

**Step-1: Create EC-2 Instance**

- Login to aws console
- Go to services and select EC2
- Now click on launch instance to create an instance.
- Now select the non-community AMI Amazon Linux 64-bit.
- Now configure the instance.
- In configuration you can add extra volumes to the server which already has 8GB of EBS available.
- Also while launching an instance you will be asked to use a key or create a key through which your instance will be accessed by you.
- Select create a new key and it will download a .pem file.
- Now review and launch the instance.
- It will take sometime before the status of instance is changed to running and after that some time to perform two status check.
- When the status check is done you can see system log and observe some server OS related things by clicking on action>instance--->get system log.
- Now your instance is running successfully.

**Step-2: Attach EBS and mount file system on it:**

- Now to create an EBS volume go to EBS category on leftside panel and select volume.
- There you will see a volume already present, this volume is the one which is allocated to the server you created.
- To allocate a new storage you need to click create volume.
- You need to make sure that the new volume is created in the same region in which your instance is running.
- While you are creating the volume you can assign the storage space needed by you (minimum is 1GB). Also, it will show a name of the volume as /dev/sdf which will then be changed by the kernels when you will use this volume.
- Now to attach this volume to the instance just click on actions and select attach option after that select the instance name you want to attach it to.
- Now to mount an extra file system to the extra volume "you" need to connect to the "instance" through putty.
- Now before using putty you need to use puttygen to convert your key file which is in .pem into .ppk file.
- In puttygen load the .pem file and save the .ppk file without any passphrase for simplicity. But assigning the passphrase will provide more security to the private key of your instance.
- Now open the putty and in the Session category on left side you need to enter the hostname, port, select SSH as connection method.
  - Host-name: <Amazon-linux user name>@<public dns of instance>
  - Port: 22
- Now go to the SSH>Auth category on the left side and select the .ppk file for authentication through private key section.

- You can save this session before selecting OPEN. The saved session will help you connect to the instance directly for the next time.
- Once you are connected to the instance check what file systems are already attached to this instance. To do that type the following command:
  - **df**
- You will observe a root file system and a temporary file system.
- Now remember that you have attached the extra disk to the instance but it is not showing as a file system because you haven't mounted any file systems to that disk.
- For checking if the disk is really attached or not type the following commands:
  - **sudo su -** : This will make you the root user.
  - **fdisk -l** : This will show you the disks attached to this server.
- You will observe one disk with 8Gb space and another one with 1 Gb space. You will also observe that the kernels has changed the name of the disk from `"/dev/sdf"` to `"/dev/xvdf"`.
- Now to mount a file system on this disk type:
  - **mkfs -t ext3 /dev/xvdf** : This will tell the system to make this disk a file system given a partitioning table.
- But this is still not mounted on the system. To see what is mounted on the system type:
  - **cat /etc/fstab** : This will show the things that a system uses when it is booted.
- So what happens that even if we mount the disk to the system it will get unmounted everytime we reboot the instance. So one way to do it is to use an "echo" command and type a string and append that string to `"/etc/fstab"`. The another way to do it is to use a "narrator". So for that to not happen type:
  - **echo "<disk name> <directory name where you want to mount the disk> <type of file system> <checking file system>" >> <file where you wanna append the string as a new line>**
  - **echo "/dev/xvdf /data ext3 noatime 0 0" >> /etc/fstab**
- Now perform the following command in that order to mount the disk.
  - **mkdir /data** : Make a directory named /data
  - **mount /data** : Mount the disk in /data directory
  - **df** : Will show the file systems that are attached to the system.

### Step-3: Install postgresql, java and Payara

- Download and install postgresql on instance using wget command. After that I have created 2 users, user lord with super user access and user serf as normal user. And also allow postgresql to have external access from that user. After this, created a database that can accessed by serf user for querying and accessing the data. In the process of setting up the database, downloaded two files postgresql.conf and pg\_hba.conf from professor's website and replaced the original files with these files.

- Similarly downloaded java jdk 8 using wget command. Now I installed payara server zip file on my machine and used scp command to copy it on ec2 instance. After doing that I connect to my instance and unzip the file.
- After that I set the master password and the administrative password for payara.
- **The administrative password is “Jnm3660”.**

#### **Step-4: make database connection pool and deploy the application**

- Configured a connection pool for database in application server. Connection pool is for connecting serf user. And added multiple keys in connection pool for connecting to the database server. At first, I have encountered an error in pinging it to the database but after some search I have noticed that my database server was not running at that time. After that I added JNDI resources to use to access the database.
- After that I deployed the application on application server. And used it. It ran perfectly.