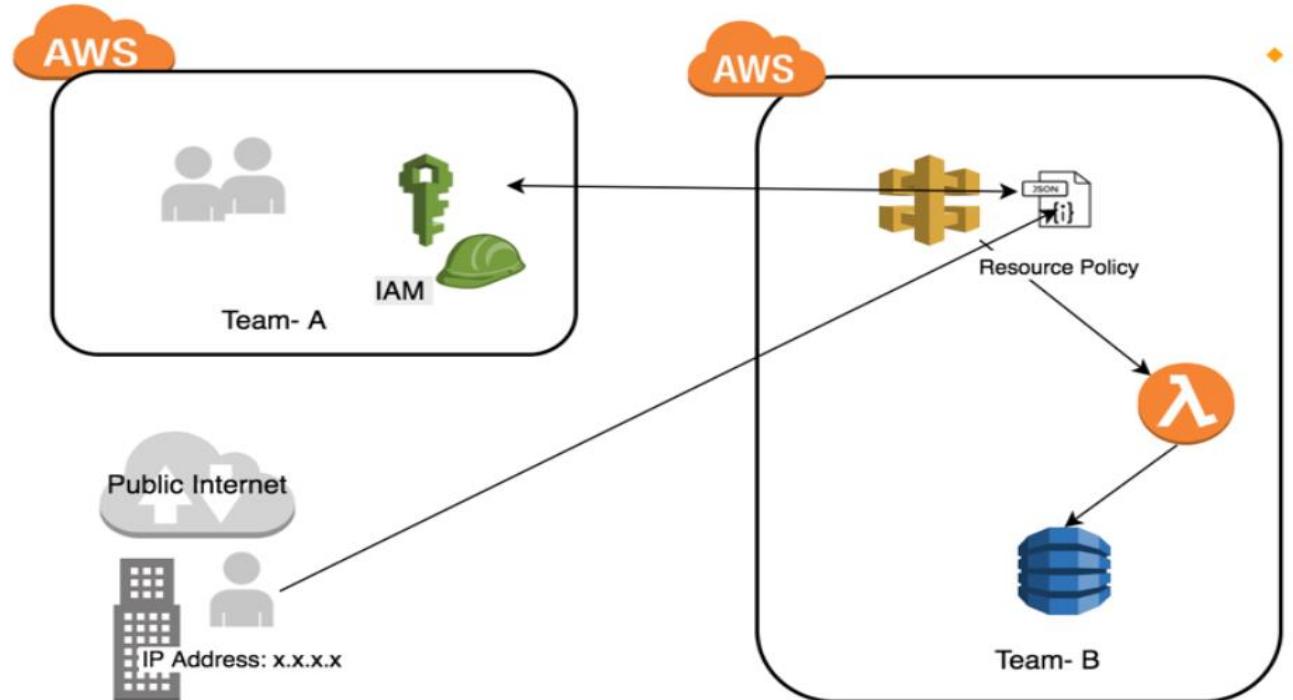
The screenshot shows the Postman interface for making a GET request to an API endpoint. The URL is `https://zs3xanyprg.execute-api.ap-south-1.amazonaws.com/test1/api-auth`. The 'Authorization' tab is selected, indicating AWS Signature authentication is being used. The 'AccessKey' field contains `AKIAZ7FSO3B5UVXHTS6U` and the 'SecretKey' field contains `jq1uGbJVxmyoxeVfLGH3wlHGBQRWqtG5Hd...`. Other tabs like 'Params', 'Headers (7)', 'Body', 'Scripts', 'Tests', 'Settings', and 'Cookies' are visible. A note on the left explains that the authorization header will be automatically generated when the request is sent. At the bottom, fields for 'AWS Region', 'Service Name', and 'Session Token' are shown with values `ap-south-1`, `execute-api`, and `Session Token` respectively.

API Gateway – Security

Using Resource Policy:

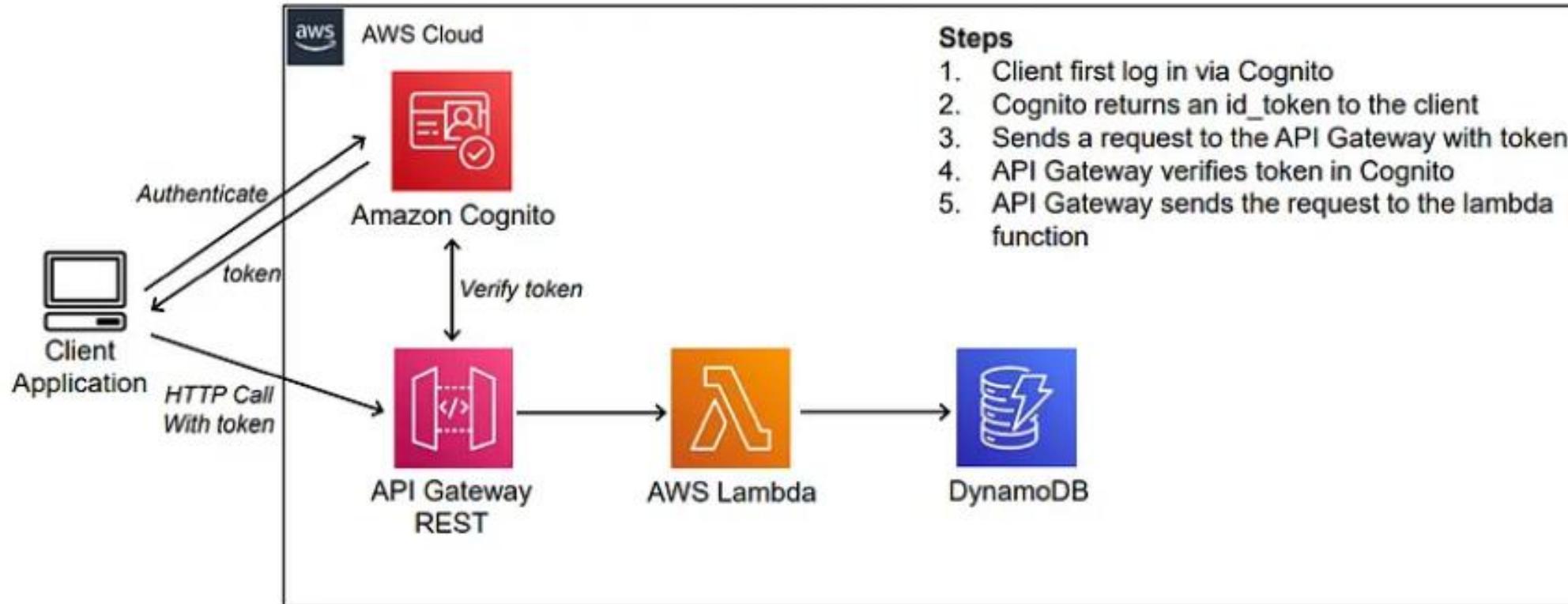
<https://github.com/CloudSihmar/api-gateway-ip-whitelisting.git>

- Allows for Cross Account Access
- Allow for a specific source IP address
- Allow for a VPC endpoint



API Gateway – Security

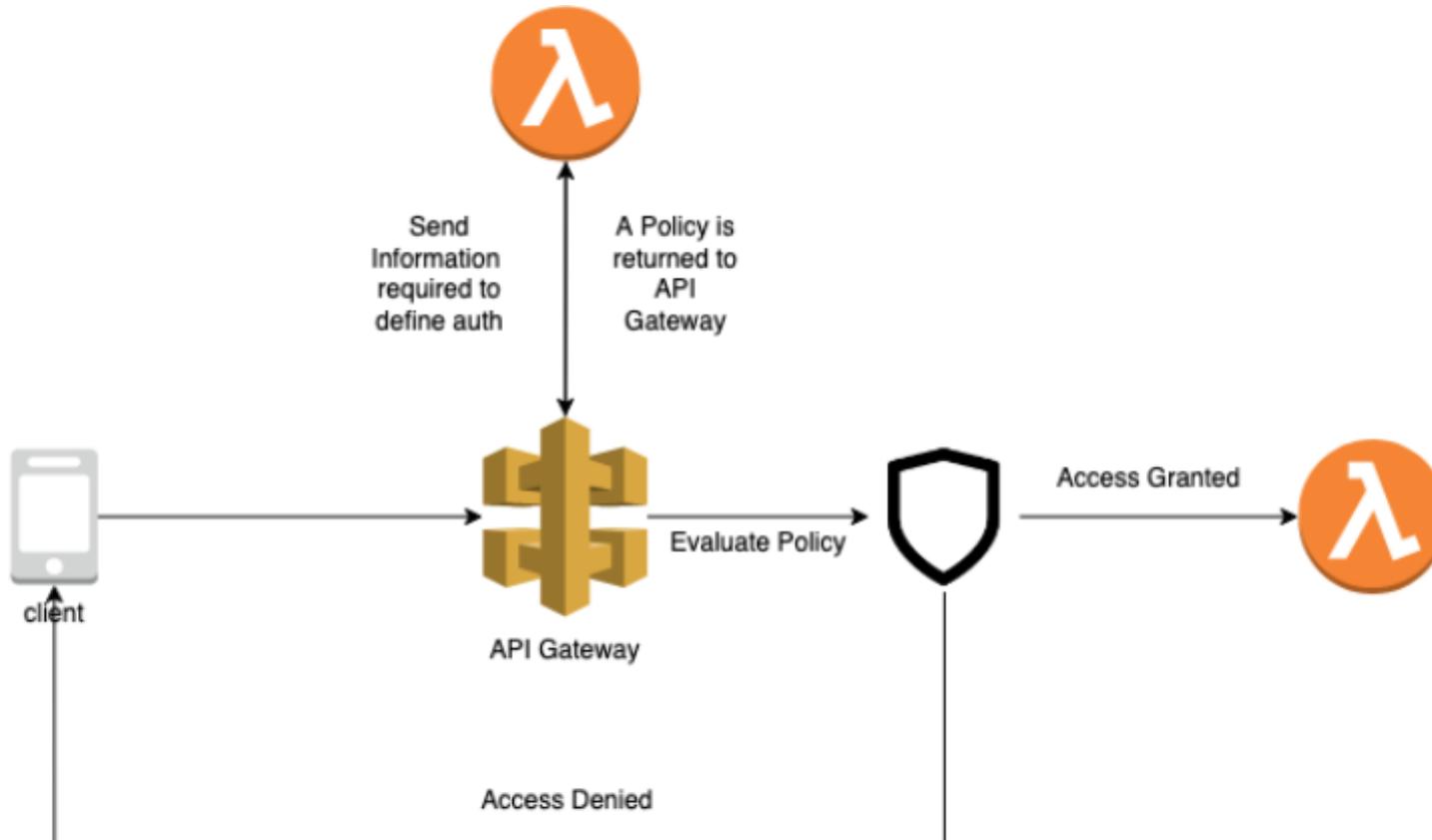
Using Cognito User Pool:



API Gateway – Security

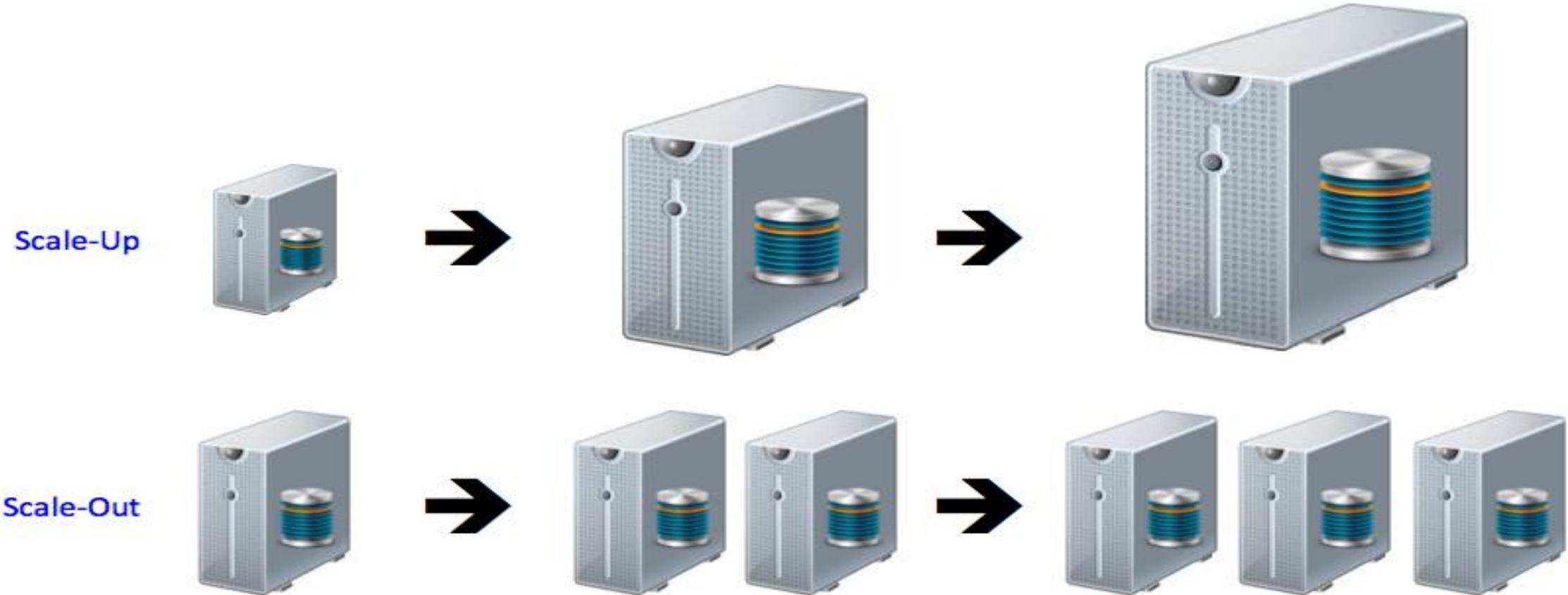
Using Lambda Authorizer:

<https://github.com/CloudSihmar/Lambda-Authorizer-API.git>

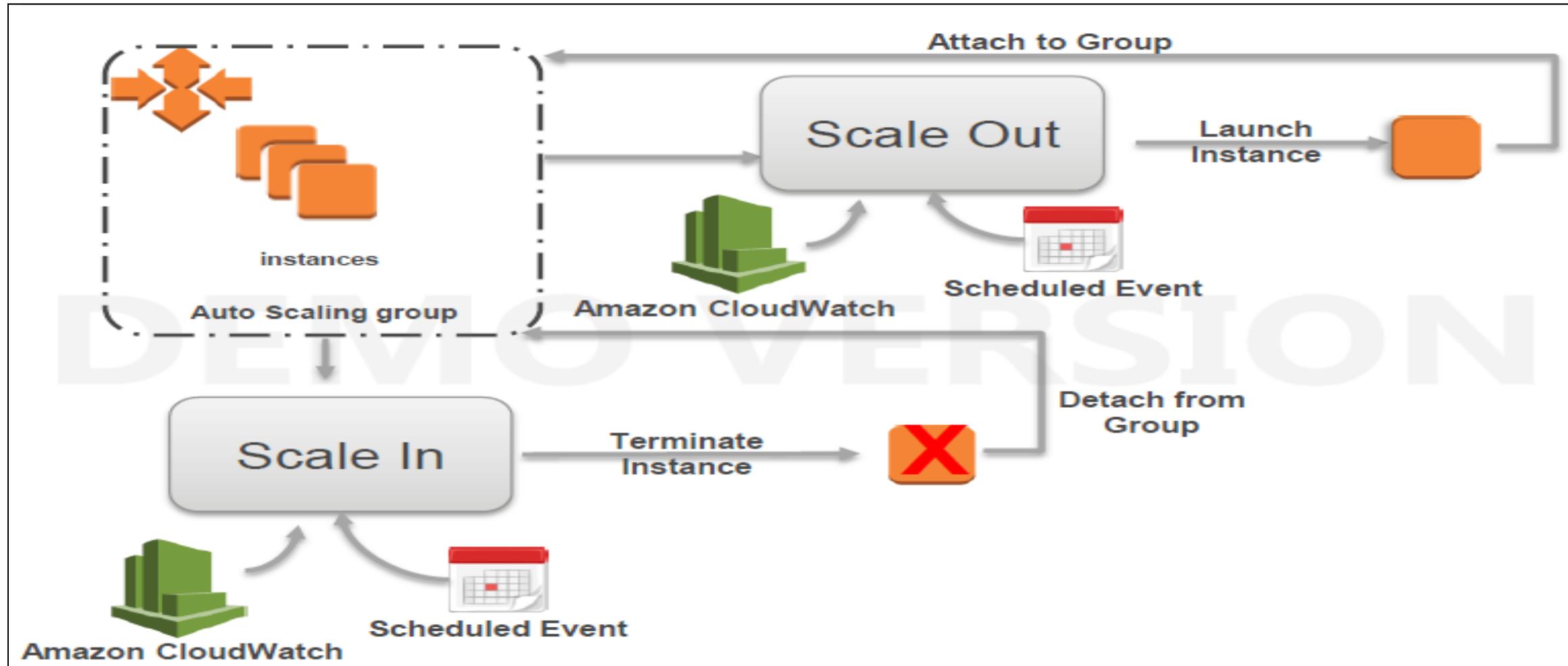


AWS AutoScaling

Auto Scaling



Auto Scaling



AWS VPC - Network Services

VPC

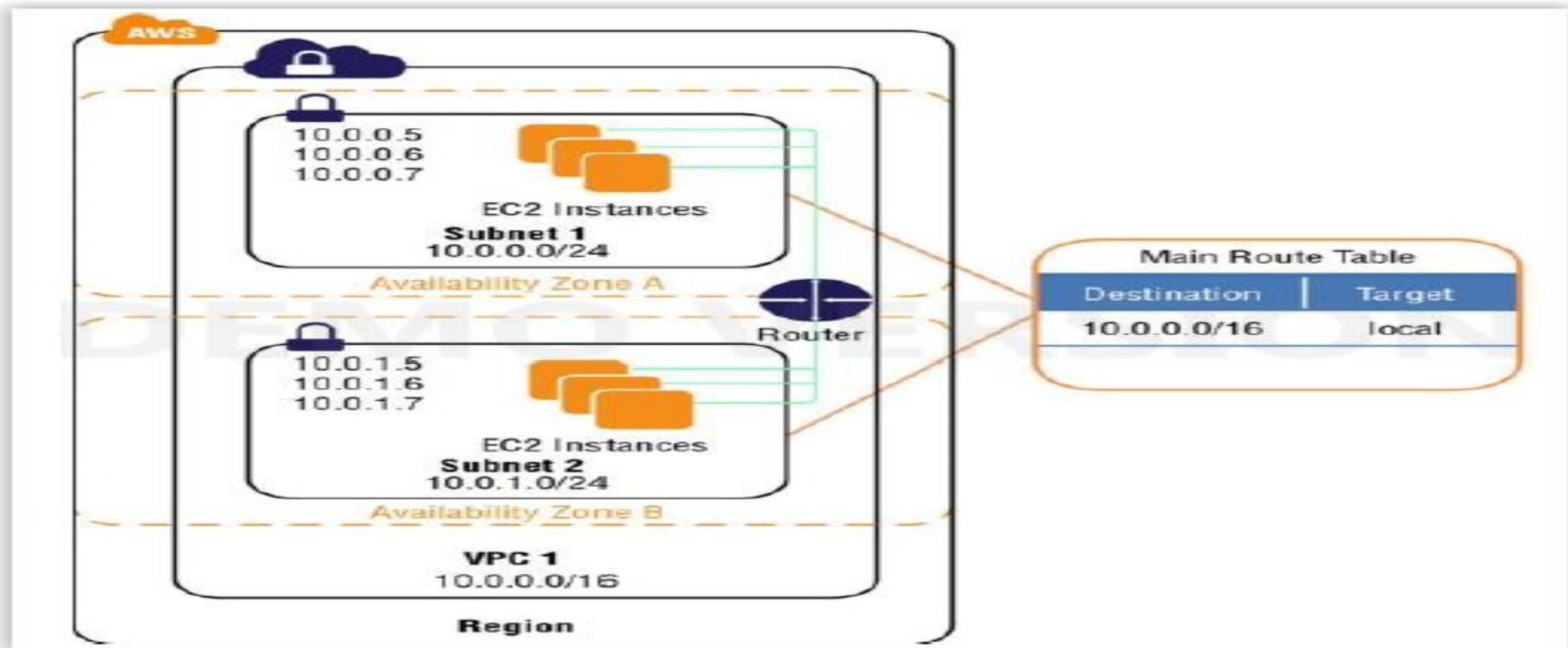
VPC stands for Virtual Private Cloud. Provision a **private, isolated virtual network** on the AWS cloud.

Amazon VPC is the networking layer for Amazon Elastic Compute Cloud (Amazon EC2), and it allows you to build your own virtual network within AWS.

