

AWS EC2 & IAM

CS516 – Cloud Computing

Computer Science Department

Maharishi International University

Maharishi International University - Fairfield, Iowa



All rights reserved. No part of this slide presentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or recording, or by any information storage and retrieval system, without permission in writing from Maharishi International University.

Main concepts

- Global infrastructure (Regions, AZs)
- VPC and subnets
- EC2 (AMI, EBS, Snapshots, SG, Pricing model)
- IAM (users, roles, permission and trust policies, STS assume role)

AWS Global Infrastructure Map

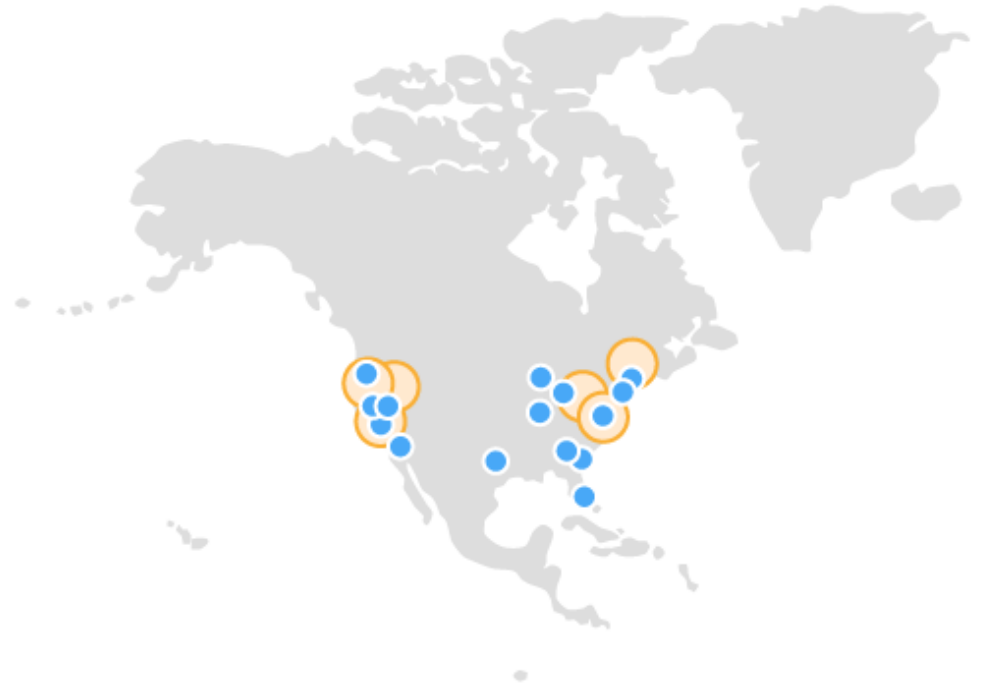


Read more about: [AWS Global Infrastructure](#)

AWS Global Infrastructure

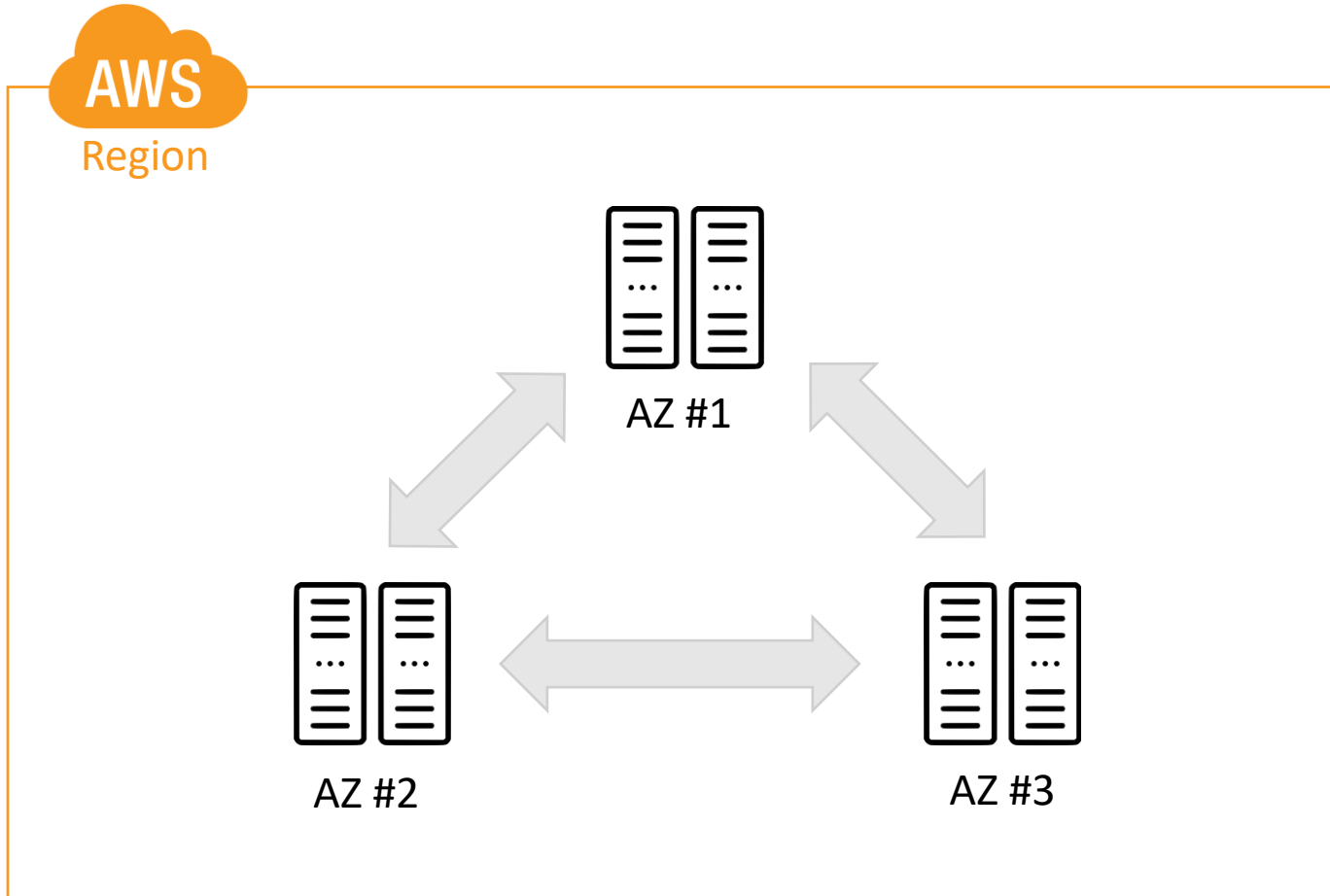
Regions - A physical location around the world where AWS ***clusters*** data centers. Usually comprised of multiple data centers.

Availability Zones - Geographical isolated data centers within a region.



AWS Region/AZ

Each Region has multiple, isolated locations known as *Availability Zones*.



Availability Zones provide redundancy for AWS resources in that region, highly available, fault tolerant, and more scalability.

AZs have low latency, high-bandwidth network connection, and supports synchronous **replication** between AZs. All traffic is **encrypted**.

High Availability is creating an architecture in such a way that the system is always available (or has the least amount of downtime as possible).

Fault Tolerant is the ability of your system to withstand failures in one or more of its components and still remain available.

Read More about: [Regions and Zones](#)

Edge servers

AWS has regions all over the world. They also have servers in 247+ countries. Those servers are not independent regions and are called edge server. Edge servers play the following roles:

1. Can find other AWS resources faster since they all are in the same AWS network.
2. Caching.
3. You can run small code like Lambda.

