# Amazon EC2

## General

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

With Amazon EC2 you launch virtual server instances on the AWS cloud.

Each virtual server is known as an “instance”.

You are limited to running up to a total of 20 On-Demand instances across the instance family, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic spot limit per region (by default).

AWS are transitioning to a vCPU based, rather than instance based, limit. This is currently being rolled out and may not feature on the exam yet.

Amazon EC2 currently supports a variety of operating systems including: Amazon Linux, Ubuntu, Windows Server, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Fedora, Debian, CentOS, Gentoo Linux, Oracle Linux, and FreeBSD.

EC2 compute units (ECU) provide the relative measure of the integer processing power of an Amazon EC2 instance.

With EC2 you have full control at the operating system layer.

Key pairs are used to securely connect to EC2 instances:

- A key pair consists of a **public key** that AWS stores, and a **private key file** that you store.
- For Windows AMIs, the private key file is required to obtain the password used to log into your instance.
- For Linux AMIs, the private key file allows you to securely SSH (secure shell) into your instance.

Metadata and User Data:

- User data is data that is supplied by the user at instance launch in the form of a script.
- Instance metadata is data about your instance that you can use to configure or manage the running instance.
- User data is limited to 16KB.
- User data and metadata are not encrypted.
- Instance metadata is available at http://169.254.169.254/latest/meta-data/ (the trailing “/” is required).
- Instance user data is available at: http://169.254.169.254/latest/user-data.
- The IP address 169.254.169.254 is a link-local address and is valid only from the instance.

- On Linux you can use the curl command to view metadata and userdata, e.g. “curl http://169.254.169.254/latest/meta-data/”.
- The Instance Metadata Query tool allows you to query the instance metadata without having to type out the full URI or category names.

# Billing and provisioning

## On demand:

- Pay for hours used with no commitment.
- Low cost and flexibility with no upfront cost.
- Ideal for auto scaling groups and unpredictable workloads.
- Good for dev/test.

## Spot:

- Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud.
- Spot Instances are available at up to a 90% discount compared to On-Demand prices.
- You can use Spot Instances for various stateless, fault-tolerant, or flexible applications such as big data, containerized workloads, CI/CD, web servers, high-performance computing (HPC), and other test & development workloads.
- You can request Spot Instances by using the Spot management console, CLI, API or the same interface that is used for launching On-Demand instances by indicating the option to use Spot.
- You can also select a Launch Template or a pre-configured or custom Amazon Machine Image (AMI), configure security and network access to your Spot instance, choose from multiple instance types and locations, use static IP endpoints, and attach persistent block storage to your Spot instances.
- **New pricing model:** The Spot price is determined by long term trends in supply and demand for EC2 spare capacity.
- You don’t have to bid for Spot Instances in the new pricing model, and you just pay the Spot price that’s in effect for the current hour for the instances that you launch.
- Spot Instances receive a two-minute interruption notice when these instances are about to be reclaimed by EC2, because EC2 needs the capacity back.
- Instances are not interrupted because of higher competing bids.
- To reduce the impact of interruptions and optimize Spot Instances, diversify and run your application across multiple capacity pools.
- Each instance family, each instance size, in each Availability Zone, in every Region is a separate Spot pool.
- You can use the RequestSpotFleet API operation to launch thousands of Spot Instances and diversify resources automatically.
- To further reduce the impact of interruptions, you can also set up Spot Instances and Spot Fleets to respond to an interruption notice by stopping or hibernating rather than terminating instances when capacity is no longer available.

