

OSL840-ASSIGNMENT1-LOHAN VADDEPALLY-150115236

Community

Search [Alt+S]

United States (N. Virginia) | voclabs/user3815291=lvaddepally@myseneca.ca @ 2616-1044-8825

EC2 > AMIs

Amazon Machine Images (AMIs) (2) info

Owned by me | Find AMI by attribute or tag

<input type="checkbox"/>	Name	AMI name	AMI ID	Source	Owner	Visibility	Status
<input type="checkbox"/>	www-for -asg1-p1		ami-07a6226efbe3894f1	261610448825/www-for -asg1-p1	261610448825	Private	✓
<input type="checkbox"/>	www-for -asg1-p3		ami-0ce1dcb0c6dfbcb7	261610448825/www-for -asg1-p3	261610448825	Private	✓

Select an AMI

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Community

Search [Alt+S]

United States (N. Virginia) | voclabs/user3815291=lvaddepally@myseneca.ca @ 2616-1044-8825

EC2 > Security Groups > sg-0545438d025645415 - ops345routersg

Volumes (1/14) info

Saved filter sets | Choose filter set | Search

Last updated 1 minute ago | Actions | Create volume

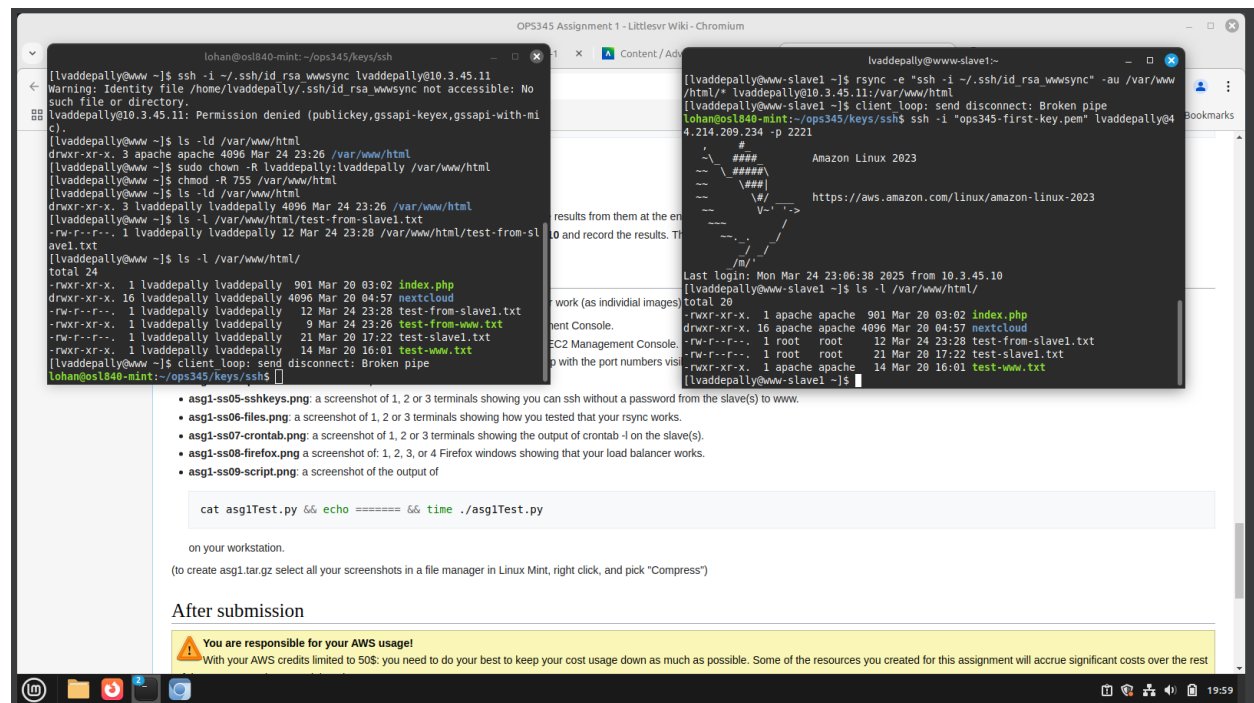
