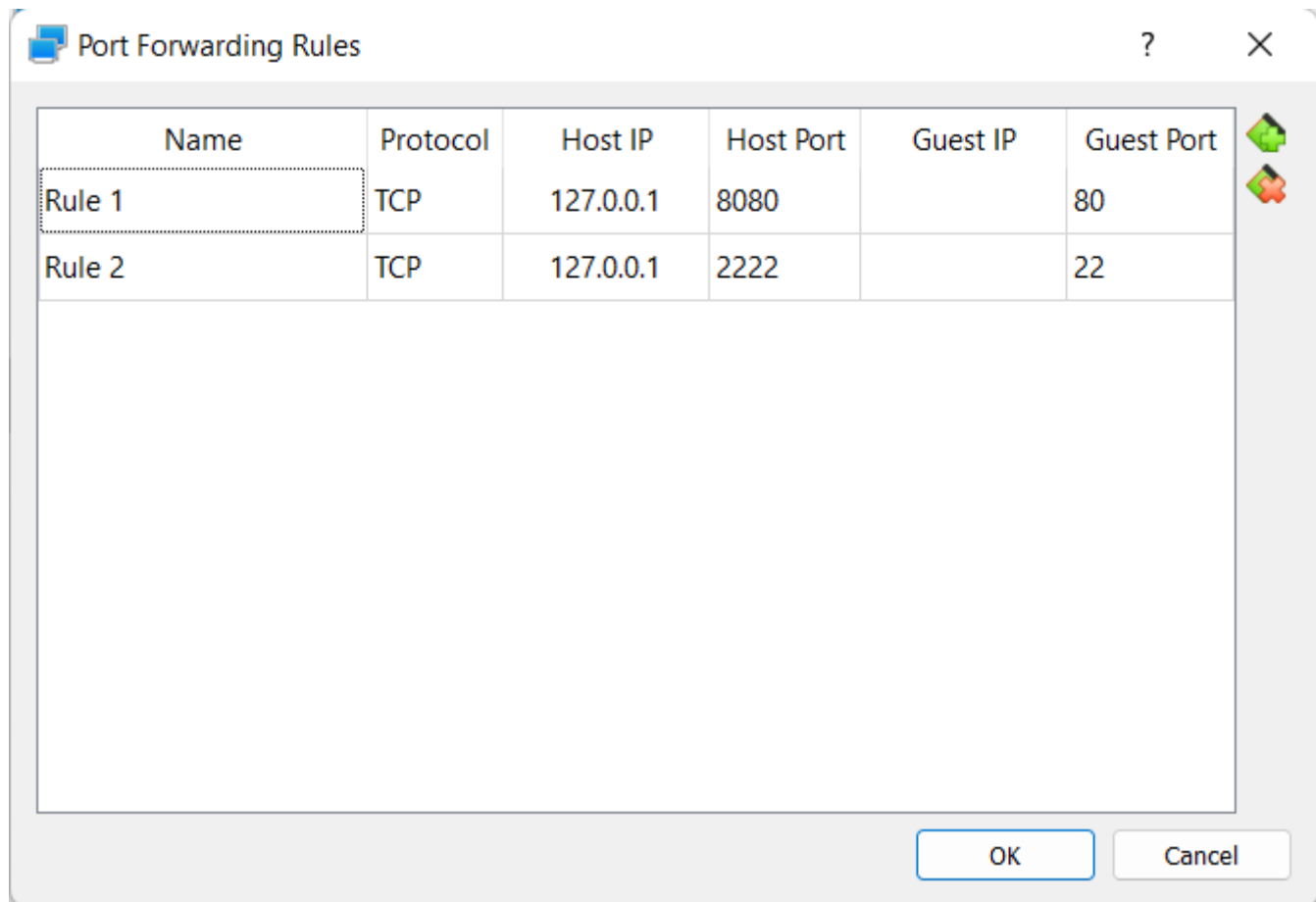


Practical Worksheet 5

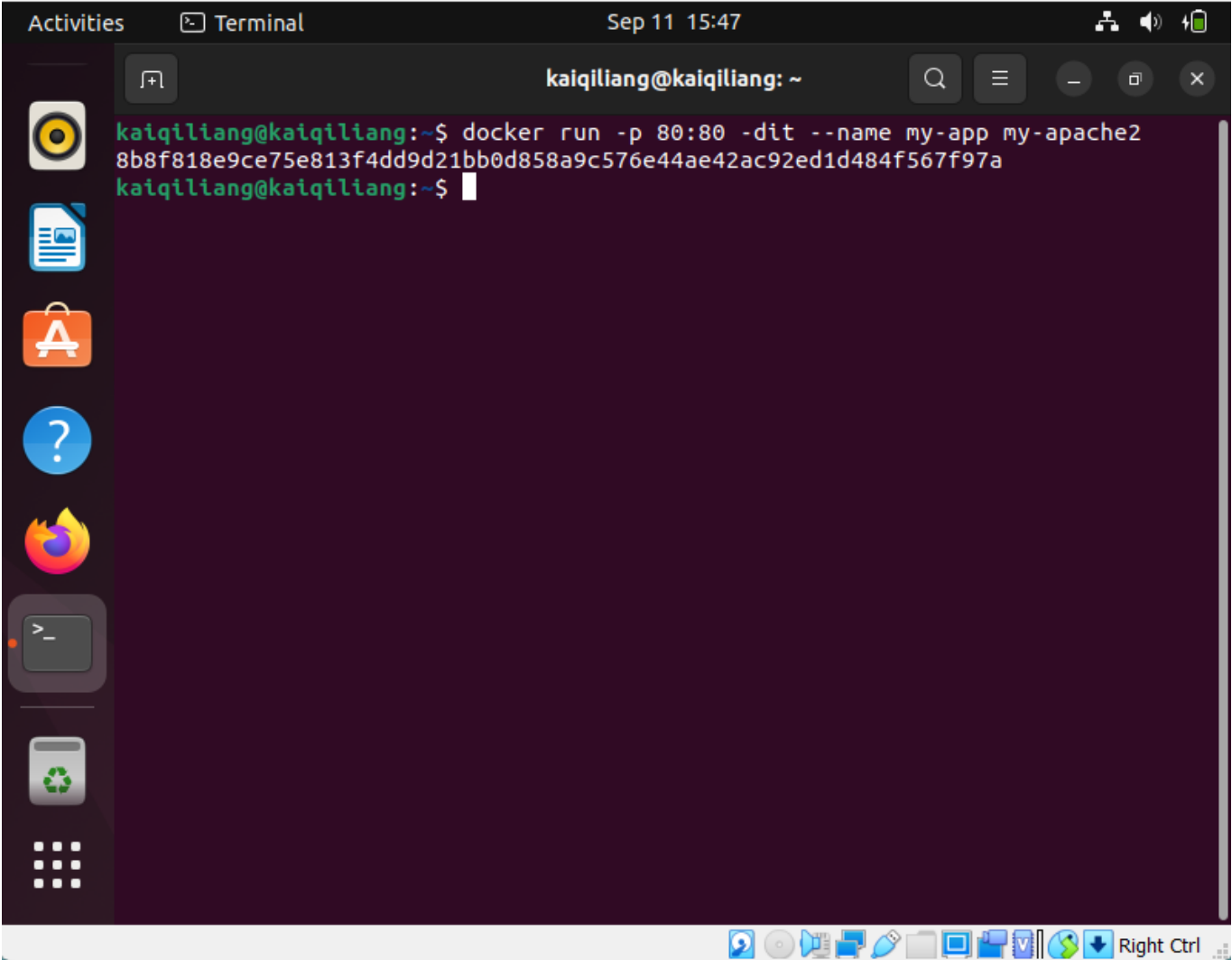
Configure inbound IP on VirtualBox VM

In the VirtualBox Manager, select the VM you want to configure, then click Settings (Golden Gear Cog) and Network. Adapter 1 should be configured as NAT. Click on Advanced and then Port Forwarding. Set up 2 rules:

- Use the localhost host IP 127.0.0.1 and host port 2222 and map that to Guest Port 22.
- Add a similar rule mapping Host Port 8080 to Guest Port 80.



