

CLOUD COMPUTING LAB

LAB 1: WEB HOSTING

Link: <https://www.youtube.com/watch?v=YNL22VWgm9Q>

Step 1: Creating EC2 instance

1. Create an EC2 instance by clicking on launch instance
2. Select Linux AMI and free tier (t2.micro) instance type
3. Configure security group
 - a. Add SSH
 - b. Add HTTP rule and to make it publicly accessible, remove ::/0 from source
 - c. Add HTTPS rule and to make it publicly accessible, remove ::/0 from source
4. Review and launch
5. Add a key pair (existing / new) and click acknowledge
 - a. If a new key pair is added give the name and download the key pair
 - b. The key pair is of pem extension
6. Launch Instance
7. Rename the created instance

Step 2: Creating S3 bucket for web application file

1. Go to S3 console
2. Click create bucket

3. Give bucket name then set region to Mumbai to avoid latency
4. Click on create and bucket is created
5. To make it public
 - a. Click on created bucket
 - b. Edit public access setting
 - c. Uncheck block all public access and ensure all the checkboxes are unchecked
 - d. Click save and confirm by typing confirm
6. Open the bucket by clicking on it
7. Click upload -> Add files -> select **zip file** -> click upload
8. To set object as public
 - a. click on the uploaded zip file
 - b. click actions -> search for make public in the drop down and click
9. go back to ec2 instance
10. Convert the downloaded pem file to ppk file using puttygen
 - Open puttygen and click on load
 - Select the downloaded pem file
 - Click on save private key
 - The downloaded file will be ppk extension
 - To access the EC2 instance and open SSH, ppk is mandatory
 - Close file
11. Open putty software
 - a. Copy paste public IPv4 address from the ec2 instance
 - b. Paste it in host name in the putty software in the format “**ec2-user@IPv4 address**”
 - c. Ensure SSH radio button is enabled

- d. Go to Category on the left side and click SSH
- e. Click on Connection -> SSH -> Auth
- f. Click browse and upload ppk file -> click open
- g. Click yes on the pop up window of security alert
- h. Terminal opens
- i. Type the following commands in the cmd

Step 1: Set to root user

`sudo su`

Step 2: Update package for EC2 instance

`yum update -y`

Step 3: Install Apache to run the website

`yum install httpd -y`

Step 4: Check the path using pwd

Step 5: Change directory to html

`cd /var/www/html`

Step 6: List the directory using ls command -> no files

Step 7: Get the files from S3

- a. Click on the zip file in s3 bucket in AWS
- b. Copy paste **Object URL** under Overview
- c. `wget s3url`

- d. use `ls` command to get the list of files -> uploaded zip file can be seen

Step 8: To unzip the uploaded zip file

- a. Unzip `filename.zip` (the same file name in the previous output along with the extension)
- b. Type `ls` -> zip file(red color) and extracted file(blue color) name is printed

Step 9: To move all the files to EC2

- a. `mv filename/* .` (filename is in blue color)
- b. type `ls` -> list out all the files in EC2
- c. to ensure if we are in the right path type `pwd`
- d. the path `“/var/www/html”` is printed

Step 10: To run Apache server

`service httpd start`

Step 11: To get the output

- a. go to EC2 instance
- b. copy IPv4 address
- c. paste the copied address in a new tab to see the website

LAB2: JAVA COMPILER

Link: <https://drive.google.com/file/d/1jmx3IPfmEUBQHk3J6M2xUK-Y2sFsetLb/view>

Note: IOPS is Input Output Operations per second – CPU burst rate – default value is 100/3000

Step 1: Create EC2 instance

1. Create an EC2 instance by clicking on launch instance
2. Select Linux AMI and free tier (t2.micro) instance type
3. Configure security group
 - a. Add SSH
 - b. Add Custom TCP rule with port range 8080 and remove ::/0
 - c. Review and launch -> add key-pair -> acknowledge -> launch
 - d. Rename the created instance

Step 2: Go to putty software

1. Copy paste IPv4 address from EC2 instance
2. Select SSH -> Auth -> Load ppk file -> Click yes to security alert -> Terminal Opens
3. Login as **ec2-user**
4. Put the following commands in the putty terminal

Step 1: Update and check java version

```
sudo yum update  
java -version
```

Step 2: To make major changes

- a. `sudo su` #to login as root user
- b. `mkdir java` #to install java create a directory
- c. `cd java` #change directory to java

Step 3: If any older version of java is found

`sudo yum remove java-VersionNumber-openjdk`

Step4: If no version of java is found

- a. `sudo yum install java-1.8.0-openjdk` #displays all the installed packages
- b. give consent by entering y to download the packages
- c. `java -version` #check the installed java version

Step 5: Download Apache Tomcat

- a. google search for tomcat
- b. click on the first link
- c. click on download and select any version (3 versions available)
- d. scroll down go to Binary Distributions -> Core -> **tar.gz** file
- e. right click on the file and copy the link
- f. go back to putty window

Step 6: To pull the data

- a. `wget` copied_link

i.e. `wget` <https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.70/bin/apache-tomcat-8.5.70.tar.gz>

- b. to verify enter **ls** command which will print the files
- c. Now we have the apache file inside the java folder