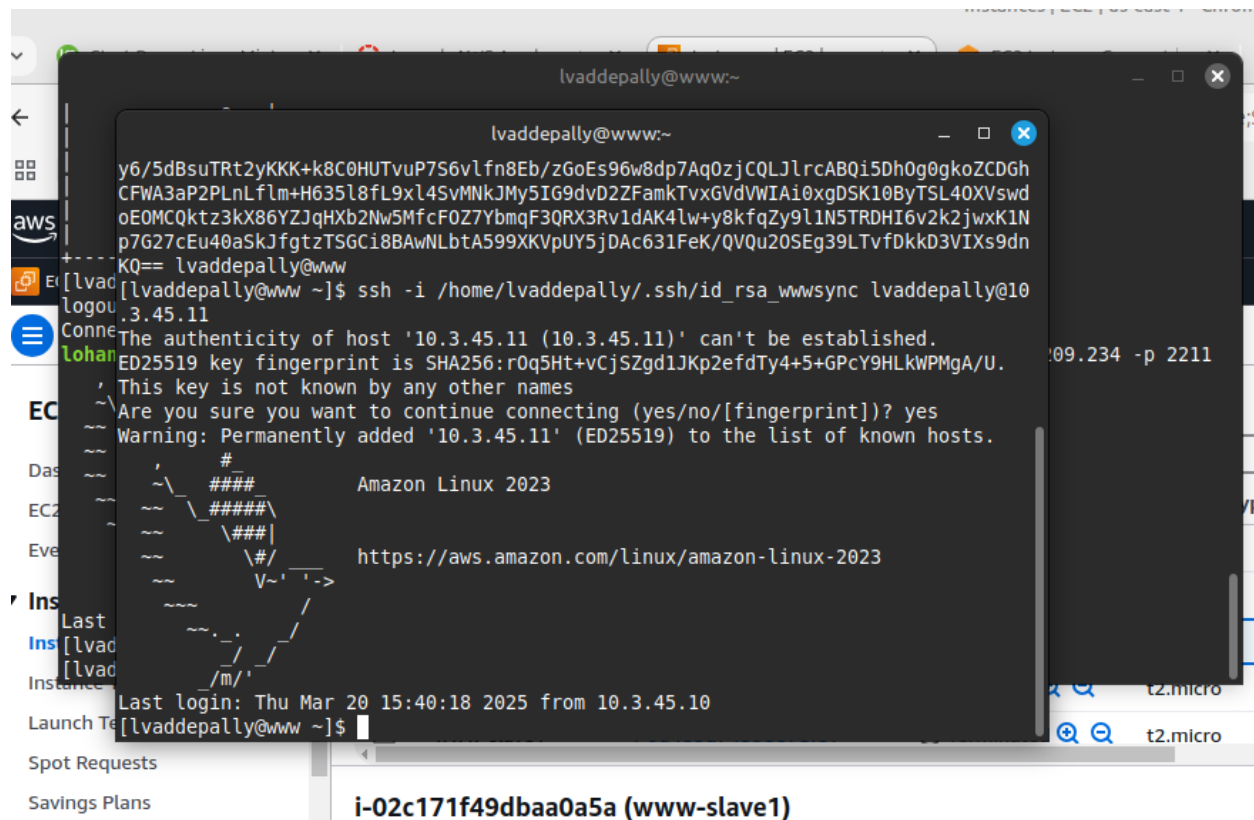
<input checked="" type="checkbox"/>	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created	Availability
<input checked="" type="checkbox"/>	wwwdataslave3	vol-0e976baefdb7a9c8f	gp3	8 GiB	3000	125	snap-0a20bef...	2025/03/24 23:36 GMT-4	us-east-1a
<input type="checkbox"/>	-	vol-01dc1b91a5909edc7	gp3	1 GiB	3000	125	snap-0578fbc...	2025/03/24 23:36 GMT-4	us-east-1a
<input type="checkbox"/>	-	vol-0a4d51d5aa3b5e3d0	gp3	1 GiB	3000	125	snap-0578fbc...	2025/03/24 23:28 GMT-4	us-east-1a
<input type="checkbox"/>	wwwdataslave2	vol-095bc669035fc3d41	gp3	8 GiB	3000	125	snap-0a20bef...	2025/03/24 23:28 GMT-4	us-east-1a
<input type="checkbox"/>	-	vol-07fda563fc29bf657	gp3	1 GiB	3000	125	snap-0bd67d2...	2025/03/20 11:19 GMT-4	us-east-1a
<input type="checkbox"/>	wwwdataslave1	vol-07eaea412a68e6a47	gp3	8 GiB	3000	125	snap-0825773...	2025/03/20 11:19 GMT-4	us-east-1a
<input type="checkbox"/>	-	vol-09ae2c225fe982faa	gp3	4 GiB	3000	125	-	2025/03/20 03:11 GMT-4	us-east-1a