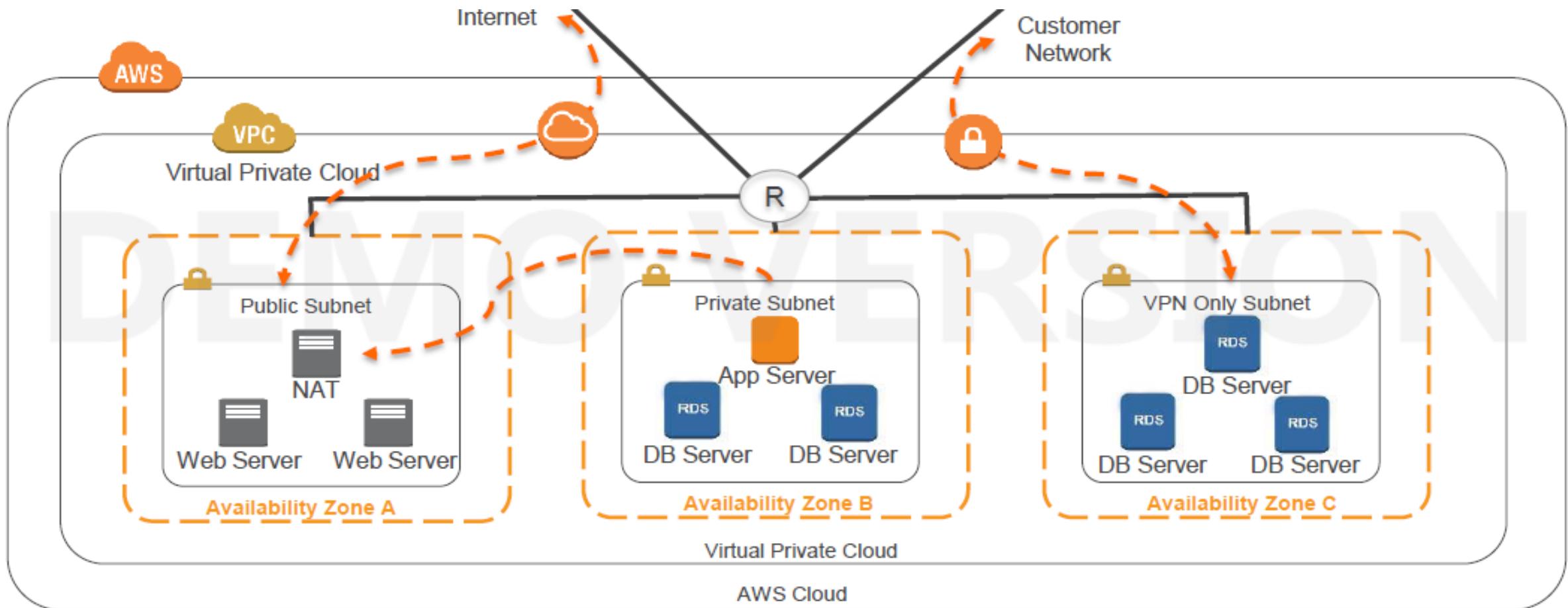
Within a region, you can create multiple Amazon VPCs, and each Amazon VPC is logically isolated.

Think of VPC as your Separate Datacenter with multiple VLANs (subnets can be treated like VLans here).

VPC



VPC/Subnets



Network Access Control List

A network access control list (ACL) is another layer of security that acts as a stateless firewall on a subnet level.

A network ACL is a numbered list of rules that AWS evaluates in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.

Operates at the subnet level (Another layer of defense).

NACL is default allow.

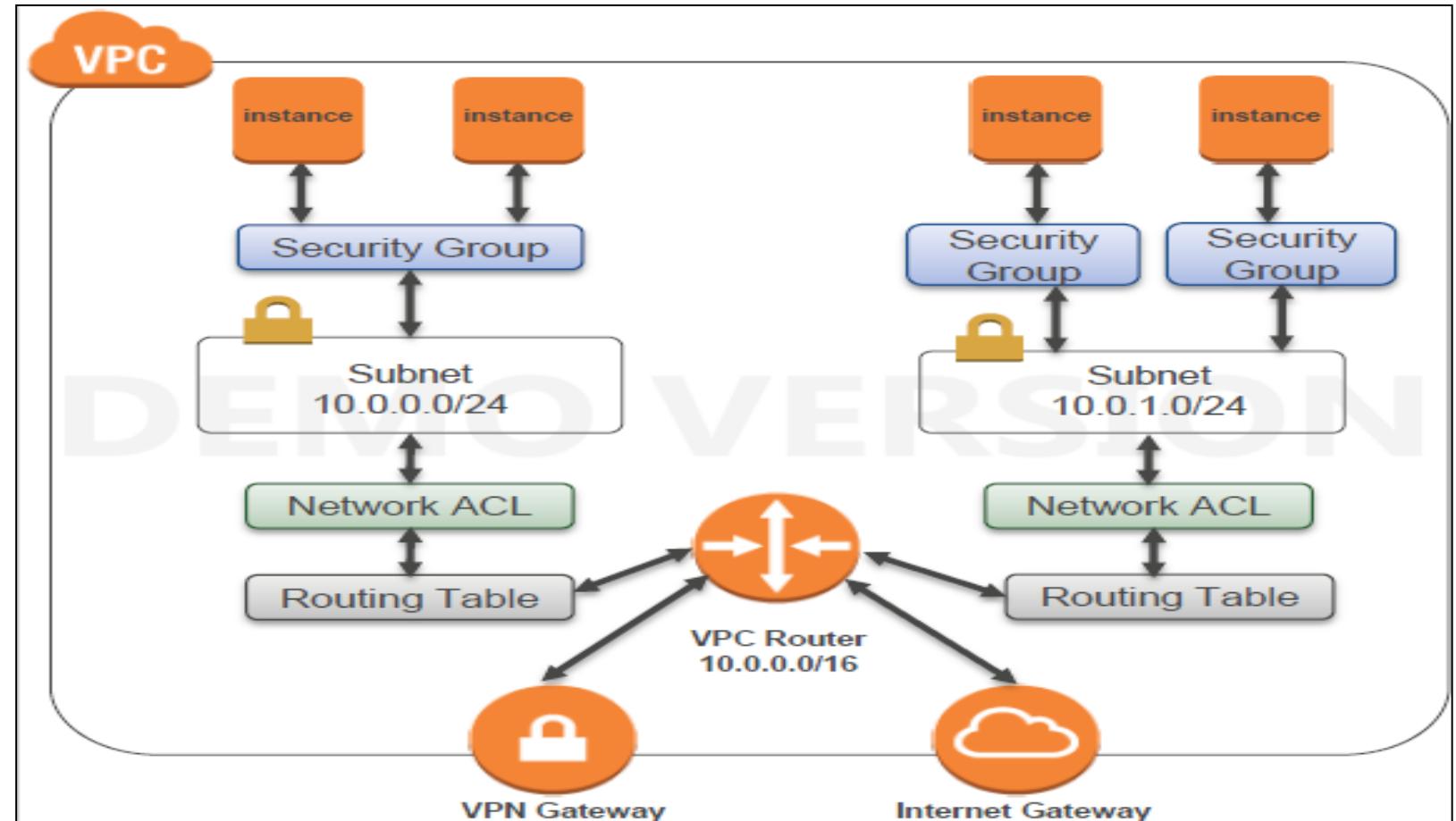
Security Layers

OS Firewall

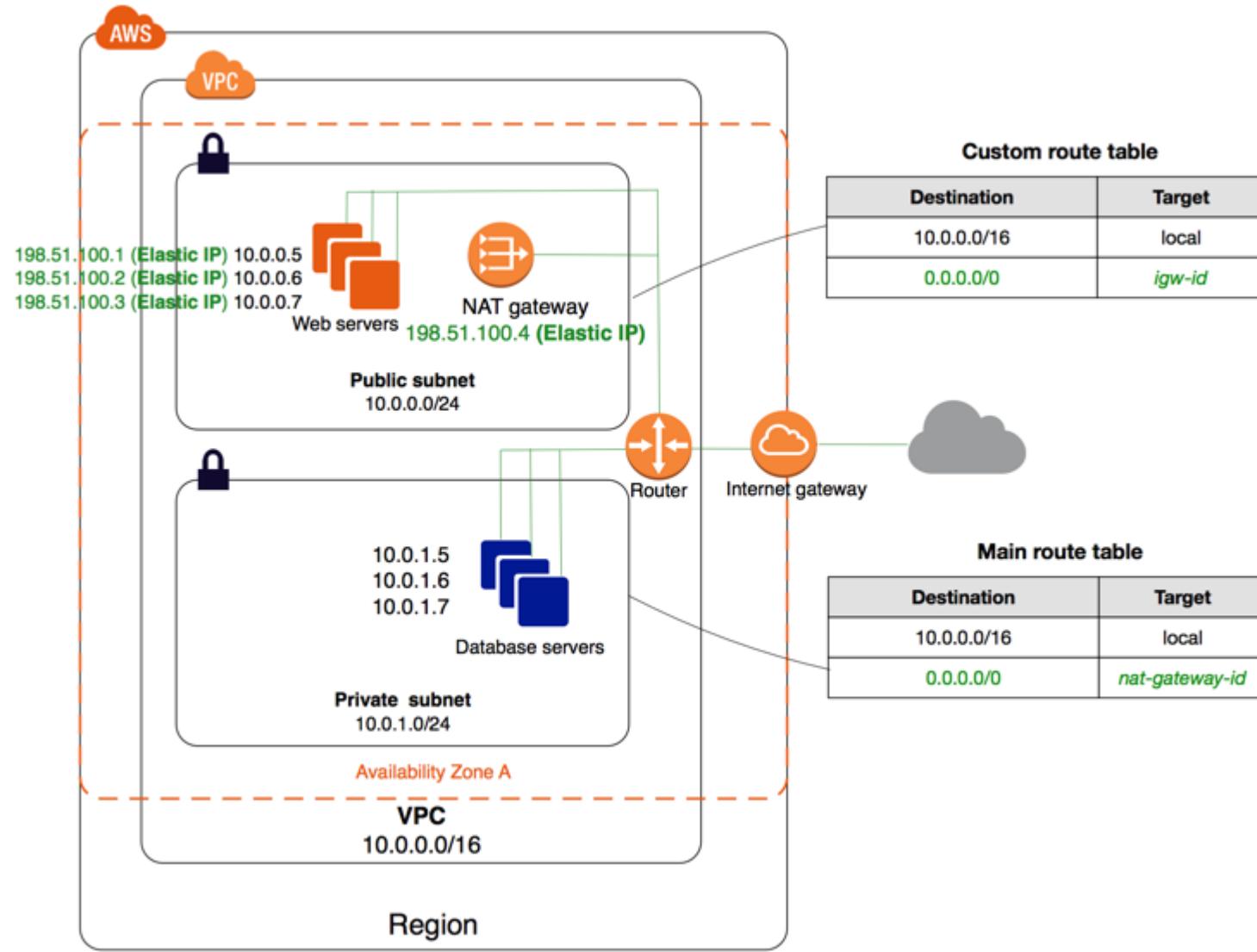
Security Group

Network ACL

Routing Tables



NAT Gateway



NAT Gateway

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

Each NAT gateway is created in a specific Availability Zone and implemented with **redundancy in that zone**.

You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply. Amazon EC2 charges for data transfer also apply.

NAT Gateway only support IPv4 IP Address.

NAT Gateway resides in a public subnet and support traffic for private subnets. We must also have to specify an Elastic IP address to associate with the NAT gateway when you create it.

After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. This enables instances in your private subnets to communicate with the internet.

NAT Gateway

If you have resources in multiple Availability Zones and they share one NAT gateway, and if the NAT gateway's Availability Zone is down, resources in the other Availability Zones lose internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps.

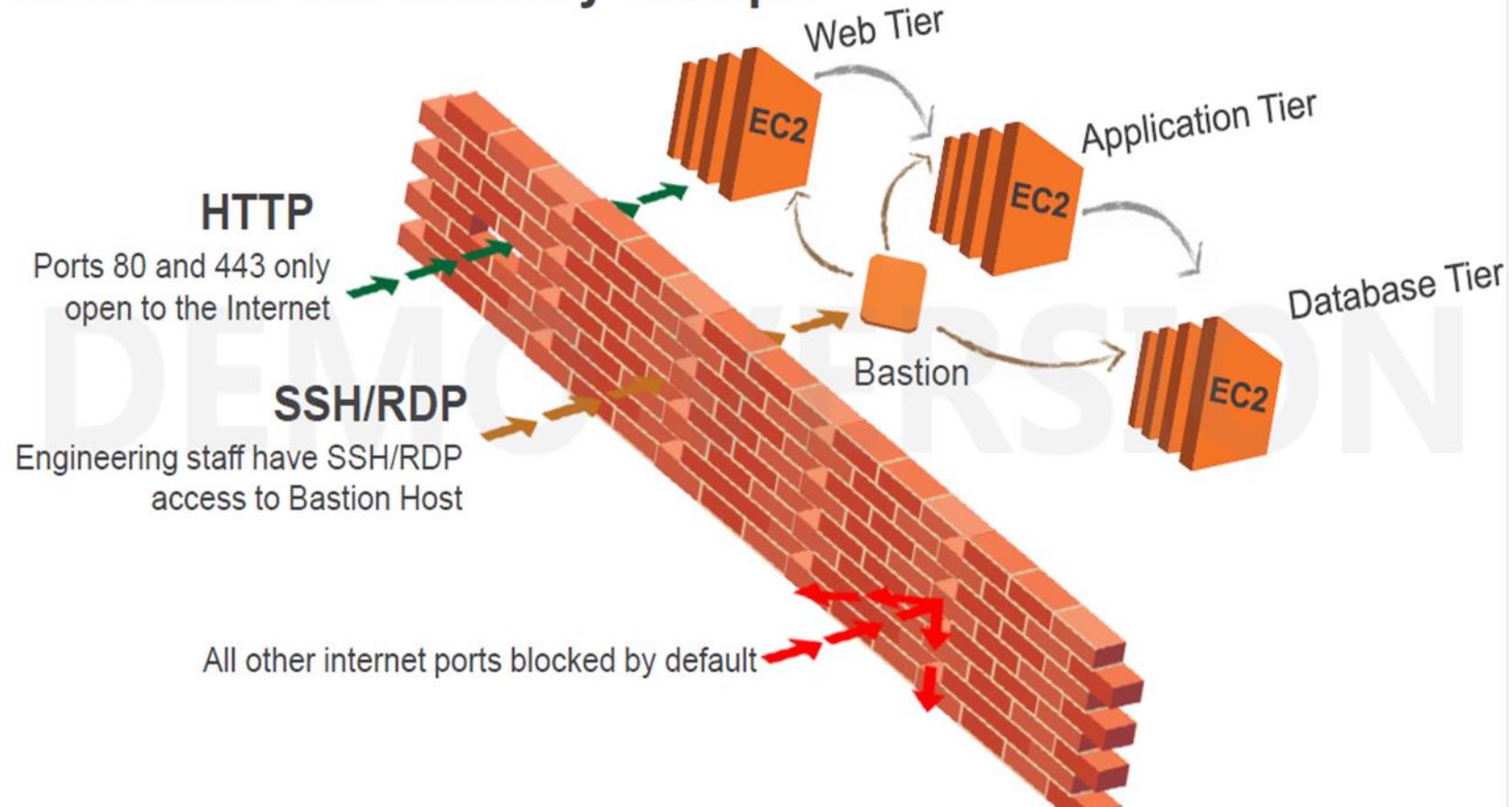
A NAT gateway supports the following protocols: TCP, UDP, and ICMP.

You cannot route traffic to a NAT gateway through a VPC peering connection, a Site-to-Site VPN connection, or AWS Direct Connect. A NAT gateway cannot be used by resources on the other side of these connections.

A NAT gateway can support up to 55,000 simultaneous connections to each unique destination. This limit also applies if you create approximately 900 connections per second to a single destination (about 55,000 connections per minute). This limit is because NAT gateway uses port range - 1024–65535, to connect to each destination IP.

Enhanced Security

AWS Multi-Tier Security Groups



Security Layers

SSL Endpoints	Security Groups	VPC
Secure Transmission Establish secure communication sessions (HTTPS) using SSL/TLS.	Instance Firewalls Configure firewall rules for instances using Security Groups.	Network Control In your Virtual Private Cloud, create low-level networking constraints for resource access. Public and private subnets, NAT and VPN support.

VPC Flow Logs

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs or Amazon S3. After you've created a flow log, you can retrieve and view its data in the chosen destination.

Flow logs can help you with a number of tasks, such as:

- Diagnosing overly restrictive security group rules
- Monitoring the traffic that is reaching your instance
- Determining the direction of the traffic to and from the network interfaces

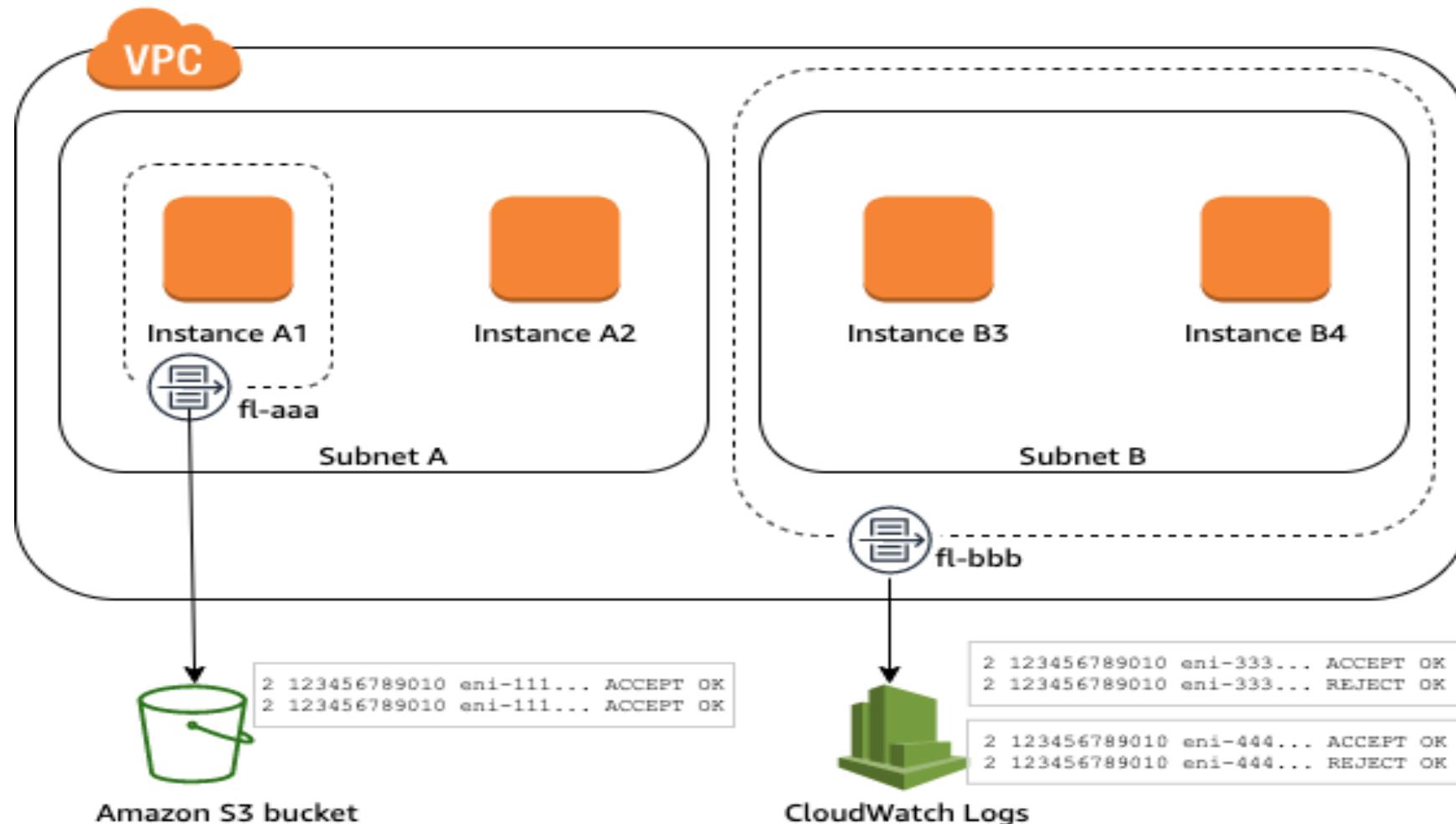
Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency. You can create or delete flow logs without any risk of impact to network performance.

Flow logs do not capture real-time log streams for your network interfaces.

By default, the record includes values for the different components of the IP flow, including the source, destination, and protocol.

When you create a flow log, you can use the default format for the flow log record, or you can specify a custom format.

VPC Flow Logs



VPC Flow Logs

Data ingestion and archival charges for vended logs apply when you publish flow logs to CloudWatch Logs or to Amazon S3.

