Lab Assignment 2

Solution

Steps to follow for executing part 1:

1. First of all create 5 ec2 instances in the same way you did in lab assignment 1, but in this use the same key which you generated previously in Lab assignment 1.
2. Now connect all these instances, and write down the below code to install NGINX.
    a. sudo -s // this will make you a super user.
    b. yum update -y // this will install updates that are necessary for installing nginx
    c. yum install -y nginx // this will install nginx
    d. service nginx start // this will start the nginx service.
    e. chkconfig nginx on
3. Now after installing NGINX you can copy the public dns of each instance and paste it in the browser to see if the nginx service is working or not, if it is working it will display a nginx page or it will display that this site cannot be reached.
4. Now before you do this just follow this Actions>inbound>edit>add rule> select “http”> click ok. Only after doing this you can get the nginx page.
5. Now once you checked that service is working properly, create an html file in every server shell only by wirtting cat > /usr/share/nginx/html/index.html. This will creat a new index.html file
6. After creating this file just paste the html code given by the professor here. Now by wirtting ”vi /usr/share/nginx/html/index.html” and pressing “i” you can actually edit the code given by the professor and change [SERVER_ID] to Server <corresponding server>.
7. Now if you copy the public dns of server instance in which you did above process you can see the change that you made replacing [SERVER_ID].
8. Do this process for every server instance.
9. Now for load balancer instance create a file like this “cat > /etc/nginx/nginx.conf” and then paste the other code that professor has given only for load balancer. Now in that also using vi command edit the code and replace [SERVER_PUBLIC_DNS_NAME] with every server public dns name.
10. After editing the code reload the load balancer by writing “/etc/init.d/nginx reload”.
11. After reloading the load balancer, paste the code starting with “#” given by the professor.
12. After this write “chmod 777 visit_server” to change the permission from read only to read-write.
13. Now visit the server by writing “ ./visit_server -d <load balancer public dns name>” to see the output of the first server weights given by the professor. The output are below:

Load balancer	Server 1	Server 2	Server 3	Server 4
1	1	1	1	1

```
-------------------------
 Summary
-------------------------
Server1 visit counts : 500
Server2 visit counts : 500
Server3 visit counts : 500
Server4 visit counts : 500
Total visit counts : 2000
```

2 1 2 3 4

```
-------------------------
 Summary
-------------------------
Server1 visit counts : 200
Server2 visit counts : 400
Server3 visit counts : 600
Server4 visit counts : 800
Total visit counts : 2000
```

3 1 2 1 2

```
-------------------------
 Summary
-------------------------
Server1 visit counts : 333
Server2 visit counts : 667
Server3 visit counts : 333
Server4 visit counts : 667
Total visit counts : 2000
```

14. Now to get the other two results just repeat all the steps after step 9-13.

Steps to follow for executing part 2:

1. For creating the instance using script I installed AWS Command Line Interface. For that go to http://docs.aws.amazon.com/cli/latest/userguide/tutorial-ec2-ubuntu.html and follow the steps.
2. By executing all the commands the output will be like below:

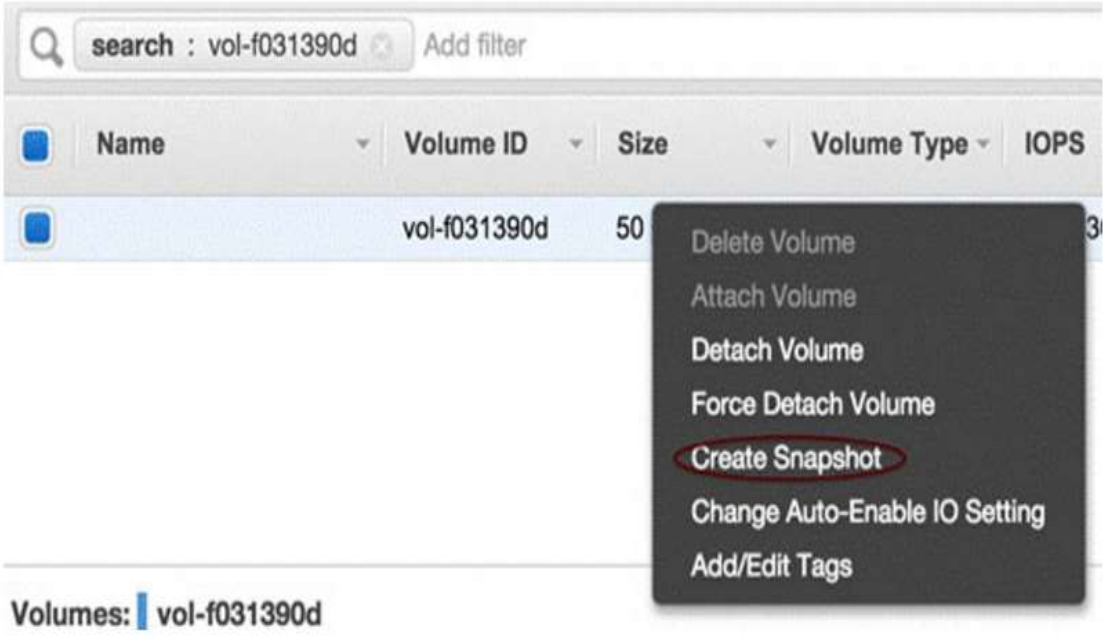
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\devan>aws configure
AWS Access Key ID [None]: AKIAJMKZH65PTWCAYVYA
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]: json

C:\Users\devan>aws ec2 create-security-group --group-name devenv-sg --description "security group for developer environment in EC2"
{
    "GroupId": "sg-6553a81e"
}

C:\Users\devan>aws ec2 authorize-security-group-ingress --group-name devenv-sg --protocol tcp --port 22 --cidr 0.0.0.0/0

C:\Users\devan>aws ec2 create-key-pair --key-name devenv-key --query 'KeyMaterial' --output text > devenv-key.pem

C:\Users\devan>chmod 400 devenv-key.pem
'chmod' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\devan>aws ec2 run-instances --image-id ami-29ebb519 --security-group-ids sg-6553a81e --count 1 --instance-type t2.micro --key-name devenv-key --query 'Instances[0]
.InstanceId'
"Instances[0].InstanceId"

C:\Users\devan>aws ec2 run-instances --image-id ami-29ebb519 --security-group-ids sg-6553a81e --count 1 --instance-type t2.micro --key-name devenv-key --query "Instances[0]
.InstanceId"
"i-0d1b1d83804ae9878"

