Introduction to EC2

What is EC2 ?

Of all the services in the AWS family, EC2 is arguably the most common and most important service. In other words, it is the backbone of AWS.

EC2 stands for Amazon Elastic Compute Cloud.

EC2 is a web service that provides secure, resizable compute capacity in the cloud.

It is a service that allows you to run application programs in the computing environment.

It is designed to make web-scale cloud computing easier for developers.

EC2 Features&Advantages

- **Features of EC2:**
  - EC2 provides you to pay only for the resources that you actually use.
  - In the past, creating a server for our business sometimes takes days, weeks, but now within 2 seconds, we can create a virtual machine that suits our wishes.
  - If you use EC2 you don't have to buy physical servers. Instead, you select a server that has more CPU capacity, RAM capacity and you can buy a server for a 5-year term. So it means you can plan for 5 years in advance.
- **Advantage of EC2:**
  - **Elasticity:** Capacity needs can be arranged within minutes.
  - **Control:** You can create, stop or terminate instances via EC2 console, CLI or SDKs easily.
  - **Reliability:** EC2 Service Level Agreement of 99.99% is committed by Amazon.

EC2 Basic Components

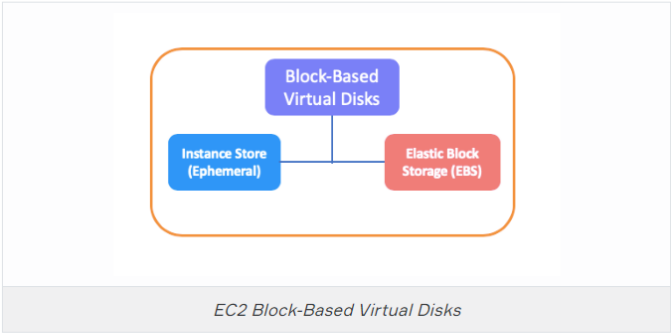
EC2 provides the following features and we will use these terms for the following lessons while launching and using EC2 :

- Virtual computing machines are known as **Instances** in AWS environment,
- Pre-configured 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),
- Generally, storage components for EC2 are known as **Amazon EBS Volumes** and **Instance Block Storage**,
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using **Security Groups**,

Next page we'll see the type of instances. But before going next page Let's remember highlights:

- EC2 is a virtual machine,
- We can adjust its capacity according to our needs,
- We pay as much as we use.

What is EC2 Volume?



EC2 Block-Based Virtual Disks

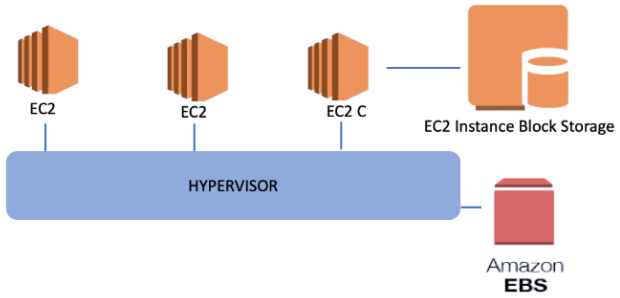
EC2 Block-Based Virtual Disks

Volume is a durable storage device that you can attach to one instance or to multiple instances at the same time. Volume is a location in which the associated machine stores its data or loads its applications. It as kind of virtual disk.

There are two main storage classes in AWS. These are File-Based Storage and Block-Based Storage. In the following lessons, we'll also see File-Based Storage (EFS e.g.) option. But, for now, we'll go on with Block-Based Storage.

AWS serves 2 basic volume options for costumers in the Block-Based Storage category. These are Instance Store (Ephemeral) and Elastic Block Storage (EBS).

Virtualization in EC2



Virtualization

Before moving to EBS, it will be more appropriate to remember the virtualization issue which is an important part of the cloud machine system.

In the virtualization environment, we have one physical server and a virtualization software on it, as seen in the figure above.

We call this driver and software layer as a **Hypervisor**. The hypervisor is a system that ensures that all information within its body can be accessed by all connected machines or storage devices.

The systems connected **directly to the hypervisor** and accessible to each machine associated with hypervisor we mentioned above is called EBS, **Elastic Block Storage**, in the AWS. Thus, if one of the physical servers fails, the virtual machine configurations running on it pass to the other physical server and continue to operate without interruption.

But, instead of being connected directly to hypervisor, **Instance Block Storage** is **connected to only the related server** on which the virtual machine is running. If the physical machine was disabled in some way, that virtual machine configuration would move to the other physical machine, but the main data would not work because it stopped at that problematic physical machine.

Instance Block Storage (Ephemeral)

This is the name given to the storage method that uses disks directly connected to the physical server on which the virtual machine is running. It may have SSD or magnetic HDD hard disk.

The **advantage** of this model is that it provides high access speed and very low latency because it is directly on the physical server to which the virtual machine is connected. Therefore, some of the Storage Optimized type D, H and I families, where disk input and output speed is important, use this type of storage solution.

The **disadvantage** is that if the virtual machine shuts down in some way, all data here is lost. If something happens to the underlying physical machine or you turn off the virtual machine, the data on these disks cannot be accessed.

EBS (Elastic Block Storage)

If you think EC2 is a virtual server in a cloud, you can say EBS is a virtual disk in a cloud.

EBS is the storage solution that can be attached to a virtual machine and can be installed in an operating system/application.

EBS also provides a 99.999% accessibility guarantee and replicates data to multiple physical devices within the same AZ, including SSD or HDD based disk infrastructures.