Flow logs do not capture all IP traffic. The following types of traffic are not logged:

- Traffic generated by instances when they contact the Amazon DNS server. If you use your own DNS server, then all traffic to that DNS server is logged.
- Traffic generated by a Windows instance for Amazon Windows license activation.
- Traffic to and from 169.254.169.254 for instance metadata.
- Traffic to and from 169.254.169.123 for the Amazon Time Sync Service.
- DHCP traffic.
- Traffic to the reserved IP address for the default VPC router. For more information, see VPC and subnet sizing.
- Traffic between an endpoint network interface and a Network Load Balancer network interface.

Managed Prefix List

A managed prefix list is a set of one or more CIDR blocks. You can use prefix lists to make it easier to configure and maintain your security groups and route tables. You can create a prefix list from the IP addresses that you frequently use, and reference them as a set in security group rules and routes instead of referencing them individually.

There are two types of prefix lists:

Customer-managed prefix lists — Sets of IP address ranges that you define and manage. You can share your prefix list with other AWS accounts, enabling those accounts to reference the prefix list in their own resources.

AWS-managed prefix lists — Sets of IP address ranges for AWS services. You cannot create, modify, share, or delete an AWS-managed prefix list.

Managed Prefix List: Demo

1. Create an instance in a VPC
2. Create a Managed Prefix list and add the CIDR ranges into it or your Public IP address.
3. Add the prefix list in the Security Group for SSH connection.
4. Test the connection
5. Remove your IP address from prefix lists
6. Test the connection again

VPC Endpoint

VPC endpoints

A **VPC endpoint** enables you to **privately** connect your VPC to supported **AWS services** and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

Instances in your VPC do not require public IP addresses to communicate with resources in the service.

Traffic between your VPC and the other service does not leave the Amazon network.

VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. They allow communication between instances in your VPC and services without imposing availability risks.

VPC endpoints

AWS currently supports two types of Endpoints

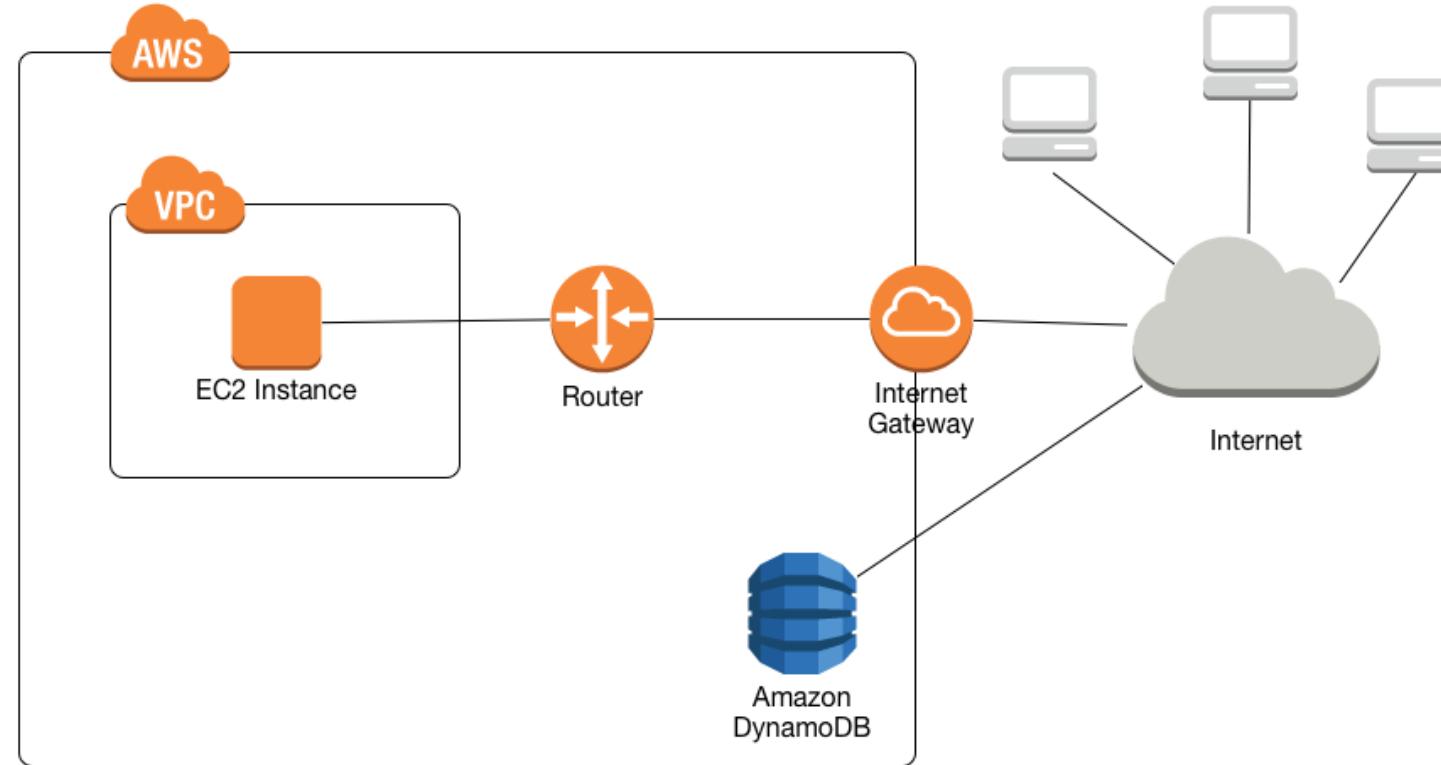
- VPC Interface Endpoints
- VPC Gateway Endpoints

VPC Endpoint policy is an IAM resource policy attached to an endpoint for controlling access from the endpoint to the specified service.. Endpoint policy, by default, allows full access to the service

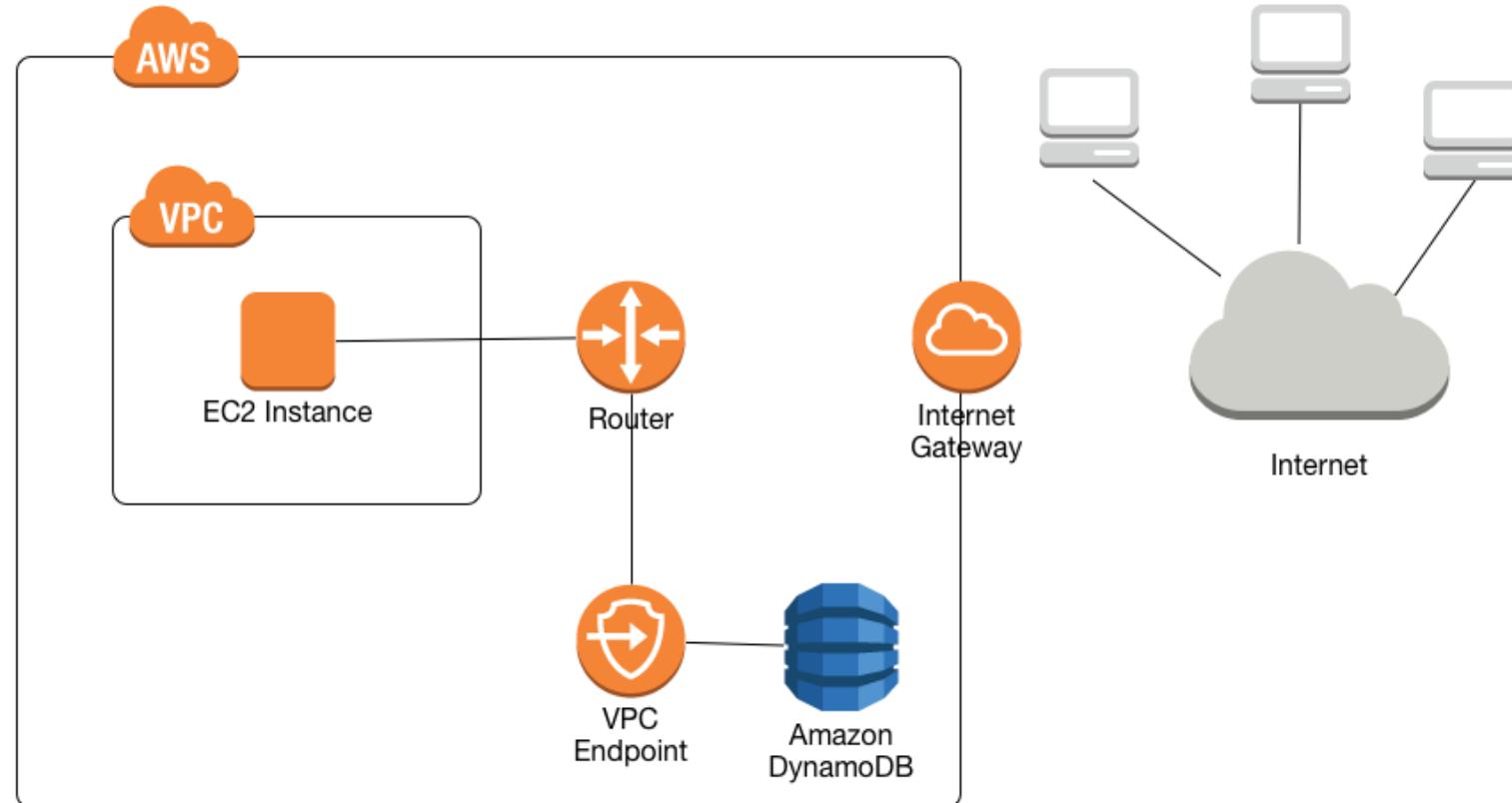
VPC Interface endpoints

- Interface endpoints enable connectivity to services over AWS PrivateLink.
- These services include some **AWS managed services**, services hosted by other **AWS customers** and **partners in their own Amazon VPCs** (referred to as endpoint services), and **supported AWS Marketplace partner services**.
- The owner of a service is a **service provider**. The principal creating the interface endpoint and using that service is a service consumer.
- An interface endpoint is a **collection of one or more elastic network interfaces** with a private IP address that serves as an entry point for traffic destined to a supported service.

VPC endpoints



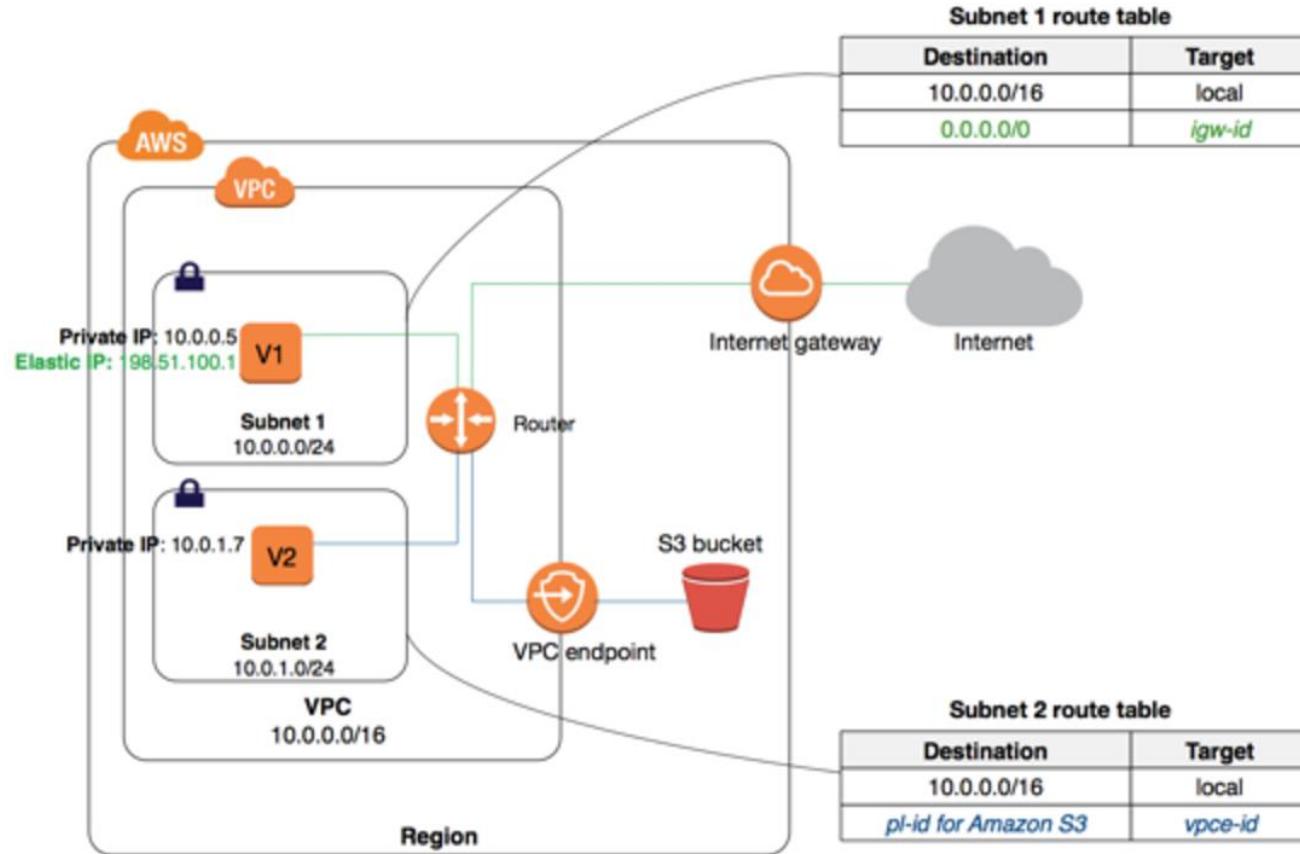
VPC endpoints



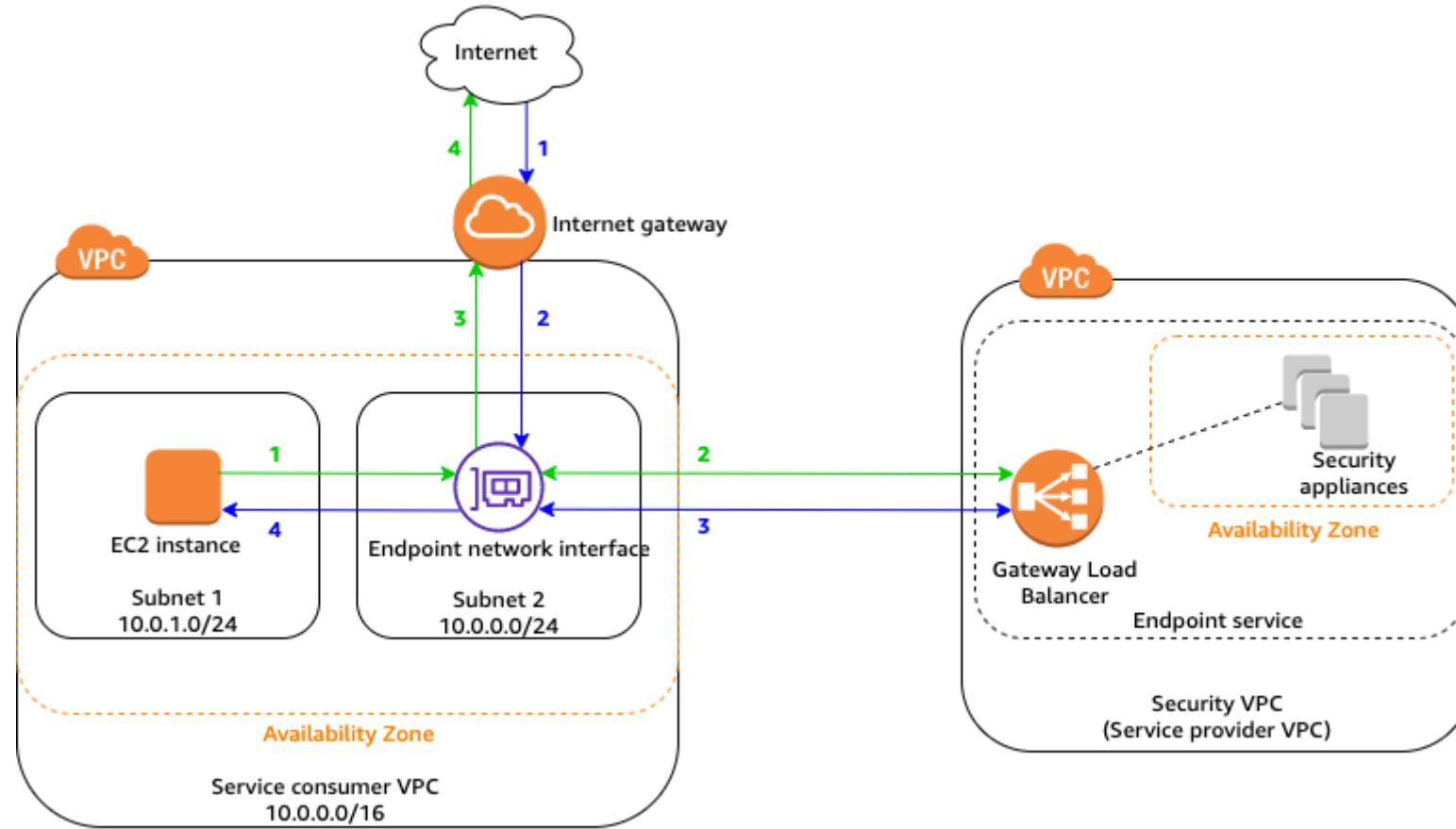
VPC Gateway endpoints

- A gateway endpoint targets specific IP routes in an Amazon VPC route table, in the form of a **prefix-list**, used for traffic destined to **Amazon DynamoDB** or **Amazon Simple Storage Service (Amazon S3)**.
Gateway endpoints do not enable AWS PrivateLink.
- Gateway endpoints are destinations that are reachable from within an Amazon VPC through prefix-lists within the Amazon VPC's route table.

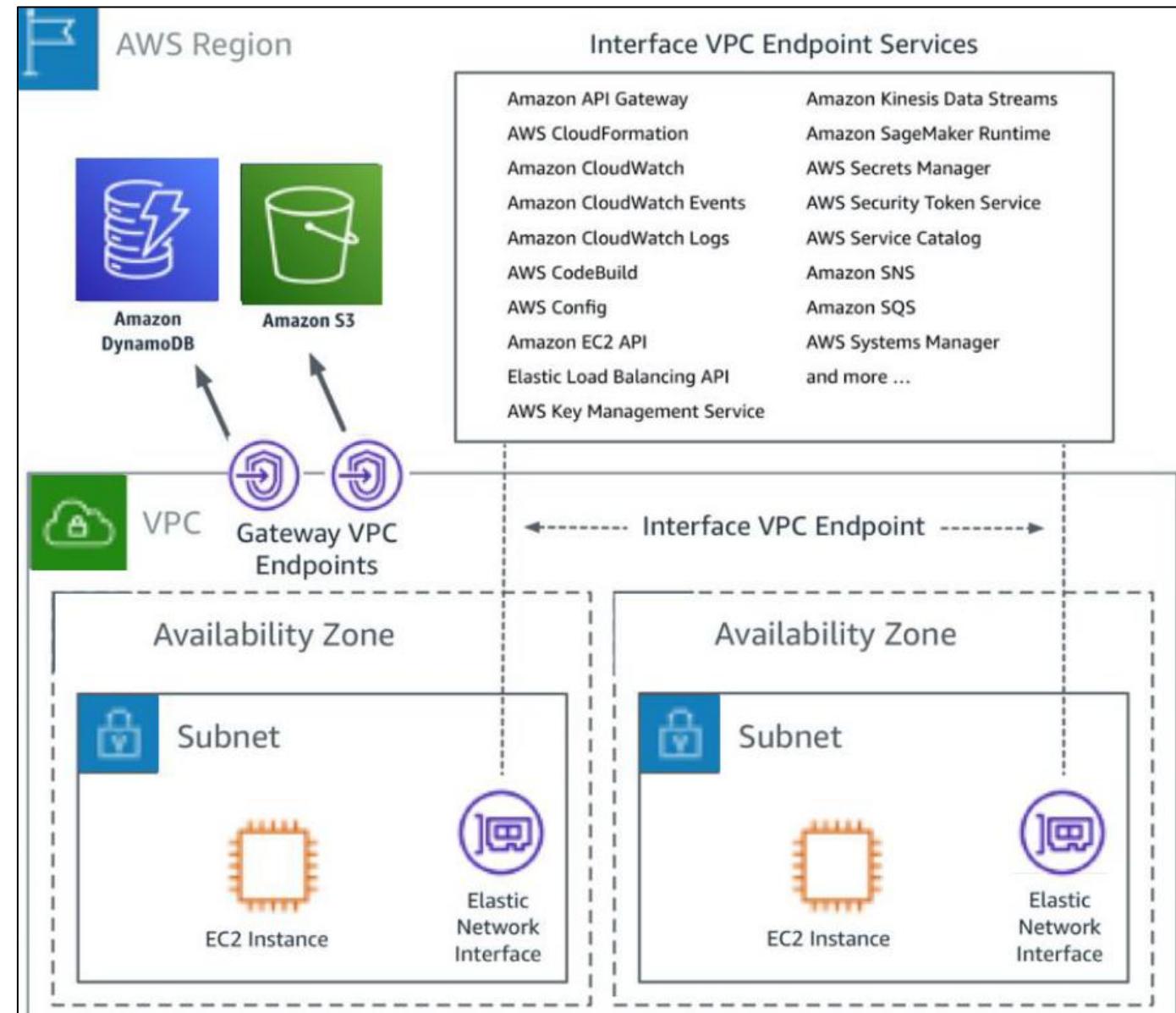
VPC Gateway Endpoints



VPC Interface Endpoints



VPC Endpoints



VPC Endpoints concept

VPC endpoint — The entry point in your VPC that enables you to connect privately to a service. The following are the different types of VPC endpoints. You create the type of VPC endpoint required by the supported service.

