AWS Solution Architect Associate Certification Training – Module 6

. Compute Services

Introduction and Features of EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. 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 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual Computing environments, known as instances.
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IPv4 addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds* (VPCs).

Benefits:

Elastic Web-Scale Computing: Amazon EC2 enables you to increase or decrease capacity within minutes, not hours or days. You can commission one, hundreds or even thousands of server instances simultaneously. Of course, because this is all controlled with web services APIs, your application can automatically scale itself up and down depending on its needs.

Completely Controlled: You have complete control of your instances. You have root access to each one, and you can interact with them as you would any machine. You can stop your instance while retaining the data on your boot partition and then subsequently restart the same instance using web

service APIs. Instances can be rebooted remotely using web service APIs. You also have access to console output of your instance.

Flexible Cloud Hosting Services: You have the choice of multiple instance types, operating systems, and software packages. Amazon EC2 allows you to select a configuration of memory, CPU, Instance storage, and boot partition size that is optimal for your choice of operating system and application. For example, your choice of operating systems includes numerous Linux distributions, and Microsoft Windows Server.

Designed for use with other Amazon Web Services: Amazon EC2 works in conjunction with Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS) and Amazon Simple Queue Service (Amazon SQS) to provide a complete solution for computing, query processing and storage across a wide range of applications.

Reliable: Amazon EC2 offers a highly reliable environment where replacement instances can be rapidly and predictably commissioned. The service runs within Amazon's proven network infrastructure and datacenters.

Secure: Amazon EC2 works in conjunction with Amazon VPC to provide security and robust networking functionality for your compute resources. Your compute instances are located in a Virtual Private Cloud (VPC) with an IP range that you specify. You decide which instances are exposed to the Internet and which remain private.

- Security Groups and networks ACLs allow you to control inbound and outbound network access to and from your instances.
- You can provision your EC2 resources as Dedicated Instances. Dedicated Instances are Amazon EC2 Instances that run on hardware dedicated to a single customer for additional isolation.
- If you do not have a default VPC you must create a VPC and launch instances into that VPC to leverage advanced networking features such as private subnets, outbound security group filtering, network ACLs and Dedicated Instances.

Inexpensive: Amazon EC2 passes on to you the financial benefits of Amazon's scale. You pay a very low rate for the compute capacity you actually consume.

Easy to start: Quickly get started with Amazon EC2 by visiting the AWS Management Console to choose preconfigured software on Amazon Machine Images (AMIs). You can quickly deploy this software to EC2 via the EC2 console.

Amazon Machine Images

An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You must specify a source AMI when you launch a instance. You can launch multiple instances from a single AMI when you need multiple

instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

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

Working with AMI – Public and private images

Amazon EC2 enables you to share your AMIs with other AWS accounts. You can allow all AWS accounts to launch the AMI (make the AMI public), or only allow a few specific accounts to launch the AMI. You are not billed when your AMI is launched by other AWS accounts; only the accounts launching the AMI are billed.

AMIs are a regional resource. Therefore, sharing an AMI makes it available in that region. To make an AMI available in a different region, copy the AMI to the region and then share it.

Sharing an AMI with all AWS accounts

After you make an AMI public, it is available in Community AMIs when you launch an instance in the same region using the console. Note that it can take a short while for an AMI to appear in Community AMIs after you make it public. It can also take a short while for an AMI to be removed from Community AMIs after you make it private again.

- Open the Amazon EC2 console
- In the navigation pane, choose AMIs.
- Select your AMI from the list, and then choose Actions, Modify Image Permissions.
- Choose Public and choose Save.

CPU and Processors of AWS Cloud Platform

Amazon Web Services (AWS) and Intel share a passion for delivering constant innovation. Together, they have developed a variety of resources and technologies for High Performance Computing, Big Data, Artificial Intelligence/Machine Learning and the Internet of Things.

