

Table of contents

- [Table of contents](#)
- [Amazon EC2 :](#)
 - [Amazon EC2 Overview](#)
 - [Launching an EC2](#)
 - [Connecting to the EC2 Instance](#)
 - [For Windows](#)
 - [For Linux](#)
 - [Troubleshooting EC2 SSH Connection](#)
 - [Executing Commands EC2 Instance](#)
 - [Installing apache web server](#)
 - [Default Linux User Name to login as per AMI](#)
- [Elastic IP](#)
 - [Allocating an Elastic IP Address to a EC2 instance.](#)
- [Modify Instance Type and Extend EBS Volume.](#)
 - [Modifying Instance type of an Amazon EBS-backed EC2](#)
 - [Extend AWS EBS volumes](#)
- [Creating Snapshots and AMI](#)
 - [Creating Snapshots for Backup of an EBS Volume.](#)
 - [Automating Snapshot Creation using Data Lifecycle Manager.](#)
 - [Creating a Linux AMI from a snapshot](#)
 - [Copying AMI or Snapshot from one region to another.](#)
- [EC2 cheat codes](#)

Amazon EC2 :

Amazon EC2 Overview

- EC2 is one of the most popular service from IAAS (Infrastructure as a service) category
- What IaaS generally stands for is, the hardware part is taken care by the vendor. We as the consumers will just have to use it.
- Let us now launch an EC2 and execute some commands on EC2.

Launching an EC2

- Log in to the AWS console and click on EC2 from the services tab
 - Click on launch instance from the EC2 dashboard
 - We will now follow the 7 steps necessary to configure and launch an instance
1. Choose an Amazon Machine Image (AMI)
 - Amazon Machine Image dictates which operating system we'll be getting along with additional packages if any . for ex- aws cli,python etc
 - We generally recomend using Amazon linux , as the support is easily available in a centralised dashboard
 - if the OS we need is not available in the dashboard , we can check in amazon marketplace option on the left hand side .

- Marketplace is a platform where vendors create their own AMIs and put it up for paid use. For ex- Palo alto.
 - We also have an option of Community AMIs , it is a platform where AWS users create their own AMIs and share it with the community for free use
 - The last option is our own AMIs . If we want to avoid doing redundant work, we can create an AMI and use it to launch new instances . This saves a lot of time . AMIs can also be used to take backups and can be used to restore in case of a disaster.
 - For now we will go with the default quickstart option and select Amazon linux 2 . (Amazon linux one is set to expire soon)
2. Choose an Instance Type
- Instance type decides the CPU,memory and Network capacity of the instance.
 - There are variety of types available to suit every use case
 - For now we will go with t2.micro , which is a free tier eligible instance type (750 hours per month for the first 1 months)
3. Configure Instance Details
- The pane allows us to select major details as the number of instances ,VPC ,subnet, spot price , tenancy and userdata(bootstrap)
 - Options like VPC, IAM role we will be exploring in later sessions
 - For now we will go with default options and click on next
4. Add Storage
- AWS offers 3 types of storages for root volume and 5 types for additional EBS volume to suit various use cases
 - One EBS Volume/disc can be scaled upto 16384 Gbs and we can attach multiple discs to an instance. Minimum we can keep 8GB in order for OS to run.
 - These discs are by default terminated when instance is terminated , but we can choose to keep them from this screen.
 - We always recommend having the volumes encrypted
5. Add Tags
- Tags give us information about the resources. This is something that we keep to know the purpose behind the instance
 - It can have the provisioner team name, project name and the purpose
 - Should be written in key-value pair. ex:- Team-Developers
6. Configure Security Group
- Security group is the firewall to your instance
 - It dictates, which IPs can reach the instance over the network.
 - We can choose the protocol , port and the source from which we want our instance to be accessible, select Type as ssh and source as My Ip
7. Review Instance Launch
- review screen is for us to check and verify the details before we launch the instance .
 - Click on launch, we'll receive a popup for a key. Click on create new key pair (for first launch, later we can use the same key), click on download and then launch.