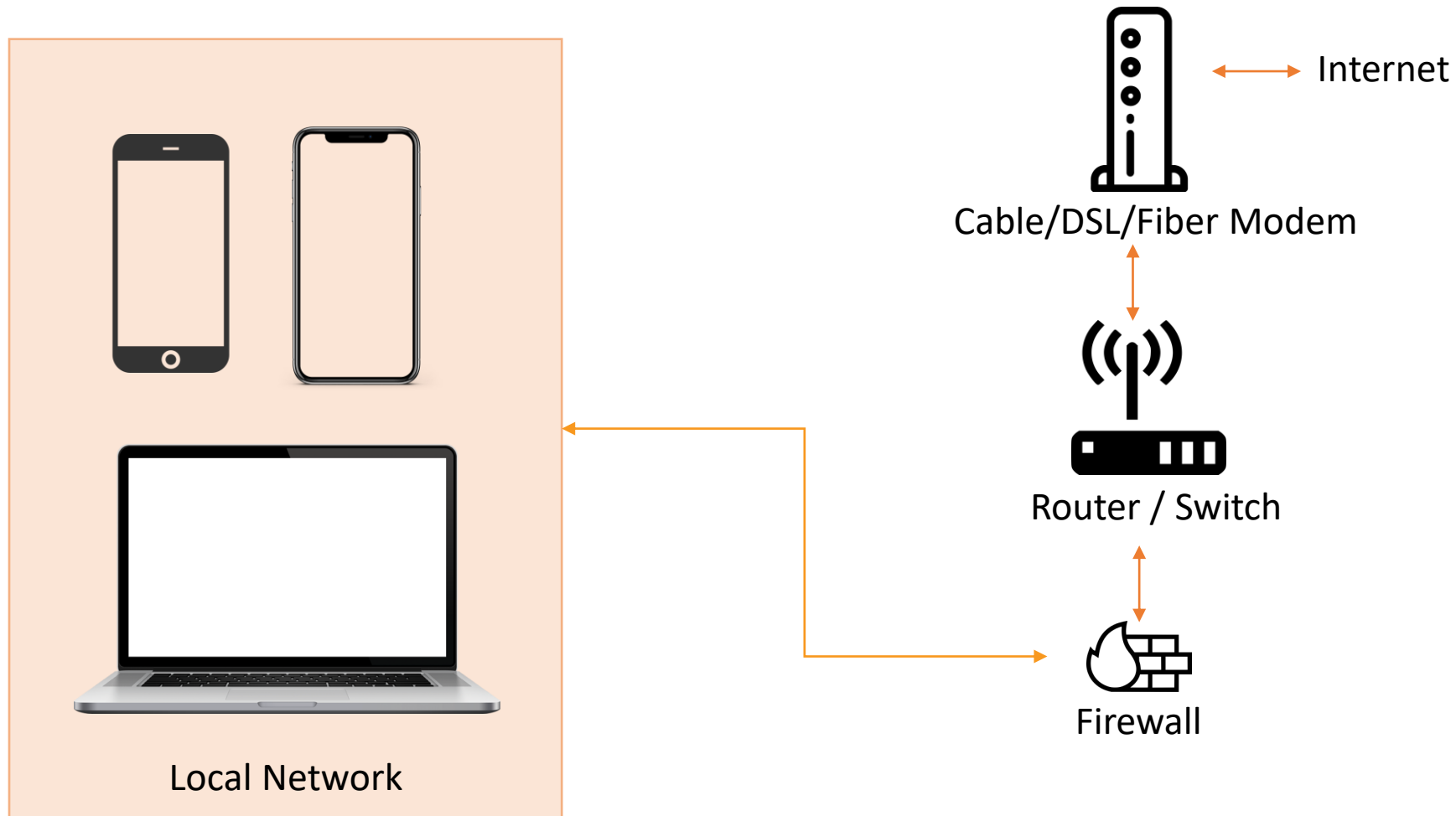
Virtual Private Cloud (VPC)

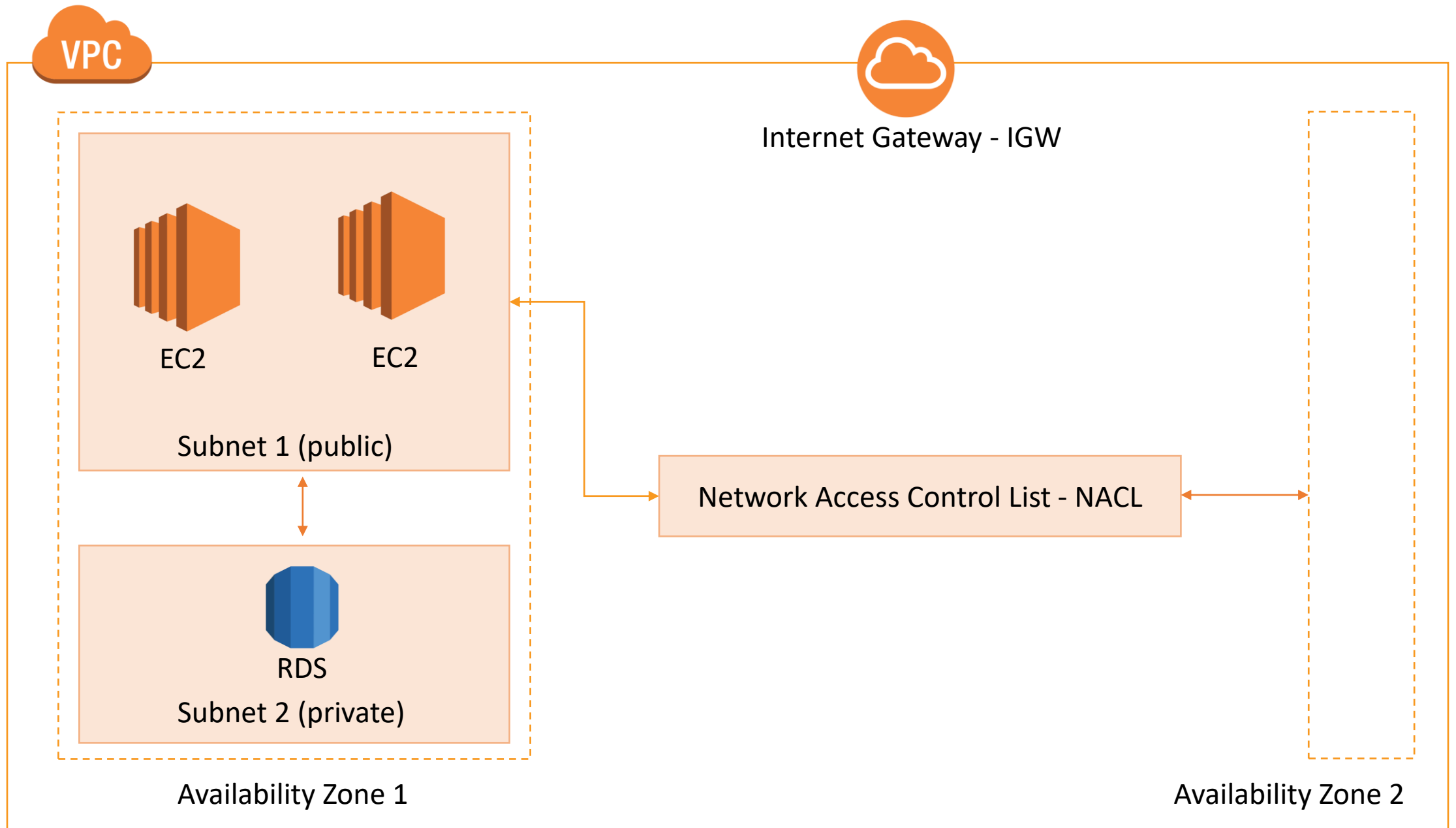
Amazon Virtual Private Cloud is an isolated virtual network where your AWS resources run. You have complete control over your virtual networking environment, including selection of your own **IP address range**, creation of **subnets**, internet gateway, and more.