Step 7: Extract the file

- a. `tar xvfz apache-tomcat-8.5.58.tar.gz`
- b. use `ls` command to view

Step 8: Change directory to the apache file as we have to work with 3 main files – bin, conf and webapps

`cd apache-tomcat-8.5.58/`

Step 9: To start the Tomcat server

#find command file inside the bin

- a. cd bin
- b. ./startup.sh #to start the server
- c. ps -ef | grep tomcat
 #to identify the keyword "tomcat" to verify if tomcat is installed and running

Step 10: To pull data from local host web browser

wget <http://localhost:8080>

Note: index.html is always the home page

Step 11: To get the Tomcat running

- a. Copy IPv4 DNS address from EC2 instance
- b. Paste the address to a new tab and add **:8080** at the end of the url
- c. Now Tomcat is installed and running
- d. Click on Manager App button in the right side of the web browser
- e. Manager App requires username and password. If entered it does not allow you because it is with default attributes.
- f. **To change the attributes modify few files**
Go back to terminal window

Step 12: Currently it is in bin folder and we should move to subfolder named **"webapps"**

- a. cd ..
- b. ls webapps/manager/META-INF/
- c. vi webapps/manager/META-INF/context.xml
 #to edit the context file
- d. Comment <Valve className> (2 lines)
- e. Press escape :wq #to close the file

Step 13: Edit the conf file

- a. vi conf/tomcat-users

- b. Insert 2 lines at the end of the conf file before the close tag of
`</tomcat-users>`
- c. `<role rolename="manager-gui/>`
`<user username="tomcat" password="tomcat" roles="tomcat,`
`manager-gui"/>`
- d. Press escape :wq

Step 14: Change the directory to the bin as it holds the startup and shutdown options

- a. `cd bin`
- b. `shutdown.sh` `#to restart shutdown and start again`
- c. `./startup.sh` `#to start`

Note: It is running successfully. Now we should be able to get into the Tomcat server.

Step 14: Check if you can enter Tomcat Manager App by giving the login credentials

- a. Username = "tomcat"
- b. Password = "tomcat"
- c. It directs to Tomcat Web Application Manager

Step 15: To deploy the war file

- a. Scroll down -> **War file to display** section -> click on choose file and upload -> deploy
- b. The war file name will be visible in the table after deploying
- c. Click on the war file name to see the output
- d. You can verify the IPv4 DNS address in both the browser as well as EC2 instance console

Note: We have edited 2 files i.e. context.xml and tomcat-users.xml

LAB 3: ELASTIC BLOCK STORE

Link: <https://drive.google.com/file/d/1woSG7JWQ84jnBzJcfWiV5cLa4xnKv5HB/view>

Steps:

1. Create EC2 instance with no changes
2. Click on EBS and add volume by clicking on CREATE VOLUME
3. Make changes to the configuration
 - a. Size = 1GiB
 - b. set the availability zone as per the EC2 instance and
 - c. add a tag with key = "name" and value="My Volume"
4. Attach the volume to the instance created by clicking on Actions -> Attach Volume -> insert instance id -> Attach
5. Status will turn to **in-use**
6. Open putty and type the commands to configure the new volume

Step 1: Creating directory and file system and mount in PUTTY

- a. `df -h` #list out file systems attached in server
- b. `sudo mkfs -t ext3 /dev/sdf` #create a file system in virtual machine
- c. `sudo mkdir /mnt/data-store` #create a directory to mount storage
- d. `sudo mount /dev/sdf /mnt/data-store` #to mount new volume
- e. `echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 12" | sudo tee -a /etc/fstab` #mount volume whenever instance get started
- f. `sudo mkfs -t ext3 /dev/sdf` #create file system of type ext3
- g. `cat /etc/fstab` #view the file

h. `df -h`

Step 2: Create a text file in PUTTY

- a. `sudo sh -c "echo Hello my name is Asha Thampi. I am creating new Volume > /mnt/data-store/asha.txt"`
- b. `cat /mnt/data-store/asha.txt`

Step 3: Create Snapshot in AWS

- a. Make a snapshot of the volume by clicking on the volume name in EBS console.
- b. Actions -> Create Snapshot
- c. Add tag and give key = "name" and value="My Snapshot"

Step 4: Delete the created text file in PUTTY

- a. `sudo rm /mnt/data-store/asha.txt`
- b. `sudo cat /mnt/data-store/asha.txt` #to confirm deletion
- c. `sudo ls /mnt/data-store/` #to confirm deletion

Step 5: Restore file from snapshot created in AWS

- a. Go to snapshots under EBS click on it
- b. Actions -> Create Volume with no modifications except availability zone if different
- c. Create tag with key = "name" and value = "Restored Volume"
- d. Go to EBS volume and click on Restored Volume and link it to the instance
- e. Actions -> Attach Volume -> Select Instance

Step 6: Restore file in PUTTY

- a. `sudo mkdir /mnt/data-restore` #new directory for restored volume
- b. `sudo mount /dev/sdg /mnt/data-restore`
- c. `ls /mnt/data-restore` #confirm file restore
- d. `cat /mnt/data-restore/asha.txt` #confirm file content restore