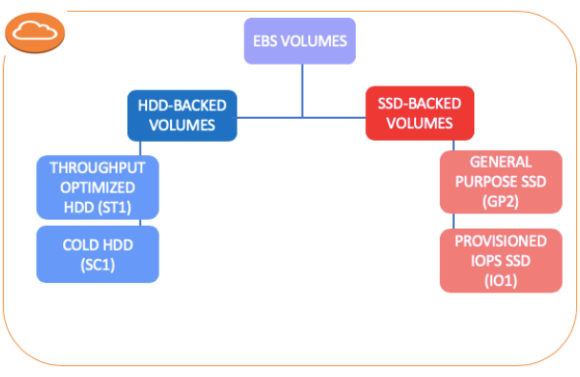
If you create a Windows or Linux EC2 instance EBS volume can be attached as Root device of volume automatically.

> Tips:
> - Although EBS has previously only allowed a single EC2 instance to be attached to a given volume, multi-attach is now possible, at least for IO1 volumes.

> Click here for more information about ebs-multi-attach

Unlike the Instance Store, EBS has snapshot support. Thus, we can save a copy of the files created with EBS at a certain time and then create new machines from this file or use this file as a backup.

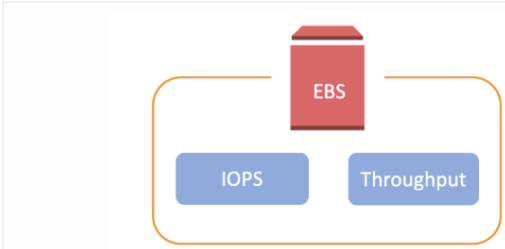
EBS Volume Types



EBS Volume Types

EBS offers us 4 different storage options. Two of them are SSD type and the remainings are HDD type as you see in the figure above.

IOPS and Throughput



IOPS vs. Throughput

Before we begin to explain EBS volume Types we should know the definition of **IOPS** and **Throughput**.

IOPS stands for **I**nput/Output **O**perations **P**er **S**econd. It is a common performance measurement used to test computer storage devices like HDD or SSD etc. In other words, IOPS is a value that specifies how many reads and writes can be made to a disk per second.

Throughput is the value that specifies how many MB of data transfer per second is allowed to a storage system.

While IOPS is related to the functional **speed** of the disk, Throughput is related to processing **capacity**. Throughput can be affected by IOPS, packet size and also network protocol.

Thus, we will have the opportunity to compare the differences between the EBS types when describing them.

SSD Type EBS

EBS General Purpose SSD (GP2):

EBS General Purpose SSD (Solid State Drive) is a fast solution for **general use** and operating system main disks. Sometimes called as GP2.

EBS Provisioned IOPS SSD (IO1):

EBS Provisioned IOPS SSD is also referred to as IO1. It is a model that is created with SSD and IOPS value can be pre-selected and provides pricing accordingly. It is therefore used for workloads such as **high-speed database applications** or **NoSQL databases**.

It is preferred when you require more than 10,000 IOPS.

HDD Type EBS

Throughput Optimized HDD (ST1):

Throughput optimized disk and frequently accessed disk is suitable for high-volume file storage. For example, Data Warehouse applications Big data, Log processing, etc.

Throughput Optimized HDD (ST1) volumes, though similar to Cold HDD (SC1) volumes, are designed to support **frequently accessed** data.

Throughput Optimized HDD is a low-cost HDD designed for those applications that require higher throughput up to 500 MB/s. The size of the Throughput Hard disk can be 500 GiB to 16 TiB. It supports up to 500 IOPS.

Cold HDD (SC1):

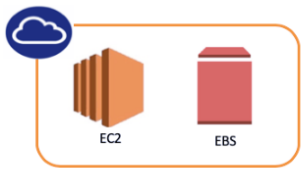
Cold HDD (SC1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. If you require infrequent access to your data and are looking to save costs, SC1 provides inexpensive block storage.

Cold HDD volumes, though similar to Throughput Optimized HDD (ST1) volumes, are designed to support **infrequently accessed** data. It is mainly used for a File Server.

The size of the Cold Hard disk can be 500 GiB to 16 TiB and can supports up to 250 IOPS.

Creating EBS (Elastic Block Store) Volumes

Usage of EBS (Elastic Block Store)



EC2&EBS

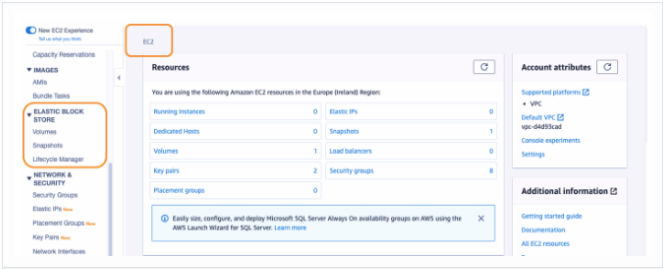
As we have seen in the previous lessons, EBS is the storage solution that can be attached to a virtual machine and can be installed in an operating system/application.

Let's assume that EC2 is a virtual server in a cloud, you can say EBS is a virtual disk in a cloud.

So, we can create an EBS volume and attach it to the virtual machine, in addition to the root EBS volume that we created while launching an instance, via the AWS Management console.

Also, we can modify the existing EBS volume on the AWS Management Console.

EC2&EBS



Elastic Block Store

As we have seen in the previous lessons, EBS is a service where we create and manage disks used in virtual machines.

Volumes, Snapshot and Lifecycle Manager options are available under the EBS menu.

- In the **Volumes** section of AWS Management console, we can see all the disks (Root Volumes and EBS) in our existing virtual machines. In addition to seeing our Root Volumes and other EBS we can also create new volumes and attach them to the instances according to our needs.
- There is a **Snapshots** section on the console just below the Volumes section. In this section, we are managed to create an image of volumes called snapshots. An EBS snapshot is a point-in-time copy of your Amazon EBS volume. We'll see detailed information on this subject in the following lessons. So we're skipping this for now.
- And finally, you can see the title of the **Lifecycle Manager** below Snapshot. Lifecycle Manager is used for helping you to automate the creation or deletion of EBS snapshots based on a schedule like you can do the same for the other storage components in AWS.