VPC is where your instances like EC2 gets its IP addresses. IP is a unique string of characters that identifies each computer using the Internet Protocol to communicate over a network. So it is reachable from the internet and can talk to resources in the same/different network.

VPC is a regional service and is associated to a single region like [most of other AWS](#) services. You cannot span a VPC across regions. If you work on a global application, you deploy that in multiple regions.

Your Home Network





Internet Gateways - IGW

A combination of hardware and software that provides your private network with a route to the Internet.

One IP for all resources in your network (VPC).

A horizontally scaled redundant and highly available VPC component that **allows communication between instances in your VPC and the Internet.**

Your default VPC already has an IGW attached.

Only 1 IGW can be attached to a VPC at a time.

An IGW cannot be detached from a VPC while there are active AWS resources in the VPC.

<input type="checkbox"/>	Name	Name	ID	State	VPC
<input type="checkbox"/>			igw-12dcdb6a	attached	vpc-af9b48d5

Read more about [Internet Gateways](#)

Subnets

A subnet is a sub-section of a network. Generally, it includes all the computers in a specific location like zip code for addressing houses.

After creating a VPC, you can add one or more subnets in each AZ. Each subnet **must reside entirely within one AZ and cannot span zones**.

Private subnet is a safe environment. The internet (outsiders) cannot directly access your resources in private subnet. In a private (secure) sub-section of VPC, you can place AWS resources, like back-end servers and databases. Resources in the same network can access resources in private subnet.

Anyone can access to resources in public subnet directly from the internet.

Subnets are written in CIDR format.

Read more about [Subnets](#)

Total address space

200.100.10.0/24
(256 addresses)

200.100.10.0	200.100.10.1
200.100.10.2	200.100.10.3
200.100.10.4	200.100.10.5
200.100.10.6	200.100.10.7
⋮	⋮
200.100.10.252	200.100.10.253
200.100.10.254	200.100.10.255

Before Subnetting

Partial address spaces

200.100.10.0/25
(128 addresses)

200.100.10.0	200.100.10.1
⋮	⋮
200.100.10.126	200.100.10.127

200.100.10.128/25
(128 addresses)

200.100.10.128	200.100.10.129
⋮	⋮
200.100.10.254	200.100.10.255

After Subnetting

VPC Security Layers

The VPC has two layers of security:

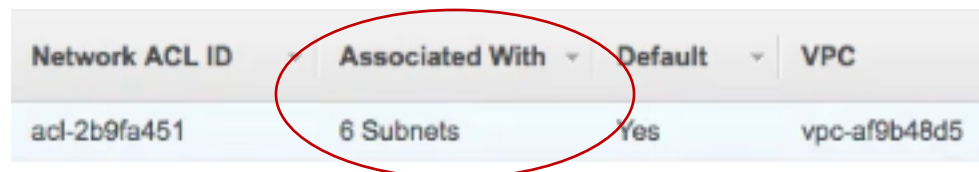
- Security Groups (SG) can be allowed to modify permission any **resource** that it is attached to. (*Instance level*)
- Network Access Control Lists (NACL) are applicable for the whole **subnet** that they are attached to. (*Subnet level*)

NACLs are **stateless** so the rules for inbound and outbound traffic are separate while SG is **stateful**.

Network Access Control Lists - NACL

Acts as a **firewall between subnets**. A network access control list (NACL) is an **optional layer of security** for your VPC that acts as a **firewall** for controlling traffic in and out of one or more **subnets**.

- Your default VPC already has an NACL in place and associated with all default subnets.



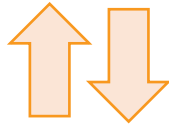
Network ACL ID	Associated With	Default	VPC
acl-2b9fa451	6 Subnets	Yes	vpc-af9b48d5

In N. Virginia region, we have 6 AZs, AWS created a subnet replicated in all AZs.

Read more about [Network ACLs](#)



The default NACL allows all traffic, both inbound and outbound



Network Access Control List - NACL



EC2

Subnet 1 (public)



EC2

Subnet 2 (public)

NACL Rules

- Rules are evaluated from lowest to highest based on rule #.
- The first rule found that applies to the traffic type is immediately applied, regardless of any rules that come after it.
- A subnet can only be associated with one NACL at a time.
- A NACL allows or denies traffic from entering a subnet. Once inside the subnet, other AWS resources may have additional security layers (security groups).

NACL Rules

- The **default NACL** allows all traffic to the default subnets.
- Any **new NACL** you create denies all traffic by default.

Inbound	Rule #	Type	Protocol	Port Range	Source	Allow / Deny	All traffic is allowed
	100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW	
	*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	
Inbound	Rule #	Type	Protocol	Port Range	Source	Allow / Deny	All traffic is denied
	90	SSH (22)	TCP (6)	22	0.0.0.0/0	DENY	
	100	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW	
	*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	

Only Allow SSH

Inbound	Rule #	Type	Protocol	Port Range	Source	Allow / Deny	Only allow SSH
	100	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW	
	*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	
Outbound	Rule #	Type	Protocol	Port Range	Destination	Allow / Deny	Allow SSH response
	100	Custom TCP Rule	TCP (6)	1024-65535	0.0.0.0/0	ALLOW	
	*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	

Only Allow HTTP

Inbound							Only allow HTTP
	Rule #	Type	Protocol	Port Range	Source	Allow / Deny	
	100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW	
Outbound	*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	Allow HTTP response
	Rule #	Type	Protocol	Port Range	Destination	Allow / Deny	
	100	HTTP (80)	TCP (6)	1024-65535	0.0.0.0/0	ALLOW	
	*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	

Notes

- The outbound traffic will use ephemeral ports 1024-65535 for the return web traffic and not port 80.
- An ephemeral port is a short-lived transport protocol port for Internet Protocol (IP) communications.

AWS Elastic Cloud Compute (EC2)

EC2 is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Basically, a virtual computer, very similar to the desktop or laptop you use at home, and commonly referred to as an **instance**.

You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.

Amazon EC2 autoscaling enables you to scale in or out to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Read more: [Amazon EC2](#)

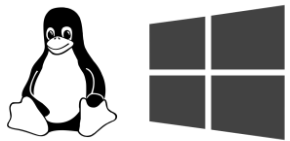
Computer and EC2 Instance



Computer



EC2 Instance



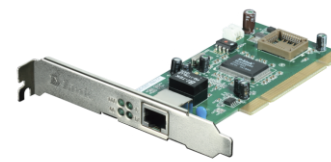
Operating System
AMIs
(Linux or Windows)



CPU & RAM
Instance Type



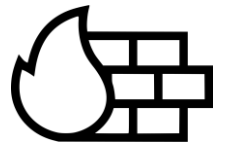
Hard Drive
EBS



Network Adapter
ENI



RAM



Firewall
Security Groups

Amazon Machine Image - AMI

Preconfigured and required to launch an EC2 instance that includes an operating system, software packages and other required settings.



Amazon Machine Image (AMI) **provides the information required to launch an instance.** You specify an AMI when you launch an instance.