## Reserved:

- Purchase (or agree to purchase) usage of EC2 instances in advance for significant discounts over On-Demand pricing.
- Provides a capacity reservation when used in a specific AZ.
- AWS Billing automatically applies discounted rates when you launch an instance that matches your purchased RI.
- Capacity is reserved for a term of 1 or 3 years.
- EC2 has three RI types: Standard, Convertible, and Scheduled.
- Standard = commitment of 1 or 3 years, charged whether it's on or off.
- Scheduled = reserved for specific periods of time, accrue charges hourly, billed in monthly increments over the term (1 year).
- Scheduled RIs match your capacity reservation to a predictable recurring schedule.
- For the differences between standard and convertible RIs, see the table below.
- RIs are used for steady state workloads and predictable usage.
- Ideal for applications that need reserved capacity.
- Upfront payments can reduce the hourly rate.
- Can switch AZ within the same region.
- Can change the instance size within the same instance type.
- Instance type modifications are supported for Linux only.
- Cannot change the instance size of Windows RIs.
- Billed whether running or not.
- Can sell reservations on the AWS marketplace.
- Can be used in Auto Scaling Groups.
- Can be used in Placement Groups.
- Can be shared across multiple accounts within Consolidated Billing.
- If you don't need your RI's, you can try to sell them on the Reserved Instance Marketplace.

| | Standard | Convertible |
|---|---|---|
| Terms | 1 year, 3 year | 1 year, 3 year |
| Average discount off On-Demand price | 40% - 60% | 31% - 54% |
| Change AZ, instance size, networking type | Yes, via ModifyReservedInstance API or console | Yes, via ExchangeReservedInstance API or console |
| Change instance family, OS, tenancy, payment options | No | Yes |
| Benefit from price reductions | No | Yes |

RI Attributes:

- Instance type – designates CPU, memory, networking capability.
- Platform – Linux, SUSE Linux, RHEL, Microsoft Windows, Microsoft SQL Server.
- Tenancy – Default (shared) tenancy, or Dedicated tenancy.
- Availability Zone (optional) – if AZ is selected, RI is reserved and discount applies to that AZ (Zonal RI). If no AZ is specified, no reservation is created but the discount is applied to any instance in the family in any AZ in the region (Regional RI).

## Comparing Amazon EC2 Pricing Models

The following table provides a brief comparison of On-demand, Reserved and Spot pricing models:

| On-Demand | Reserved | Spot |
|---|---|---|
| No upfront fee | Options: No upfront, partial upfront or all upfront | No upfront fee |
| Charged by hour or second | Charged by hour or second | Charged by hour or second |
| No commitment | 1-year or 3-year commitment | No commitment |
| Ideal for short term needs or unpredictable workloads | Ideal for steady-state workloads and predictable usage | Ideal for cost-sensitive, compute intensive use cases that can withstand interruption |

## Dedicated hosts:

- Physical servers dedicated just for your use.
- You then have control over which instances are deployed on that host.
- Available as On-Demand or with Dedicated Host Reservation.
- Useful if you have server-bound software licences that use metrics like per-core, per-socket, or per-VM.
- Each dedicated host can only run one EC2 instance size and type.
- Good for regulatory compliance or licensing requirements.
- Predictable performance.
- Complete isolation.
- Most expensive option.
- Billing is per host.

## Dedicated instances:

- Virtualized instances on hardware just for you.
- Also uses physically dedicated EC2 servers.
- Does not provide the additional visibility and controls of dedicated hosts (e.g. how instance are placed on a server).
- Billing is per instance.
- May share hardware with other non-dedicated instances in the same account.
- Available as On-Demand, Reserved Instances, and Spot Instances.
- Cost additional $2 per hour per region.

The following table describes some of the differences between dedicates instances and dedicated hosts:

| Characteristic | Dedicated Instances |
| --- | --- |
| Enables the use of dedicated physical servers | X |
| Per instance billing (subject to a $2 per region fee) | X |
| Per host billing | |
| Visibility of sockets, cores, host ID | |
| Affinity between a host and instance | |
| Targeted instance placement | |
| Automatic instance placement | X |
| Add capacity using an allocation request | |

Partial instance-hours consumed are billed based on instance usage.

Instances are billed when they're in a running state – need to stop or terminate to avoid paying.

Charging by the hour or second (by the second with Linux instances only).

Data between instances in different regions is charged (in and out).