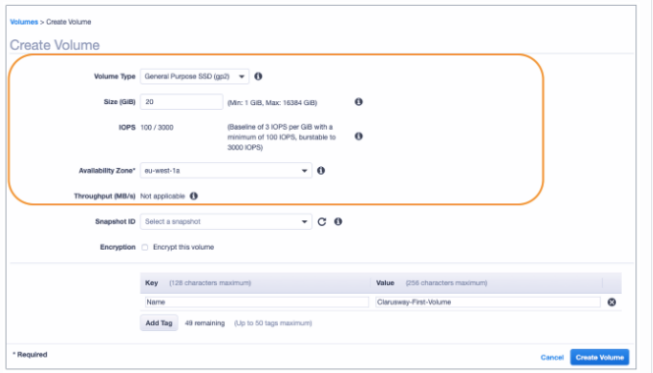
Creating Volumes-1

Go to Create Volume

When we created EC2, one volume was already created automatically known as **Root Volume**. Now let's create an additional Volume.

First Click **Volumes** on the left-hand menu and then select **Create Volume** respectively.

Creating Volumes-2



Creating a Volume Page

- **Volume Type:**

There are 5 options including General Purpose SSD (gp2) Provisioned IOPS SSD, Cold HDD, Throughput Optimized HDD and Magnetic (standard). Let's continue with General Purpose SSD.

- **Disk Size:**

You can choose from 1GB to 16384 GB of capacity for the volume. For now, let's set it as 20 GB.

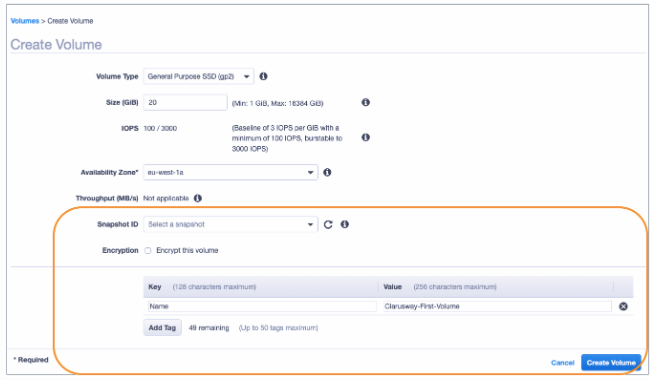
- **Availability Zone (AZ):**

  If you want to connect a disk to a virtual machine, the number one rule must be the same AZ as that virtual machine. Let's select **eu-west-1b** to be the same in the Volume creation menu.

- **Throughput:**

This setting cannot be edited because we chose the **General-Purpose** volume, it is the same for **Provisioned IOPS SSD** and **Magnetic** options. But, if we selected **Throughput Optimized** option, it would be specified here.

Creating Volumes-3



Creating Volume

- **Snapshot ID:**

We will select the **Snapshot ID** as the default. Because we haven't created any snapshot that can be also used for creating volumes. We will see the snapshot in the following lessons.

- **Encryption:**

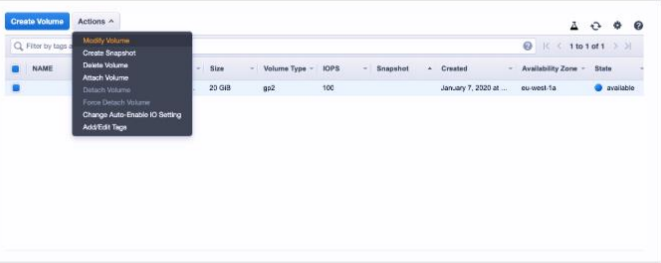
This part is important. If we check this flag here, we can encrypt this disk at startup. As you remember, volumes can be encrypted when we create Linux virtual machines.

We are also able to encrypt the EBS volumes that we created later on.

- **Add Tags:**

We can name the volume here. Let's write Volume's name as Clarusway-First-Volume.

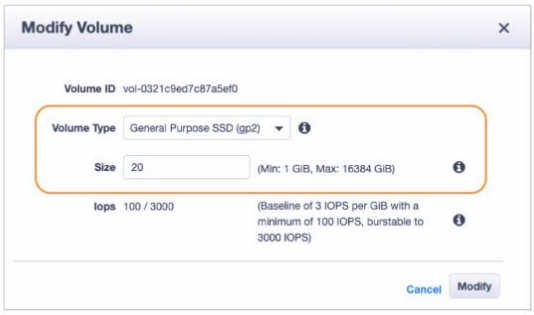
Modifing Volume Settings



Action Menu

Thanks to **Action** menu on top, you can also modify the settings of the volume that we created before.

So, first, click **Action** menu and select the **Modify Volume** option. Next, you'll see this page seen below.



Modify Volume

As you see in the figure above, you are able to modify **Volume Type** and **Size** of the existing volume here, and you can instantly see your changes on the console screen. This provides flexibility to the user.

Amazon Machine Image (AMI)

Amazon Machine Image (AMI)

Amazon AMI

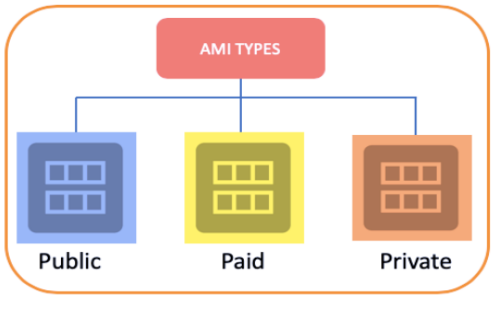
After determining our machine model and disk, it's time to cover another important component for the operating system and platform. This is called the Amazon Machine Image (AMI) in the AWS.