Connecting to the EC2 Instance

For Windows

1. We will first convert the key to suitable ppk file.

2. Download putty.exe and open puttygen once it is installed
3. Click on load , select all files from the bottom right corner , and select the key which we downloaded.
4. Click on save private key . Save without passphrase.
5. Open putty
6. in hostname copy and paste the public ip from the instance details from aws console.
7. Expand ssh on the left index and click on auth . Load the ppk file and click on open.
8. When you receive username prompt , enter 'ec2-user' and press enter.

For Linux

1. Navigate to the folder in which key is stored and use

```
sudo chmod 400 *keyname*.pem
```

2. Use below command to connect to instance

```
ssh -i *keyname*.pem ec2-user@*publicipoftheinstance*
```

Troubleshooting EC2 SSH Connection

- If you receive connection timed out error : check the security group.
- If server refused key error is received, check the username and key.

Executing Commands EC2 Instance

- Once we are connected to the server, we can test out some linux commands like below

```
free -h
```

```
lsblk
```

```
df -h
```

- Above commands give us insights into the memory and storage available.

Installing apache web server

- We will now install a web server on the EC2 for testing

1. Install apache web server package

```
sudo yum install httpd -y
```

2. Navigate to the home directory file and create index.html file for home page

```
sudo su -  
cd /var/www/html/  
echo "this is our homepage" >> index.html
```

3. Let us start the web server

```
service httpd start  
service httpd status  
systemctl enable httpd
```

4. Once the service is started, navigate to the ec2 dashboard in aws console.
5. Select the security group associated with the instance.
6. Add inbound rule for http type and source as anywhere.
7. Paste the public ip of the instance in another tab in the browser.
8. The above content of the index.html file on the EC2 instance should be displayed on Browser Webpage.

Default Linux User Name to login as per AMI

- For Amazon Linux 2 or the Amazon Linux AMI, the user name is **ec2-user**.
- For a CentOS AMI, the user name is **centos**.
- For a Debian AMI, the user name is **admin**.
- For a Fedora AMI, the user name is **ec2-user** or **fedora**.
- For a RHEL AMI, the user name is **ec2-user** or **root**.
- For a SUSE AMI, the user name is **ec2-user** or **root**.
- For an Ubuntu AMI, the user name is **ubuntu**.

Elastic IP

Allocating an Elastic IP Address to a EC2 instance.

- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance
 - Open the Amazon EC2 console, choose **Elastic IPs**.
 - Choose **Allocate Elastic IP address**.
 - For Scope, choose either VPC, choose **Amazon's pool of IP addresses**
 - Choose **Allocate**.
- To associate an Elastic IP address with an instance.
 - Select Elastic IP and choose **Actions, Associate Elastic IP address**.
 - For Resource type, choose **Instance** for which it has to be attached.
- Test by stopping and starting the EC2 instance. The Elastic IP will be attached and be allocated in AWS Account.
- This is used in case the Public IP of the Instance is to be configured some other application or code.

- Once Testing is done, to avoid incurring any charges, Disassociate Elastic IP from Instance and Delete the EIP from your Account.

Modify Instance Type and Extend EBS Volume.

Modifying Instance type of an Amazon EBS-backed EC2

- In case we find our EC2 instance is over-utilized(the instance type is too small) or under-utilized (the instance type is too large), we can change the size of the instance.
- For example, if your t2.micro instance is too small for its workload, you can change it to another instance type that is appropriate for the workload.
 - Select the instance and choose **Actions, Instance State, Stop.**
 - Once Stopped, choose **Actions, Instance Settings, Change Instance Type.**
 - From Instance Type, select the instance type that you want.
 - Choose Apply and choose **Actions, Instance State, Start.**
- Verify the configuration applied to the EC2 instance with New Instance Type.

Note : Make sure Resources that do not come under Free Tier Limit are deleted.