Gateway endpoint - You can use gateway load balancer endpoints to privately and securely inject in-line network and security services, such as firewalls, intrusion detection and prevention systems, monitoring, analytics and others, running outside your VPC into your traffic flow.

Interface endpoint - You can use interface endpoints to privately and securely access services like AWS services, internal application services or SaaS services that are running outside your VPC.

Gateway Load Balancer endpoint

Endpoint service — Your own application or service in your VPC. Other AWS principals can create an endpoint from their VPC to your endpoint service.

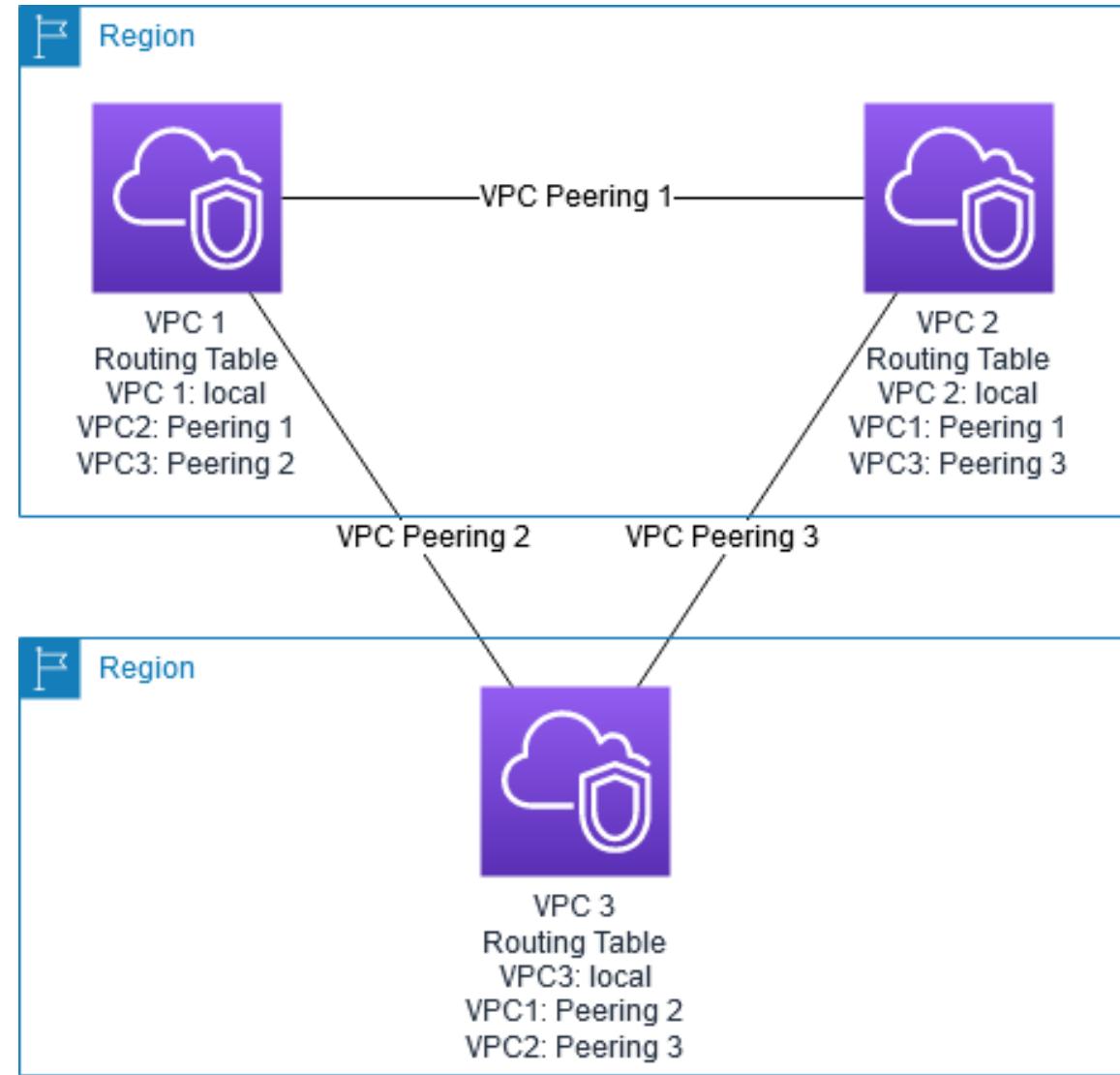
AWS PrivateLink — A technology that provides private connectivity between VPCs and services.

LAB: VPN Endpoint

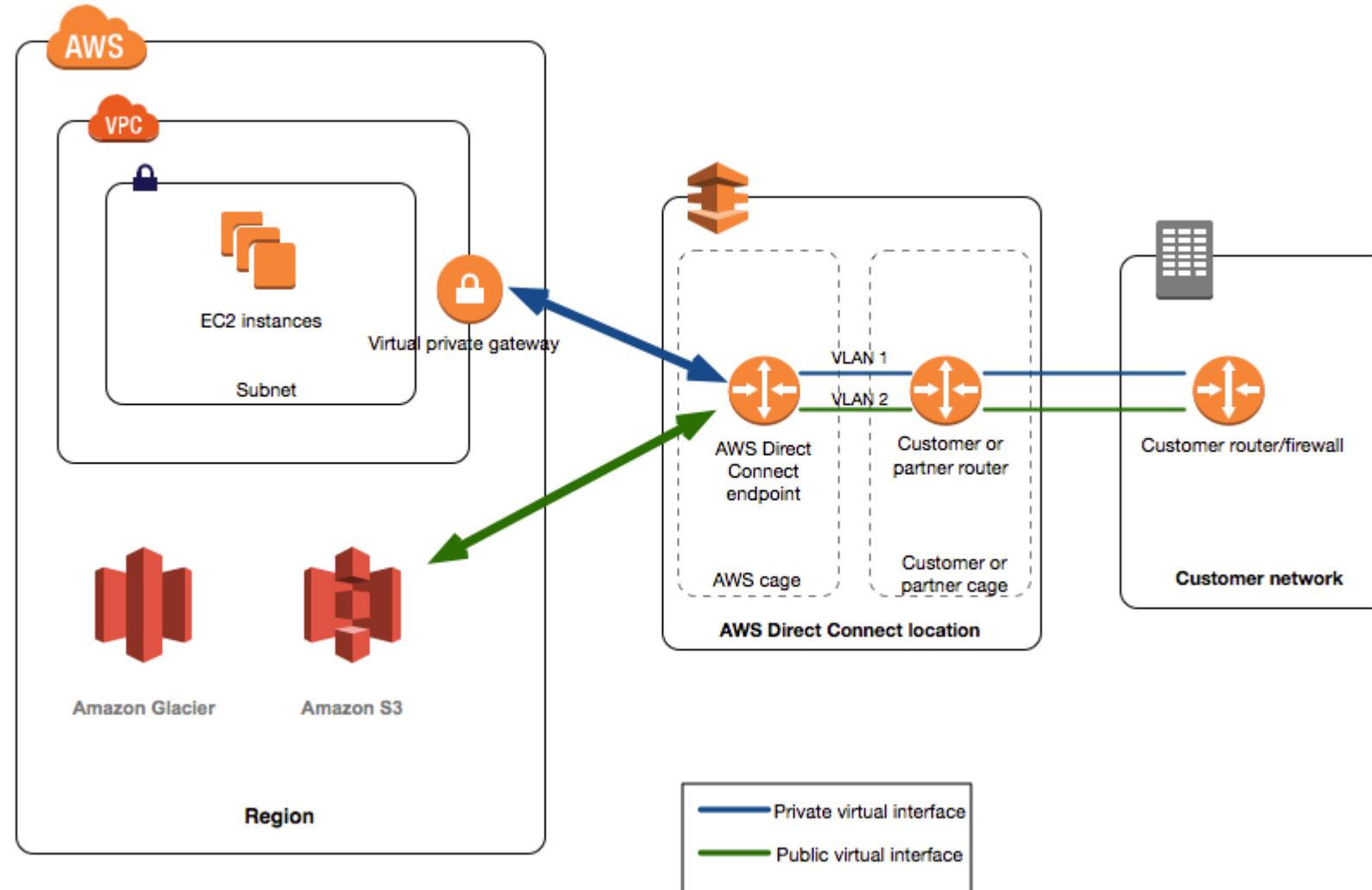
Create a VPN Endpoint for your network:

1. Go to VPC Service and select endpoints
2. Click on create endpoint
3. Select AWS Service and Service name DynamoDB (Observe type as Gateway)
4. Select your VPC and route tables
5. Assign policy as full policy
6. Assign Name tag under tags
7. Click on create
8. Observe Prefixes in managed Prefix list
9. Observe changes in route table entries
10. Login to a server in your VPC and run below commands:
 - traceroute -n -T -p 443 dynamodb.us-east-2.amazonaws.com
 - traceroute dynamodb.us-east-2.amazonaws.com
 - ping dynamodb.us-east-2.amazonaws.com
11. Delete Endpoint and run above commands to observe the difference.

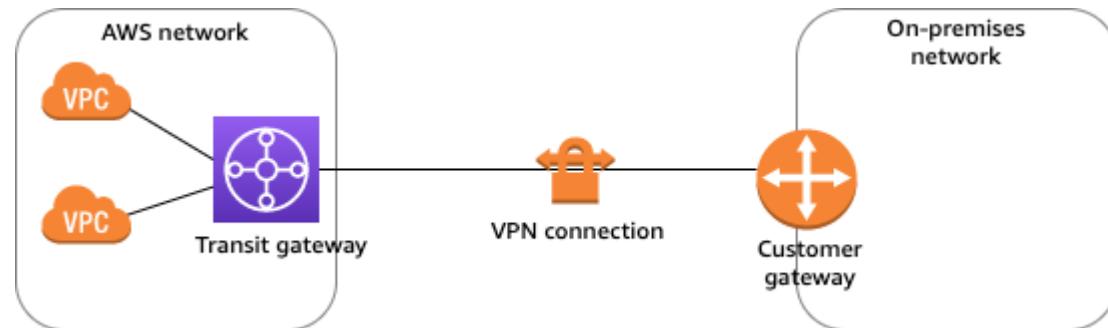
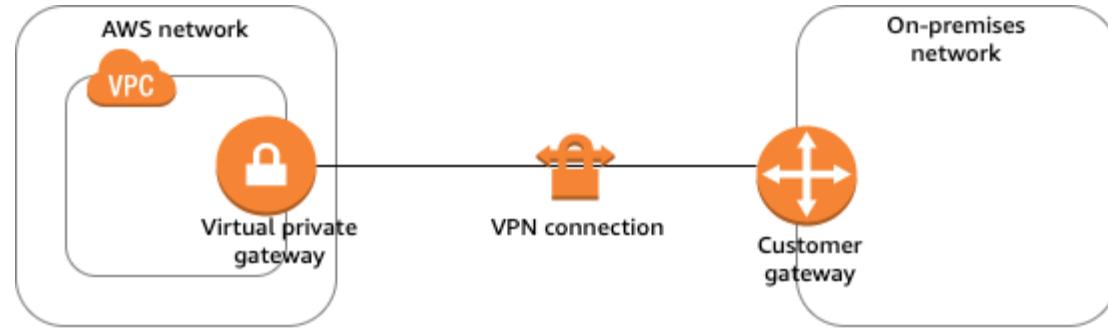
VPC Peering



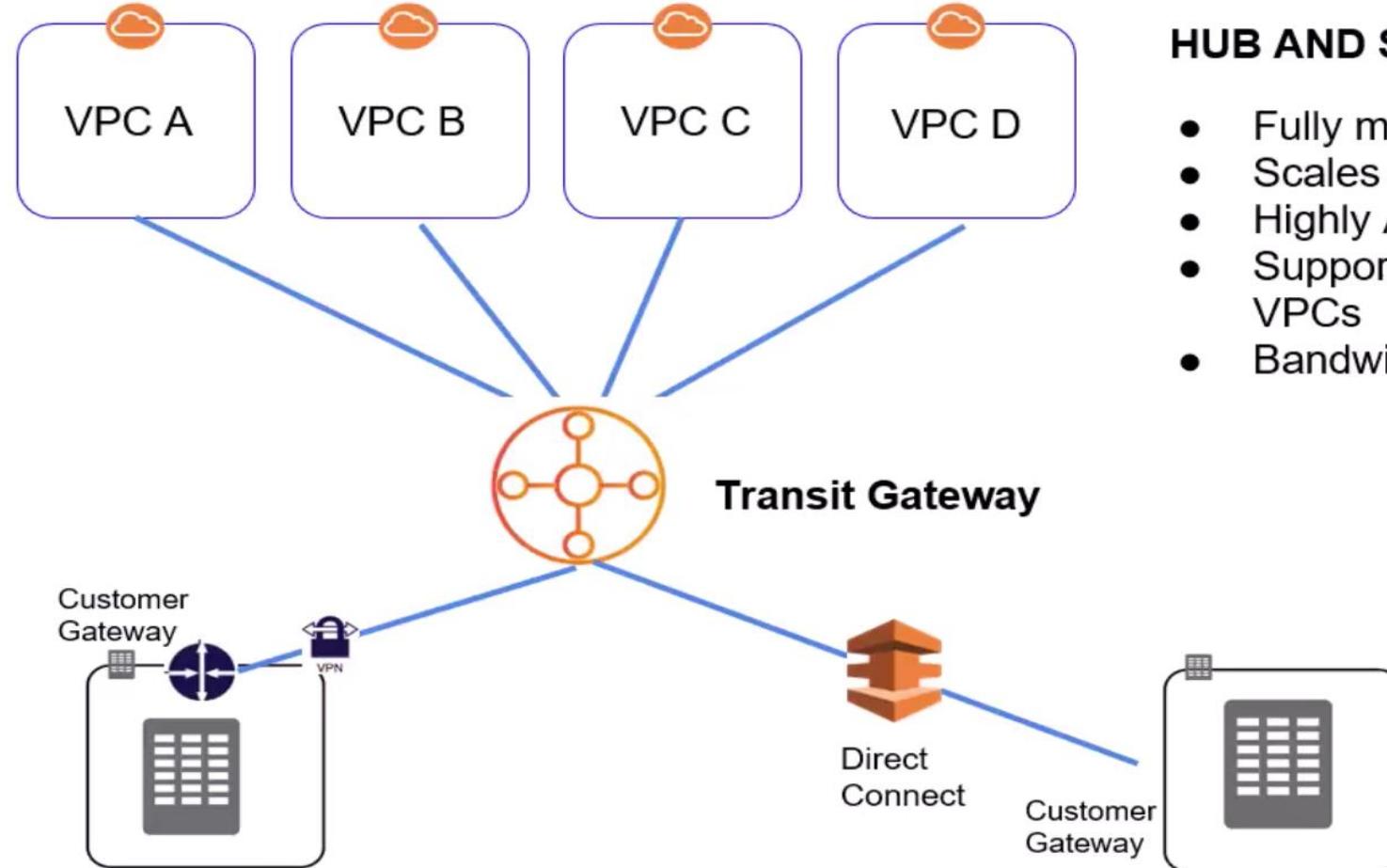
AWS Direct Connect



AWS VPN Connect



AWS Transit Gateway



HUB AND SPOKE TOPOLOGY

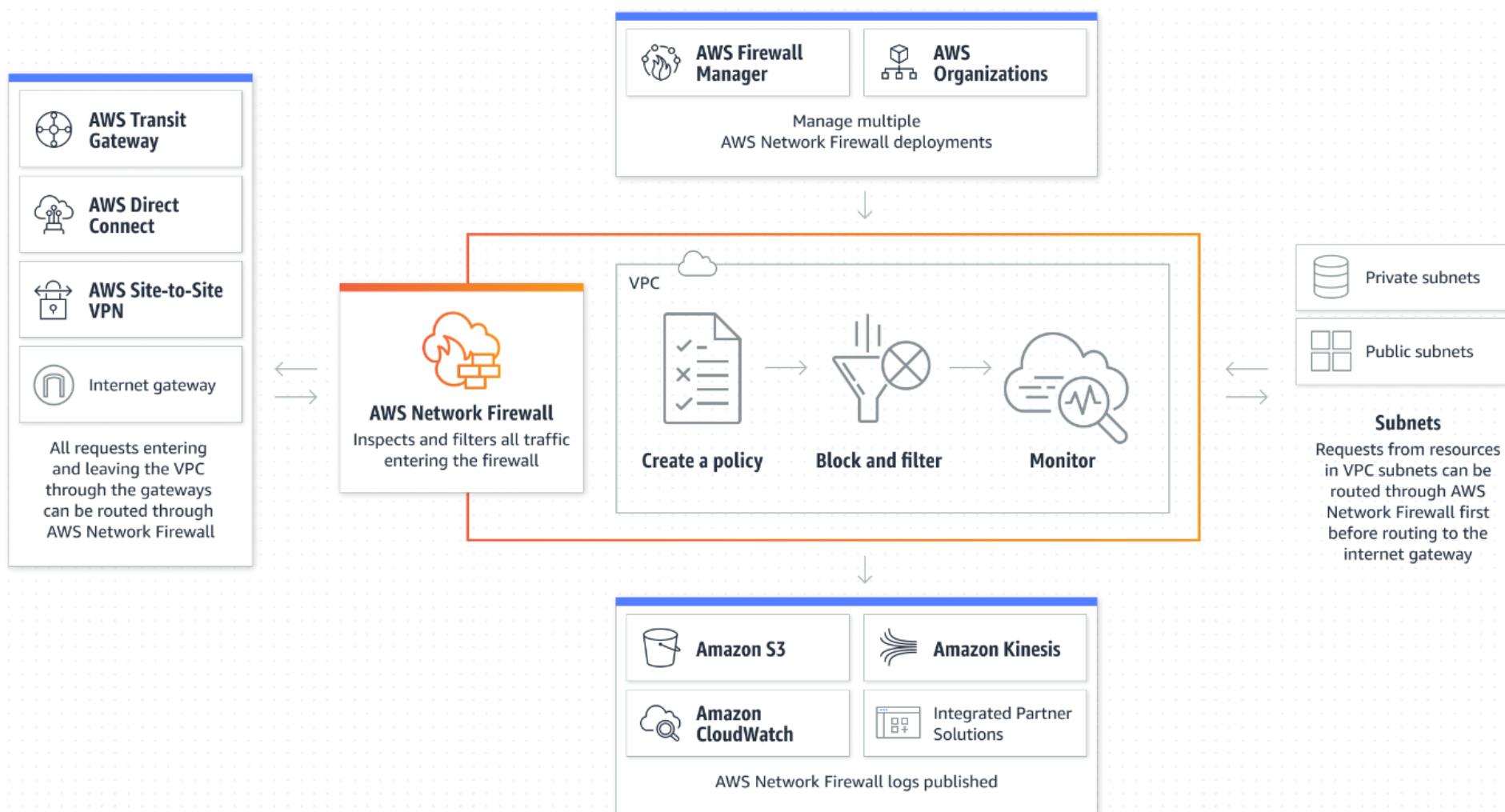
- Fully managed gateway
- Scales automatically
- Highly Available
- Supports attaching upto 5000 VPCs
- Bandwidth upto 50 gbps