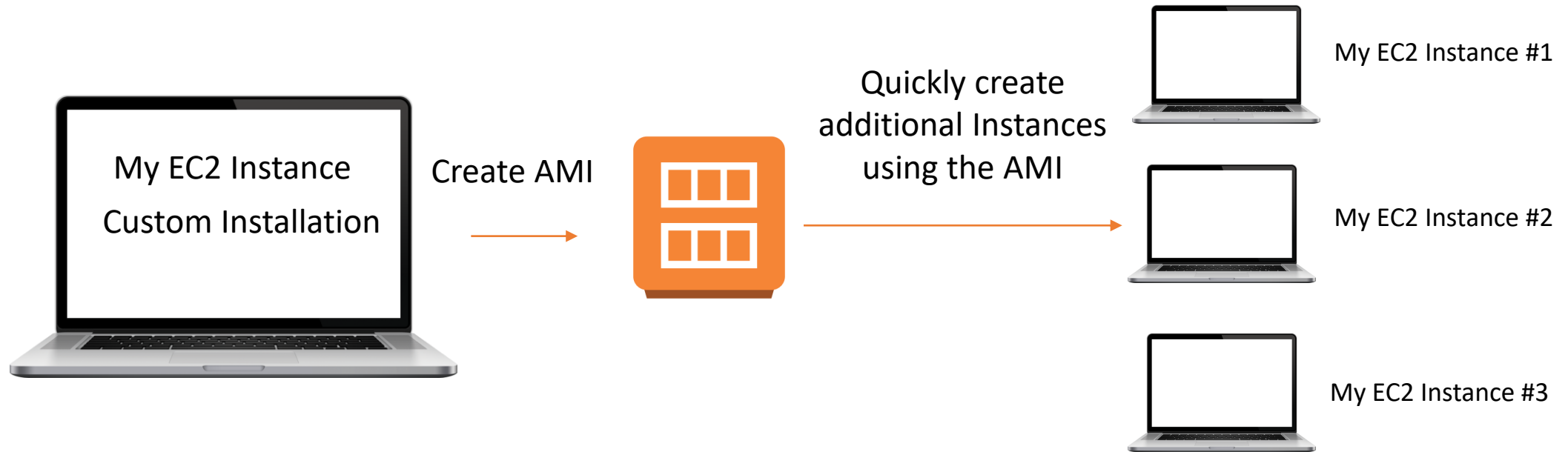
Read more: [AMI](#)

Custom AMI

Assume that you need to run your app in hundreds of servers. It is not possible to configure every single server one by one. Instead, you can create a custom AMI and use it on as many servers as you want. A custom AMIs contain:

- Operating System (OS)
- Packages of the OS
- Dependencies for your application.
- Your application configuration.

Understanding AMI



Instance Types

The Instance Type is the **CPU** (Central Processing Unit) of your instance.

When you launch an instance, the instance type that you specify determines the hardware of the host computer used for your instance.

Each instance type offers different **compute**, **memory**, and **storage**, **graphic** capabilities and are grouped in instance families based on these capabilities.

Read more: [Instance Types](#)

General-purpose and memory-optimized instance type

General-purpose instances provide a balance of compute, memory, and networking resources. It can be used for web servers and code repositories. General-purpose instance types start with **T** and **M**.

Memory-optimized instances are good for applications that process large amounts of data in memory such as caching. Memory-optimized instance types start with **R** and **X**.

Compute-optimized instance type

Compute-optimized instances are ideal for compute-bound applications that benefit from high-performance processors. It is good for batch processing workloads, media transcoding, high-performance web servers, high-performance computing (**HPC**), scientific modeling, dedicated gaming servers, ad server engines, machine learning inference, and other compute-intensive applications. Compute-optimized instance types start with **C**.

Accelerated computing instance type

Accelerated computing instances use hardware accelerators (GPUs) or co-processors to perform functions such as floating point number calculations, graphics processing, or data pattern matching.

Both accelerated computing and compute-optimized instance types provide high compute power. But the way how they provide the compute power is different. Accelerated computing instances have GPUs which are accelerators whereas the compute-optimized ones have good high-performance processors. Accelerated computing instance types start with **P** and **G**.

Storage-optimized instance type

Storage-optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications. Storage-optimized instance types start with **I** and **D**.

Do not configure EC2 as a database! Instead, use Relation Database Service which provides instance types designated for databases.

Elastic Block Storage - EBS

EBS is a storage volume for an EC2 instance. It provides block level storage volumes for use with EC2 instances.

EBS volumes are **highly available and reliable** storage volumes that can be attached to any running instance that is **in the same Availability Zone**.

EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist **independently** from the life of the instance. It charges independently.

Read more: [Amazon EBS Volumes](#)



=



Input/Output Operations Per Second - IOPS

IOPS is the amount of data that can be written to or retrieved from EBS per second.

The operations are measured in KiB, and the underlying drive technology determines the maximum amount of data that a volume type counts as a single IO.

More IOPS means better volume performance (faster R/W speeds).

More expensive. Recommended for production environment.

Types of EBS volumes

- **Provisioned IOPS** – The fastest and most expensive where you can specify IOPS. Good for production servers.
- **General Purpose** – It provides moderate amount of speed. You can not specify IOPS. The IOPS depends on the volume size. Good for development environment servers.
- **HDD and Magnetic** – The cheapest option and slow. It is good when storing large amounts of data and archiving and you don't need much performance.

Create Additional Volume

[Volumes](#) > Create Volume

Create Volume

Volume Type General Purpose SSD (gp2) ▼ ⓘ

Size (GiB) 100 (Min: 1 GiB, Max: 16384 GiB)

IOPS 300 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)

Availability Zone* us-east-1a ▼ ⓘ

Throughput (MB/s) Not applicable ⓘ

Snapshot ID Select a snapshot ▼ ↺ ⓘ

Encryption ☐ Encrypt this volume ⓘ

Amazon Elastic File System (Amazon EFS)

It is an elastic file system that lets you share file data without provisioning or managing storage.

Amazon EFS is designed to provide massively parallel **shared** access to thousands of Amazon EC2 instances.

Amazon EFS is well suited to support a broad spectrum of use cases from home directories to business-critical applications.

You need to set a Security Group that allows access from a fleet of EC2 instances on the EFS volume.

Read more: [Amazon EFS](#)



Block vs File vs Object storages

File storage stores data as a single piece of information **in a folder** to help organize it among other data.

Block storage takes a file apart **into singular blocks** of data and then stores these blocks as separate pieces of data. Faster than the file storage.

Object storage **is a flat structure** in which files are spread out among hardware. Unlimited scaling.

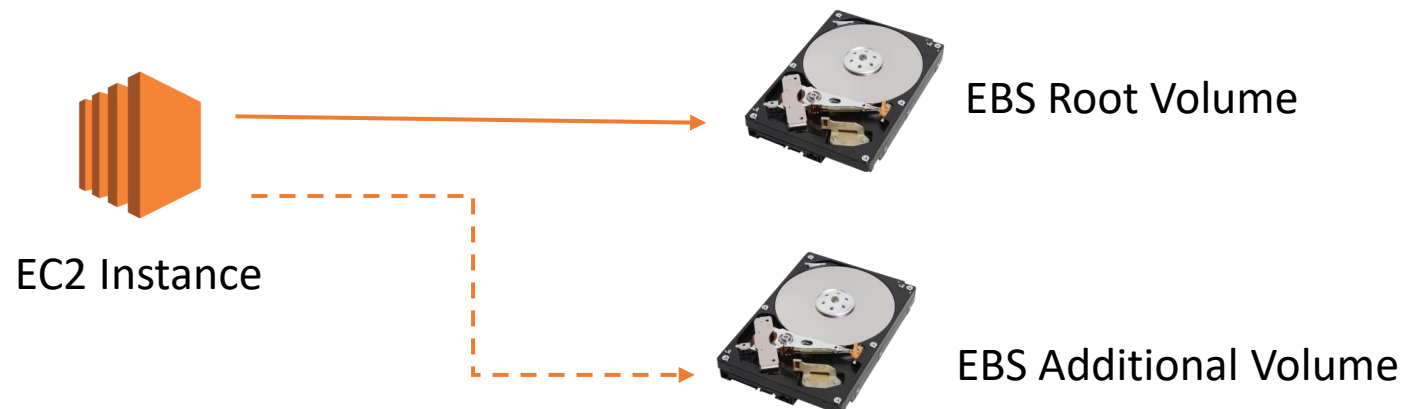
Read more: [File storage, block storage, or object storage?](#)

Root vs. Additional EBS Volumes

Every EC2 instance must have a **root volume** that the AMI is restored.