What is Amazon Machine Image (AMI) ?

AMI is a virtual image used to create a virtual machine within an EC2 instance. In other words, it is a virtual machine template containing predefined operating system and application files.

When creating a virtual machine, you will get an operating system and application list according to the features presented in that template by choosing which template to create this machine. All AMI provides a template for the root volume of an instance. You can copy the AMI and create another instance also.

AMI Types



AMI Types

Public:

These are the Public Shared Community AMIs and the AMIs managed by the Amazon itself. This package **covers common server features**.

For example, UBUNTU, one of the most famous community-driven distributions in the Linux world, has its own AMIs and you get a server with UBUNTU installed.

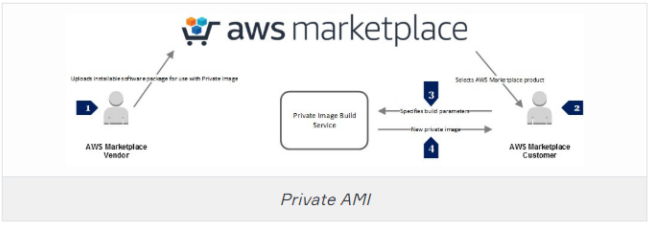
And also The Amazon Linux AMI, which Amazon has prepared itself, and includes several AWS applications similar to AWS CLI, as well as the basic Linux operating system. All of these and more are AMIs available to everyone.

Paid:

Another type of AMI is paid versions that we call Paid. These are ready-made packages **created by various companies or independent developers**, including various applications as well as the operating system.

For example, an application creator creates a Linux server image with its own application installed in the AWS store called Amazon Market Place and sets a price. You can buy it by accepting this price

Private:



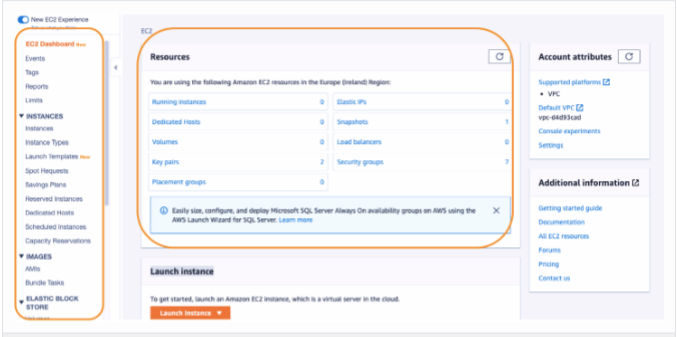
Private AMI

There are also AMIs that we can create and manage with AWS Marketplace and Private Image BuildService. It's now in public beta and enables AWS customers to purchase your installable software products through AWS Marketplace. Then you can install those products and specify them with **Private Image Build Service** for your IT needs as you see in the picture above.

Launching an Instance

EC2 Console

Now that we have learned the components of EC2, we can move on to the hands-on section to create a virtual machine and practice what we have learned.



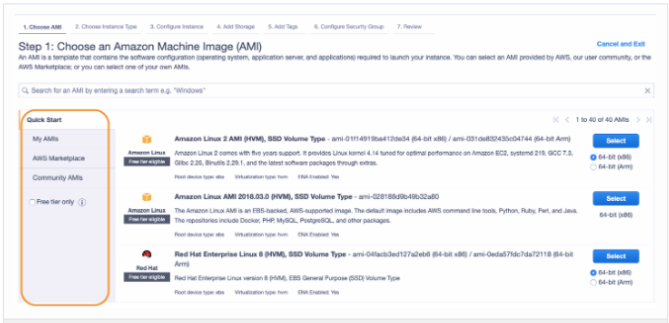
EC2 Console

When we open the EC2 console; (From Services or AWS Browser)

- On the left-hand menu, you will find all EC2-related services, sub-services, and their settings.
- In the middle, we can see how many resources we have and the details of them.
- Directly below you'll see Launch Instance button.
- And also Scheduled Events, Migrate a Machine, Quick ID Filter, Service Health, and Availability Zone Status are listed respectively.

Let's click the **Launch Instance** button and start to build our first virtual machine. We can also make the same process from **Instance** button that is located left-hand menu of the console.

Step 1 Choose an Amazon Machine Image (AMI)



Choose an AMI

The left-hand side of the console shows a variation of AMI we want to create an instance with. We have 4 basic options here.

Quick Start:

Here are common use virtual machines that Amazon offers.

My AMIs:

In this option, we can use the AMI Templates that we have already created.

AWS Marketplace:

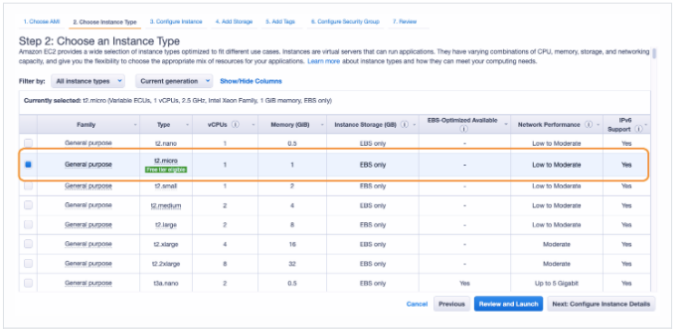
We can also go to the AWS marketplace and choose one of the AMIs offered by other companies for a fee.

Community AMIs:

Here are the official AMIs of some communities and companies. For example, if we want to install a virtual machine from Red Hat community, we can choose an AMI provided and updated by the Red Hat company.

If we click **Free tier only** button we can only display free machines.

