AWS Training Info:

==================

**Notes:**

Security Group:

Virtual firewall

Graphical user interface

Description automatically generated

MAC User:

Open Terminal:

* + Open an SSH client.
  + Locate your private key file. The key used to launch this instance is FirstEC2Instance.pem
  + Run this command, if necessary, to ensure your key is not publicly viewable.  
     chmod 400 FirstEC2Instance.pem
  + Connect to your instance using its Public DNS:
  + ec2-3-101-57-254.us-west-1.compute.amazonaws.com
  + Example:
  + ssh -i "FirstEC2Instance.pem" ec2-user@ec2-3-101-57-254.us-west-1.compute.amazonaws.com
  + ssh -i "myFirstpem.cer" ec2-user@54.183.0.81

Once login into EC2 follow the command:

* + sudo su
  + yum update -y (update ec2 instance related)

Windows User:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Download:

**putty.exe** (the SSH and Telnet client itself)

**puttygen.exe** (a RSA and DSA key generation utility)

Go to start button

Click on Putty and then choose putty gen for generate ppk file.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Once successfully imported click on "save private key" and store this as "sample.ppk".

Close Putty key generator window.

Open Putty terminal

Graphical user interface, text, application

Description automatically generated

Select "sample.ppk" file which you stored in local folder by clicking Browse button.(above ss)

Graphical user interface, text, application

Description automatically generated

Host name should be give **EC2 public ipv4 address** and click open button.(above ss)

After click on open you will see this EC2 terminal screen (below ss)

Text

Description automatically generated

<https://sumantmishra.medium.com/how-to-deploy-node-js-app-on-aws-with-github-db99758294f1>

#!/bin/bash

Yum update -y

Security group

Https

* + **NodeJS application runs on EC2:**

* + Install node version manager (nvm) by typing the following at the command line.

*curl -o-*[*https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh*](https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh)*| bash*

* + Activate nvm by typing the following at the command line.

*. ~/.nvm/nvm.sh*

* + Use nvm to install the latest version of Node.js by typing the following at the command line.

*nvm install node*

* + *Test that node and npm are installed and running correctly by typing the following at the terminal:*

*node -v*

*npm -v*

* + To install git, run below commands in the terminal window:

*sudo yum update -y*

*sudo yum install git -y*

* + Just to verify if system has git installed or not, please run below command in terminal:

*git — version*

* + Run below command to clone the code repository from Github:

*git clone*[*https://github.com/sumant-mishra/node-app.git*](https://github.com/sumant-mishra/node-app.git)

This will create a folder with name node-app.

*Install Dependencies:*

* + Now, move to the folder node-app by running below command in the terminal window:

*cd node-app*

* + To install dependencies, run below command in the terminal:

*npm install*

* + *To start the application, run the below command in the terminal:*

*node index.js*

***Configure security group to access via public URL:***

* + *By default, nobody can access the application without configuring the Inbound traffic configurations for the EC2 instance. To configure Inbound traffic for the EC2 instance, follow the below steps:*

Select the EC2 Instance and click on the security group link in the Description section.

Graphical user interface, text, application, email

Description automatically generated

* + *By clicking on the Edit button available in the Inbound tab, it will open Edit Inbound rules popup. By default, it will show SSH configurations. Since our application is configured for port number 3100, we need to add a new rule “Custom TCP Rule”. Enter port range as 3100 and select Source as “Anywhere”. After saving the changed rules, it will allow us to access our application from anywhere.*

Graphical user interface

Description automatically generated

***Access the application in browser***

* + *Now, we are ready and can access our application using both Public IP or Public DNS and port 3100 in browsers. For example,*
    - [*http://ec2-52-53-232-255.uswest-1.compute.amazonaws.com:3100*](http://ec2-52-53-232-255.uswest-1.compute.amazonaws.com:3100)

EC2 Assignments:

Graphical user interface, text, application

Description automatically generated

Commands:

Once login Ec2 instance:(after step 1)

>$ df

>$ lsblk

Command to mount disk(s):

* + List the attached disks: lsblk

You should see the list of unmounted disks and partitions. Note you need to prefix the disk with /dev/ path while mounting.

* + Format the disk with ext4 fs: sudo mkfs -t ext4 /dev/xvdf (For Formatting)
  + sudo mkdir /mnt/mydisk (For create filesystem)
  + Mount the disk at same directory : sudo mount /dev/xvdf /mnt/mydisk
  + cd /mnt/mydisk
  + ls -lrt
  + sudo vim test
  + df (can see the /dev/xvdf mount with /mnt/mydisk volume)
  + UnMount : sudo unmount -l /mnt/mydisk
  + Df (can't see the unmount the volume /dev/xvdf bos its unmounted)

After Step 8

Text

Description automatically generated

Note:

* + Snapshots are cheaper as compared to volumes(EBS). Bcos its stored in S3.
  + Once snapshots are created from the volume and you can still get the data back after mount the volume. Example: sudo mount /dev/xvdf /mnt/myseconddisk (this has already file which is test)

Registration Page:

SignUp Button. ---> rest api --> POST singup\_service

SignIn Button --> rest api --> GET singIn\_service

Forgot Password (link or button) --> rest api --> PATCH forgot\_pwd\_service

Delete User (link or button). --> rest api --> DELETE user