You can add **additional EBS Volumes** to an instance at anytime. Any additional volume can be attached or detached from the instance at any time.

Additional EBS volumes are NOT deleted (the default) and you still pay when the instance is terminated whereas root volume gets deleted when the instance is terminated.

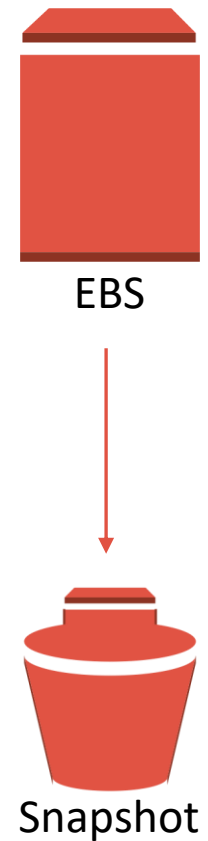


Snapshots

A snapshot is an **image** of an EBS volume that can be stored as an **incremental backup** of the volume or used to create a duplicate.

A snapshot is not an active EBS volume. You cannot attach or detach a snapshot to an EC2 instance.

To restore a snapshot, you need to create a new EBS volume using the snapshot as its template.



AMI vs Snapshots

AMI

An entire EC2 instance definition that includes all EBS snapshots plus some metadata like kernel, AMI name, description, block device mappings, and more.

Could be used to deploy your applications on different machines easily.

EBS Snapshot

Backup of a single volume.

You need to recover and mount a snapshot to an EC2 instance.

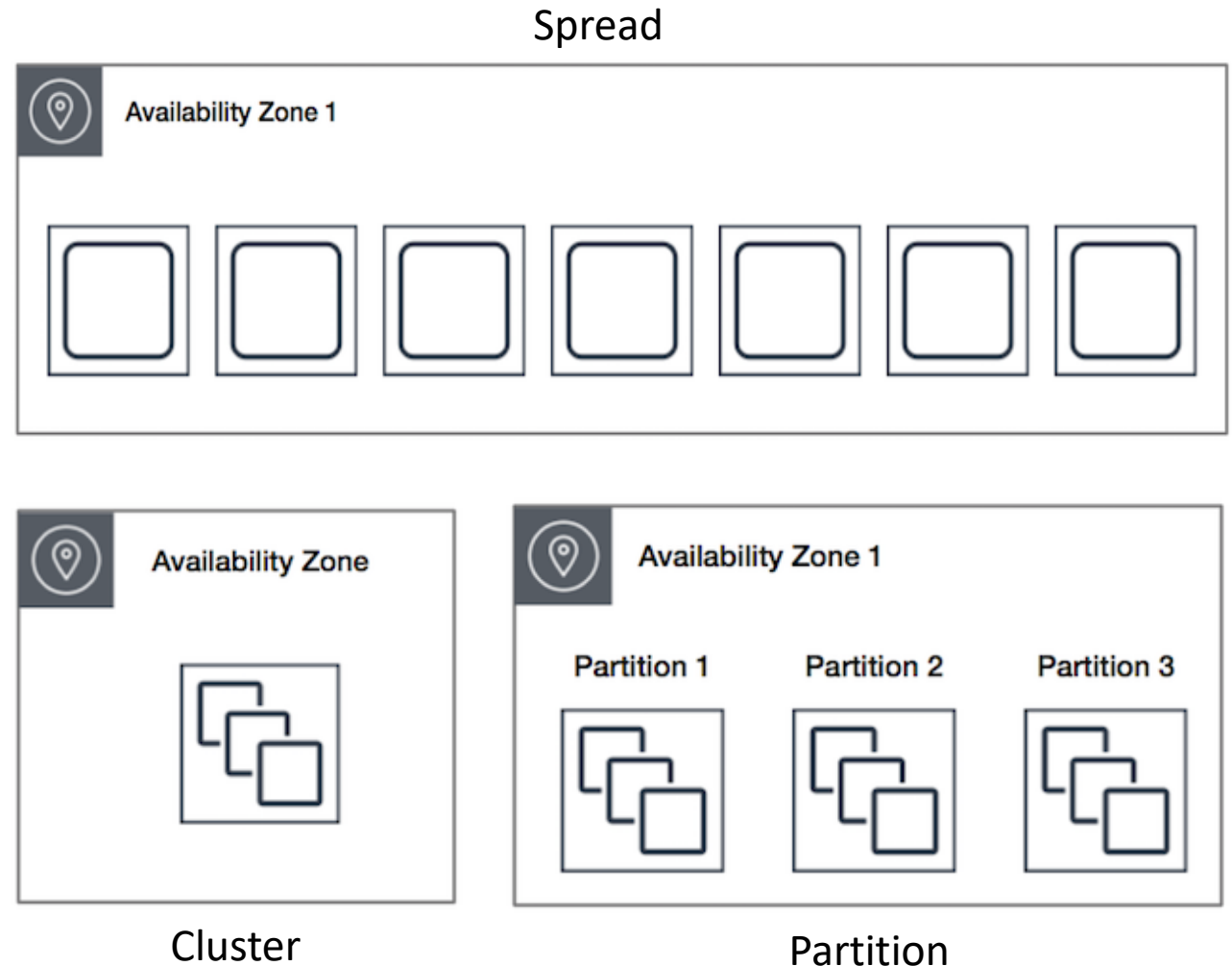
Placement groups

A placement group is a logical grouping of instances within a single AZ that benefit from low network latency and high network throughput.

Cluster – Packs instances close together. Low latency and high performance.

Partition – Spreads instances across logical partitions. One partition don't share the underlying hardware. Large distributed and replicated workloads such as Hadoop, Cassandra, and Kafka.

Spread – Places a small group of instances across distinct underlying hardware to reduce correlated failure.



EC2 Instance Options - On-Demand

On-demand purchasing allows you to choose any instance type you like and provision/terminate it at any time (on-demand).

The most expensive purchasing option.

You are only charged when it is running (billed by the hour).

Read more: [EC2 pricing](#)

EC2 Instance Options - Reserved

Reserved purchasing allows you to purchase an instance for a set time period of 1 or 3 years.

This allows for a significant price discount over using on-demand.

You can select to pay upfront, partial upfront, no upfront.

Once you buy a reserved instance, you own it for the selected time period and are responsible for the entire price - regardless of how often you use it.

About 20% savings with one-year term and 30% savings with three-year term.

EC2 Instance Options - Spot

Cheapest but not reliable.

Amazon sells unused instances for short amounts of time at a substantial discount.

Spot pricing is a way for you to bid on an instance type, then only pay for and use that instance when the spot price is equal to or below your bid price.