Intel Processor Overview

Intel® Xeon® Scalable Processor Family

Intel® Platinum Xeon Scalable processor families are the foundation of new services being deployed by AWS. AWS instances based on Intel® processors are ready to serve unique and innovative new workloads that demand better data protection, faster processing of greater data volumes, and service flexibility without a hit to performance. These processors feature:

Intel® Advanced Vector Extension 512 (Intel® AVX-512) which offers accelerated application performance 2x better than previous generation technologies, enabling significant improvements in workload speed and data application.

Intel® Trusted Execution Technology (Intel® TXT) which remains Intel's technology for establishing more secure platforms. With Intel® One Touch Activation comes an added level of protection for geographic needs like regional and county-specific data sovereignty regulations.

Intel® Xeon® E7 Processor Family

Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) allow you to enable encryption for enhanced data security without paying a performance penalty.

Intel® Advanced Vector Extensions 2 (Intel® AVX2) can double the floating-point performance for compute-intensive workloads over Intel® AVX, and it provides additional instructions useful for compression and encryption.

Intel® Transactional Synchronization Extensions (Intel® TSX) enabling faster application execution time, boosting the performance of in-memory transactional data processing.

Intel® Xeon® E5 Processor Family

Intel® Xeon® processor E5 family, based on the Haswell microarchitecture, has better branch prediction--making it efficient at prefetching instructions and data--along with other improvements that can boost existing applications' performance by 30% or more.

P state and C state control provides the ability to individually tune each core's performance and sleep states to improve application performance.

Intel® AVX2 can double the floating-point performance for compute-intensive workloads over Intel® AVX, and provides additional instructions useful for compression and encryption.

Machine type classification in EC2 and use cases of each machine type

Instance Types

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 capabilities and are grouped in instance families based on these

capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance.

Amazon EC2 provides each instance with a consistent and predictable amount of CPU capacity, regardless of its underlying hardware.

Amazon EC2 dedicates some resources of the host computer, such as CPU, memory, and instance storage, to a particular instance. Amazon EC2 shares other resources of the host computer, such as the network and the disk subsystem, among instances. If each instance on a host computer tries to use as much of one of these shared resources as possible, each receives an equal share of that resource. However, when a resource is underused, an instance can consume a higher share of that resource while it's available.

Each instance type provides higher or lower minimum performance from a shared resource. For example, instance types with high I/O performance have a larger allocation of shared resources. Allocating a larger share of shared resources also reduces the variance of I/O performance. For most applications, moderate I/O performance is more than enough. However, for applications that require greater or more consistent I/O performance, consider an instance type with higher I/O performance.

Current Generation Instances:

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.

Instance Family	Current Generation Instance Types
General purpose	a1.medium | a1.large | a1.xlarge | a1.2xlarge | a1.4xlarge | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge | m5.large | m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5.metal | m5a.large | m5a.xlarge | m5a.2xlarge | m5a.4xlarge | m5a.12xlarge | m5a.24xlarge | m5d.large | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge | m5d.metal | t2.nano | t2.micro | t2.small | t2.medium | t2.large | t2.xlarge | t2.2xlarge | t3.nano | t3.micro | t3.small | t3.medium | t3.large | t3.xlarge | t3.2xlarge
Compute optimized	c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c5.large | c5.xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.18xlarge | c5d.xlarge | c5d.2xlarge | c5d.4xlarge | c5d.9xlarge | c5d.18xlarge | c5n.large | c5n.xlarge | c5n.2xlarge | c5n.4xlarge | c5n.9xlarge | c5n.18xlarge
Memory optimized	r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge | r5.large | r5.xlarge | r5.2xlarge | r5.4xlarge | r5.12xlarge | r5.24xlarge | r5.metal | r5a.large | r5a.xlarge | r5a.2xlarge | r5a.4xlarge | r5a.12xlarge | r5a.24xlarge | r5d.large | r5d.xlarge | r5d.2xlarge | r5d.4xlarge | r5d.12xlarge | r5d.24xlarge | r5d.metal | u-6tb1.metal | u-9tb1.metal | u-12tb1.metal | x1.16xlarge | x1.32xlarge | x1e.xlarge | x1e.2xlarge | x1e.4xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge | z1d.large | z1d.xlarge | z1d.2xlarge | z1d.3xlarge | z1d.6xlarge | z1d.12xlarge | z1d.metal
Storage optimized	d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | h1.2xlarge | h1.4xlarge | h1.8xlarge | h1.16xlarge | i3.large | i3.xlarge | i3.2xlarge | i3.4xlarge | i3.8xlarge | i3.16xlarge | i3.metal
Accelerated computing	f1.2xlarge | f1.4xlarge | f1.16xlarge | g3s.xlarge | g3.4xlarge | g3.8xlarge | g3.16xlarge | p2.xlarge | p2.8xlarge | p2.16xlarge | p3.2xlarge | p3.8xlarge | p3.16xlarge | p3dn.24xlarge