C:\Users\devan>aws ec2 describe-instances --instance-ids i-0d1b1d83804ae9878 --query "Reservations[0].Instances[0].PublicIpAddress"
"52.38.90.175"
```

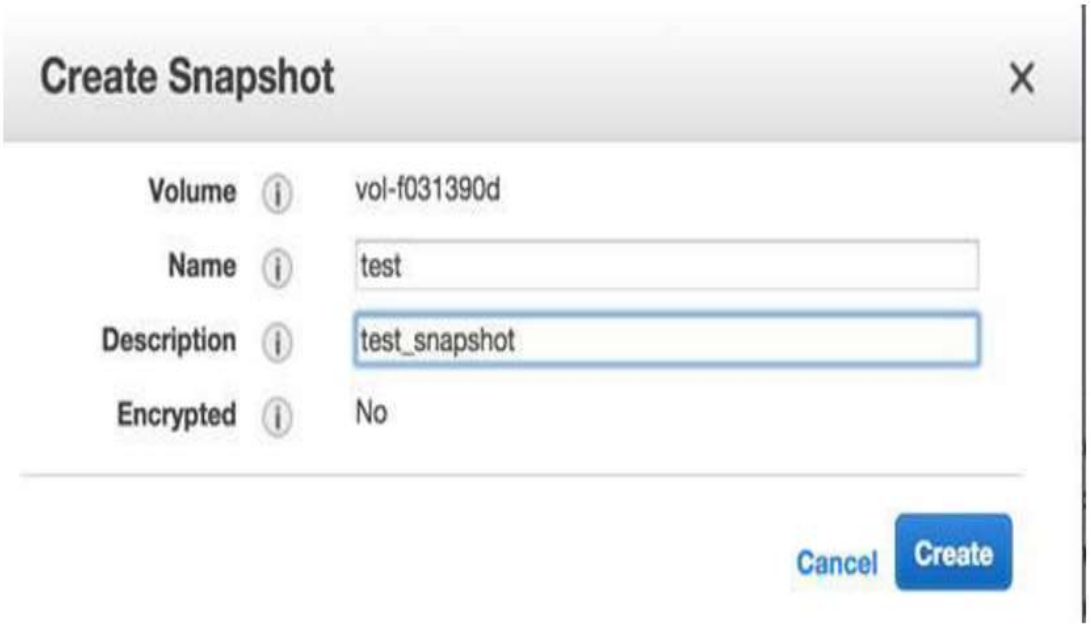
3. The first step is to install aws CLI to run the script to create an instance.
4. After installing the CLI configure the CLI by writing “aws configure”, then give the access key id and the secret access key you got when downloaded the CLI from above site.
5. Create a group id by writing the below command:
   aws ec2 create-security-group --group-name devenv-sg --description "security group for development environment in EC2"
   ```
   {
     "GroupId": "sg-b018ced5"
   }
   ```
   aws ec2 authorize-security-group-ingress --group-name devenv-sg --protocol tcp --port 22 --cidr *0.0.0.0/0*
6. Now create a key pair by writing “**ec2 create-key-pair --key-name devenv-key --query 'KeyMaterial' --output text > devenv-key.pem”.**
7. In this way an instance is created using AWS CLI.
8. Now, for tcpdump part you need to install tcpdump. To do this you need to write “yum install tcpdump” in your load balancer shell.
9. After installing tcpdump execute the command in load balancer shell “sudo tcpdump” to see the number of packets exchanged. You will get the output like below.