Regional Data Transfer rates apply if at least one of the following is true, but are only charged once for a given instance even if both are true:

- The other instance is in a different Availability Zone, regardless of which type of address is used.
- Public or Elastic IP addresses are used, regardless of which Availability Zone the other instance is in.

# Instance types

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases.

Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications.

Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

| Category | Families | Purpose/Design |
|---|---|---|
| General Purpose | A1, T3, T3a, T2, M5, M5a, M4 | General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads |
| Compute Optimized | C5, C5n, C4 | Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors |
| Memory Optimized | R5, R5a, R4, X1e, X1, High Memory, z1d | Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory |
| Accelerated Computing | P3, P2, G4, G3, F1 | Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating-point number calculations, graphics processing, or data pattern matching |
| Storage Optimized | I3, I3en, D2, H1 | This instance family provides Non-Volatile Memory Express (NVMe) SSD-backed instance storage optimized for low latency, very high random I/O performance, high sequential read throughput and provide high IOPS at a low cost |

## Options when Launching Instances

Choose whether to auto-assign a public IP – default is to use the subnet setting.

Can add an instance to a placement group.

Instances can be assigned to IAM roles which configures them with credentials to access AWS resources.

Termination protection can be enabled and prevents you from terminating an instance.

Basic monitoring is enabled by default (5 minute periods), detailed monitoring can be enabled (1 minute periods, chargeable).

Can define shared or dedicated tenancy.

T2 unlimited allows applications to burst past CPU performance baselines as required (chargeable).

Can add a script to run on startup (user data).

Can join to a directory (Windows instances only).

There is an option to enable an Elastic GPU (Windows instances only).

Storage options include adding additional volumes and choosing the volume type.

Non-root volumes can be encrypted.

Root volumes can be encrypted if the instance is launched from an encrypted AMI.

There is an option to create tags (or can be done later).

You can select an existing security group or create a new one.

You must create or use an existing key pair – this is required.

# Amazon Machine Images

An Amazon Machine Image (AMI) provides the information required to launch an instance.

An AMI includes the following:

- A template for the root volume for the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

AMIs are regional. You can only launch an AMI from the region in which it is stored. However, you can copy AMI's to other regions using the console, command line, or the API.

Volumes attached to the instance are either EBS or Instance store:

- Amazon Elastic Block Store (EBS) provides persistent storage. EBS snapshots, which reside on Amazon S3, are used to create the volume.
- Instance store volumes are ephemeral (non-persistent). That means data is lost if the instance is shut down. A template stored on Amazon S3 is used to create the volume.

# Networking

Networking Limits (per region or as specified):

| Name | Default L |
| --- | --- |
| EC2-Classic Elastic IPs | 5 |
| EC2-VPC Elastic IPs | 5 |
| VPCs | 5 |
| Subnets per VPC | 200 |
| Security groups per VPC | 500 |
| Rules per VPC security group | 50 |
| VPC security groups per elastic network interface | 5 |
| Network interfaces | 350 |
| Network ACLs per VPC | 200 |
| Rules per network ACL | 20 |
| Route tables per VPC | 200 |
| Entries per route table | 50 |
| Active VPC peering connections | 50 |
| Outstanding VPC peering connection requests | 25 |
| Expiry time for an unaccepted VPC peering connection | 168 |

**IP Addresses**

There are three types of IP address that can be assigned to an Amazon EC2 instance:

- Public – public address that is assigned automatically to instances in public subnets and reassigned if instance is stopped/started.
- Private – private address assigned automatically to all instances.
- Elastic IP – public address that is static.

Public IPv4 addresses are lost when the instance is stopped but private addresses (IPv4 and IPv6) are retained.

Public IPv4 addresses are retained if you restart the instance.

Elastic IPs are retained when the instance is stopped.

Elastic IP addresses are static public IP addresses that can be remapped (moved) between instances.

All accounts are limited to 5 elastic IP's per region by default.

AWS charge for elastic IP's when they're not being used.

An Elastic IP address is for use in a specific region only.

You can assign custom tags to your Elastic IP addresses to categorize them.