AWS Network Firewall

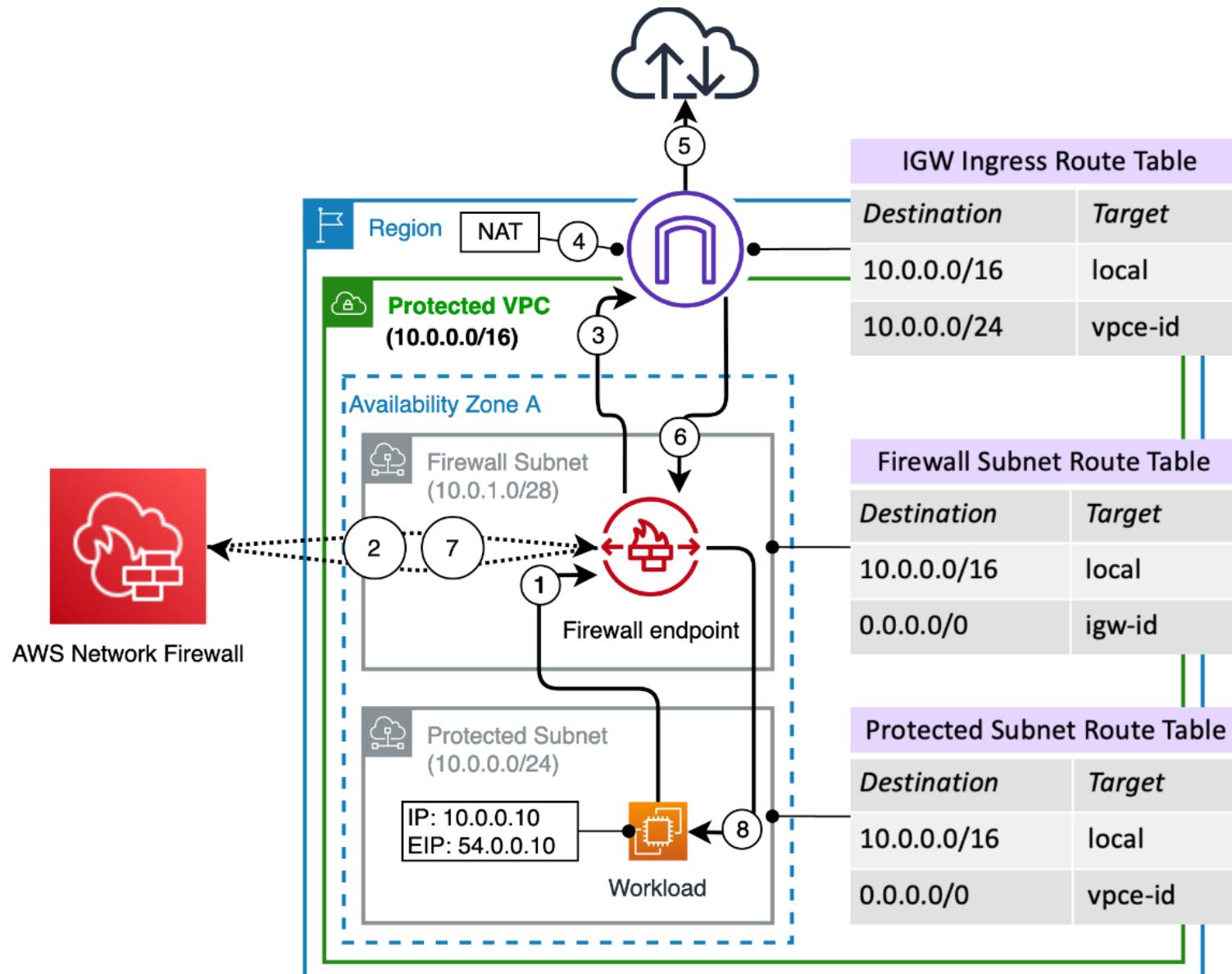
Network Firewall

- Scalable, highly available, managed network Firewall and intrusion detection/prevention service
- Applies blanket protection for your entire VPC
- Any application Type or Protocol
- What all it inspects
 - Traffic coming from Internet to VPC
 - Traffic outgoing to Internet from VPC
 - VPC to VPC Traffic
 - Traffic coming through AWS Direct Connect & VPN as well
- It uses the open-source intrusion prevention system(IPS), Suricata for Stateful inspection.
- It supports Suricata compatible rules

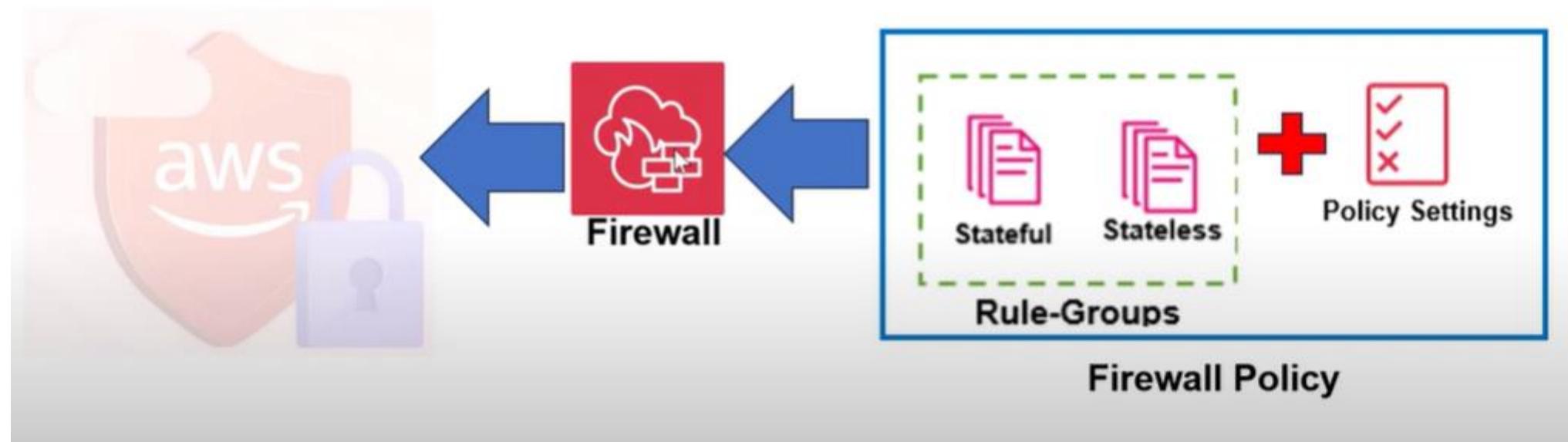
Network Firewall Architecture



Network Firewall Architecture



Network Firewall Components



Rule Groups

- Stateless Rule Group
 - Stateless rule groups evaluate packets in isolation
 - Define standard network connection attributes for examining a packet on its own, with no additional context
- Stateful Rule Group:
 - Stateful rule groups evaluate them in the context of their traffic flow
 - Network Firewall uses a Suricata rules engine to process all stateful rules
 - You can write any of your stateful rules in Suricata compatible format
 - Alternatively, for domain list rules and for very basic rules, you can use an easy entry form provided by network firewall.

Stateless Rule Group Action

- Pass
- Drop
- Forward to stateful rules

Stateful Rule Group Action

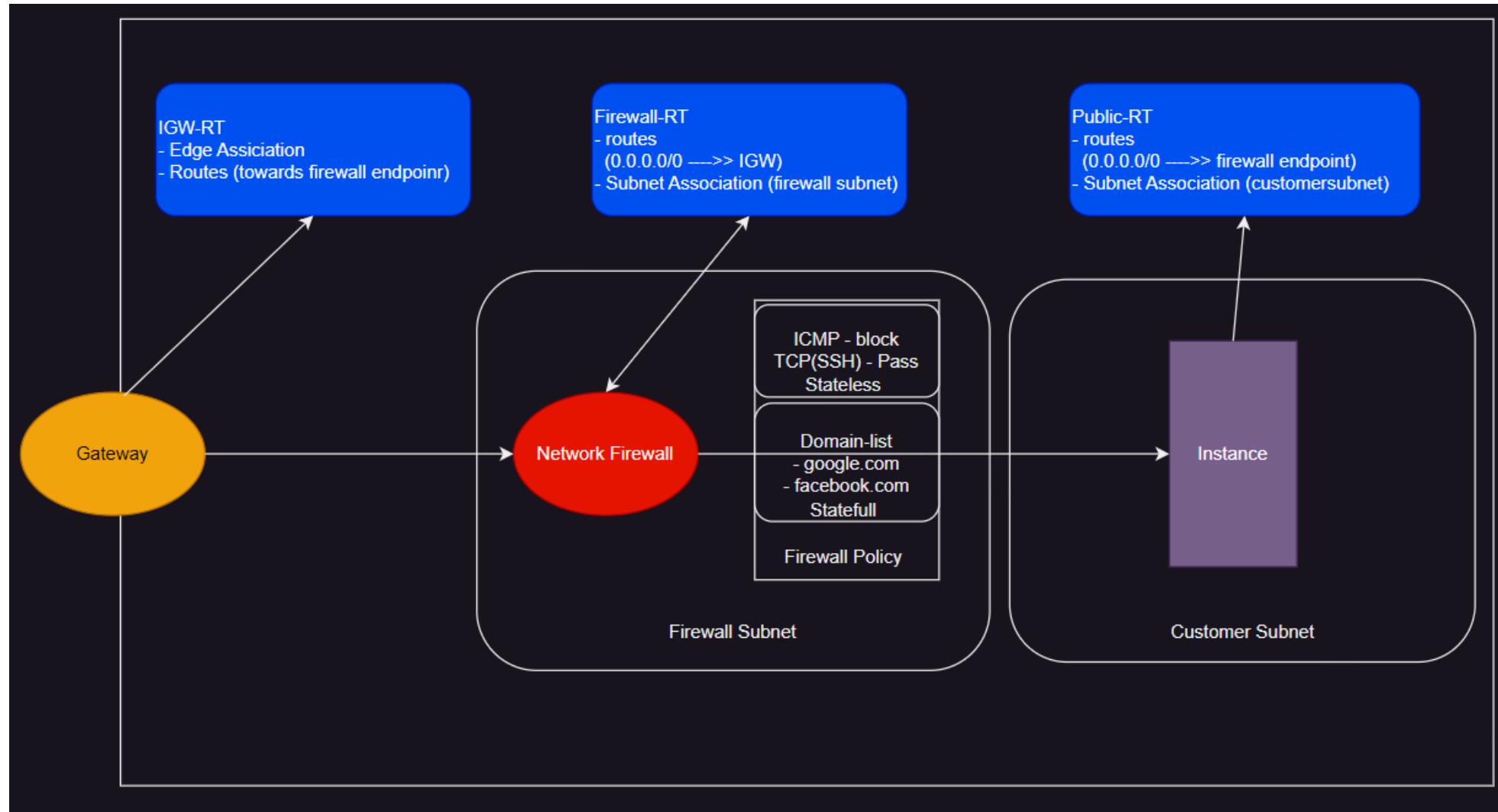
The option for stateful action settings vary by the rule type

- Standard rules and Suricata compatible settings
 - Pass
 - Drop or Alert
 - Reject (only for TCP traffic)
- Domain Lists
 - Allow
 - Deny

Demo

- Create a VPC (20.0.0.0/16)
- Create a public subnet (20.0.2.0/24)
- Create a Security Group to allow RDP, HTTP/HTTPS, ICMP traffic
- Create an Internet Gateway and attach to the VPC
- Create a route table for the subnet to send the traffic to internet via Internet Gateway
- Create Firewall Subnet (20.0.4.0/28)
- Create Stateful Rule Group (domain list)- to block traffic to few sites
- Create Stateless Rule Group for ICMP and RDP Traffic
- Change the routing for traffic to move through the firewall subnet

Demo



AWS Route 53

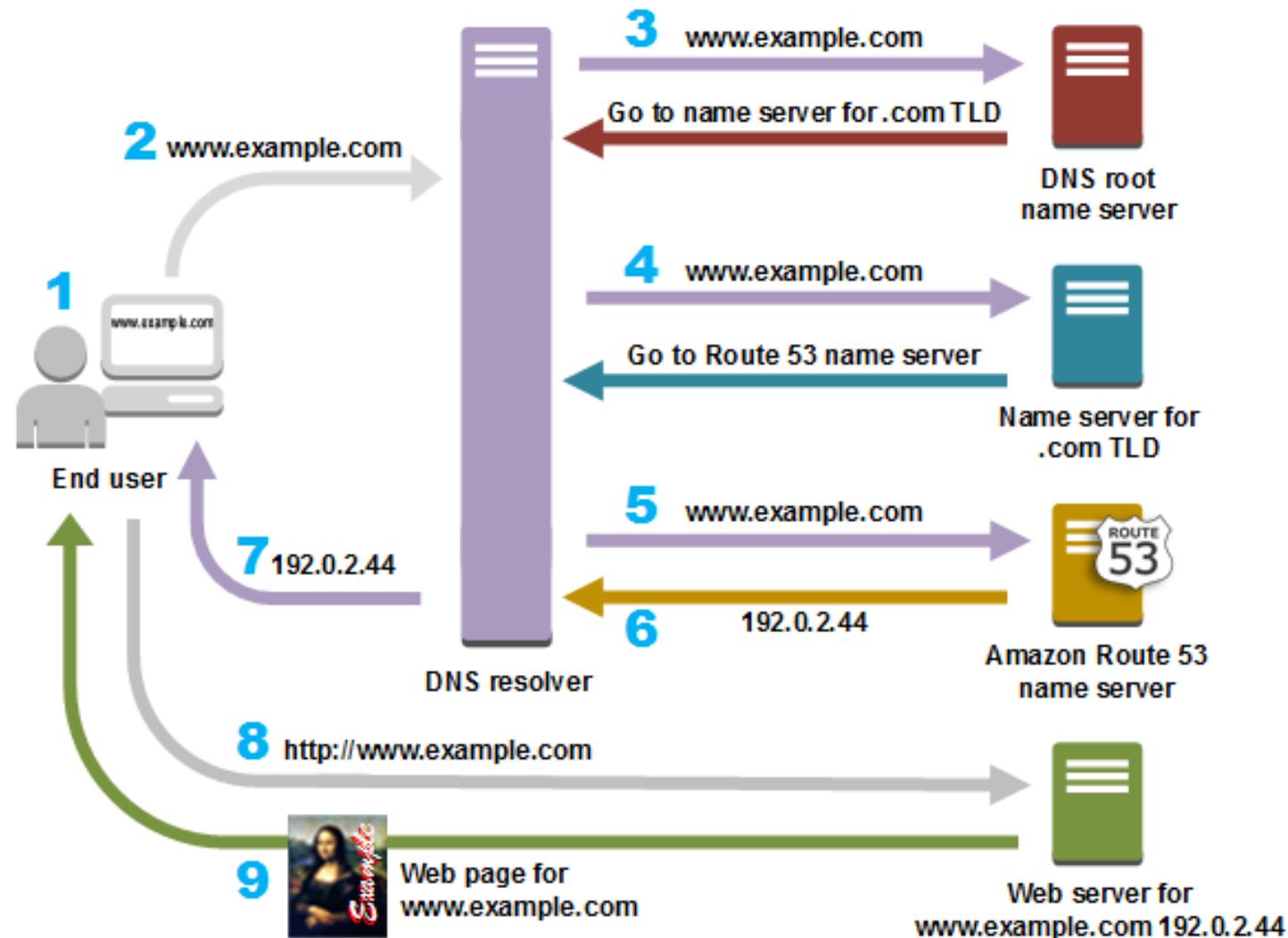
Route 53

- Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service.
- Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS.
- Amazon Route 53 is fully compliant with IPv6 as well.
- You can use Amazon Route 53 to configure DNS health checks to route traffic to healthy endpoints or to independently monitor the health of your application and its endpoints.
- Amazon Route 53 Traffic Flow makes it easy for you to manage traffic globally through a variety of routing types, including Latency Based Routing, Geo DNS, Geoproximity, and Weighted Round Robin—all of which can be combined with DNS Failover in order to enable a variety of low-latency, fault-tolerant architectures.
- You can easily manage how your end-users are routed to your application's endpoints—whether in a single AWS region or distributed around the globe.
- Amazon Route 53 also offers Domain Name Registration

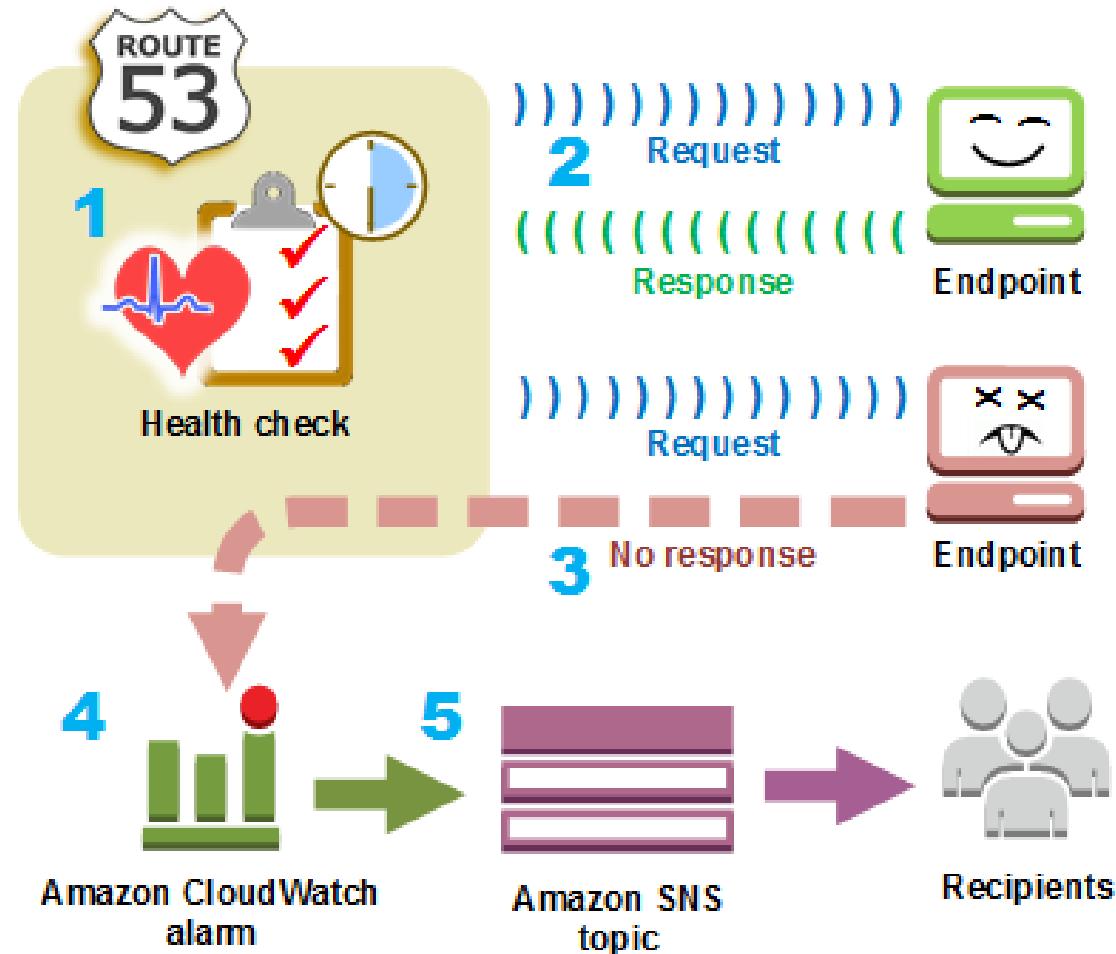
Key features

- Resolver
- Traffic flow
- Latency based routing
- Geo DNS
- Private DNS for Amazon VPC
- DNS Failover
- Health Checks and Monitoring
- Amazon Route 53 can monitor the health and performance of your application as well as your web servers and
- Domain Registration
- Existing Resource Integration

DNS Routing



Health Check



AWS Cognito

AWS Cognito

We want to give our users an identity so that they can interact with our application.