Spot prices fluctuate based on supply and demand.

A provisioned instance automatically terminates when the spot price is greater than your bid price.

You can use Spot instance along with other types of instances in your cluster or a fleet of servers for a fault-tolerant application. So you save a lot.

IP Addressing

An IP address is the EC2 instance address on the network.

Private IP Address:

EC2 instance receive the private IP from the subnet.

All EC2 instances have a private IP address.

Private IP addresses allow instances to communicate with resource in the same network.

Public IP address:

All EC2 Instances can be launched with or without a public IP address.

Public IP addresses are required for the instance to communicate with the Internet.

Read more: [IP Addressing](#)

Elastic IP

Elastic IP is static public IP.

When you stop and then start an EC2 instance, it may change its public IP. If you need to have a fixed public IP for your instance, you need an Elastic IP.

- An Elastic IP is a public IPv4 IP you own.
- You can attach it to one instance at a time.
- By default, you can have 5 elastic IPs in AWS.
- It charges when you are NOT using. When using, it is free.

Security Groups (SG)

Security groups are found on the **instance level**. They act as a **virtual firewall** that controls the traffic for one or more instances.

When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that **allow traffic** to or from its associated instances.

You can attach multiple SGs to an instance. All the rules from the security groups that are associated with the instance are evaluated.

Read more: [Security Groups](#)

Inbound/Outbound Rules

Name

Group ID

Group Name

VPC ID

sg-b63da5fa

default

vpc-af9b48d5

Description

Inbound

Outbound

Tags

Edit

Type ⓘ

Protocol ⓘ

Port Range ⓘ

Source ⓘ

Description ⓘ

All traffic

All

All

your IP

Description

Inbound

Outbound

Tags

Edit

Type ⓘ

Protocol ⓘ

Port Range ⓘ

Destination ⓘ

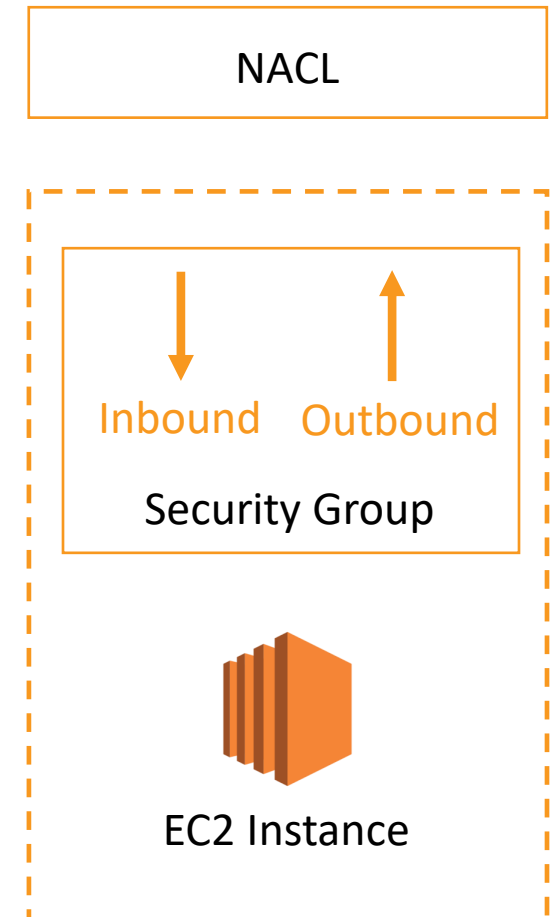
Description ⓘ

All traffic

All

All

0.0.0.0/0



Create Security Group

When you create a new Security Group, by default:

- All inbound traffic is denied
- All outbound traffic is allowed. Developers don't often touch the outbound rule.

Create Security Group

Security group name

Description

VPC

vpc-af9b48d5 (default)

Security group rules:

Inbound

Outbound

Type	Protocol	Port Range	Source	Description
This security group has no rules				

Add Rule

Inbound

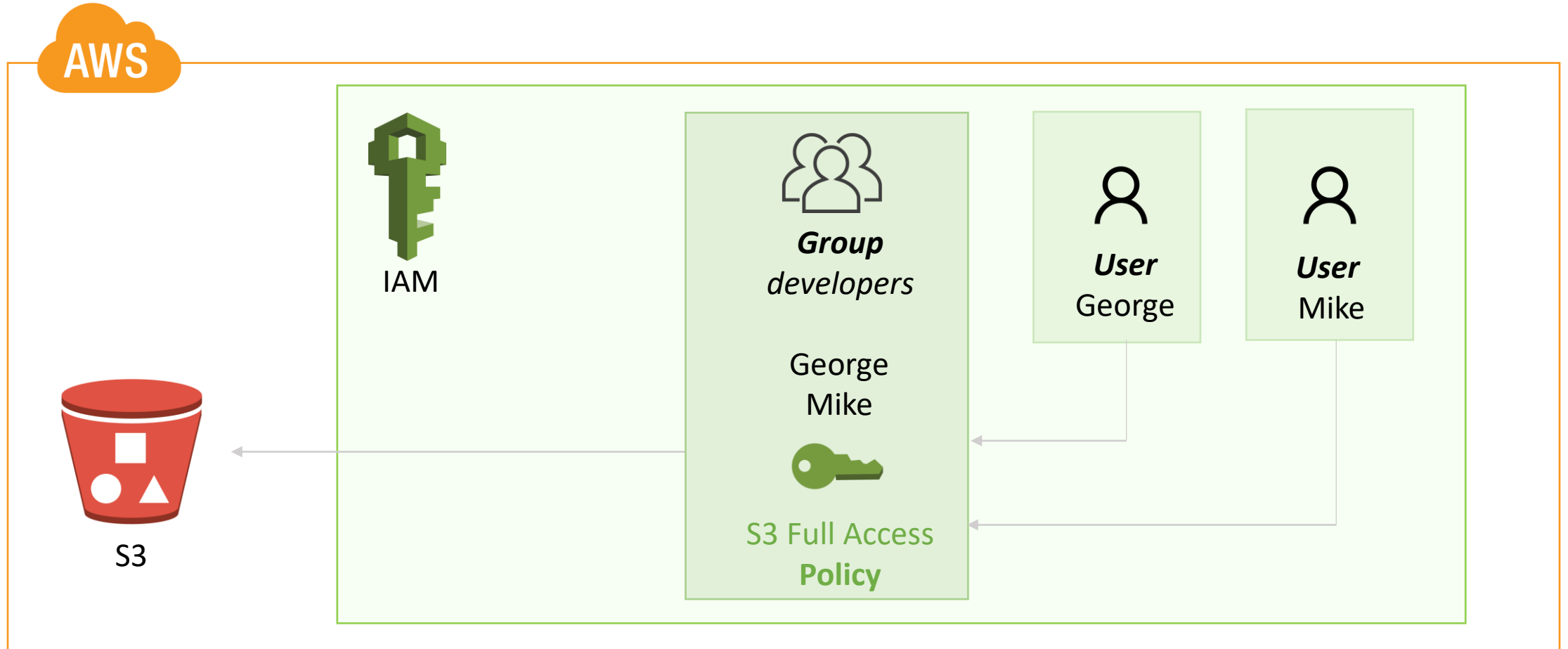
Outbound

Type	Protocol	Port Range	Destination	Description
All traffic	All	0 - 65535	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