Step 2 Choose an Instance Type



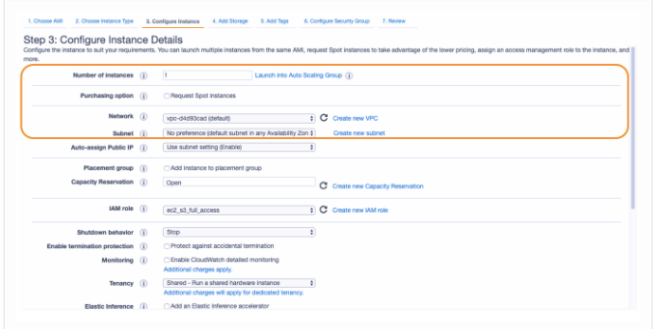
Instance Type Page

In this section, we will choose what kind of virtual machine configuration we will install on AMI.

All EC2 Instance types are listed here. If we plan to run programs that require high capacity, we can choose more advanced but paid versions.

We choose our free tier option **t2.micro** and go on with clicking **Next**

Step 3 Configure Instance-1



Configure Instance Page

Now let's see what our options mean. However, we will leave some options that we haven't explained yet as default.

Number of instances:

It defines how many EC2 instances you want to create for your configuration. Let's pick one.

Purchasing Option:

Here you have the option to request for Spot Instances.

We leave it as unchecked for now.

Network:

We can choose a network, create a VPC and select our own IP address range.

VPC closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS. **Leave it as default for now**.

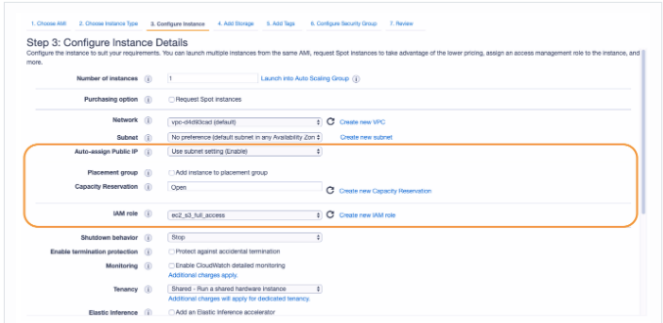
Subnet:

A subnet is a range of IP addresses in your VPC that can be used to isolate different EC2 resources from each other or from the Internet.

If we created a VPC before, we could have determined which subnet we want from here. So we **leave it as default**.

Network and Subnet options can be better understood when we examine the VPC title, we will not go into these issues for now.

Step 3 Configure Instance-2



Configure Instance Page

Auto-assign Public IP:

If we select the Auto-assign Public IP option, AWS will assign a Public IP to the instance so that we can connect to this machine via the internet. So this option must be selected.

Placement Group:

From here we can insert this machine into a mounting group. We can launch an instance in a placement group to benefit from greater redundancy or higher networking throughput. You can select an existing group or create a new one. We **leave it as unchecked**.

Capacity Reservation:

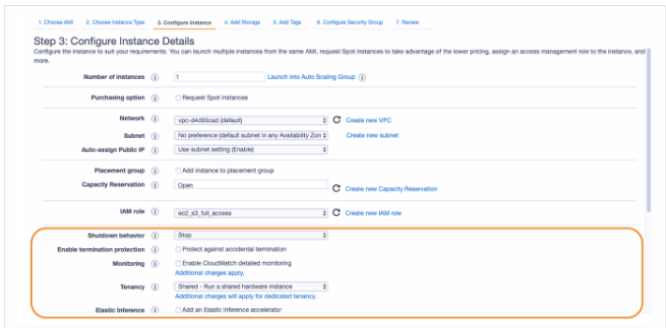
Capacity Reservations reserve capacity for your EC2 instances in a specific Availability Zone. You can launch instances into a Capacity Reservation if they have matching attributes. We **leave it as default**.

IAM Role:

An IAM role automatically deploys AWS credentials to resources that assume it. You can select the instance profile that contains the required IAM role.

Leave it as default for now.

Step 3 Configure Instance-3



Configure Instance Page

Shutdown Behavior:

It defines the behavior of the instance type. You can either stop or terminate the instance when you shut down this machine. Now, we leave it as Stop.

Enable Termination Protection:

When **Protect Against Accidental Termination** option is selected, you cannot terminate this machine via the API or management console accidentally. The machine will not be wiped until this protection is removed. Let's **leave it as is**

Monitoring:

Normally these machines are observed every 5 minutes. If we choose this option here, this time will be reduced to 1 min. But, when we choose this, there will be charged.

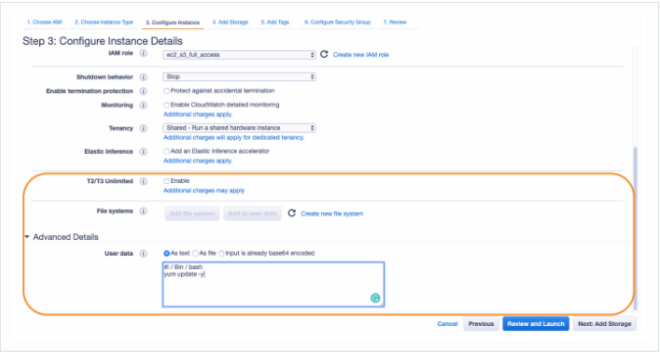
Tenancy:

In this section, there is a dedicated host or a shared system option. You can choose to run your instances on physical servers fully dedicated for your use. We learned it in the previous lesson-Pricing Instance Type Model. SInce we use on-demand instance, we choose the shared one for now.

Elastic Inference Accelerator:

Elastic Inference provides cost-efficient hardware acceleration for deep learning inference for all EC2 instance types, at a fraction of the cost of standalone GPU instances. It is an advanced feature of EC2 so, now leave it as unchecked.

