[Instructions] Getting Started with EC2

Let us get started with AWS EC2.

* EC2 stands for Elastic Cloud Compute
* It is nothing but a virtual machine provisioned from AWS Data Center for you. The Virtual Machines are provisioned from availability zones within a region.
* EC2 instances are provisioned using the instance type. An instance type is nothing but a combination of pre-configured CPU, Memory, and Instance Store.
* On top of CPU, Memory, and Instance Store, we also need to choose Operating System, at least one EBS for the root file system, VPC for networking, security groups for firewall security, and EC2 Key Pair to login into the system.
* EC2 instances can be in a running or stopped state. You will be charged only when the instance is up and running (except for storage).
* You can attach additional EBS Volumes to add more storage and you can create snapshots for the volumes periodically for backups.

[Instructions] Create EC2 Key Pair

Let us understand how to create EC2 Key Pair so that we can connect to the EC2 instance from external clients such as our PC or Mac.

* We can either create a key pair or use an existing one while creating an EC2 instance.
* It is not good practice to create key pairs while creating EC2 instances.

Here are the instructions to create a key pair.

* Login into AWS Web Console.
* Search for EC2 and go to EC2 Console.
* On the left sidebar, you have EC2 Key Pair under Security and Networking. Click on EC2 Key Pair and then click on “Create Key Pair”.
* Make sure to download the pem file and save it to the standard location on our PC or Mac. You need this later to connect to the EC2 instance.

Let us understand what happens when we create a Key Pair.

* A random private key and the associated public key will be created.
* Private Key files will be provided to you for download.
* Public Key will be managed somewhere inside AWS repositories.
* When you launch an ec2 instance with a specific key pair, the public key will be added to authorized\_keys on the server under standard user.

[Instructions] Launch EC2 Instance

Let us understand how we can launch an EC2 instance using AWS Web Console. Follow the video to understand how to create an EC2 instance using Ubuntu 18.04.

* You can leverage free tier instances to learn how to create instances.
* Go to AWS Management Console and then to EC2 Console.
* Click on Launch Instance and follow the steps as demonstrated.

[Instructions and Commands] Connecting to EC2 Instance

Let us understand how to connect to EC2 Instance using SSH from Mac or Linux. We can use the Mac or Linux terminal to connect to remote servers via SSH.

Here is the command for reference.

1. ssh -i ~/.ssh/itvaws.pem \
2. ubuntu@ec2-34-232-68-20.compute-1.amazonaws.com

[Instructions and Commands] Security Groups Basics

Let us understand the basics of security groups. AWS EC2 Security Groups facilitate us to define Firewall rules to block unintended traffic into EC2 instances from external systems.

* We need to choose an existing security group or create a new one while launching an ec2 instance.
* By default, SSH Daemon will be running on Linux-based ec2 instances on AWS using port 22.
* The default rule as part of most of the security groups is to open port 22 so that we can connect from remote machines using SSH.
* We will be able to login using SSH only.

Let us perform these tasks to understand the relevance of security groups.

* Make sure you can connect to the instance that is launched earlier using SSH.
* Go to the security group associated with the ec2 instance and delete the rule related to port 22.
* Try connecting to the ec2 instance using SSH. It will be struck.
* Go back and open port 22 for everyone and validate by using SSH.
* Install Apache on the Ubuntu machine. Login and run these commands to install and start the Apache Web server.

1. sudo apt update
2. sudo apt install -y apache2
3. sudo systemctl status apache2
4. # Hit Ctrl+C to come out of the command output
6. # Confirm that the apache server is running on port 80 using telnet
7. telnet localhost 80
8. telnet ec2-34-232-68-20.compute-1.amazonaws.com 80

* Go to the browser and try to access Apache Web Server using http://<public\_ip>. Here is the link for your reference - [http://ec2-34-232-68-20.compute-1.amazonaws.com](http://ec2-34-232-68-20.compute-1.amazonaws.com/)
* The link will not work. Now go to the security group and the new rule for port number 80 using **My IP**.
* Now the link should work without any issues. It will launch the default Apache Web Server Page for Ubuntu.

[Instructions] Public and Private IP Addresses

Let us understand the concepts behind public and private IP addresses associated with AWS EC2 Instances.

* Typically 2 IP Addresses and corresponding DNS aliases will be attached to each AWS EC2 Instance.
* Public DNS aliases start with **ec2** and Private DNS aliases start with **ip**.
* Public DNS alias or underlying Public IP address can be used to access EC2 instances or services running on it via the internet from outside of AWS.
* Private DNS alias or underlying Private IP address can be used for internal communication between EC2 instances within AWS VPC.
* By default, the Public DNS alias or Public IP address is ephemeral. It means if you stop and start an EC2 instance, most likely the Public DNS alias and Public IP address will change.

[Instructions] EC2 Life Cycle

Let us go through the life cycle of EC2 Instance. EC2 instance will be in one of these states as long as it is not terminated.

* Running
* Stopped
* Restarting

If you stop the EC2 instance, the public IP might be reset as it is ephemeral by default. You need to lease elastic IP and assign it to the EC2 instance so that public IP does not change.