Cognito User Pools:

- Sign in functionality for app users
- Integrate with API Gateway & Application Load Balancer

Cognito Identity Pools (Federated Identity):

- Provide AWS credentials to users so they can access AWS resources directly
- Integrate with Cognito User Pools as an identity provider

Cognito Sync:

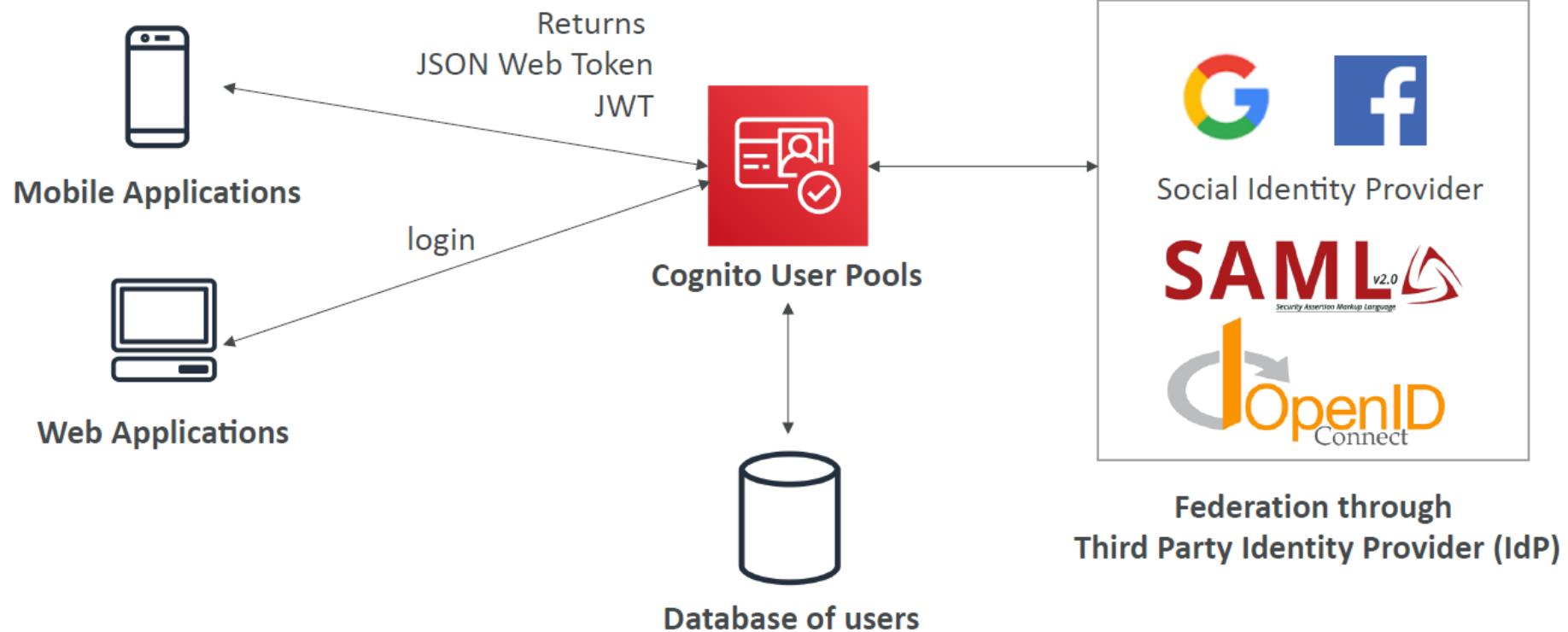
- Synchronize data from device to Cognito.
- Is deprecated and replaced by AppSync
- Cognito vs IAM: “hundreds of users”, “mobile users”, “authenticate with SAML”

AWS Cognito

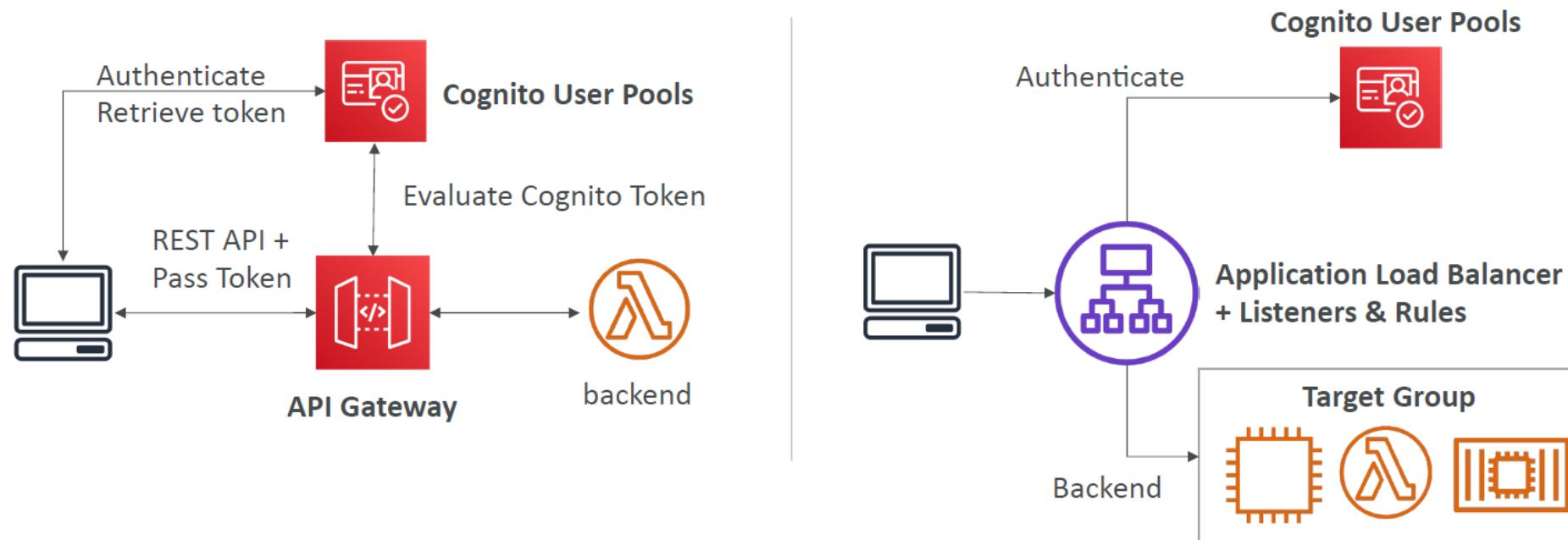
Cognito User Pools (CUP) – User Features

- Create a serverless database of user for your web & mobile apps
- Simple login: Username (or email) / password combination
- Password reset
- Email & Phone Number Verification
- Multi-factor authentication (MFA)
- Federated Identities: users from Facebook, Google, SAML...
- Feature: block users if their credentials are compromised elsewhere
- Login sends back a JSON Web Token (JWT)

AWS Cognito Usage



User Pool Integrations



UserPool Lambda Trigger

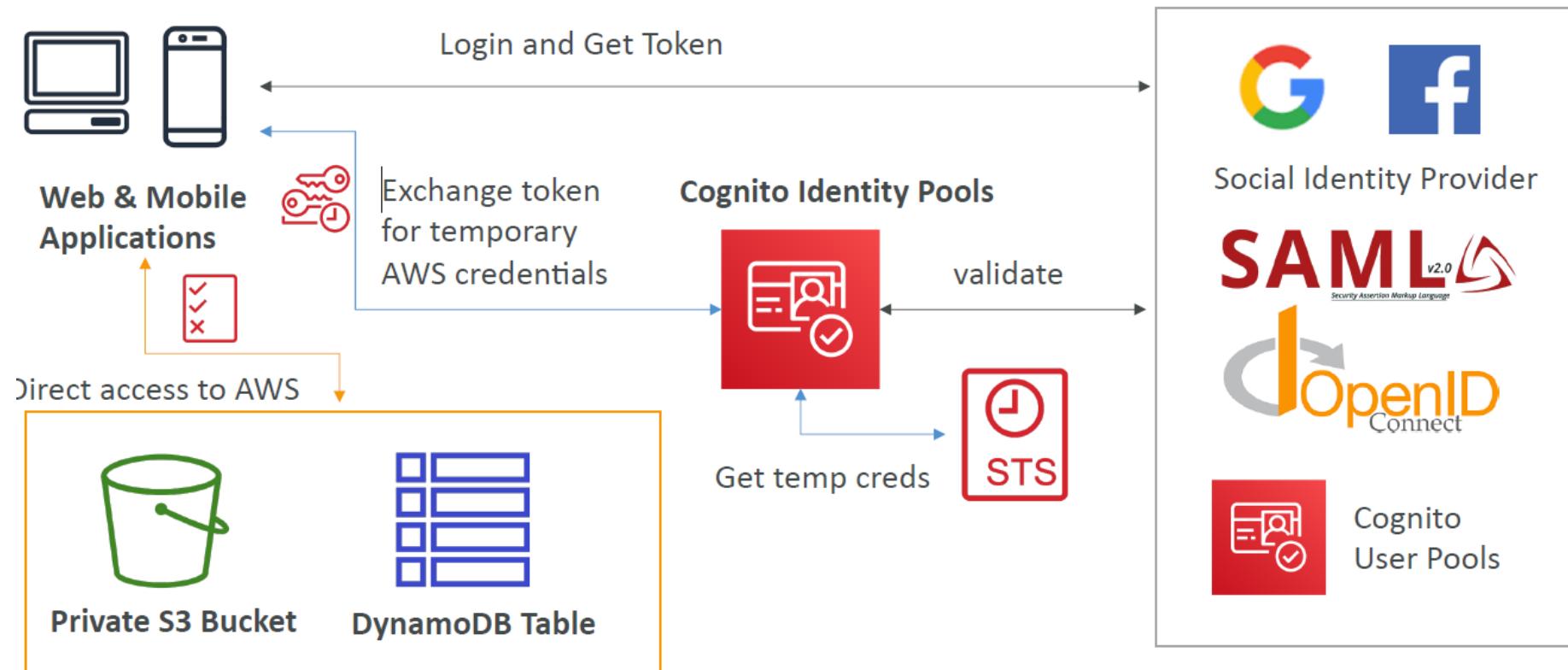
User Pool Flow	Operation	Description
Authentication Events	Pre Authentication Lambda Trigger	Custom validation to accept or deny the sign-in request
	Post Authentication Lambda Trigger	Event logging for custom analytics
	Pre Token Generation Lambda Trigger	Augment or suppress token claims
Sign-Up	Pre Sign-up Lambda Trigger	Custom validation to accept or deny the sign-up request
	Post Confirmation Lambda Trigger	Custom welcome messages or event logging for custom analytics
	Migrate User Lambda Trigger	Migrate a user from an existing user directory to user pools
Messages	Custom Message Lambda Trigger	Advanced customization and localization of messages
Token Creation	Pre Token Generation Lambda Trigger	Add or remove attributes in Id tokens

Cognito Identity Pool

Get identities for “users” so they obtain temporary AWS credentials

- Your identity pool (e.g identity source) can include:
- Public Providers (Login with Amazon, Facebook, Google, Apple)
- Users in an Amazon Cognito user pool
- OpenID Connect Providers & SAML Identity Providers
- Developer Authenticated Identities (custom login server)
- Cognito Identity Pools allow for unauthenticated (guest) access
- Users can then access AWS services directly or through API Gateway
- The IAM policies applied to the credentials are defined in Cognito
- They can be customized based on the user_id for fine grained control

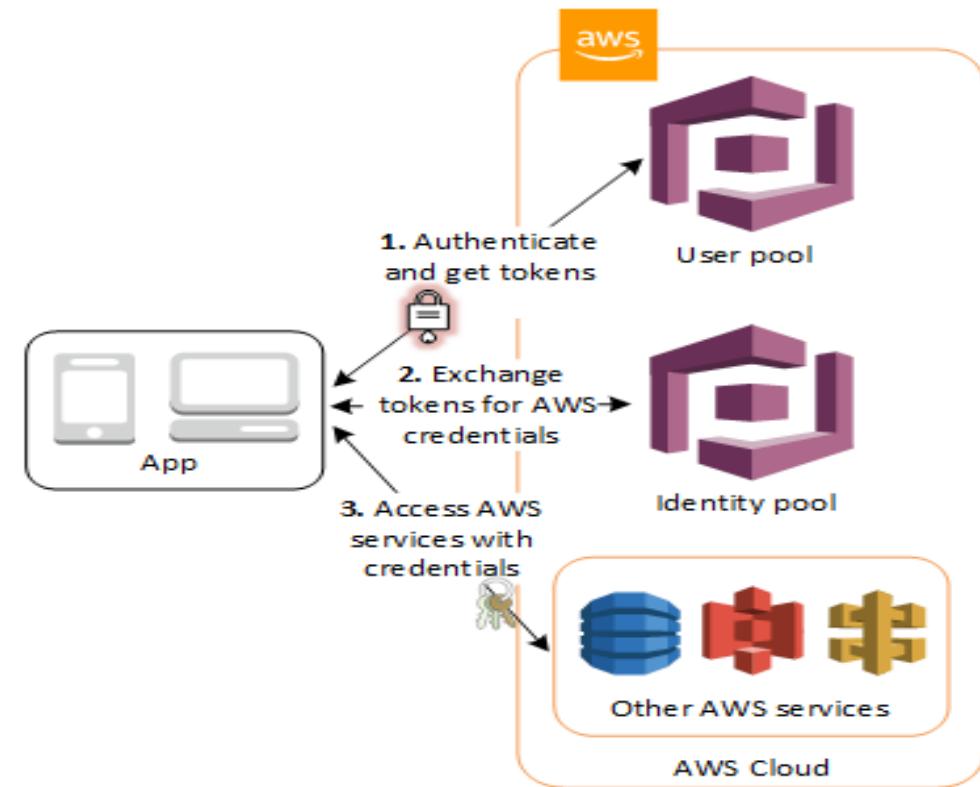
Identity Pool



Cognito

Amazon Cognito provides authentication, authorization, and user management for your web and mobile apps. Your users can sign in directly with a user name and password, or through a third party such as Facebook, Amazon, Google or Apple.

The two main components of Amazon Cognito are **user pools** and **identity pools**. User pools are user directories that provide sign-up and sign-in options for your app users. Identity pools enable you to grant your users access to other AWS services. You can use identity pools and user pools separately or together.



AWS Monitoring Services

Cloud Watch

A monitoring service for AWS cloud resources and the applications you run on AWS

Visibility into resource utilization, operational performance, and overall demand patterns

Custom application-specific metrics of your own

Accessible via AWS Management Console, APIs, SDK, or CLI

Monitor other AWS resources

View graphics and statistics

Set Alarms

Performance Insights

Amazon RDS Performance Insights monitors your Amazon RDS DB instance load so that you can analyze and troubleshoot your database performance.

It provides you a database performance dashboard where you can quickly assess performance on relational database workloads.

If you have more than one database on the DB instance, performance data for all of the databases is aggregated for the DB instance.

You can gain insight of your database with “Performance Insight” on below area:

Top SQL queries causing load on DB

Average Active session

hosts

users

AWS Auditing

Cloud Trail

A monitoring service for AWS Account Activity

You can use Cloud Trail to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure.

It logs all activities made through anywhere i.e AWS Management Console, AWS Command Line Interface, AWS SDKs and APIs.

By default Maintains data for last 90 days. You can download the trail data every three months to have backup.

You can Create a trail to retain a record of specific events (As longer as you want)

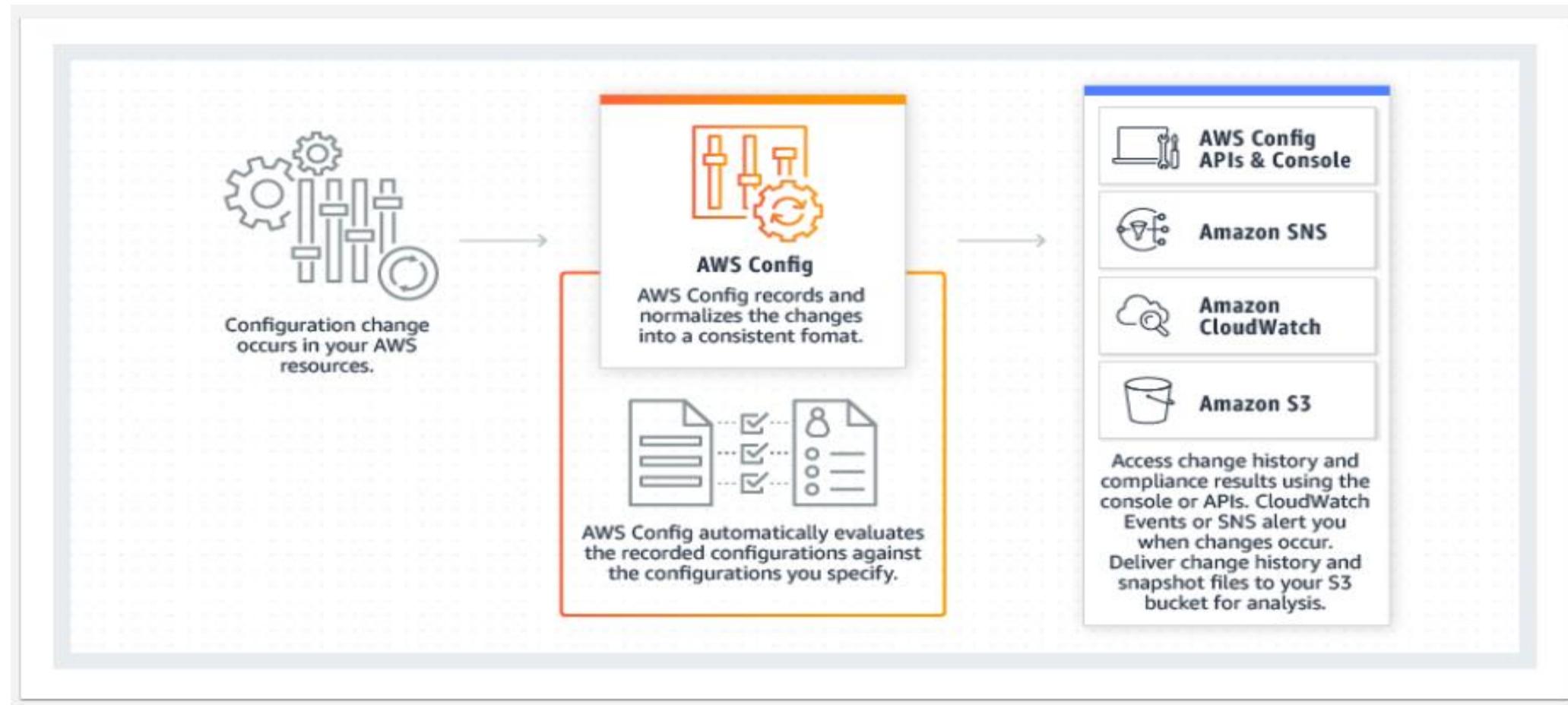
Default 90 days logging is free, creating trail will be chargeable.

AWS Config: Continuous audit and compliance

AWS Config

- AWS Config provides a detailed view of the configuration changes of AWS resources in your AWS account.
- You'll be able to continuously monitor and record configuration changes of your AWS resources
- AWS Config allows you to continuously audit and assess the overall compliance of your AWS resource configurations with your organization's policies and guidelines.
- AWS Config provides a detailed view of the resources associated with your AWS account, including
 - how they are configured,
 - how they are related to one another,
 - and how the configurations and their relationships have changed over time.
- Use AWS Config for:
 - Resource Administration
 - Auditing and Compliance
 - Security Analysis

AWS Config



AWS Advisory Services

Trusted Advisor

- An online resource to help you reduce cost, increase performance, and improve security by optimizing your AWS environment.
- Trusted Advisor provides real time guidance to help you provision your resources following AWS best practices.



Cost Optimization



Performance



Security



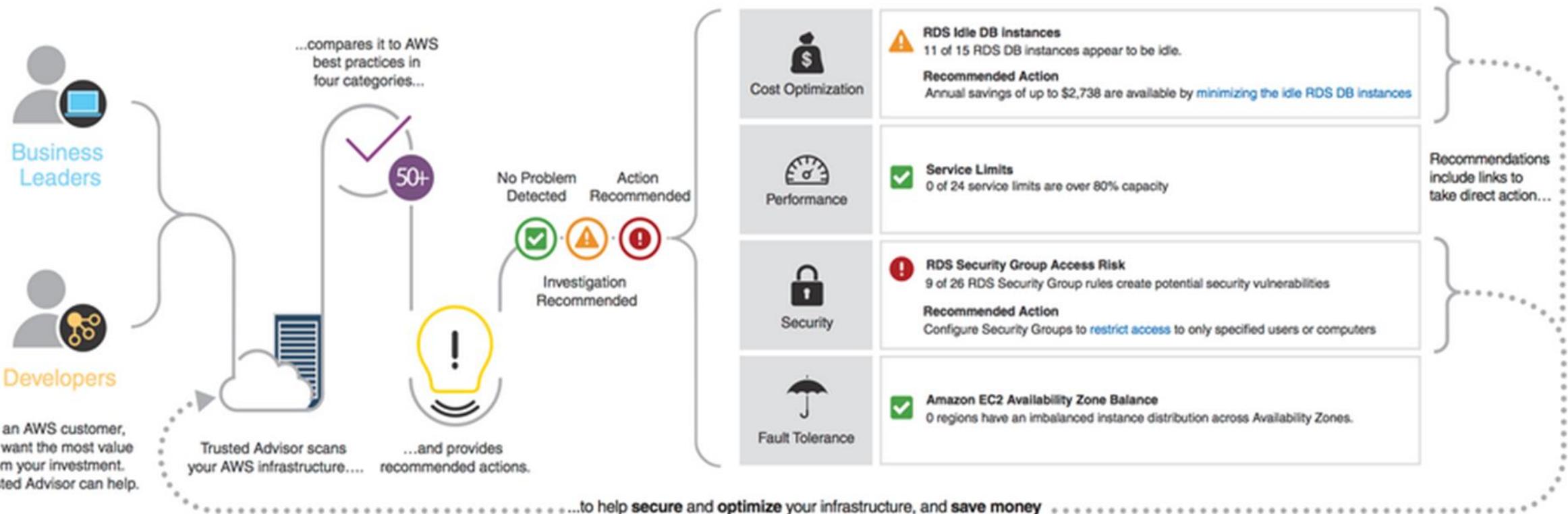
Fault Tolerance



Service Limits

Trusted Advisor

An Introduction to AWS Trusted Advisor



AWS WAF

OWASP top 10

Injection. Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Broken Authentication. Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

Sensitive Data Exposure. Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

XML External Entities (XXE). Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

Broken Access Control. Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

Security Misconfiguration. Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

Cross-Site Scripting (XSS). XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Insecure Deserialization. Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

Using Components with Known Vulnerabilities. Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

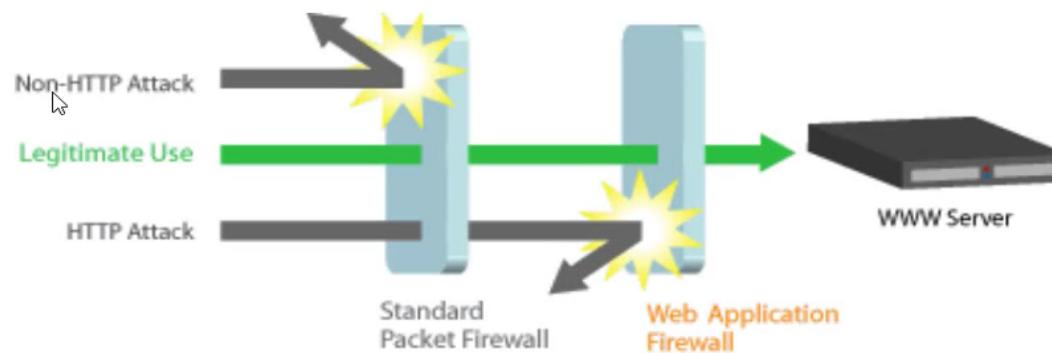
Insufficient Logging & Monitoring. Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

WAF

We all know about Firewalls and in some way might have worked as well.