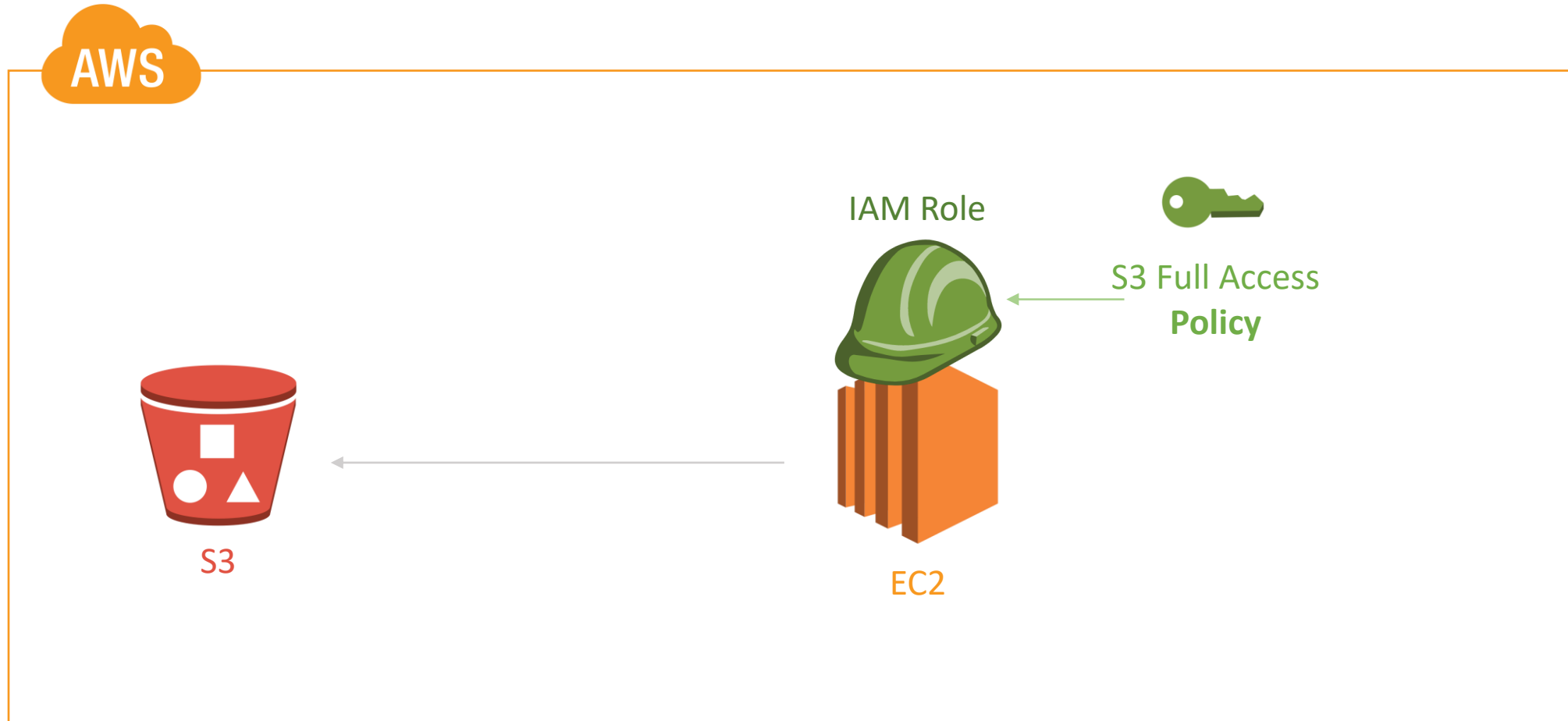
Add Rule

NACL	Security Group
NACL can be understood as the firewall or protection for the subnet.	Security group can be understood as a firewall to protect EC2 instances.
These are stateless, meaning any change applied to an incoming rule isn't automatically applied to an outgoing rule. Example: If a request comes through port 80, it should be explicitly indicated that its outgoing response would be the same port 80.	These are stateful, which means any changes which are applied to an incoming rule is automatically applied to a rule which is outgoing. Example: If the incoming port of a request is 80, the outgoing response of that request is also 80 (it is opened automatically) by default.
NACL supports allow and deny rules. Denial of rules can be explicitly mentioned, so that when the layer sees a specific IP address, it blocks connecting to it.	SG supports only allow rules, and the default behavior is denial of all.
In case of NACL, the rules are applied in the order of their priority, wherein priority is indicated by the number the rule is assigned. This means every rule is evaluated based on the priority it has.	In case of a security group, all the rules are applied to an instance.

AWS IAM Group



AWS IAM Role



IAM Policies

IAM Policies are **permissions** that you can assign to any User, Group, and Roles.

We don't attach a IAM Policy to a Service, instead we would need to use a **Role**.

There are **AWS managed** policies and **user managed** policies.

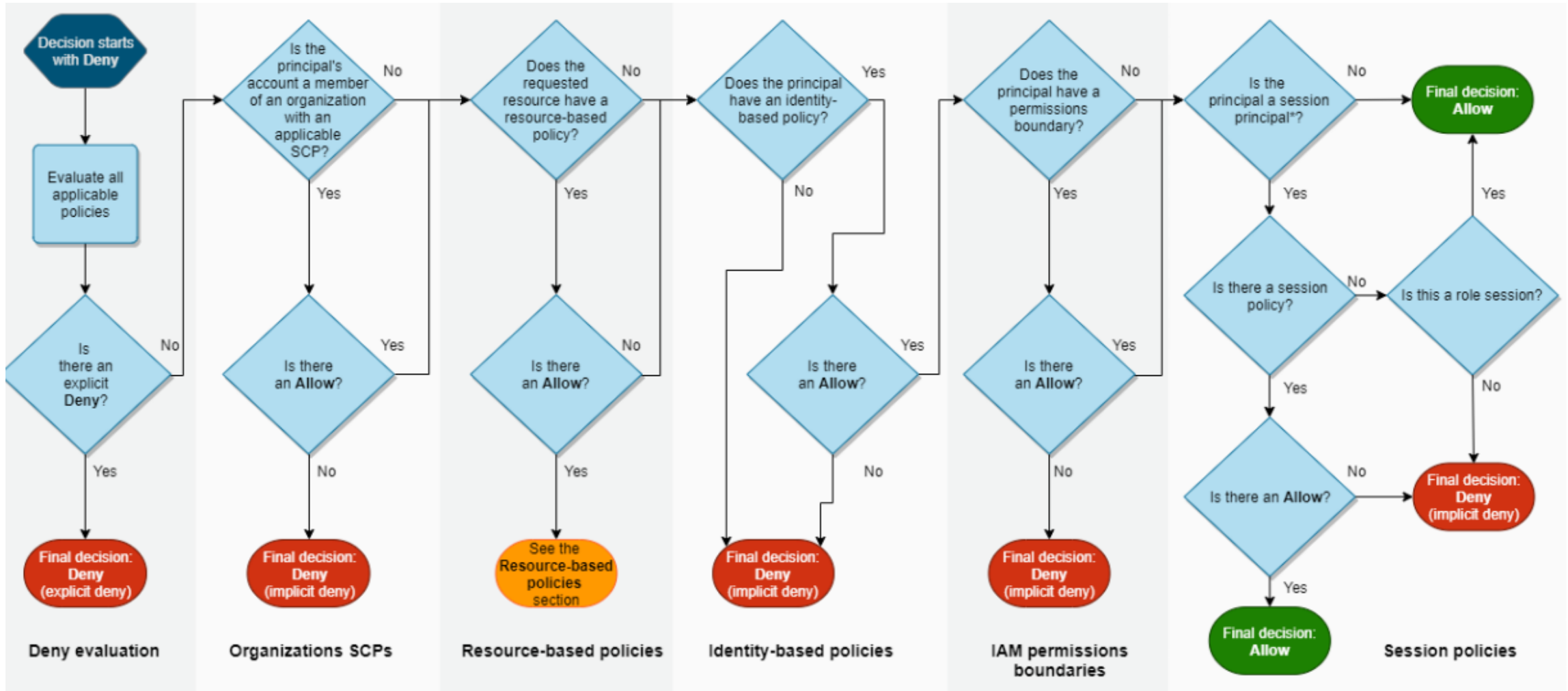
Identity-based policies
– Attach policies to IAM identities (users, or roles)

Resource-based policies
– Attach inline policies to resources (S3). Has a **principal** tag.

