

PROJECT-3

Step1:Create two linux instances,Use the first free linux AMI

The screenshot displays the AWS Management Console interface for the us-east-2 region. The left sidebar shows the navigation menu with options like EC2 Dashboard, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The main content area shows a list of EC2 instances. Two instances are listed: Linux2 and Linux1, both in a 'running' state. Below the list, the 'Description' tab is selected, showing details for the instances.

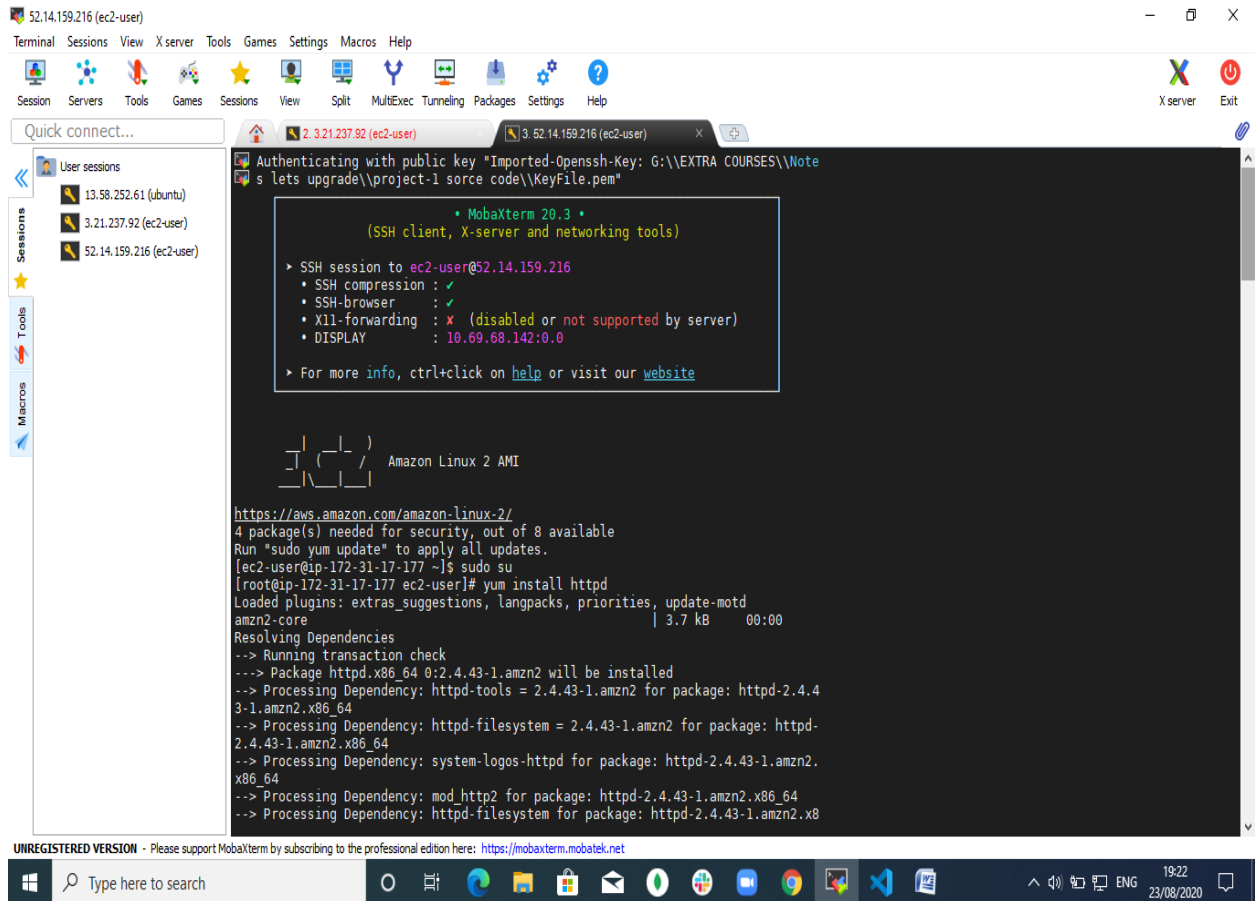
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Linux2	i-0988d104a71181750	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-52-14-159-216.us-...
Linux1	i-09406e14fd0fb6652	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-3-21-237-92.us-aa...