Step 3 Configure Instance-4



Configure Instance Page

T2 / T3 Unlimited:

Enabling T2/T3 Unlimited allows applications to burst beyond the baseline for as long as needed at any time. If the average CPU utilization of the instance is at or below the baseline, the hourly instance price automatically covers all usage.

No need to choose this option. Now we can leave it as default.

File System:

This option specifies Amazon EFS file systems to mount to your instance. We will see it later.

Advanced Details:

User Data:

You can specify user data to configure an instance or run a configuration script during launch. This section provides an important service for us. For example, If you write any command code here, it will make all operating system and package updates available to us when this machine is installed and turned on.

So we'll write a 2-line code.

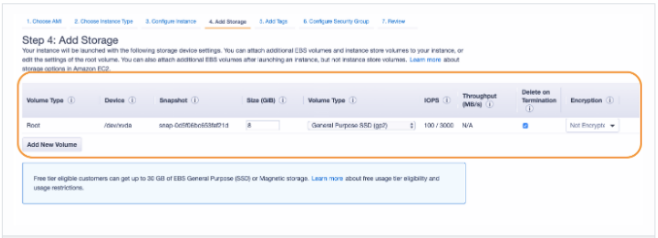
```
#!/bin/bash
yum update -y
```

- The first line tells that commands will be run in Bash Shell
- And the second line is Linux yum package manager. It provides us to update the machine automatically without any other confirmation when it starts thanks to the **-y** command. Now all the updates will be ready when we start the machine.

Let's continue with **Next** and add a virtual disk to this machine.

automatically

Step 4-Add Storage



Add Storage Page

Basically two kinds of disks can be attached to EC2 instance. The first one is **default volume** that will be attached when you create an instance. It's already seen here as named Root.

And the second one can be added manually via **Add New Volume** tab.

In Root Volume you can see the size of the disk, in other words, how many GB it will be. The default value is 8 GB and we can increase if we want.

You can also select what kind of **Volume Type** you want. General-Purpose is available for us.

Delete on termination is checked means that the termination of an EC2 instance will also delete EBS volume.

By the way, pay attention to the Encrypted part. You cannot encrypt the root volumes of EC2 virtual machines after created unless taking a snapshot. Leave it as is, for now. We will see this method in the following lesson.

After setting up our root disk, we can add other disks to the virtual machine by clicking the **Add New Volume** tab. You can add as many new virtual disks as you want, change their size, and then change the volume types if you want to.

Now let's go to the next section

Step 5 Add Tags

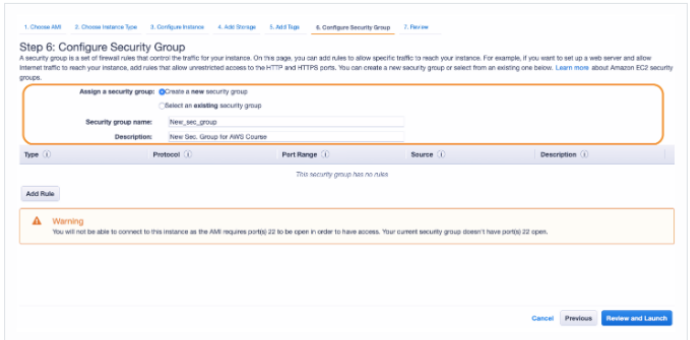
Add Tags Page

Tags are created as key-value pairs and allow you to more easily follow the resources you create in AWS world.

For example, we enter **Name** for Key Tab and **First_Instance** for Value Tab.

You can give more than one tag to each instance.

Step 6 Configure Security Group-1



Configure Security Group Page

The Security Group can be roughly defined as a firewall that stands in front of this virtual machine.

By default, **all inbound traffic is blocked**. In addition, **all outbound traffic is allowed** automatically. But, according to our purpose, we can provide access to this machine through various ports such as HTTP, HTTPs, etc.

To do this, we first need to create a security group and assign it to the virtual machine. Now let's create a new security group and enter name and description.

In the Security Group page, you'll see 2 options:

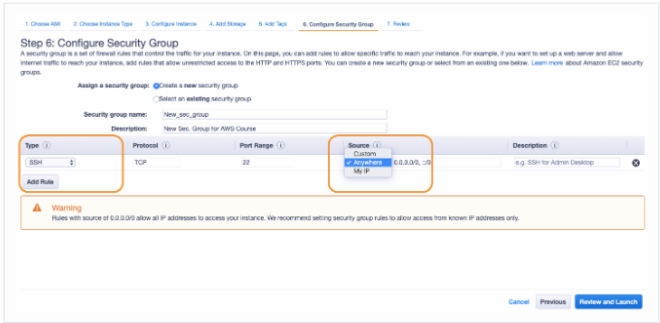
- Create a New Security Group
- An Existing Security Group

If you haven't created any security group before, you should select create a new one. And this option is already selected by default.

So, enter the name as **New_sec_group** and then write the security group description (New Sec. Group for AWS Course).

After that, you'll see Rule Menu. We can add new rules by clicking **Add Rule** here.

Step 6 Configure Security Group-2



Configure Security Group Page

First, we can choose SSH, because we want to access this machine remotely with SSH. It's automatically arranged Protocol as TCP and port range 22.

And finally, for the source part, we have 3 options here. **Custom**, **Anywhere** and **MyIP**.