By default, EC2 instances come with a private IP assigned to the primary network interface (eth0).

Public IP addresses are assigned for instances in public subnets (VPC).

Public IP addresses are always assigned for instances in EC2-Classic.

DNS records for elastic IP's can be configured by filling out a form.

Secondary IP addresses can be useful for hosting multiple websites on a server or redirecting traffic to a standby EC2 instance for HA.

You can choose whether secondary IP addresses can be reassigned.

You can associate a single private IPv4 address with a single Elastic IP address and vice versa.

When reassigned the IPv4 to Elastic IP association is maintained.

When a secondary private address is unassigned from an interface, the associated Elastic IP address is disassociated.

You can assign or remove IP addresses from EC2 instances while they are running or stopped.

All IP addresses (IPv4 and IPv6) remain attached to the network interface when detached or reassigned to another instance.

You can attach a network interface to an instance in a different subnet as long as its within the same AZ.

The following table compares the different types of IP address available in Amazon EC2:

| Name | Description |
| --- | --- |
| Public IP address | Lost when the instance is stopped<br>Used in Public Subnets<br>No charge<br>Associated with a private IP address on the instance<br>Cannot be moved between instances |
| Private IP address | Retained when the instance is stopped<br>Used in Public and Private Subnets |
| Elastic IP address | Static Public IP address<br>You are charged if not used<br>Associated with a private IP address on the instance<br>Can be moved between instances and Elastic Network Adapters |

## Elastic Network Interfaces

An elastic network interface (referred to as a network interface in this documentation) is a logical networking component in a VPC that represents a virtual network card.

A network interface can include the following attributes:

- A primary private IPv4 address from the IPv4 address range of your VPC.
- One or more secondary private IPv4 addresses from the IPv4 address range of your VPC.
- One Elastic IP address (IPv4) per private IPv4 address.
- One public IPv4 address.
- One or more IPv6 addresses.
- One or more security groups.
- A MAC address.
- A source/destination check flag.
- A description.

You can create and configure network interfaces in your account and attach them to instances in your VPC.

You cannot team by adding ENIs to an instance.

eth0 is the primary network interface and cannot be moved or detached.

By default, eth0 is the only Elastic Network Interface (ENI) created with an EC2 instance when launched.

You can add additional interfaces to EC2 instances (number dependent on instances family/type).

An ENI is bound to an AZ and you can specify which subnet/AZ you want the ENI to be added in.

You can specify which IP address within the subnet to configure or leave it be auto-assigned.

You can only add one extra ENI when launching but more can be attached later.

ENIs can be “hot attached” to running instances.

ENIs can be “warm-attached” when the instance is stopped.

ENIs can be “cold-attached” when the instance is launched.

If you add a second interface AWS will not assign a public IP address to eth0 (you would need to add an Elastic IP).

Default interfaces are terminated with instance termination.

Manually added interfaces are not terminated by default.

You can change the termination behaviour.

## Enhanced Networking – Elastic Network Adapter (ENA)

Enhanced networking provides higher bandwidth, higher packet-per-second (PPS) performance, and consistently lower inter-instance latencies.

Enhanced networking is enabled using an Elastic Network Adapter (ENA).

If your packets-per-second rate appears to have reached its ceiling, you should consider moving to enhanced networking because you have likely reached the upper thresholds of the VIF driver.

AWS currently supports enhanced networking capabilities using SR-IOV.

SR-IOV provides direct access to network adapters, provides higher performance (packets-per-second) and lower latency.

Must launch an HVM AMI with the appropriate drivers.

Only available for certain instance types.

Only supported in VPC.

## Elastic Fabric Adapter (EFA)

An Elastic Fabric Adapter is an AWS Elastic Network Adapter (ENA) with added capabilities.

An EFA can still handle IP traffic, but also supports an important access model commonly called OS bypass.

This model allows the application (most commonly through some user-space middleware) access the network interface without having to get the operating system involved with each message.

