

Launch Types:

1. Instances:

-> Displays all the EC2 instances launched in the region.

AZ = me-central-1c

me -> Middle East

central - 1 -> exact location (UAE)

c -> AZ (3rd AZ)

2. Instance Types:

-> Displays all the available & supported EC2 instances with sizes & configurations, available for the selected region.

3. Launch Template:

-> We can create templates & define the configuration settings which are used for launching an instance, eg: OS, Instance Type, Security group, etc; thus automating the process of instance launch.

4. Spot Requests:

-> For requesting unused EC2 instances, which can reduce the EC2 costs significantly.

-> But there's a warning:

AWS can reclaim the instance by terminating the spot instance at any time, if they need the capacity back, with only 2 minutes of warning.

5. Saving Plans:

-> A flexible pricing model that offers low prices on EC2 instances.

-> Customers can reduce the EC2 costs by making a commitment to a consistent amount of usage in USD/hours over a term of 1 or 3 years.

6. Reserved Instances:

-> It is a billing discount model on EC2 usage.

-> Lets customers purchase EC2 capacity in advance for 1 or 3 years with a significant discount.

-> Reducing EC2 costs by making a commitment for a specific instance configuration, instance type and in a particular region

7. Dedicated Hosts:

-> It is a physical server- fully dedicated to an organization.

-> Unlike normal EC2 instances, a Dedicated Host gives users full control over the underlying infrastructure and you do not share the resources with any customers.

8. Capacity Reservation:

-> Capacity reservation ensures that you have guaranteed EC2 instance capacity available for launch when needed, in a specific AZ.

-> Basically it is like reserving compute capacity for future use in a particular AZ.

Connection of Instance using SSH:

-> SSH : Secure Shell.

-> Accessing instance present in a Data Center & you want to connect to the VM/instance through your local machine; basically remotely, using CLI

Make sure these thing before connecting;

- i. EC2 instance is running
- ii. 'pem' private key file is downloaded in the local machine.
- iii. Security group allows inbound rule of SSH (port22)

Steps:

1. Launch the Instance.
2. Click on 'Connect' and select 'SSH client'.
3. Copy the SSH example at the bottom:

`ssh -i "key.pem" ubuntu@ec2-3-28-45-75. mecentra-1.compute.amazonaws.com`

Secure Shell identify key pair private key pair Username to connect public IP address of Instance Region-code Link of AWS

4. Go to 'Private key' in the local machine where it was downloaded & just select it, do not open it.
5. Go to path and then paste the SSH example link there and press Enter.
6. A prompt will come, press 'Yes' and instance is connected!

Assignment

1. Create an Ubuntu instance with free eligible instance type.
2. Along with root volume, add additional storage of 10 GB.
3. Create a .pem key.
4. Now, while connecting instance, use ssh client.
5. When connected, inside the instance, write a script to update the system, install java and maven and run the script.
6. Close the instance.
7. Go to EC Instance connect and try to connect again.
8. Verify if the script is still present in instance and if yes, verify if the tools are installed.

Assignment

1. Create a Launch Template with Ubuntu as OS and instance type as any free eligible.
2. Create .pem key
3. Launch an instance with the template only
4. Verify if the instance is running and if yes, connect with ssh client.
5. Create 2 folders inside it and create 2 files in both of them.
6. Add content in all files using vi editor
7. Remove folder1 and rename folder2 and give any new name.
8. Delete the instance.

Connection of Instance through PuTTY

-> Local machine is Windows, there we cannot directly use SSH to connect to the remote machine. That is where PuTTY software comes into play. It is also a SSH client but as an software.

1. '.ppk' key pair should be there.
(Putty Private Key)
2. Create an Instance with .ppk private key file.
3. Click on Instance and check for 'Details'.
4. Copy the Public IP of the instance.
5. Go to PuTTY software and under 'Host Name', paste the public IP and click on 'Save'.
6. From left side panel of PuTTY, select 'SSH' and then select 'Auth' (Authorization) & select 'Credentials'.
7. Browse for the .ppk file and click on 'Open'.
8. A confirmation page will pop-up, then click on 'Accept'.
9. A 'login-as' page will pop-up, there give the username of the instance and press 'Enter'.

Creating a Windows Instance and connecting to it through RDP:

RDP -> Remote Desktop Protocol Client

Root volume for Ubuntu instance -> 8gb

Root volume for Windows instance -> 30gb

Root volume for macOS instance -> 100gb

Steps:

1. Create a Windows instance with Windows Server Base OS (select a large instance) and create a .pem key file.
2. After launching the instance, click on 'Connect' & select RDP Client.
3. Select 'Download RDP file'.
4. To connect to Windows Instance, click on 'Get Password'.
5. Upload the 'private key' and click on 'Decrypt Password' to generate a password.
6. Open the rdp file, click on 'Connect' and provide the password and then click on 'OK'.

System Check:

1. System status check:

-> verifies underlying AWS hardware & infrastructure which is hosting your EC2 instances.

2. Instance Status check:

-> verifies the health of your EC2 instance itself and monitors the software & connectivity of the individual IP.

3. EBS status check:

-> evaluates the health of both Root volume & attached volumes for potential issues.