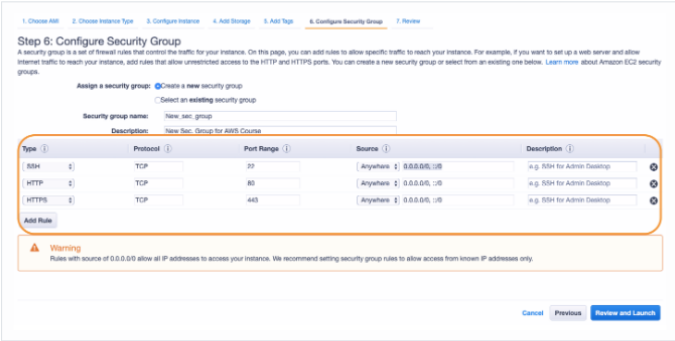
Anywhere: In this option, the machine can be accessed from all over the world.

Custom: Only specific IP that we choose is available to access for this machine

MyIP: Only your current IP is allowed to access to this machine.

We will now set up anywhere option. 0.0.0.0/0, :: / 0. So, this virtual machine can be accessed from this TCP 22 port from anywhere on the public internet.

Step 6 Configure Security Group-3



Configure Security Group Page

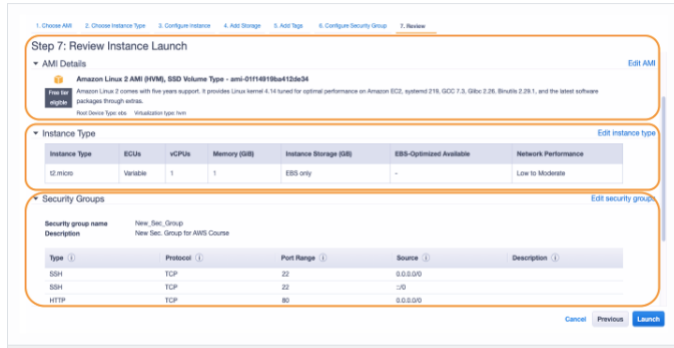
Now, let's add 2 more rules.

The first one will be an HTTP rule. Because we will define this machine as a web server later. As of second one, we can choose the HTTPS security protocol. We choose **anywhere** as a source for both rules.

So we created a new security group, wrote a description to this group, and arranged access from all over the world via SSH, HTTP, and HTTPS ports.

Now click **Review and Launch** tab for the last screen.

Step 7 Review Instance Launch



Review Instance Launch Page

This page is a summary of all the settings we have selected.

Let's check all the settings we did.

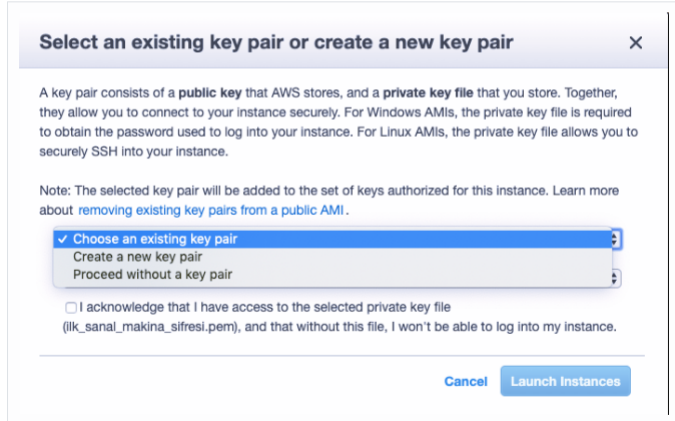
If everything is OK, click Launch to start creating this virtual machine.

Step 7 Review Instance Launch- Create a Key Pair

After review, you need to create a key pairs. Because in the AWS, we connect to the virtual machines with these key pairs.

These two keys are **Public Key** and **Private Key**. The Public Key will be stored in AWS and we will download the Private Key on our computer.

We need to keep this file(Private Key) well because if we lose this file, we won't be able to connect to this virtual machine again.

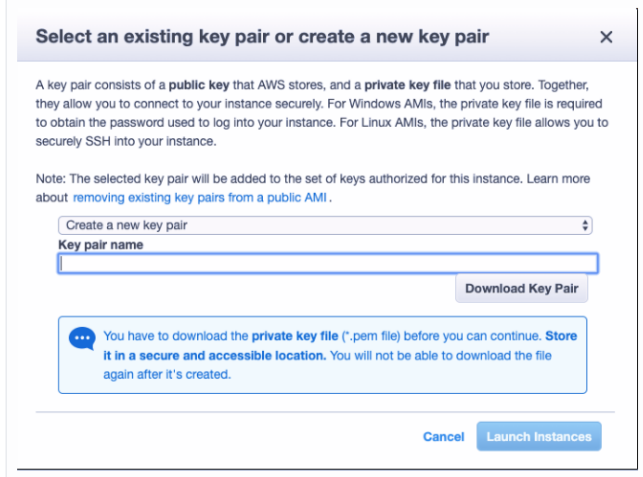


Key Options

While creating a Private Key You have three options. You can choose;

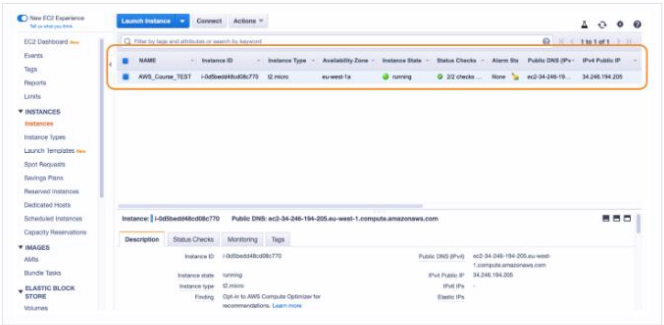
- Create a New Key Pair,
- Choose an Existing Key Pair,
- Proceed Without Key Pair options.

If you created any instance before you have already one key pair. So, you can choose **Choose an Existing Key Pair** option.