Previous Generation Instances:

Amazon Web Services offers previous generation instances for users who have optimized their applications around these instances and have yet to upgrade. AWS encourage you to use the latest generation of instances to get the best performance, but AWS continue to support these previous generation instances. If you are currently using a previous generation instance, you can see which current generation instance would be a suitable upgrade.

Instance Family	Previous Generation Instance Types
General purpose	m1.small | m1.medium | m1.large | m1.xlarge | m3.medium | m3.large | m3.xlarge | m3.2xlarge
Compute optimized	c1.medium | c1.xlarge | cc2.8xlarge | c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge
Memory optimized	m2.xlarge | m2.2xlarge | m2.4xlarge | cr1.8xlarge | r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge
Storage optimized	hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge
GPU optimized	g2.2xlarge | g2.8xlarge
Micro	t1.micro

The configuration of EC2 Instances – Placement Groups, Capacity reservation, Tenancy, Shutdown behavior, Metadata

Amazon EC2 Instance Configuration:

When you plan and configure EBS volumes for your application, it is important to consider the configuration of the instances that you will attach the volumes to. In order to get the most performance out of your EBS volumes, you should attach them to an instance with enough bandwidth to support your volumes, such as an EBS-optimized instance or an instance with 10 Gigabit network connectivity. This is especially important when you stripe multiple volumes together in a RAID configuration.

Placement Groups:

You can launch or start instances in a placement group, which determines how instances are placed on underlying hardware. When you create a placement group, you specify one of the following strategies for the group:

- ✓ *Cluster* – clusters instances into a low-latency group in a single Availability Zone
- ✓ *Partition* – spreads instances across logical partitions, ensuring that instances in one partition do not share underlying hardware with instances in other partitions
- ✓ *Spread* – spreads instances across underlying hardware

Cluster Placement groups: A cluster placement group is a logical grouping of instances within a single Availability Zone. A placement group can span peered VPCs in the same Region. The chief benefit of a cluster placement group, in addition to a 10 Gbps flow limit, is the non-blocking, non-oversubscribed, fully bi-sectional nature of the connectivity. In other words, all nodes within the placement group can talk to all other nodes within the placement group at the full line rate of 10 Gpbs flows and 25 aggregate without any slowing due to over-subscription.

The following image shows instances that are placed into a cluster placement group.

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. To provide the lowest latency and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see Enhanced Networking.

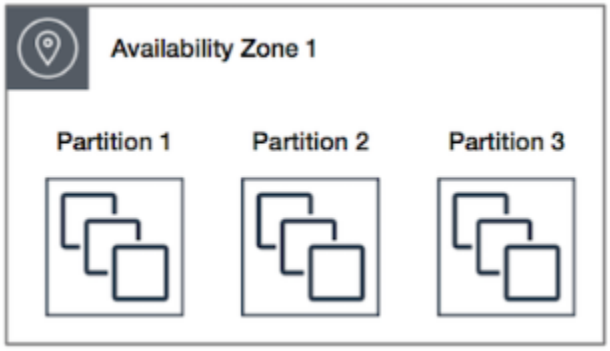
We recommend that you launch the number of instances that you need in the placement group in a single launch request and that you use the same instance type for all instances in the placement group. If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group that already has running instances, stop and start all of the instances in the placement group, and try the launch again. Restarting the instances may migrate them to hardware that has capacity for all the requested instances.

