Scribo
Page No.:
Date: 9 / 6 / 22

AWS Server. Complete steps.

steps to be followed.

1) login to AWS management Console.

2) Create an AWS account.
   - ↳ E-Mail address
   - ↳ Account User Name.
     (Email id verification)

3) User Password Creation.
   - ↳ Continue.

4) Fill in the Details.
   - ↳ Credit / Debit Card
     Address
     Phone no. etc.

5) Personalize your experience
   - ↳ student (Role)
     websites & web Apps (intrested in)

★ Account is Created.

Scribo
Page No.:
Date: / /

6) Then Sign in to Console
↳ Root user login

7) Click on ⇒ Switch to new Console Home.

8) Change Data Centre to 'Mumbai'

9) Click on services → Compute → EC2

EC2 is used to create Virtual server on cloud.

10) Now we have to Create an instance (virtual Server)

EC2 → instances

Select launch instance

11) Give name to the instance.
Eg:- IET-Node1's server
→ launch instance

Select → Create new key Pair
(give key pair Name)
⇒ This will create a key pair ~~Na~~
• Pem file

Select .ppk file (Public Private key)

12) → Select ⇒ Create security group
☑ Allow SSH ~~traffic~~ from

☑ Allow HTTP Traffic from internet

⇒ Launch instance.
(Now the instance is successfully launched)

13) Click on the ID that we see.
We can now see the instance (running)
click on the check box & then connect

Scribo
Page No.:
Date: / /

13) Now we see instance ID & public IP address. (Dynamic)

Now click on connect

★★ Once we click the Connect button. we can see a linux command Prompt (linux Terminal) for mumbai Data Centre.

14) Now we have to update the linux server using "Sudo yum Update" Command.

15) Now we have to check if we have git in the instance or not.
Command ⇒ Git -v

16) if we dont have git installed
Command ⇒ Sudo yum install git.

★ These procedures are needed to be performed for every instance.

Scribo
Page No.:
Date: / /

17) Now To Clone git hub repo. to host a website on our server.

18) Command ⇒ git clone <the given URL>

~~github~~ git Clone https://github.com/Ravi Tambde/webapp.git.

19) ⇒ LS (To check webapp folder created or not)

20) ⇒ Cd webapp (Change directory to webapp)

21) ⇒ LS (To check the contents in the folder of webapp)

↳ Dockerfile package.json README.md server.Js.

Server.Js is a JavaScript file. for that we need Node.Js file.

Now we have to install Nodejs

Command
⇒ Sudo curl -SL https://rpm.nodesource.com/setup_12.x | sudo bash - (setting path for nodejs)

⇒ npm install (To download library)

Scribo
Page No.:
Date: / /

NodeJs successfully installed.

22) ⇒ sudo yum -y install nodejs
↳ node -v (to check version)

23) → node server.js -d (detach mode)
Server is running on Port 8081

24) Now to check public server IP add.

Copy the Public IP address → To go new tab → http://IP address:8081 enter.

★ If the tab is not running that means firewall is not set properly

Then go to instance ID → security → security group. → Edit inbound rules.

Custom → Anywhere. IPV4

★ By configuring this rule we tell the firewall to accept the request from HTTP & send it to server

Scribe
Page No.:
Date: / /

select All traffic instead of HTTP.

Then save Rule.

★★ If we do not set firewall Rules Then our request will not go the server.

25) After <u>step 23</u> Press <u>Ctrl+C</u>

This will stop the server. Now if The server is stopped the site will also stop working.

— x —

26) To stop/Terminate the instance.

go to instances → Click on the check box → instance state → stop instance.

if we stop then it will be there using some space on the server.

if we terminate it will be deleted. (Permanantly.)

Scribo
Page No.:
Date: / /

Now to run a Java file in AWS server

first make a folder
→ mkdir Java.
→ Cat Hello.Java.

Cd Java.
nano hello.Java.
(This will open an editor in linux)

Write a hellowrld prog. in that.

```
Public class Java {
    public static void main(string
                    args[])
{
  Sysout ("Helloworld");
}
}
```

—— X ——

But first we have to check if Java is installed or not.

Search on google how to install Java on AWS EC2 instance. (Cloud Affaire)

Scribo
Page No.:
Date: / /

Copy the link given for Amazon linux.

installing Jdk.

⇒ Java -v.
Cat hello.Java.

Class name & file name should be same.

Scribo
Page No.:
Date: / /

Docker & Containerization.

1) Create a new instance with same procedure as previous.

2) Do not install node JS as we want to create an OS for server.

3) Search for docker in linux. if not available

4) Then search for docker version.
   ⇒ sudo yum info docker

5) Now install the docker
   ⇒ sudo yum install docker

After installation check version
docker -v
version :- 20.10.13

6) To check wether the docker is running or not (Enable docker)

⇒ sudo systemctl enable docker.
~~service~~

7) To Start docker:-

⇒ sudo systemctl start docker.service

8) To stop docker:-

sudo systemctl stop docker.service

—— x ——

☆ Now we created the docker level now we want to run an application on the server. for that we need to give source file, json, docker file

without docker file it cannot create image!

The docker file will contain instruction for the Container which will instruct the Container ~~or~~ for how to Create the images.

Scribo
Page No.:
Date: / /

To create the image we first clone the git repo on our linux server to access that docker file.

Command ⇒ git clone https://< ~~path~~ link of repo>

we can see webapp folder

⇒ cd webapp

* Now to build the image is a task of a ~~f~~ docker file ~~whi~~ we have to give command to docker.

Command ⇒ sudo docker build -t webapp

it will install node js & npm file in background not on the linux but on the container.

To check wether image is created or not

⇒ sudo docker image.

Scribo
Page No.:
Date: / /

Now the image is created but it has to run. if it isnt running then it will not create instance.

cmd ⇒ sudo docker run --name expstcvet
-d -p 8081:8081
inside outside.
(webapp)
folder name.

⇒ sudo docker ps.

it will show image name, when it was created, status, ports. etc.

☆ After this our virtual machine is running we can now give the ip address to the customer & they can run it on their machine.

— x —

Now if we want to stop that website. just stop the container.

Scribo
Page No.:
Date: / /

To stop that container.
(we need ID of cont)

⇒ sudo docker kill <Container ID>

This command will stop the on going image on the container & eventually the site will stop working

— x —

To restart the image just give the command for 'run' with diff name.

⇒ sudo docker run --name webserver -d -p 8081:8081 webapp

The site will be accessible.

* In the same way we can run files of Javascript, Java runtime, python django, 3 different websites will run indipendently using docker commands.

Scribo
Page No.:
Date: / /

To remove all images from docker.

⇒ sudo docker image prune --all

for all images

- for a particular image just name it.