Firewall works on the Layer 3 and Layer 4 of the OSI model.

Main aim of firewall: Block malicious and unauthorized traffic.



WAF

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources.

It works at layer 7, unlike Security Groups and NACL, which works at Layer 4.

AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns you define.

You can get started quickly using Managed Rules for AWS WAF, a pre-configured set of rules managed by AWS or AWS Marketplace Sellers. The Managed Rules for WAF address issues like the **OWASP Top 10 security risks**. These rules are regularly updated as new issues emerge. AWS WAF includes a full-featured API that you can use to automate the creation, deployment, and maintenance of security rules.

With AWS WAF, you pay only for what you use. The pricing is based on how many rules you deploy and how many web requests your application receives. There are no upfront commitments.

AWS WAF

We can combine multiple statements into rules to precisely target requests.

WAF provides two primary rule types: **Regular Rule & Rate-Based rule**

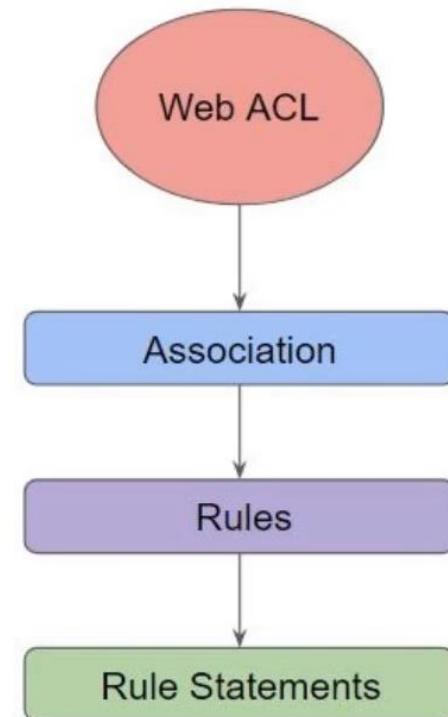
Let's look into sample regular rule:

1. If a request comes from 172.30.0.50
2. Request has SQL-like code

Rules with multiple statements can be AND, OR, NOT

Rate-Based rule = Regular Rule + Rate limiting feature

1. If a request comes from 172.30.0.50
2. If requests exceeds 1000 request in 10 minutes



AWS WAF

We can combine multiple statements into rules to precisely target requests.

WAF provides two primary rule types: **Regular Rule & Rate-Based rule**

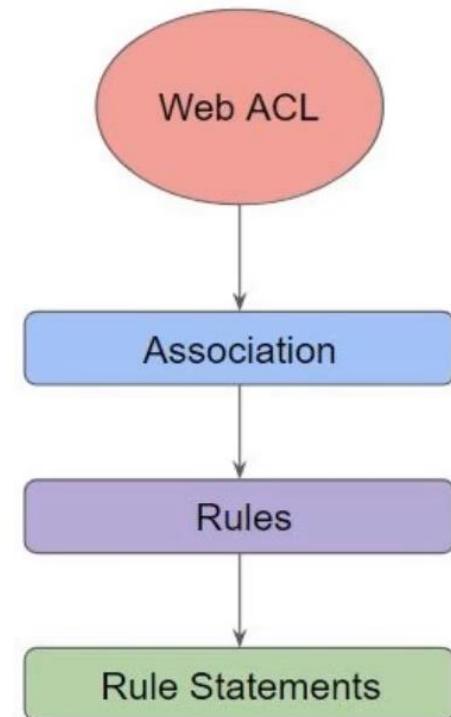
Let's look into sample regular rule:

1. If a request comes from 172.30.0.50
2. Request has SQL-like code

Rules with multiple statements can be AND, OR, NOT

Rate-Based rule = Regular Rule + Rate limiting feature

1. If a request comes from 172.30.0.50
2. If requests exceeds 1000 request in 10 minutes



AWS Shield – DDoS Prevention

Types of DDoS attack

#	Layer	Application	Description	Vector Example
7	Application	Data	Network process to application	HTTP floods, DNS query floods
6	Presentation	Data	Data representation and encryption	SSL abuse
5	Session	Data	Interhost communication	N/A
4	Transport	Segments	End-to-end connections and reliability	SYN floods
3	Network	Packets	Path determination and logical addressing	User Datagram Protocol (UDP) reflection attacks
2	Datalinks	Frames	Physical addressing	N/A
1	Physical	Bits	Media, signal, and binary transmission	N/A

AWS Shield

AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS.

AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection.

There are two tiers of AWS Shield - **Standard** and **Advanced**.

All AWS customers benefit from the automatic protections of AWS Shield Standard, at no additional charge.

AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target your web site or applications.

When you use AWS Shield Standard with Amazon CloudFront and Amazon Route 53, you receive comprehensive availability protection against all known infrastructure (Layer 3 and 4) attacks.

AWS Shield

AWS Shield Advanced provides additional detection and mitigation against large and sophisticated **DDoS** attacks, near real-time visibility into attacks, and integration with **AWS WAF, a web application firewall**.

AWS Shield Advanced also gives you 24x7 access to the AWS DDoS Response Team (DRT) and protection against DDoS related spikes in your Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 charges.

AWS Shield Advanced is available globally on all Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 edge locations. You can protect your web applications hosted anywhere in the world by deploying Amazon CloudFront in front of your application. Your origin servers can be Amazon S3, Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), or a custom server outside of AWS. You can also enable AWS Shield Advanced directly on an Elastic IP or Elastic Load Balancing (ELB)

Penetration Testing

Penetration Testing

AWS customers can carry out security assessments or penetration tests against their AWS infrastructure without prior approval for 8 services, listed in the next section under “Permitted Services.”

Stress test and DDoS Simulation can be performed with prior approval and policy adherence.

Please ensure that these activities are aligned with the policy set out below.

Note: Customers are not permitted to conduct any security assessments of AWS infrastructure, or the AWS services themselves. If you discover a security issue within any AWS services in the course of your security assessment, please contact AWS Security immediately.

If AWS receives an abuse report for activities related to your security testing, they will forward it to you. When responding, please provide the root cause of the reported activity, and detail what you've done to prevent the reported issue from recurring.

Penetration Testing

Permitted Services

Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers

Amazon RDS

Amazon CloudFront

Amazon Aurora

Amazon API Gateways

AWS Lambda and Lambda Edge functions

Amazon Lightsail resources

Amazon Elastic Beanstalk environments

Prohibited Activities

DNS zone walking via Amazon Route 53 Hosted Zones

Denial of Service (DoS), Distributed Denial of Service (DDoS), Simulated DoS, Simulated DDoS (These are subject to the DDoS Simulation Testing policy)

Port flooding

Protocol flooding

Request flooding (login request flooding, API request flooding)

AWS Resource Access Manager (RAM)

AWS RAM

AWS Resource Access Manager (AWS RAM) lets you share your resources with any AWS account or through AWS Organizations.

If you have multiple AWS accounts, you can create resources centrally and use AWS RAM to share those resources with other accounts.

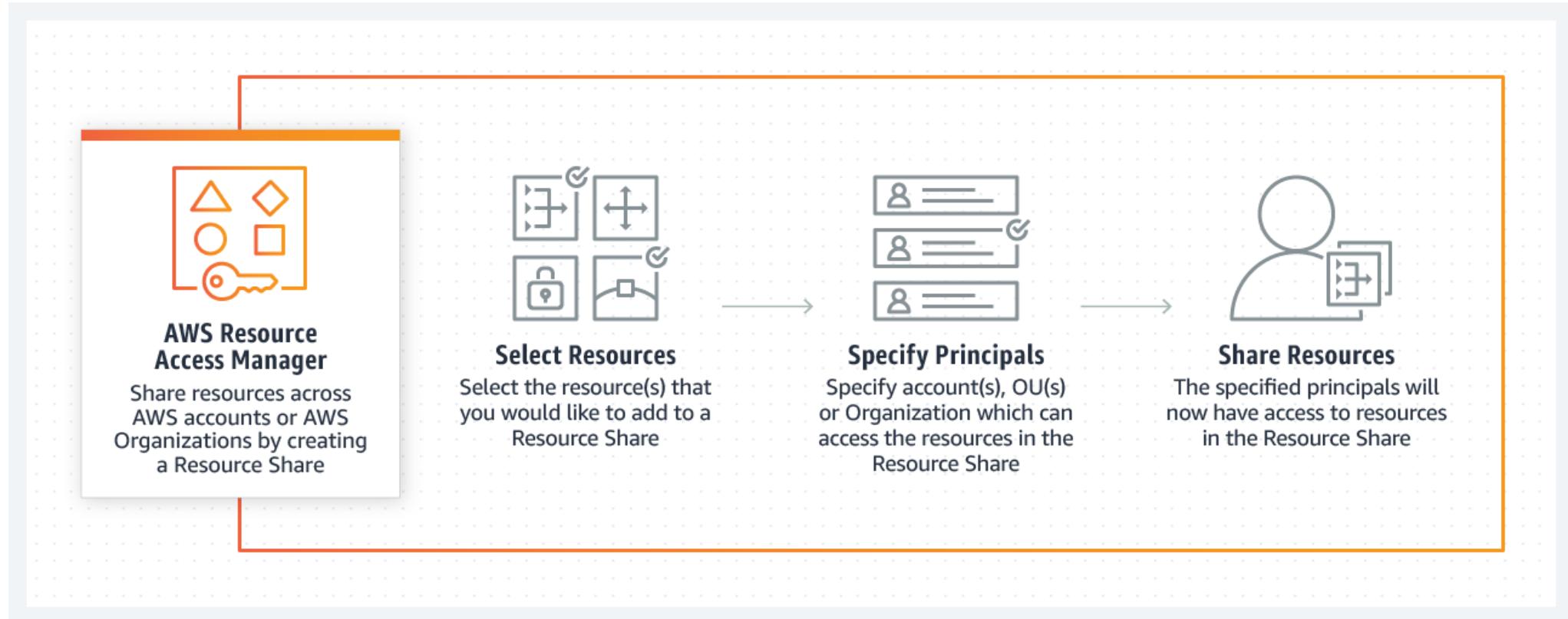
AWS RAM offers the following benefits:

Reduces operational overhead—Create resources centrally and use AWS RAM to share those resources with other accounts. This eliminates the need to provision duplicate resources in every account, which reduces operational overhead.

Provides security and consistency—Govern consumption of shared resources using existing policies and permissions, to achieve security and control. AWS RAM offers a consistent experience for sharing different types of AWS resources.

Provides visibility and auditability—View usage details for shared resources through integration with Amazon CloudWatch and AWS CloudTrail. AWS RAM provides comprehensive visibility into shared resources and accounts.

AWS RAM



Vulnerability Assessment

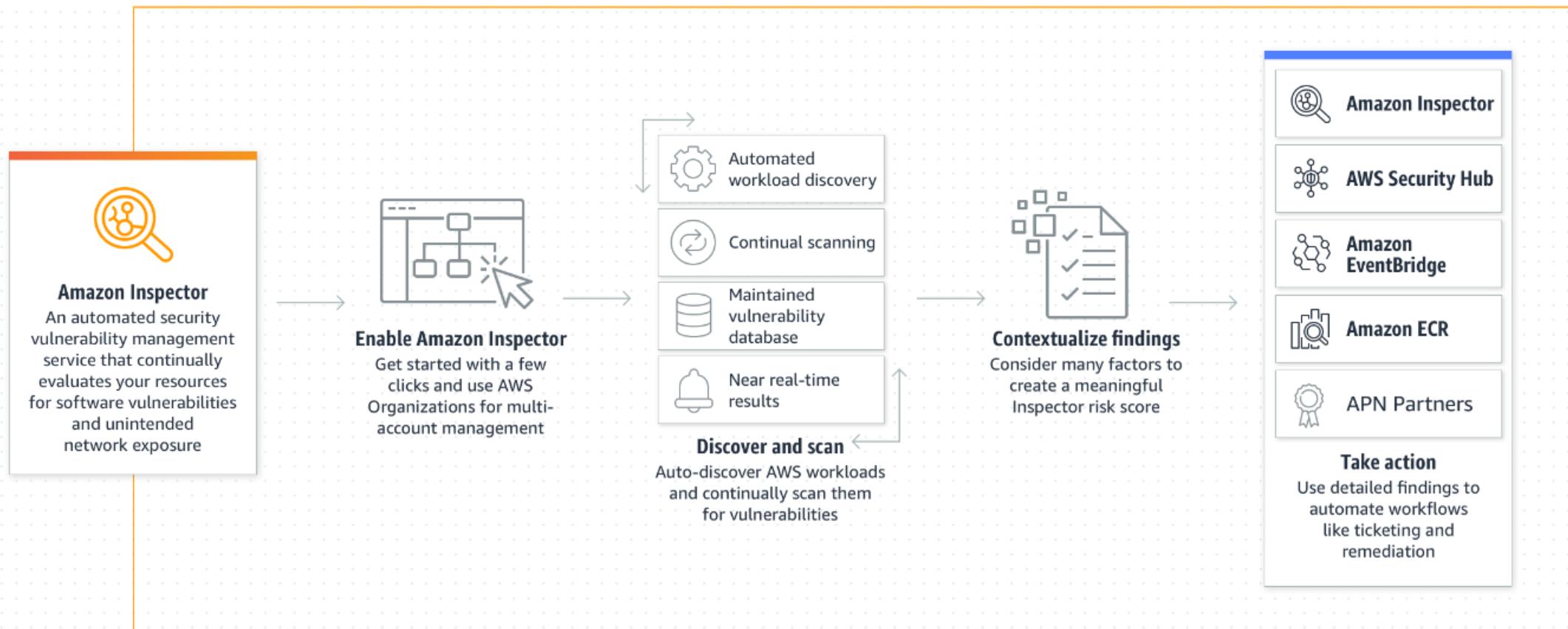
Amazon Inspector

- Amazon Inspector tests the network accessibility of your Amazon EC2 instances and the security state of your applications that run on those instances.
- You can use Amazon Inspector to assess your assessment targets (collections of AWS resources) for potential security issues and vulnerabilities.
- Amazon Inspector compares the behavior and the security configuration of the assessment targets to selected security rules packages (Security checks).
- After performing an assessment, Amazon Inspector produces a detailed list of security findings that is organized by level of severity.
- With Amazon Inspector, you can automate security vulnerability assessments throughout your development and deployment pipelines or for static production systems. This allows you to make security testing a regular part of development and IT operations.

Amazon Inspector

- **Configuration scanning and activity monitoring engine** – Amazon Inspector provides an agent that analyzes system and resource configuration. It also monitors activity to determine what an assessment target looks like, how it behaves, and its dependent components. The combination of this telemetry provides a complete picture of the target and its potential security or compliance issues.
- **Built-in content library** – Amazon Inspector includes a built-in library of rules and reports. These include checks against best practices, common compliance standards, and vulnerabilities. The checks include detailed recommended steps for resolving potential security issues.
- **Automation through an API** – Amazon Inspector can be fully automated through an API. This allows you to incorporate security testing into the development and design process, including selecting, executing, and reporting the results of those tests.

Amazon Inspector



Amazon Inspector

- **Common Vulnerabilities and Exposures**
- **CIS Operating System Security Configuration Benchmarks**
- **Security Best Practices**
- **Network Reachability**

Amazon Inspector

The screenshot shows the Amazon Inspector interface with the following details:

Summary info
Viewing data from all accounts

Environment coverage
Your accounts, instances, and repositories that are enabled with Inspector.

Accounts	Instances	Repositories
100% 3 / 3 accounts	83% 5 / 6 Instances	100% 1 / 1 repository

Critical findings
All active critical findings in your environment.

ECR container	EC2 instance	Network reachability
134 Critical 1277 total findings	2 Critical 1233 total findings	0 Critical 1 total finding

Risk based remediations
Vulnerabilities impacting the most instances and images.

Package name	Critical	All
openssl	16	93
libgd2	15	38
imagemagick	13	212
graphite2	12	51
curl	12	129

[View all vulnerabilities](#)

Threat Detection

AWS Guard Duty

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following data sources: **VPC Flow Logs**, **AWS CloudTrail event logs**, and **DNS logs**.

Intelligent threat detection system

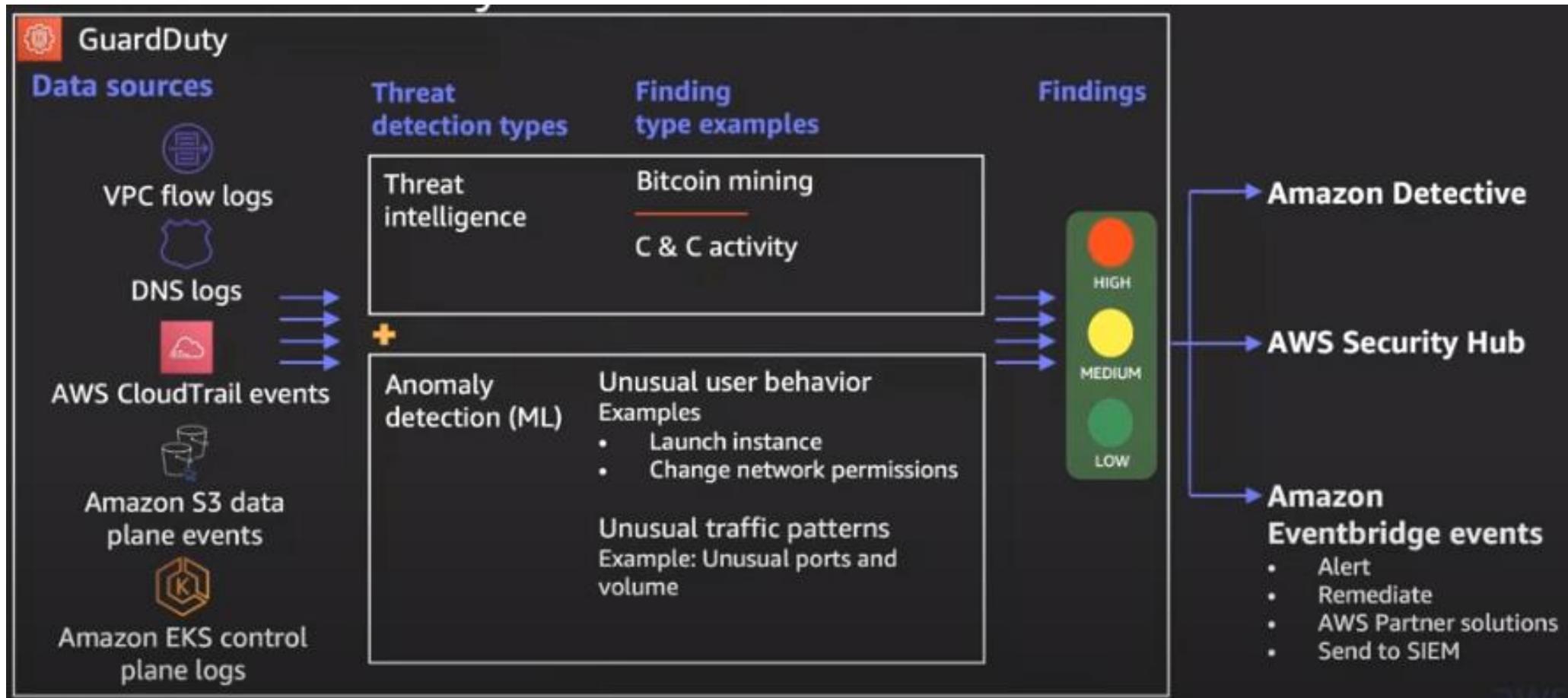
It uses **threat intelligence feeds**, such as lists of **malicious IPs** and **domains**, and **machine learning** to identify unexpected and potentially **unauthorized and malicious activity** within your AWS environment. This can include issues like **escalations of privileges**, **uses of exposed credentials**, or **communication with malicious IPs, URLs, or domains**. For example, GuardDuty can detect compromised EC2 instances serving **malware or mining bitcoin**.