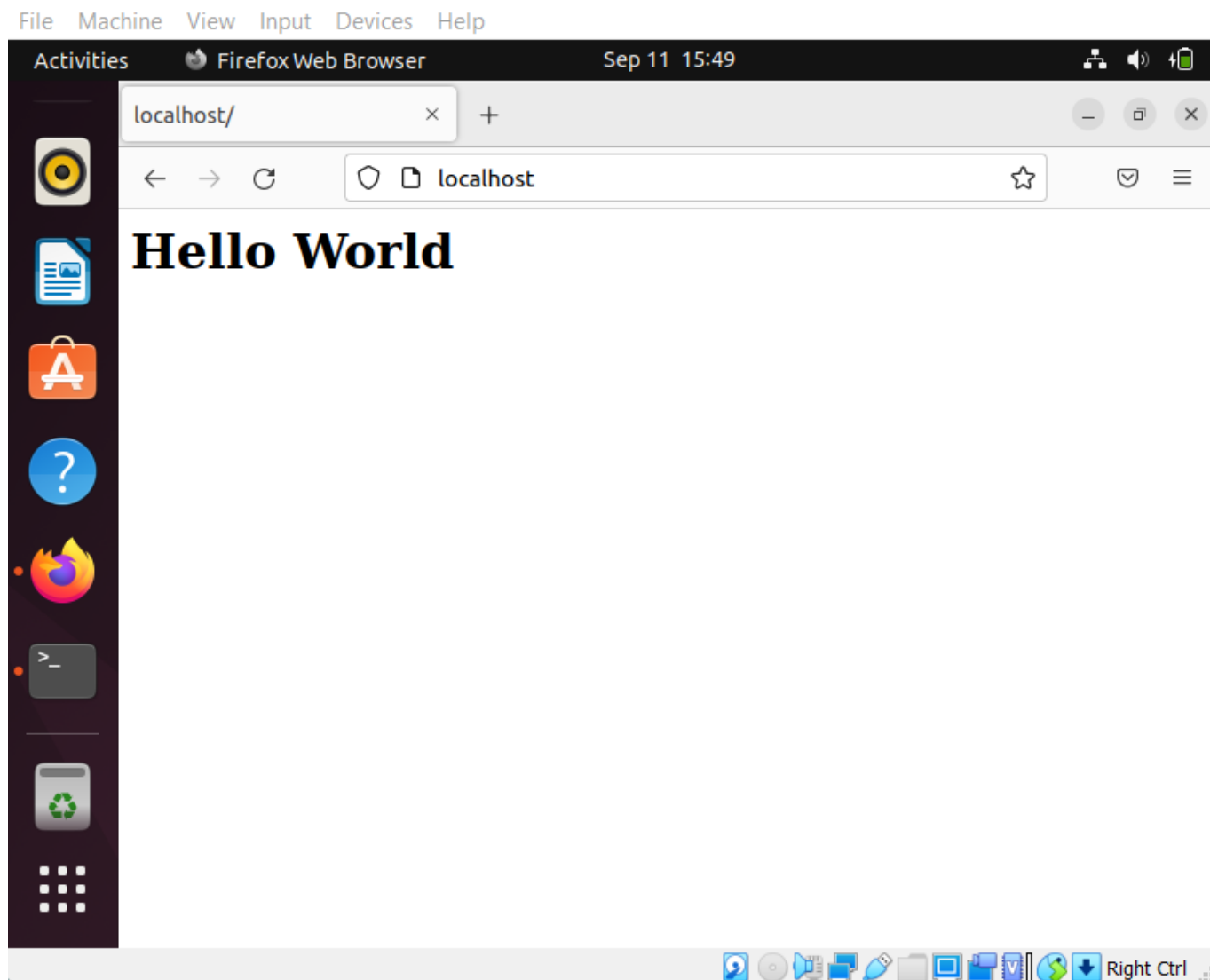
Test the NAT ports by running **apache2** docker app on the VM.



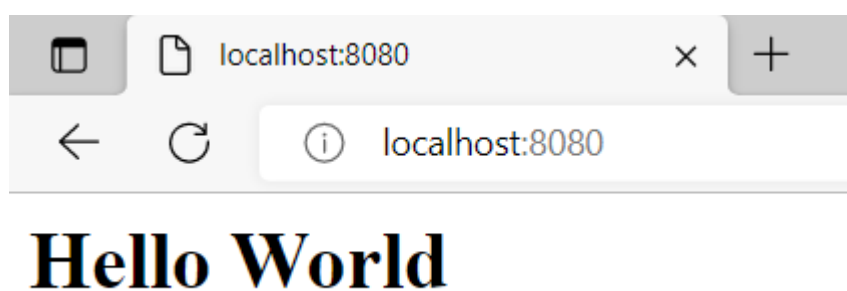
A terminal window titled "Terminal" with a dark background. The window shows a command being executed in a shell. The prompt is "kaiqiliang@kaiqiliang: ~". The command is "docker run -p 80:80 -dit --name my-app my-apache2". The output is a long alphanumeric string: "8b8f818e9ce75e813f4dd9d21bb0d858a9c576e44ae42ac92ed1d484f567f97a". The prompt returns to "kaiqiliang@kaiqiliang:~\$". The window has a title bar with "Activities", "Terminal", and the date/time "Sep 11 15:47". There are icons for search, menu, and window management. On the left side, there is a vertical dock with icons for a terminal, a file manager, a web browser, and a terminal. At the bottom, there is a system tray with icons for network, volume, and other system functions.

```
kaiqiliang@kaiqiliang:~$ docker run -p 80:80 -dit --name my-app my-apache2
8b8f818e9ce75e813f4dd9d21bb0d858a9c576e44ae42ac92ed1d484f567f97a
kaiqiliang@kaiqiliang:~$
```