Instances: i-0988d104a71181750 (Linux2), i-09406e14fd0fb6652 (Linux1)

Description | Status Checks | Monitoring | Tags

- i-0988d104a71181750: ec2-52-14-159-216.us-east-2.compute.amazonaws.com
- i-09406e14fd0fb6652: ec2-3-21-237-92.us-east-2.compute.amazonaws.com

Step2:Launch both instances using MobaXterm



3.21.237.92 (ec2-user)

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

home/ec2-user/

Sessions

- Name
- ..
- .ssh
- .bash_logout
- .bash_profile
- .bashrc

Tools

Macros

Sftp

Remote monitoring

☐ Follow terminal folder

```
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>
<body>
<div>
<h3>Server2</h3>
<form action="action_page.php" method="post">
<div>
<label for="uname"><b>Username</b></label>
<input
  type="text"
  placeholder="Enter Username"
  name="uname"
  required
/>
<label for="psw"><b>Password</b></label>
<input
  type="password"
  placeholder="Enter Password"
  name="psw"
  required
/>
<button type="submit">Login</button>
</div>
</form>
</div>
</body>
</html>

[root@ip-172-31-25-194 html]#
[root@ip-172-31-25-194 html]# service start httpd
The service command supports only basic LSB actions (start, stop, restart, try-restart, reload, force-reload, status). For other actions, please try to use systemctl.
[root@ip-172-31-25-194 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-25-194 html]#
```

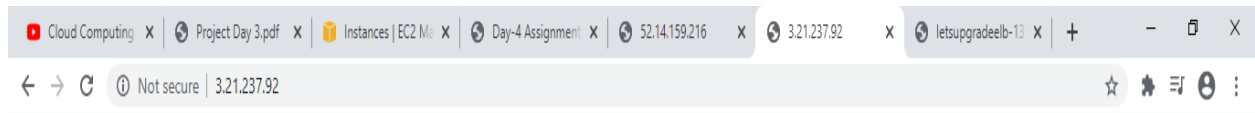
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Type here to search

19:21
23/08/2020

Step4:Host html login webpage on both servers

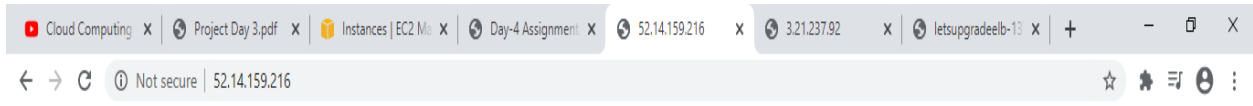
Step5:Check is application is deployed on both servers by copy pasting the public ip of the servers into the browser.