Which one will win if there are similar identity and resource-based policies?

Read More about: [IAM Policies](#)

1. Deny policies always win.
2. The resource-based policies takes precedence over identity-based policies.



IAM Policy structure

The **Action** element is the specific API action for which you are granting or denying permission

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  }]
}
```

The **Effect** element can be Allow or Deny

The **Resource** element specifies the resource that's affected by the action

The **Condition** element is optional and can be used to control when your policy is in effect

IAM JSON policy elements: Condition

The Condition element (or Condition block) lets you specify conditions for when a policy is in effect.

In the Condition element, you build expressions in which you use condition operators (equal, less than, etc.) to match the condition keys and values in the policy against keys and values in the request context.

Learn more: [IAM JSON policy elements: Condition](#)

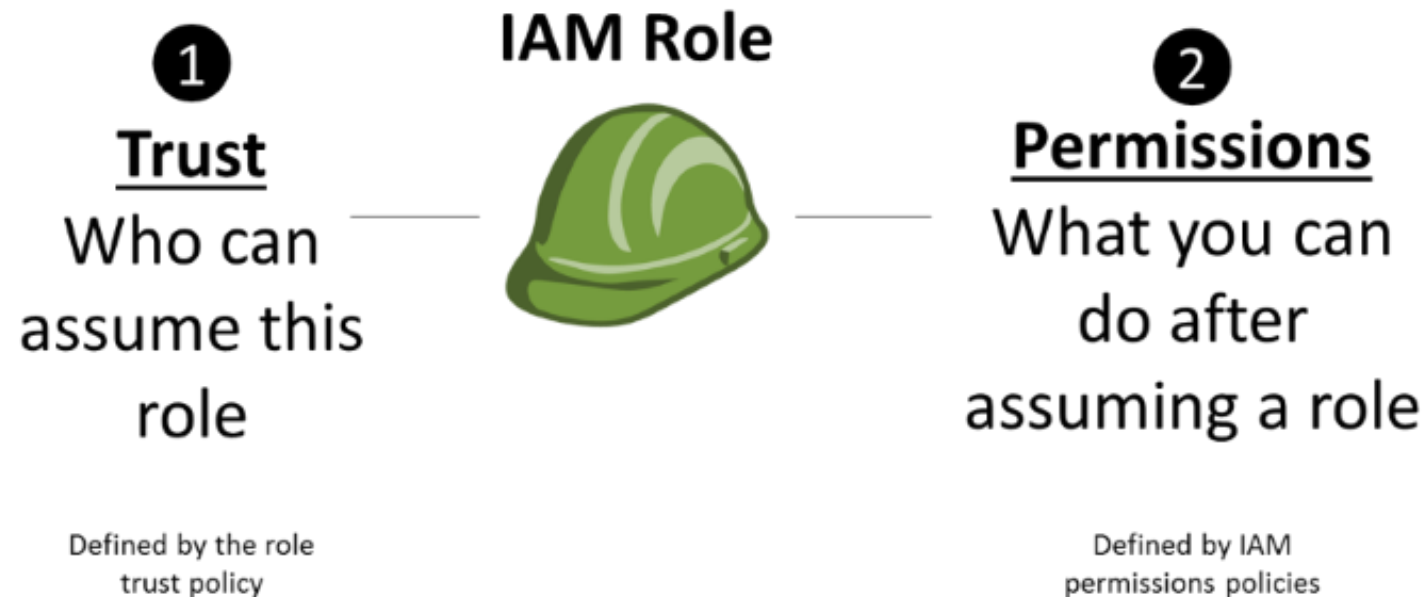
```
"Condition" : { "{condition-operator}" : { "{condition-key}" : "{condition-value}" }}
```

```
"Condition" : { "StringEqualsIgnoreCase" : { "aws:username" : "johndoe" }}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "rds:DescribeGlobalClusters"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:RebootDBInstance",
        "rds:StartDBInstance",
        "rds:StopDBInstance"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/Department": "DBAdmins",
          "rds:db-tag/Environment": "Production"
        }
      }
    }
  ]
}
```


IAM role

IAM role is similar to IAM user but you assign that to a resource, not a developer. There is a trust policy that defines who can assume (use) that role. It is similar to IAM user in a way that we assign permission policies to it.



Trust relationships in AWS console

In this example, support.amazonaws.com (AWS's support team member) can assume the AWSServiceRoleForSupport role in my account. There is no conditions. If you check the permission policies, then you will find the AWS support can do in your account.

[Roles](#) > [AWSServiceRoleForSupport](#)

Summary

This service-linked role cannot be deleted in IAM. [Learn more](#)

Role ARN	arn:aws:iam:::role/aws-service-role/support.amazonaws.com/AWSServiceRoleForSupport
Role description	Enables resource access for AWS to provide billing, administrative and support services Edit
Instance Profile ARNs	
Path	/aws-service-role/support.amazonaws.com/
Creation time	2021-05-23 21:53 EST
Last activity	Not accessed in the tracking period

Permissions

Trust relationships

Tags

Access Advisor

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

Trusted entities

The following trusted entities can assume this role.

Trusted entities

The identity provider(s) support.amazonaws.com

Conditions

The following conditions define how and when trusted entities can assume the role.

There are no conditions associated with this role.

AWS Security Token Service (AWS STS)

Under the hood, tokens are generated and used to access AWS services. STS is a web service that enables you to request **temporary** credentials for IAM role.

The temporary credentials consist of:

1. **Access key ID** - Access keys are long-term credentials for an IAM user. Like username.
2. **Secret access key** - Like password.
3. **Session token** - Validates temporary credentials.
4. **Duration** - defines how long the temporary credentials lasts. Most cases 12 hours. 15-min is min. You will not see these in the permanent tokens for users.

STS - AssumeRole

The underlying service makes an implicit “AssumeRole” call to STS on behalf of the resource and fetch the temporary tokens. It all happens under the hood. For instance, when fetching images from S3 in EC2, EC2 makes the AssumeRole call to STS, then receives the token and provides the token to S3. You can also make the AssumeRole call programmatically to retrieve temporary tokens for the IAM role.

```
https://sts.amazonaws.com/  
?Version=2011-06-15  
&Action=AssumeRole  
&RoleSessionName=testAR  
&RoleArn=arn:aws:iam::123456789012:role/demo  
&PolicyArns.member.1.arn=arn:aws:iam::123456789012:policy/demopolicy1
```

IAM User vs Role

IAM User	IAM Role
IAM entity assigned to a person .	IAM entity assigned to a service . It has a trust policy that specifies what services can use the role.
Tokens are permanent. It is on you to rotate that regularly.	Tokens are temporary. Tokens are generated by AWS STS.

Identity federation

Identity federation grants external identities secure access to resources in your AWS account. These external identities can come from your corporate identity provider such as Microsoft Active Directory.

Federated users (external identities) are users you manage outside of AWS in your corporate directory, but to whom you grant access to your AWS account using **temporary security credentials**.

Federated users and temporary security credentials STS