Create New Key Pair Page

Instance is Ready



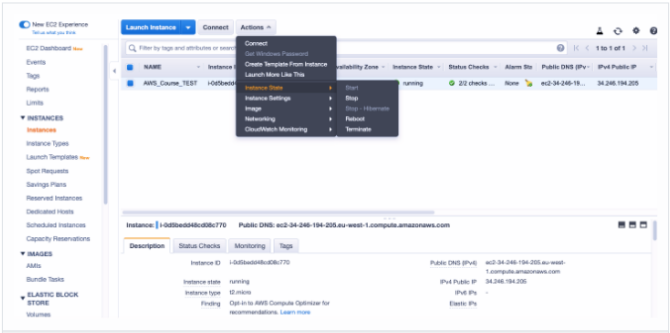
Instance is Ready

We have created our first EC2 instance. You can see the status of your instance from the EC2 Dashboard as seen above.

As you see, our instance's state seems **Running**. It means that this instance is ready to use.

Let's see, which option AWS offers while using this instance.

Instance Actions Menu



Actions Menu

We can rearrange most of the features selected while creating the machine via Action Menu. But we will use and explain many of them in the following chapters.

Now, we can focus on the subject of **Instance State**

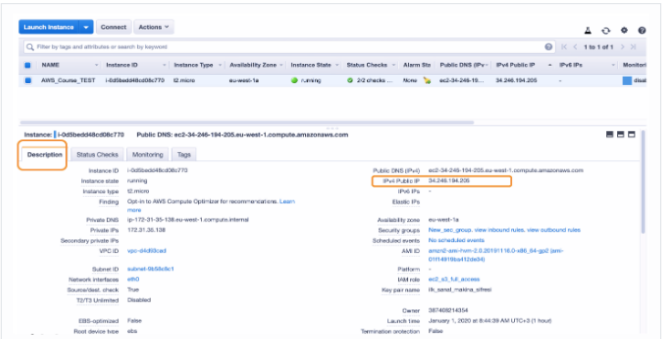
From this screen, we can simply restart or stop the instance, or we can terminate it. There are **Instance State**, **Stop**, **Reboot** or **Terminate** options are available here.

But, what is the difference between **Stop** and **Terminate** options of instance?

If you stop the instance you can restart it later. But if you terminate it, it means you delete the instance permanently. So, you can't use this instance anymore. Thus, we usually prefer to stop the instance.

Instances Console-Description

In this part, let's get to know our machine features and console.



Description Page

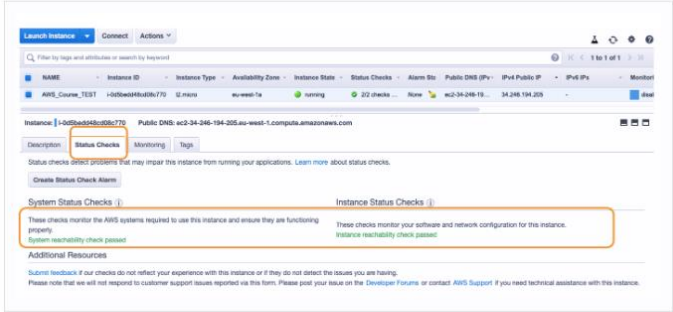
- **Description:**

In this section, we can access all the information that we set up while launching the instance including instance state, instance type, root device type, availability zone, security groups, etc.

You can also see **Public IP** seen on the right-hand side.

Public IP is especially important for us. Because we will connect to this machine with this Public IP. This is the IP of this machine that appears on the internet.

Instances Console-Status Checks



Status Checks Page

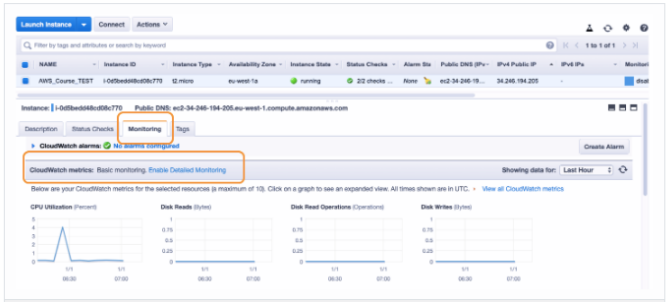
- **Status Checks:**

Here we can see the status of the virtual machine.

If a problem occurs at the underlying physical server and hypervisor layer, we can get information from here.

If there was a problem in the system or hypervisor, we would see these problems in the **System Status Checks** and **Instance Status Checks** sections. So everything seems okay for now.

Instances Console-Monitoring



Monitoring Page

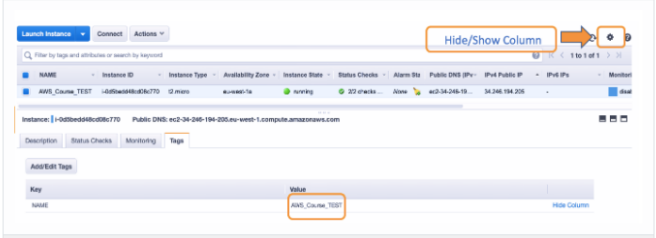
- **Monitoring:**

In this section, we can access the data about the virtual machine.

As you remember, we did not select **Detailed Monitoring** when launching the virtual machine. That's why this CloudWatch service checks this machine every 5 minutes and prints the information it creates on this screen.

If we have chosen **Enable Detailed Monitoring**, 5 minute-period of Monitoring would be reduced to 1 minute.

Instances Console-Tags



Tags Page

- **Tags:**

We can see the tag and names we edited when creating the virtual machine here.

For each instance, there can multiple tags also.

So, you can choose which tag you want to appear via **Hide/Show Column** tab on the top right of the page.

Now instance is ready and the menus are familiar. So, it's time to connect to the machine