Elastic Fabric Adapter (EFA) is a network interface for Amazon EC2 instances that enables customers to run applications requiring high levels of inter-node communications at scale on AWS.

Its custom-built operating system (OS) bypass hardware interface enhances the performance of inter-instance communications, which is critical to scaling these applications.

With EFA, High Performance Computing (HPC) applications using the Message Passing Interface (MPI) and Machine Learning (ML) applications using NVIDIA Collective Communications Library (NCCL) can scale to thousands of CPUs or GPUs.

As a result, you get the application performance of on-premises HPC clusters with the on-demand elasticity and flexibility of the AWS cloud.

EFA is available as an optional EC2 networking feature that you can enable on any supported EC2 instance at no additional cost.

## ENI vs ENA vs EFA

When to use ENI:

- This is the basic adapter type for when you don’t have any high performance requirements.
- Can use with all instance types.

When to use ENA:

- Good for use cases that require higher bandwidth and lower inter-instance latency.
- Supported for limited instance types (HVM only).

When to use EFA:

- High Performance Computing.
- MPI and ML use cases.
- Tightly coupled applications.
- Can use with all instance types.

## Placement Groups

Placement groups are a logical grouping of instances in one of the following configurations.

Cluster – clusters instances into a low-latency group in a single AZ:

- A cluster placement group is a logical grouping of instances within a single Availability Zone.
- Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both, and if the majority of the network traffic is between the instances in the group.

Spread – spreads instances across underlying hardware (can span AZs):

- A spread placement group is a group of instances that are each placed on distinct underlying hardware.
- Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other.

Partition — divides each group into logical segments called partitions:

- Amazon EC2 ensures that each partition within a placement group has its own set of racks.
- Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.
- Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks.

The table below describes some key differences between clustered and spread placement groups:

| | Clustered | Spread | Partition |
|---|---|---|---|
| What | Instances are placed into a low-latency group within a single AZ | Instances are spread across underlying hardware | Instances are grouped into logical segments called partitions which use distinct hardware |
| When | Need low network latency and/or high network throughput | Reduce the risk of simultaneous instance failure if underlying hardware fails | Need control and visibility into instance placement |
| Pros | Get the most out of enhanced networking Instances | Can span multiple AZs | Reduces likelihood of correlated failures for large workloads. |
| Cons | Finite capacity: recommend launching all you might need up front | Maximum of 7 instances running per group, per AZ | Partition placement groups are not supported for Dedicated Hosts |

Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same underlying hardware.

Recommended for applications that benefit from low latency and high bandwidth.

Recommended to use an instance type that supports enhanced networking.

Instances within a placement group can communicate with each other using private or public IP addresses.

Best performance is achieved when using private IP addresses.

Using public IP addresses the performance is limited to 5Gbps or less.

Low-latency 10 Gbps or 25 Gbps network.

Recommended to keep instance types homogenous within a placement group.

Can use reserved instances at an instance level but cannot reserve capacity for the placement group.

The name you specify for a placement group must be unique within your AWS account for the Region.

You can't merge placement groups.

An instance can be launched in one placement group at a time; it cannot span multiple placement groups.

On-Demand Capacity Reservation and zonal Reserved Instances provide a capacity reservation for EC2 instances in a specific Availability Zone. The capacity reservation can be used by instances in a placement group. However, it is not possible to explicitly reserve capacity for a placement group.

Instances with a tenancy of host cannot be launched in placement groups.

# IAM Roles

IAM roles are more secure than storing access keys and secret access keys on EC2 instances.

IAM roles are easier to manage.

You can attach an IAM role to an instance at launch time or at any time after by using the AWS CLI, SDK, or the EC2 console.

IAM roles can be attached, modified, or replaced at any time.

Only one IAM role can be attached to an EC2 instance at a time.

IAM roles are universal and can be used in any region.

# Bastion/Jump Hosts

You can configure EC2 instances as bastion hosts (aka jump boxes) in order to access your VPC instances for management.

Can use the SSH or RDP protocols to connect to your bastion host.

Need to configure a security group with the relevant permissions.