Server2

Username Password



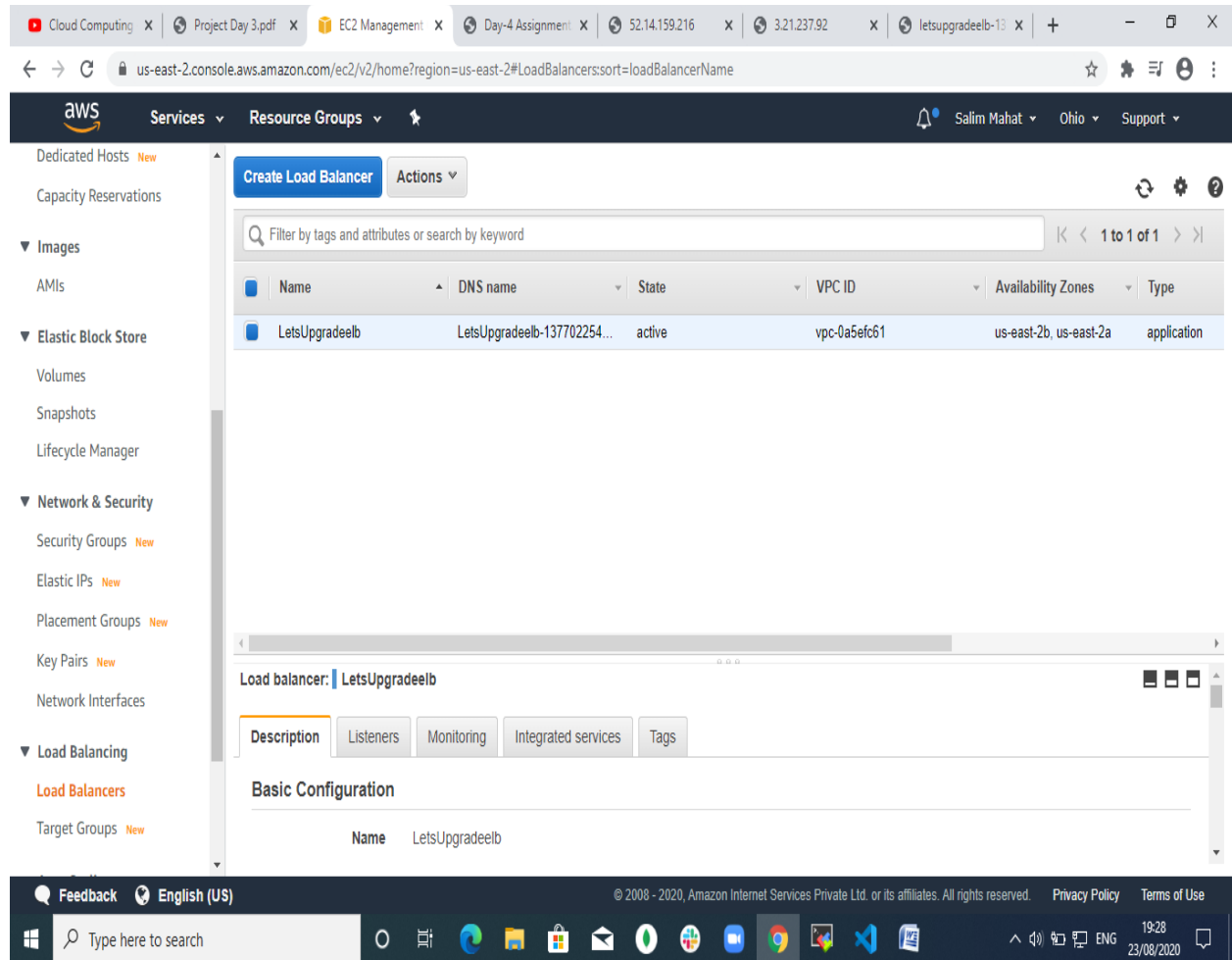


Server1

Username Password



Step6:Create a application Load balancer with the above two instances as targets



The screenshot shows the AWS Management Console interface. The left sidebar contains navigation links for various services, including Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area displays the 'Load Balancers' page. At the top, there is a 'Create Load Balancer' button and an 'Actions' dropdown. Below this is a search bar and a table listing the load balancers. The table has columns for Name, DNS name, State, VPC ID, Availability Zones, and Type. One load balancer, 'LetsUpgradeelb', is listed with a state of 'active'. Below the table, the 'Description' tab is selected, showing the 'Basic Configuration' section with the name 'LetsUpgradeelb'.

Name	DNS name	State	VPC ID	Availability Zones	Type
LetsUpgradeelb	LetsUpgradeelb-137702254...	active	vpc-0a5efc61	us-east-2b, us-east-2a	application

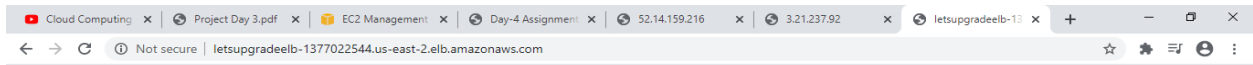
Load balancer: LetsUpgradeelb

Description | Listeners | Monitoring | Integrated services | Tags

Basic Configuration

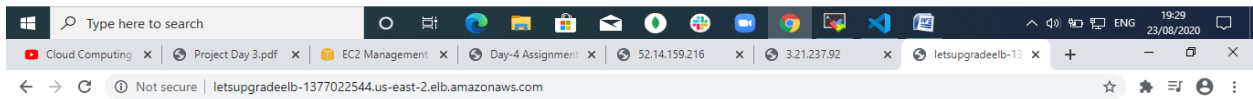
Name LetsUpgradeelb

Step7:Check the functioning of ELB



Server2

Username Password



Server1

Username Password