Volume ID: vol-0e976baefdb7a9c8f (wwwdataslave3)

Details	Status checks	Monitoring	Tags
Volume ID vol-0e976baefdb7a9c8f (wwwdataslave3)	Size 8 GiB	Type gp3	Status check ✓ Okay
AWS Compute Optimizer finding ⚠ This user is not authorized to call AWS	Volume state ✓ In-use	IOPS 3000	Throughput 125

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

OSL840-ASSIGNMENT1-LOHAN VADDEPALLY-150115236



OSL840-ASSIGNMENT1-LOHAN VADDEPALLY-150115236

The screenshot shows a terminal window with a cron job running. The cron job is a script that checks the date and writes the current date to a file. The script is located at `/var/www/html/cron-test-www.txt`. The terminal output shows the script running at 00:11:35 UTC 2025 and 00:17:03 UTC 2025. The script writes the date to the file. The terminal also shows the user logging in from 10.3.45.21. The web browser shows a lab page with instructions on how to set up iptables to do the load balancing. The page includes a list of steps: 1. Add two new rules to send 50% of the incoming requests for port 80 to www, and the rest to www-slave-1. 2. Test that your load balancer works by looking at the logs on both web servers and reloading your webpage in Firefox. After about some number of requests from Firefox the new requests will be directed to the other servers, and back and forth, more-or-less randomly. 3. You can also see the private IP address on your web page change: that's the actual IP address of the server processing the request, not the IP address of the load balancer. The page also includes a code block for the iptables rules:

```
iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -m statistic --mode random --probability 0.5 -j DNAT --to-destination 10.3.45.11:80
iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -j DNAT --to-destination 10.3.45.21:80
```

The page also includes a code block for the tail command:

```
tail -f /var/log/httpd/access_log
```

The page also includes a section titled "Part 3: Two more slaves".

The screenshot shows a terminal window with the output of the `iptables -L -n -t nat` command. The output shows the following rules:

```
Chain PREROUTING (policy ACCEPT)
target prot opt source destination
DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:2221 to:10.3.45.21:22
DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:2211 to:10.3.45.11:22
DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:80 statistic mode random probability 0.500000000000 to:10.3.45.11:80
DNAT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:80 to:10.3.45.21:80

Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Chain POSTROUTING (policy ACCEPT)
target prot opt source destination
MASQUERADE all -- 0.0.0.0/0 0.0.0.0/0
```

The terminal also shows the user logging in from 142.114.229.38. The web browser shows a lab page with instructions on how to set up iptables to do the load balancing. The page includes a list of steps: 1. Add two new rules to send 50% of the incoming requests for port 80 to www, and the rest to www-slave-1. 2. Test that your load balancer works by looking at the logs on both web servers and reloading your webpage in Firefox. After about some number of requests from Firefox the new requests will be directed to the other servers, and back and forth, more-or-less randomly. 3. You can also see the private IP address on your web page change: that's the actual IP address of the server processing the request, not the IP address of the load balancer. The page also includes a code block for the iptables rules:

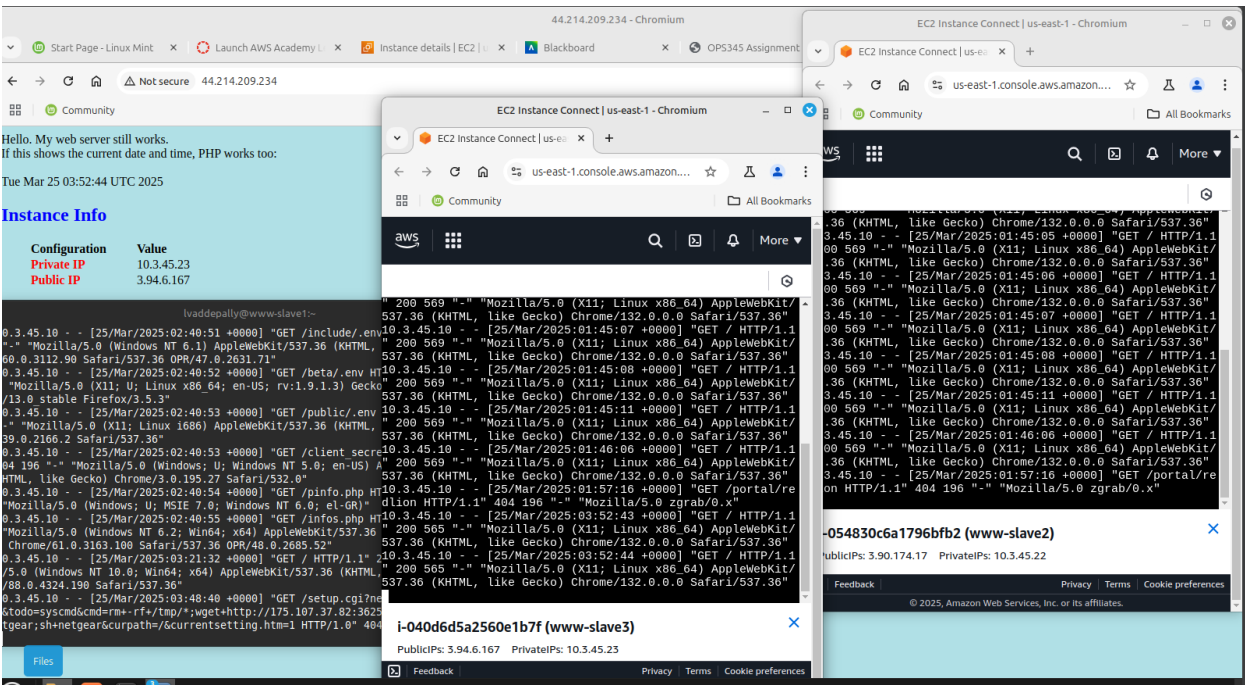
```
iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -m statistic --mode random --probability 0.5 -j DNAT --to-destination 10.3.45.11:80
iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -j DNAT --to-destination 10.3.45.21:80
```

The page also includes a code block for the tail command:

```
tail -f /var/log/httpd/access_log
```

The page also includes a section titled "Part 3: Two more slaves".

OSL840-ASSIGNMENT1-LOHAN VADDEPALLY-150115236



1. Delete the two images you created.
2. Delete all three slave virtual machines.

- **asg1-ss03-routersg.png**: a screenshot of the ops345routersg security group with the port numbers visible.

```

lohan@osl840-mint: ~
elif ip == '10.3.45.21':
    numSlave1 += 1
elif ip == '10.3.45.22':
    numSlave2 += 1
elif ip == '10.3.45.23':
    numSlave3 += 1

print("Hits on main www server      (10.3.45.11):", numMain)
print("Hits on www-slave1 server    (10.3.45.21):", numSlave1)
print("Hits on www-slave2 server    (10.3.45.22):", numSlave2)
print("Hits on www-slave3 server    (10.3.45.23):", numSlave3)

=====
^[Traceback (most recent call last):
  File "/home/lohan/./asg1Test1.py", line 23, in <module>
    curlOutput = output.read()
    ~~~~~^
KeyboardInterrupt
^C
real    0m28.905s
user    0m1.993s
sys     0m2.769s

lohan@osl840-mint:~$

```

1. Delete the two images you created.
2. Delete all three slave virtual machines.
3. Delete the three extra storage devices which were allocated to these slaves.

```

Lohan@osl840-mint:~$ vi asg1Test1.py
Lohan@osl840-mint:~$ chmod +x asg1Test1.py
Lohan@osl840-mint:~$ ./asg1Test1.py
Hits on main www server      (10.3.45.11): 133
Hits on www-slave1 server    (10.3.45.21): 152
Hits on www-slave2 server    (10.3.45.22): 150
Hits on www-slave3 server    (10.3.45.23): 152
Lohan@osl840-mint:~$ cat asg1Test1.py && echo ===== && time ./asg1Test1.py
#!/usr/bin/env python3
# asg1Test.py
# Test for OPS345 Assignment 1
# Author: Andrew Smith
# Student changes by: L Vaddepally

```