Can use auto-assigned public IPs or Elastic IPs.

Can use security groups to restrict the IP addresses/CIDRs that can access the bastion host.

Use auto-scaling groups for HA (set to 1 instance to just replace if it fails).

Best practice is to deploy Linux bastion hosts in two AZs, use auto-scaling and Elastic IP addresses.

# EC2 Migration

VM Import/Export is a tool for migrating VMware, Microsoft, XEN VMs to the Cloud.

Can also be used to convert EC2 instances to VMware, Microsoft or XEN VMs.

Supported for:

- Windows and Linux.
- VMware ESX VMDKs and (OVA images for export only).
- Citrix XEN VHD.
- Microsoft Hyper-V VHD.

Can only be used via the API or CLI (not the console).

Stop the VM before generating VMDK or VHD images.

AWS has a VM connector plugin for vCenter:

- Allows migration of VMs to S3.
- Then converts into a EC2 AMI.
- Progress can be tracked in vCenter.

# Monitoring

EC2 status checks are performed every minute and each returns a pass or a fail status.

If all checks pass, the overall status of the instance is **OK.**

If one or more checks fail, the overall status is **impaired.**

System status checks detect (StatusCheckFailed_System) problems with your instance that require **AWS** involvement to repair.

Instance status checks (StatusCheckFailed_Instance) detect problems that require **your** involvement to repair.

Status checks are built into Amazon EC2, so they cannot be disabled or deleted.

You can, however create or delete alarms that are triggered based on the result of the status checks.

You can create Amazon CloudWatch alarms that monitor Amazon EC2 instances and automatically perform an action if the status check fails.

Actions can include:

- Recover the instance (only supported on specific instance types and can be used only with StatusCheckFailed_System).
- Stop the instance (only applicable to EBS-backed volumes).
- Terminate the instance (cannot terminate if termination protection is enabled).
- Reboot the instance.

It is a best practice to use EC2 to reboot instance rather than the OS (create a CloudWatch record).

CloudWatch Monitoring frequency:

- Standard monitoring = 5 mins.
- Detailed monitoring = 1 min (chargeable).

# Tags

A tag is a label that you assign to an AWS resource.

Used to manage AWS assets.

Tags are just arbitrary name/value pairs that you can assign to virtually all AWS assets to serve as metadata.

Each tag consists of a key and an optional value, both of which you define.

Tagging strategies can be used for cost allocation, security, automation, and many other uses. For example, you can use a tag in an IAM policy to implement access control.

Enforcing standardized tagging can be done via AWS Config rules or custom scripts. For example, EC2 instances not properly tagged are stopped or terminated daily.

Most resources can have up to 50 tags.

# Resource Groups

Resource groups are mappings of AWS assets defined by tags.

Create custom consoles to consolidate metrics, alarms and config details around given tags.

# High Availability Approaches For Compute

Up-to-date AMIs are critical for rapid fail-over.

AMIs can be copied to other regions for safety or DR staging.

Horizontally scalable architectures are preferred because risk can be spread across multiple smaller machines versus one large machine.

Reserved instances are the only way to guarantee that resources will be available when needed.

Auto Scaling and Elastic Load Balancing work together to provide automated recovery by maintaining minimum instances.

Route 53 health checks also provide “self-healing” redirection of traffic.

# Migration

AWS Server Migration Service (SMS) is an agent-less service which makes it easier and faster for you to migrate thousands of on-premises workloads to AWS.

AWS SMS allows you to automate, schedule, and track incremental replications of live server volumes, making it easier for you to coordinate large-scale server migrations.

Automates migration of on-premises VMware vSphere or Microsoft Hyper-V/SCVMM virtual machines to AWS.

Replicates VMs to AWS, syncing volumes and creating periodic AMIs.

Minimizes cutover downtime by syncing VMs incrementally.

Supports Windows and Linux VMs only (just like AWS).

The Server Migration Connector is downloaded as a virtual appliance into your on-premises vSphere or Hyper-V environments.