Extend AWS EBS volumes

- In order to extend the volume size, follow these simple steps:
 - Under EC2 service, navigate to **Volumes** under ELASTIC BLOCK STORE menu.
 - Choose the volume that you want to resize, click on **Actions > Modify Volume**
 - Set the new size for your EBS volume (in this case extended an 8GB volume to 20GB), Click on modify.
- Now, we need to extend the partition itself.
- SSH to the EC2 instance for to which extended EBS Volume is attached.
- To list our block devices:

```
lsblk
```

- The size of the root volume reflects the new size, while the size of the partition reflects the original size and must be extended before you can extend the file system.
- Make sure below command has correct device name(/dev/xvda) and partition number(1).

```
sudo growpart /dev/xvda 1

#verify the partition size
lsblk
```

Creating Snapshots and AMI

Creating Snapshots for Backup of an EBS Volume.

- Open the Amazon EC2 console and Choose Snapshots under Elastic Block Store in the navigation pane.

- Choose Create Snapshot, For Select resource type, choose Volume.
- For Volume, select the volume, Enter a description for the snapshot.
- (Optional) Choose Add Tag to add tags to your snapshot.
- Choose **Create Snapshot**.

Automating Snapshot Creation using Data Lifecycle Manager.

- Create a Snapshot lifecycle policy in Amazon DLM:
 - Tag your volumes or instances to be used by DLM policy.
 - Navigate to EC2 > under Elastic Block Store, select Lifecycle Manager.
 - Select Create Snapshot Lifecycle Policy and Enter a Description for the policy.
 - Select a resource type.
 - **Select Volume** - to schedule snapshots for volumes with a specific tag.
 - **Select Instance**- to schedule snapshots for all volumes attached to an instance with a specific tag.
 - Select the tags associated with the Amazon EBS volume or Amazon EC2 instance.
 - Add a Schedule name to your policy. Any snapshot created with this policy is automatically tagged with the schedule name entered here.
 - Specify the number of hours that will elapse between policy runs also Specify the policy run start time.
 - Set the **Retention rule** to the maximum number of snapshots that you want to retain. >The oldest snapshot beyond this number will be deleted.
 - Select an **IAM role** that has snapshot create and delete permissions. The Default role has the required permissions.
 - Review the Policy Summary and Select Enable policy to start the policy runs at the next scheduled time and Select Create Policy.

Creating a Linux AMI from a snapshot

- From Amazon EC2 console under Elastic Block Store, choose Snapshots.
- Choose the specific snapshot and choose Actions, Create Image.
- In the Create Image from EBS Snapshot dialog box, complete the fields to create your AMI, then **choose Create**.
 - Architecture: Choose x86_64 for 64-bit.
 - Root device name: Enter the appropriate name for the root volume. (/dev/xvda) Virtualization type: Choose hardware virtual machine (HVM) virtualization.
- Launch EC2 instance using this AMI.

Copying AMI or Snapshot from one region to another.

- AMI and Snapshots are only available to the region where they are created.
- However, if they have to be made available in different AWS Regions, AMI and Snapshots needs to be Copied from one Region to another.
- To copy a snapshot to another Region :
 - Select the snapshot to copy, and then choose Copy from the Actions list.

- Destination region: Select the Region where you want to write the copy of the snapshot. choose Copy.
- To copy a AMI to another Region :
 - Select the AMI to copy, and then choose Copy from the Actions list.
 - Destination region: Select the Region where you want to write the copy of the AMI. choose Copy.
- To view the progress of the copy process, switch to the destination Region, and then refresh the Snapshots page.
- You can launch an instance from the copied AMI in another region.
- Also, a volume can be created from the copied snapshot in another region.

EC2 cheat codes

- **Region** : A geographical location wherein multiple availability zones are present
- **Availability zone(AZ)**: Data center consisting of large network of physical servers
- **Instance** : A virtual server launched in AWS's region inside a availability zone
- **AMI** : Amazon Machine Image , Consider them as OS + additional package bundle
- **EBS** : Elastic Block Storage , Hard disc or storage option for your EC2 . Can be scaled up but cant be scaled down
- **Root volume** : disc on which the operating system will run.
- **Security Group** : Firewall to your instance. First thing to check if we are receiving connection related issues
- **Snapshot** - Cold backup of your EBS volume . In the background stored in S3 , but cant be seen in the bucket by us
- **Key pair** - Private key , need to be used to login to your instance. Can only be downloaded once
- **Tags** - Used to add information about the AWS resource