Continuously monitor your AWS environment for suspicious activity and generate findings.

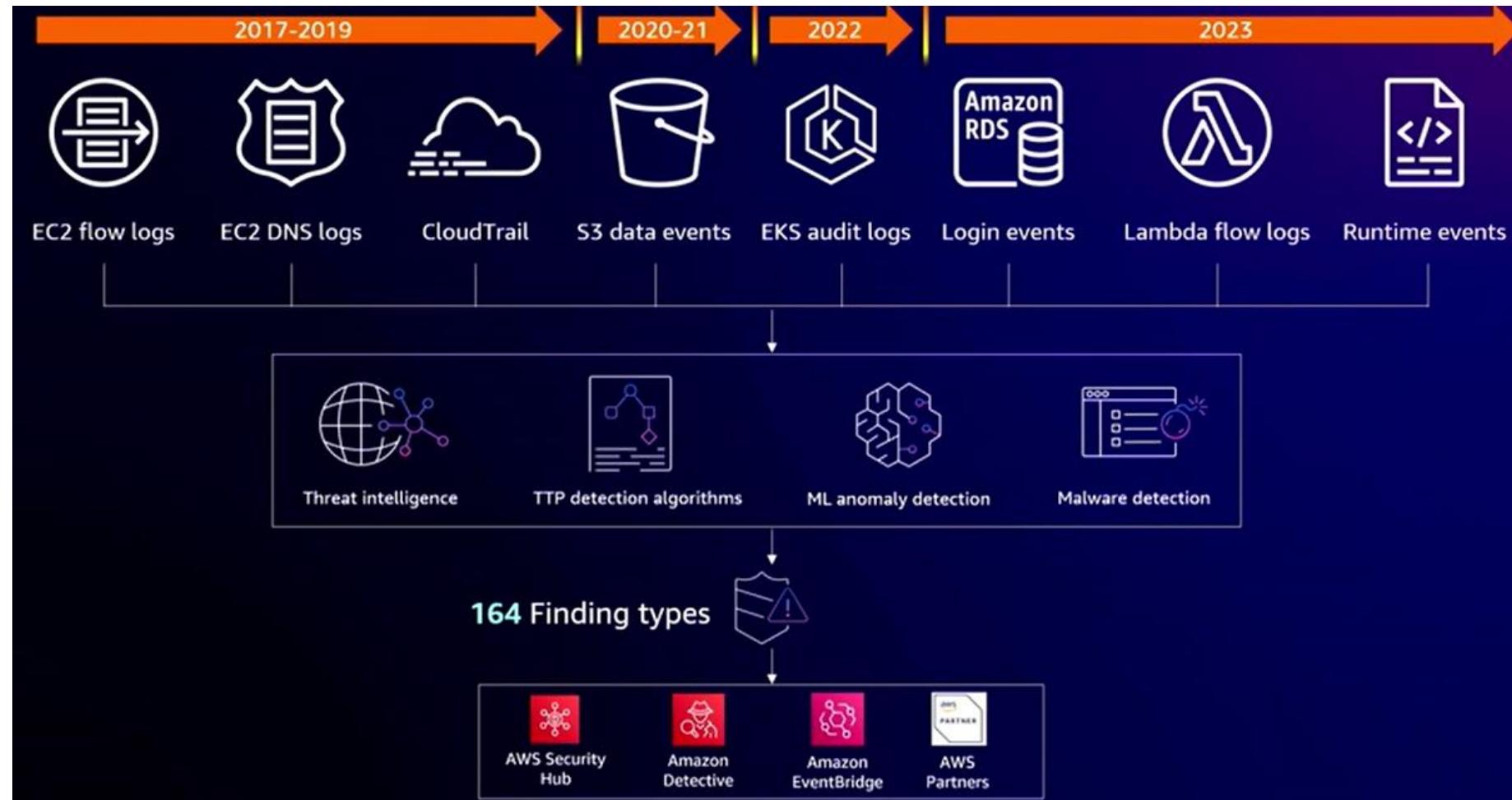
Analyze multiple data sources, including AWS CloudTrail events and VPC Flow Logs.

Customize GuardDuty by adding your own threat lists and trusted IP lists.

AWS Guard Duty



AWS Guard Duty



AWS Guard Duty S3 Protection

Policy	Malicious access	Anomalous user behavior
<ul style="list-style-type: none">• Bucket made public• Block public access disabled• Logging disabled• Root credentials used	<ul style="list-style-type: none">• Data discovery, exfiltration, or modification from:<ul style="list-style-type: none">• Tor• Leaked instance credentials• Malicious IPs	<ul style="list-style-type: none">• Unusual location• Unusual bucket• Unusual volume• Unusual error volume

AWS Guard Duty EKS Protection



AWS Guard Duty

It also monitors AWS account access behavior for signs of compromise, such as unauthorized infrastructure deployments, like instances deployed in a Region that has never been used, or **unusual API calls**, like a password policy change to reduce password strength.

GuardDuty continuously analyzes **VPC Flow Logs** and **DNS requests** and responses to identify malicious, unauthorized, or unexpected behavior in your AWS accounts and workloads.

Findings are displayed in the GuardDuty console and contain a detailed description of the security issue. You can also retrieve your generated findings by calling the **GetFindings** and **ListFindings** API operations.

You can create Suppression rules to create very specific combinations of attributes to suppress findings. For e.g. suppress logs for a particular instance.

You can specify a list of trusted IPs and a list of threat IPs to suppress or specifically generate findings for same

Finding Types:

https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-active.html

AWS Guard Duty ECS Runtime Monitoring

- Fully manages runtime threat detection for serverless containers, removing organizational complexity
- Uniquely detect entire attack chains of events with machine learning and integrated threat intelligence
- Accurately detect and respond to threats early before they escalate to broader business impacting breaches
- Get the coverage and visibility required to protect your container workloads from an evolving threat landscape

AWS Guard Duty ECS Runtime Monitoring

	EKS	ECS	AWS Fargate	EC2
Node Level Network (IP & DNS)	✓	✓	NA	✓
Malware Detection (EBS)	✓	✓	NA	✓
Control Plane	✓	✓	✓	✓
Container level network (IP & DNS)	✓	✓	✓	✓
Container Specific Threats	✓	✓	✓	✓
Process event threat detection	✓	✓	✓	✓

Data Privacy: Discovery & Classification

Amazon Macie

Amazon Macie is a security service that uses **machine learning & pattern matching** to automatically discover, classify, and protect sensitive data in AWS. Macie recognizes sensitive data such as **personally identifiable information (PII)** or **intellectual property**.

A Real effective tool to identify your S3 loopholes at one place.

Macie automatically detects a large and growing list of sensitive data types, including personal identifiable information (PII) such as names, addresses, and credit card numbers. It maintains a growing list of sensitive data types that include common **personally identifiable information (PII)** and other sensitive data types as defined by data privacy regulations, such as GDPR, PCI-DSS, and HIPAA.

Scalable on-demand and automated sensitive data discovery jobs

Ability to add custom-defined data types using regular expressions to enable Macie to discover proprietary or unique sensitive data for your business.

Continual evaluation of your Amazon S3 environment for any unencrypted buckets, publicly accessible buckets, or buckets shared outside your AWS Organization.

Amazon Macie

❖Data Discovery and Classification

Amazon Macie enables you to identify business-critical data and analyze access patterns and user behavior as follows:

- Continuously monitor new data in your AWS environment
- Use artificial intelligence to understand access patterns of historical data
- Automatically access user activity, applications, and service accounts
- Use natural language processing (NLP) methods to understand data
- Intelligently and accurately assign business value to data and prioritize business-critical data based on your unique organization
- Create your own security alerts and custom policy definitions

Amazon Macie

❖ Data Security

Amazon Macie enables you to be proactive with security compliance and achieve preventive security as follows:

- Identify and protect various data types, including PII, PHI, regulatory documents, API keys, and secret keys
- Verify compliance with automated logs that allow for instant auditing
- Identify changes to policies and access control lists
- Observe changes in user behavior and receive actionable alerts
- Receive notifications when data and account credentials leave protected zones
- Detect when large quantities of business-critical documents are shared internally and externally

Amazon Macie

Macie supports a subset of the regex pattern syntax provided by the Perl Compatible Regular Expressions (PCRE) library.

When Amazon Macie analyzes data in an S3 bucket, it performs a deep inspection that factors the file or storage format for the data. Macie can analyze and detect sensitive data in many different formats, including commonly used compression and archive formats. This support applies to both managed data identifiers and custom data identifiers.

File or storage type	Description	File name extensions
Big data	Apache Avro object containers and Apache Parquet files	–
Compression or archive	GNU Zip compressed archives, TAR archives, and ZIP compressed archives	.gz, .gzip, .tar, .zip
Document	Adobe Portable Document Format files, Microsoft Excel workbooks, and Microsoft Word documents	.doc, .docx, .pdf, .xls, .xlsx
Text	Non-binary text files such as comma-separated values (CSV) files, Hypertext Markup Language (HTML) files, JavaScript Object Notation (JSON) files, plain-text documents, tab-separated values (TSV) files, and Extensible Markup Language (XML) files	.csv, .htm, .html, .json, .tsv, .txt, .xml, and others (depending on the type of non-binary text file)

When Macie analyzes a compressed or archive file, it inspects both the full file and the contents of the file. To inspect the file's contents, it decompresses the file, and then inspects each extracted file that uses a supported format. It does this for as many as 1,000,000 files and up to a nested depth of 10 layers.

SIEM

Security Hub

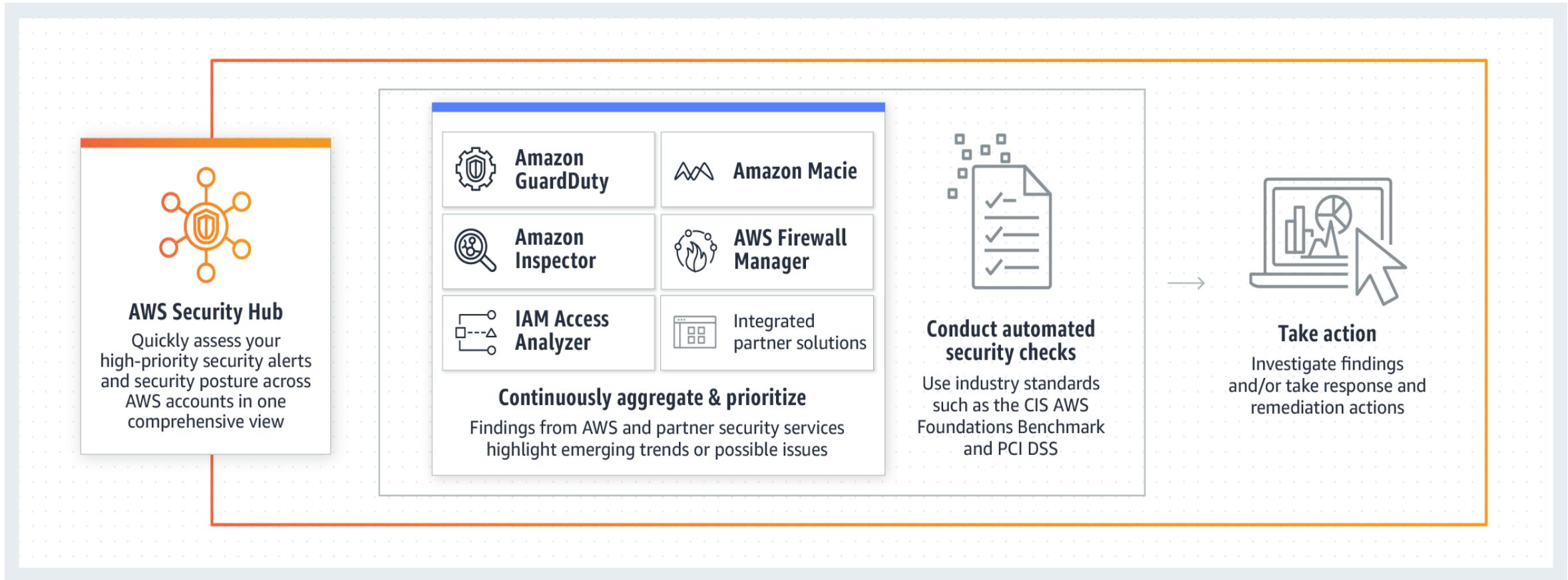
AWS Security Hub gives you a comprehensive view of your high-priority security alerts and security posture across your AWS accounts.

A single place that **aggregates, organizes, and prioritizes** your security alerts, or findings, from multiple AWS services, such as **Amazon GuardDuty, Amazon Inspector, Amazon Macie, AWS Identity and Access Management (IAM) Access Analyzer, and AWS Firewall Manager**, as well as from AWS Partner solutions.

AWS Security Hub continuously monitors your environment using automated security checks based on the AWS best practices and industry standards that your organization follows.

You can also take action on these security findings by investigating them in Amazon Detective or by using Amazon CloudWatch Event rules to send the findings to ticketing, chat, Security Information and Event Management (SIEM), Security Orchestration Automation and Response (SOAR), and incident management tools or to custom remediation playbooks.

Security Hub



Secret Manager

AWS Secrets Manager

- You can use Secrets Manager to store, rotate, monitor, and control access to secrets such as database credentials, API keys, and OAuth tokens.
- Enable secret rotation using built-in integration for MySQL, PostgreSQL, and Amazon Aurora on Amazon RDS. You can also enable rotation for arbitrary secrets using AWS Lambda functions.
- To retrieve secrets, you simply replace hardcoded secrets in applications with a call to Secrets Manager APIs, eliminating the need to expose plaintext secrets.
 - ❖ Rotate secrets safely
 - ❖ Manage access with fine-grained permissions
 - ❖ Secure and audit secrets centrally
 - ❖ Pay as you go

OWASP Top 10

- A document containing the 10 most critical security concerns for web application security
- Released every few years (historical releases have been 2003, 2004, 2007, 2010, 2013, 2017, 2021)
- Historically released as a PDF

1. Broken Access Control

If **authentication and access restriction** are not properly implemented, it's easy for attackers to take whatever they want. With broken access control flaws, **unauthenticated or unauthorized** users may have access to **sensitive files and systems**, or even user privilege settings.

Penetration testing can detect missing authentication but cannot determine the misconfigurations that lead to the exposure. One of the benefits of the increasing use of Infrastructure as Code (IaC) tools is the ability to use scanning tools to detect configuration errors leading to access control failures.

Weak access controls and issues with credentials management in applications are preventable with secure coding practices, as well as preventative measures like locking down administrative accounts and controls and using multi-factor authentication.

2. Cryptographic Failure

Common errors such as using hardcoded passwords, outdated cryptographic algorithms, or weak cryptographic keys can result in the exposure of sensitive data

Scanning images for hardcoded secrets, and ensuring that data is properly encrypted at rest and in transit can help mitigate exposing sensitive data to attackers.

3. Injection

Injection attacks occur when attackers exploit vulnerabilities in web applications that **accept untrusted data**. Common types include **SQL injection** and **OS command injection**. This category now also includes **Cross Site Scripting (XSS)**. By inserting malicious code into input fields, attackers can execute unauthorized commands, access sensitive databases, and potentially gain control over systems.

Application security testing can reveal injection flaws and suggest remediation techniques such as stripping special characters from user input or writing parameterized SQL queries.

4. Insecure Design

Insecure design is a new category in the 2021 OWASP Top Ten which focusses on fundamental design flaws and ineffective controls as opposed to weak or flawed implementations.

Creating secure designs and secure software development lifecycles requires a combination of culture, methodologies and tools. Developer training, robust **threat modelling**, and an organizational library of secure design patterns should all be implemented to reduce the risks of insecure designs creating critical vulnerabilities.

5. Security Misconfiguration

Application servers, frameworks, and cloud infrastructure are highly configurable, and security misconfigurations such as too broad permissions, insecure default values left unchanged, or too revealing error messages can provide attackers easy paths to compromise applications.

The 2023 Veracode State of Software Security reported that misconfiguration errors were reported in 70% or more applications that had introduced a new vulnerability in the last year.

To reduce misconfiguration risks organizations should routinely harden deployed application and infrastructure configurations and should scan all infrastructure as code components as part of a secure SDLC.

6. Vulnerable and Outdated Components

Modern applications are built using a large number of **third-party libraries** (which themselves are dependent on other libraries), and frequently run on open-source frameworks. In a modern application there may be orders of magnitude more code from libraries and components than written by an organization's developers.

As might be expected with any software, vulnerabilities in libraries and components will routinely be discovered, patched, and new versions released. The challenges of identifying all the components in use, keeping track of their vulnerability status, and routinely rebuilding and testing deployed software is both essential and onerous. Perhaps this is why so many organizations are still running vulnerable software in production.

8. Software and Data Integrity Failures

The tools used to build, manage, or deploy software are increasingly common vectors of attack. A CI/CD pipeline that routinely builds, tests and deploys software can also be used to inject malicious code (or libraries), create insecure deployments, or steal secrets.

As discussed above in ‘Vulnerable and Outdated Components’ modern applications use many third-party components, often pulling them from third party repositories.

Organizations can mitigate this threat by ensuring both the security of the build process, and the components pulled into the build. Adding in code scanning and software component analysis steps into a software build pipeline can identify malicious code or libraries.

7. Identification and Authentication Failures

Identifying and authorizing users and non-human clients is a fundamental security practice. It goes without saying that weaknesses in a way an application allows access or identifies users is a critical vulnerability.

While mitigation starts with secure coding practices, tools to detect and prevent credential stuffing and brute force attacks are also useful protections.

9. Security Logging and Monitoring Failures

Having adequate logging and monitoring in place is essential in both detecting a breach early, hopefully limiting the damage, and in incident forensics to establish the scope of the breach, and to determine the method of compromise.

Simply generating the data is obviously insufficient, organizations must have adequate collection, storage, alerting and escalation processes. Organizations should also verify that these processes are working correctly – using Dynamic Application Security Testing (DAST) tools like Veracode DAST, for instance, should produce significant logging and alerting events.

10. Server-Side Request Forgery (SSRF)

Modern web applications commonly fetch additional content or data from a remote resource. If an attacker can influence the destination resource, and the application does not validate the supplied URL, then a crafted request may be sent to a target destination.

Mitigating SSRF attacks is done using familiar methods such as sanitizing user input, using explicit allow lists, and inspecting request responses before they are returned to clients.

- The OWASP Application Security Verification Standard (ASVS) Project provides a basis for testing web application technical security controls and also provides developers with a list of requirements for secure development.
- It can be used as a checklist when testing web applications. For Example if you were testing a logon page you can reference the ASVS and see exactly how the logon page should behave vs how it does behave.
- Provided in multiple formats e.g PDF, Word, CSV
- <https://github.com/OWASP/ASVS/tree/v4.0.3?tab=readme-ov-file#latest-stable-version---403>

- Concise information on application security topics e.g. what to do and what not to do
- For example, when it comes to authentication this is a huge topic that could be daunting to first try and understand. The authentication cheat sheet is concise and breaks down exactly where to focus.
- Cheat sheets also exist for the OWASP Top 10 and OWASP ASVS which index which cheat sheets are relevant so can be directly correlated between the two.
- <https://cheatsheetseries.owasp.org/>

- Produced by CIS (Centre for internet security)
- Non profit with the aim of securing organisations against **cyber threats**
- CIS benchmarks are **configuration guidelines** for various products e.g **software, operating systems, network devices, mobile devices** etc
- Be aware as they can make your life easier e.g. need to ensure a server is hardened? Check against specific OS's CIS benchmark
- They also provide hardened images e.g. out of the box for Azure, making your life easier as they are pre-hardened
- Some tools will test against benchmarks

Further Security Concepts

Standards and Compliances

Few standards which focuses around Information Security and touchbase Cloud:

- **ISO 27001** looks to certify that the Information Security management can address relevant risks and elements that is appropriate based on risks
- **ISO 27002** is the framework for best practice
- **SOC I, II, III** Service Organization Control defines a comprehensive approach to auditing and assesses the provider's controls and their effectiveness
- **NIST 800-53**: Goal is to ensure that appropriate security requirements and security controls are applied to all US Federal government information and information systems
- **ISO/IEC 17788/17789** to address cloud computing concepts
- **FIPS 140** addresses uses of encryption and cryptography
- **PCI-DSS, HIPPA, GDPR, PDPA** and other regulations/LAW- to protect PII.

Data Privacy Acts

PII (Personal Identifiable Information) must be treated as secret and same get enforced with multiple regulatory breaches, in case not handled properly. Multiple regulators (based on Geographical regions) are there to protect Personal Rights:

Law/Regulations

EU Data Protection Directive 95/46 (Now General Data Protection Regulation (GDPR))

Australian privacy Act

Canada Personal Information protection & Electronic Document Act (PIPEDA)

EFTA & Switzerland

APEC Privacy Framework

Standards

ISO/IEC 27017:2015

PCI DSS

Note: Remember, its Data Owner who is always at stake.

<https://forms.gle/o78W31owEbePmRMQ6>



THANK
YOU