Partition Placement Groups: A partition placement group is a group of instances spread across partitions. Partitions are logical groupings of instances, where contained instances do not share the same underlying hardware across different partitions.

The following image shows instances in a single Availability Zone that are placed into a partition placement group with three partitions, Partition 1, Partition 2, and Partition 3. Each partition comprises multiple instances. The instances in each partition do not share underlying hardware with the instances in the other partitions, limiting the impact of hardware failure to only one partition.



Partition placement groups can be used to spread deployment of large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct hardware to reduce the likelihood of correlated failures. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.

In addition, partition placement groups offer visibility into the partitions—you can see which instances are in which partitions. You can share this information with topology-aware applications, such as HDFS, HBase, and Cassandra, which use this information to make intelligent data replication decisions for increasing data availability and durability.

A partition placement group can have a maximum of seven partitions per Availability Zone. The number of instances that can be launched into a partition placement group is limited only by the limits of your account. Partition placement groups can also span multiple Availability Zones in the same Region.

If you start or launch an instance in a partition placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

Spread Placement Groups: A spread placement group is a group of instances that are each placed on distinct underlying hardware. The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The instances do not share underlying hardware with each other.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same underlying hardware. Spread placement groups provide access to distinct hardware, and are therefore suitable for mixing instance types or launching instances over time.

A spread placement group can span multiple Availability Zones, and you can have a maximum of seven running instances per Availability Zone per group.

If you start or launch an instance in a spread placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

On Demand Capacity Reservations:

On-Demand Capacity Reservations enable you to reserve capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage capacity reservations independently from the billing discounts offered by Reserved Instances (RI). By creating Capacity Reservations, you ensure that you always have access to EC2 capacity when you need it, for as long as you need it. Capacity Reservations can be created at any time, without entering into a one-year or three-year term commitment, and the capacity is available immediately. When you no longer need the reservation, cancel the Capacity Reservation to stop incurring charges for it.

When you create a Capacity Reservation, you specify the Availability Zone in which you want to reserve the capacity, the number of instances for which you want to reserve capacity, and the instance attributes, including the instance type, tenancy, and platform/OS. Capacity Reservations can only be used by instances that match their attributes. By default, they are automatically used by running instances that match the attributes. If you don't have any running instances that match the attributes of the Capacity Reservation, it remains unused until you launch an instance with matching attributes.

In addition, you can use your Regional RIs with your Capacity Reservations to benefit from billing discounts. This gives you the flexibility to selectively add capacity reservations and still get the Regional RI discounts for that usage. AWS automatically applies your RI discount when the attributes of a Capacity Reservation match the attributes of an active Regional RI.

Differences between Capacity Reservations and RI's:

	Capacity Reservations	Zonal RIs	Regional RIs
Term	No commitment required. Can be created and cancelled as needed.	Require fixed one-year or three-year commitment.	
Capacity benefit	Reserve capacity in a specific Availability Zone.	Reserve capacity in a specific Availability Zone.	Do not reserve capacity in an Availability Zone.
Billing discount	No billing discount. Instances launched into a Capacity Reservation are charged at their standard On-Demand rates. However, Regional RIs can be used with Capacity Reservations to get a billing discount.	Provide billing discounts.	
Instance Limits	Limited to your On-Demand Instance limits per Region.	Limited to 20 per Availability Zone. A limit increase can be requested.	Limited to 20 per Region. A limit increase can be requested.