As long as EC2 Instance is stopped you will not be charged for the instance. But, if you use EBS for storage, you have to pay for it.

[Instructions] Allocating and Assigning Elastic IP Addresses

Let us understand how we can configure a static elastic IP address for EC2 Instance.

* Go to AWS Web Console and then go to EC2 Console.
* Go to **Elastic IPs** under **Network & Security**.
* Click on **Allocate Elastic IP address** to lease the IP address so that we can associate with the ec2 instance created earlier.
* Select newly allocated elastic IP address, go to **Actions** and click on **Associate Elastic IP address**.
* Select the EC2 instance in the drop-down related to Instance.
* Now validate by running SSH using newly associated Public DNS Alias and make sure that you can connect to EC2 instance.

You can also try accessing Apache Web Server that is supported earlier via **http**.

[Instructions and Commands] Managing EC2 Using AWS CLI

Let us understand how we can manage EC2 Instances using AWS CLI.

* We should be able to take care of all the tasks related to managing EC2 Instances via AWS CLI. Here are some of the examples.
  + Create Key Pair or use existing Key Pair.
  + Create EC2 Instances
  + Attach Security Groups to EC2 Instances
  + Manage Security Group Rules
  + Stop, Restart as well as Terminate EC2 Instance.
  + Allocate and Attach Elastic IP.
  + Describe an instance to get all the metadata associated with it.
* You can get help on **aws ec2** by using the following command.

1. aws ec2 help

Let us warm up by performing some basic tasks against EC2 Instances using AWS CLI. We will be using the **itvadmin** profile as the user account have Administrator Access on AWS Account.

* Describe instances and get the instance ids.

1. aws ec2 describe-instances \
2. --profile itvadmin \
3. --region us-west-1
5. aws ec2 describe-instances \
6. --profile itvadmin \
7. --region us-west-1 | \
8. grep -i instanceid
10. # You can use one of the instance ids and get instance status
11. aws ec2 describe-instance-status \
12. --instance-id i-07c085b765f162233 \
13. --profile itvadmin \
14. --region us-west-1

* Stop the instance and validate whether the instance is stopped or not.

1. aws ec2 stop-instances \
2. --instance-id i-07c085b765f162233 \
3. --profile itvadmin \
4. --region us-west-1
6. aws ec2 describe-instance-status \
7. --instance-id i-07c085b765f162233 \
8. --profile itvadmin \
9. --region us-west-1

* Start the instance and validate whether the instance is started or not.

1. aws ec2 start-instances \
2. --instance-id i-07c085b765f162233 i-00f80143dc2e77b85 \
3. --profile itvadmin \
4. --region us-west-1
6. aws ec2 describe-instance-status \
7. --instance-id i-07c085b765f162233 i-00f80143dc2e77b85 \
8. --profile itvadmin \
9. --region us-west-1

* List Elastic IPs that are allocated so far.

1. aws ec2 describe-addresses \
2. --profile itvadmin \
3. --region us-west-1

[Instructions and Commands] Upgrade or Downgrade EC2 Instances

Let us understand how to upgrade or downgrade EC2 Instance using AWS Management Console.

* Increasing Memory or CPU or both is called Upgrading the EC2 Instance.
* Decreasing Memory or CPU or both is called Downgrading the EC2 Instance.
* Here we are talking about vertical scaling. Upgrading or Downgrading the same instance with a different configuration is vertical scaling.
* Adding more servers or removing some of the servers from a cluster of servers is called horizontal scaling. We typically talk about horizontal scaling with services like EMR, ECS, EKS, etc.

Let us go through the steps to upgrade as well as downgrade the EC2 Instance.

* Login into AWS Management Console
* Go to EC2 Console and choose the instance to which you want to upgrade.
* Go to **Instance State** and stop the instance. You can also use CLI to stop the instance.
* Go to **Actions**, then choose **Instance Settings** and then click on **Change Instance Type**.
* As of now, the EC2 Instance is of type **t2.micro**. Let us upgrade it to **t2.medium**.
* Start the server and login via SSH to confirm that you can log in to the server.
* Once validated, you can shutdown, downgrade to **t2.micro** and start the server.
* You can then log in once again to validate.
* We can run the below commands to confirm the memory and CPU configuration from within the server.

1. free -h
2. lscpu

* You can also run these commands to change the instance type using the command line.

1. aws ec2 stop-instances \
2. --instance-id i-07c085b765f162233 \
3. --profile itvadmin \
4. --region us-west-1
6. aws ec2 modify-instance-attribute \
7. --instance-id i-07c085b765f162233 \
8. --instance-type t2.micro \
9. --profile itvadmin \
10. --region us-west-1
12. aws ec2 describe-instances \
13. --instance-id i-07c085b765f162233 \
14. --profile itvadmin \
15. --region us-west-1
17. aws ec2 start-instances \
18. --instance-id i-07c085b765f162233 \
19. --profile itvadmin \
20. --region us-west-1