```
19:44:35.743804 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 191616:192320, ack 2657,
 win 262, length 704
19:44:35.743964 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [P.], seq 2657:2721, ack 190698, w
in 259, length 64
19:44:35.782004 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [.], ack 2721, win 262, length 0
19:44:35.826742 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [.], ack 191616, win 256, length 0
19:44:35.826769 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 192320:193296, ack 2721,
 win 262, length 976
19:44:35.826876 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 193296:193520, ack 2721,
 win 262, length 224
19:44:35.826913 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 193520:193760, ack 2721,
 win 262, length 240
19:44:35.885714 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [.], ack 192320, win 259, length 0
19:44:35.885741 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 193760:194240, ack 2721,
 win 262, length 480
19:44:35.915625 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [.], ack 193520, win 255, length 0
19:44:35.915648 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 194240:194704, ack 2721,
 win 262, length 464
19:44:35.915730 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 194704:194928, ack 2721,
 win 262, length 224
19:44:35.973065 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [.], ack 194240, win 259, length 0
19:44:35.973092 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 194928:195408, ack 2721,
 win 262, length 480
19:44:35.973208 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 195408:195632, ack 2721,
 win 262, length 224
19:44:36.003069 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [.], ack 194928, win 257, length 0
19:44:36.003094 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 195632:196112, ack 2721,
 win 262, length 480
19:44:36.003197 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 196112:196336, ack 2721,
 win 262, length 224
19:44:36.064902 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [.], ack 195632, win 259, length 0
19:44:36.064929 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 196336:196816, ack 2721,
 win 262, length 480
19:44:36.065032 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 196816:197040, ack 2721,
 win 262, length 224
19:44:36.092873 IP host-155-246-210-159.dhcp.stevens-tech.edu.57990 > ip-172-31-5-48.us-west-2.compute.internal.ssh: Flags [.], ack 195632, win 259, options
[nop,nop,sack 1 {196112:196336}], length 0
19:44:36.092897 IP ip-172-31-5-48.us-west-2.compute.internal.ssh > host-155-246-210-159.dhcp.stevens-tech.edu.57990: Flags [P.], seq 197040:197520, ack 2721,
 win 262, length 480
^C
831 packets captured
833 packets received by filter
0 packets dropped by kernel
[root@ip-172-31-5-48 ec2-user]#
```

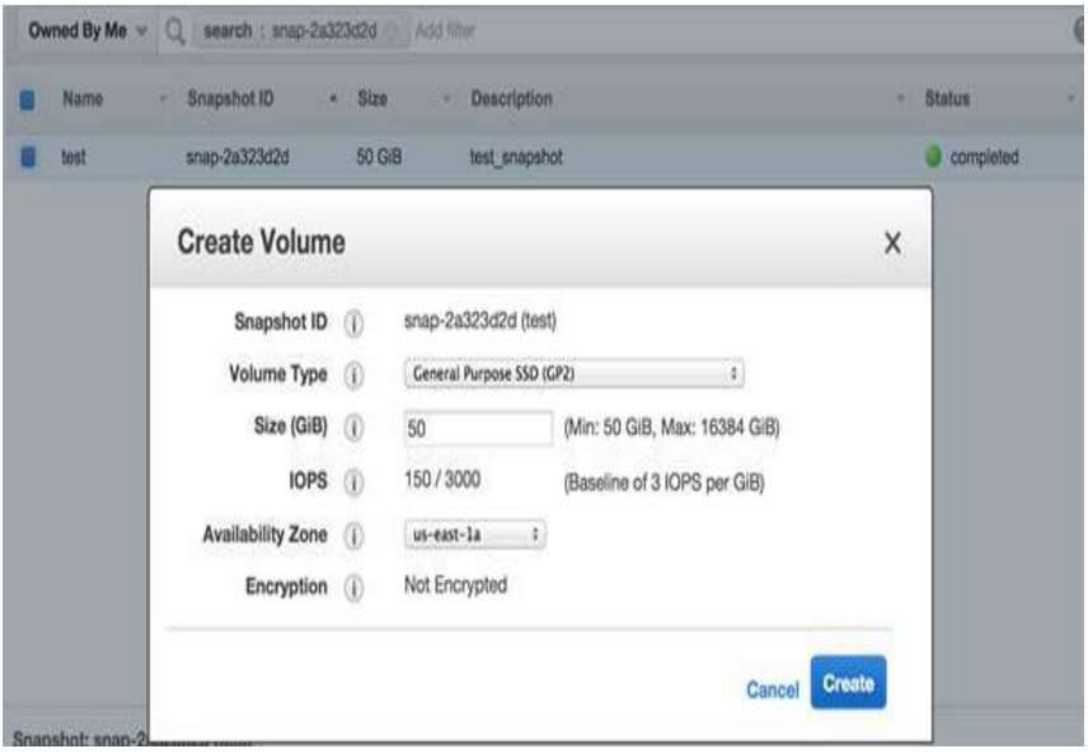
Steps to follow for executing part 3:

1. First, create a new instance in which you will retrieve the backed-up file from the source instance.
2. Now from source instance create a snapshot of the volume in which your file is stored.





3. Now after creating the snapshot just create a volume of this snapshot and attach it to the new instance.



4. Now before attaching the new volume to new instance just detach the volume that is already associated with the new instance.
5. After attaching the volume connect your new instance and see if the file created in source instance is retrieved or not. The output will be like below:

```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key" from agent
Last login: Tue Apr  4 21:40:13 2017 from host-155-246-210-159.dhcp.stevens-tech
.edu

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
12 package(s) needed for security, out of 43 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2017.03 is available.
[ec2-user@ip-172-31-8-113 ~]$ ls -lrt
total 4
-rw-rw-r-- 1 ec2-user ec2-user 33 Apr  4 21:41 test
[ec2-user@ip-172-31-8-113 ~]$ cat test
Hello! This is  a test program!!
[ec2-user@ip-172-31-8-113 ~]$
```