Capacity Reservations Limits: The number of instances for which you are allowed to reserve capacity is based on your account's On-Demand Instance limit. You can reserve capacity for as many instances as that limit allows, minus the number of instances that are already running.

Capacity Reservation Limitations and Restrictions:

Before you create Capacity Reservations, take note of the following limitations and restrictions.

- Active and unused Capacity Reservations count towards your On-Demand Instance limits
- Capacity Reservations can't be shared across AWS accounts
- Capacity Reservations are not transferable from one AWS account to another
- Zonal RI billing discounts do not apply to Capacity Reservations
- Capacity Reservations can't be created in Placement Groups
- Capacity Reservations can't be used with Dedicated Hosts

Dedicated Instances: Dedicated Instances are Amazon EC2 instances that run in a virtual private cloud (VPC) on hardware that's dedicated to a single customer. Dedicated Instances that belong to different AWS accounts are physically isolated at the hardware level. In addition, Dedicated Instances that belong to AWS accounts that are linked to a single payer account are also physically isolated at the hardware level. However, Dedicated Instances may share hardware with other instances from the same AWS account that are not Dedicated Instances.

Security groups

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch, you can specify one or more security groups otherwise, we use the default security

group. You can add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

Security Groups Rules:

The rules of a security group control the inbound traffic that's allowed to reach the instances that are associated with the security group and the outbound traffic that's allowed to leave them.

The following are the characteristics of security group rules:

- By default, security groups allow all outbound traffic.
- Security group rules are always permissive; you can't create rules that deny access.
- Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules.

Private IP, Public IP, Elastic IP

Elastic IP: An Elastic IP address is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

An Elastic IP address is a public IPv4 address, which is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet; for example, to connect to your instance from your local computer.

Private IP: When EC2 instances are launched, the primary elastic network interface is assigned a reserved private IP address from the default VPC DHCP pool.

The private IP address stays assigned to the network interface until it is deleted. The instance's primary network interface cannot be removed; it stays assigned to the instance until the instance is deleted. It is not possible to remove or change the private IP address of the primary network interface, but it is possible to add more private IP addresses to the network interface.

Public IP: A public IP address is an IPv4 address that's reachable from the Internet. You can use public addresses for communication between your instances and the Internet. Each instance that receives a public IP address is also given an external DNS hostname; for example,ec2-203-0-113-25.compute-1.amazonaws.com.

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 description

Network Interface Basics:

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

You can also modify the attributes of your network interface, including changing its security groups and managing its IP addresses.

Every instance in a VPC has a default network interface, called the primary network interface (eth0). You cannot detach a primary network interface from an instance. You can create and attach additional network interfaces. The maximum number of network interfaces that you can use varies by instance type.

IP Addresses per Network Interface per Instance Type

The following table lists the maximum number of network interfaces per instance type, and the maximum number of private IPv4 addresses and IPv6 addresses per network interface. The limit for IPv6 addresses is separate from the limit for private IPv4 addresses per network interface. Not all instance types support IPv6 addressing. Network interfaces, multiple private IPv4 addresses, and IPv6 addresses are only available for instances running in a VPC.

Instance Type	Maximum Network Interfaces	IPv4 Addresses per Interface	IPv6 Addresses per Interface
a1.medium	2	4	4
a1.large	3	10	10
a1.xlarge	4	15	15
a1.2xlarge	4	15	15
a1.4xlarge	8	30	30
c1.medium	2	6	IPv6 not supported
c1.xlarge	4	15	IPv6 not supported
c3.large	3	10	10
c3.xlarge	4	15	15
c3.2xlarge	4	15	15
c3.4xlarge	8	30	30
c3.8xlarge	8	30	30
c4.large	3	10	10
c4.xlarge	4	15	15
c4.2xlarge	4	15	15
c4.4xlarge	8	30	30

and so on...

Key pairs

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a key pair.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

Resource Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a *key* and an optional *value*, both of which you define. Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags you've assigned to it. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level.