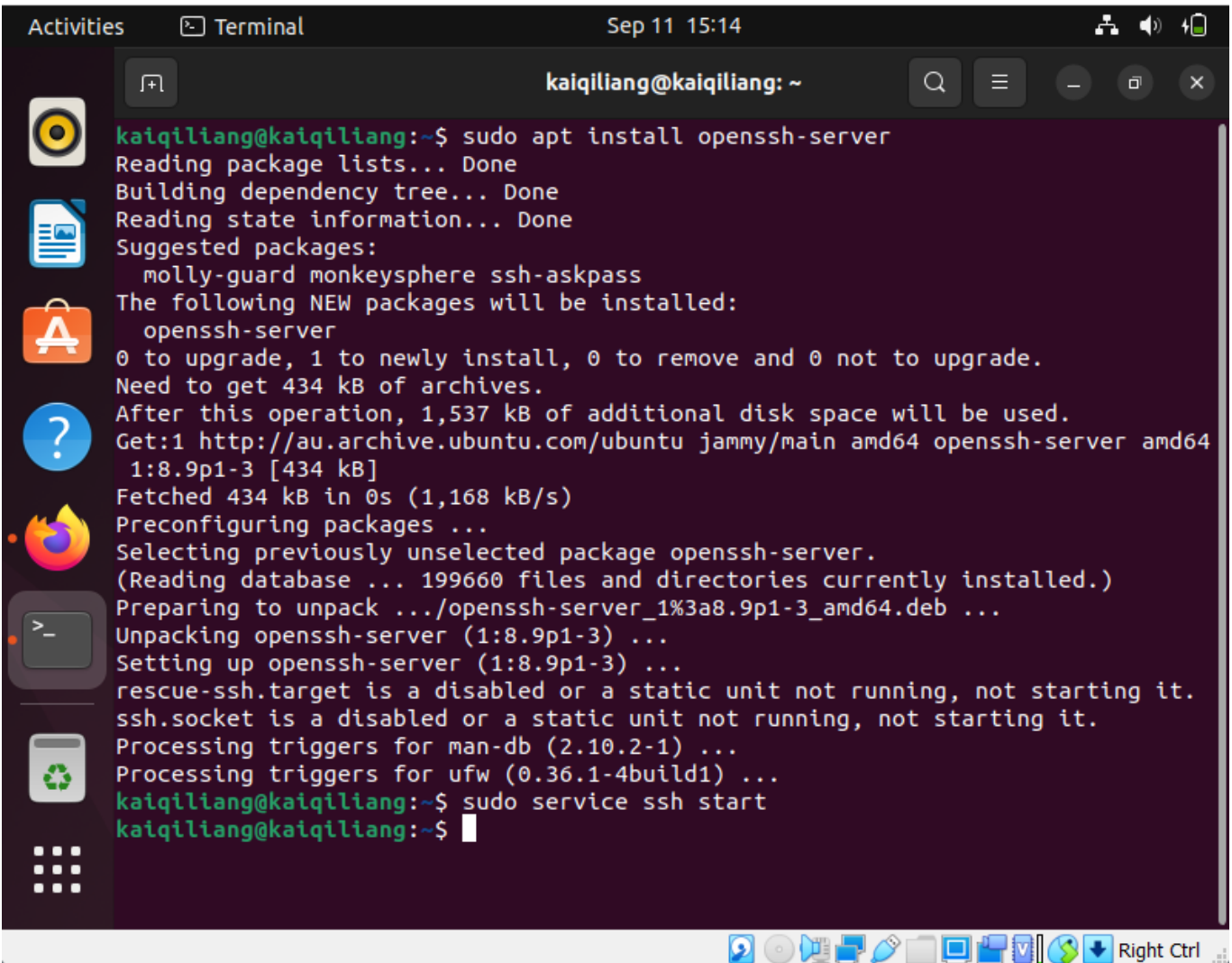
Now open localhost on the VM.



Open localhost:8080 in a browser on the local machine and it should show the exact same thing as on the VM.



Enable ssh to the VM by installing `openssh-server` and start the ssh service.



The screenshot shows a terminal window titled "Terminal" with the user "kaiqiliang@kaiqiliang: ~". The user has run the command `sudo apt install openssh-server`. The terminal output shows the package list being read, the dependency tree being built, and the state information being read. It then lists suggested packages: `molly-guard`, `monkeysphere`, and `ssh-askpass`. The following NEW packages will be installed: `openssh-server`. The terminal shows the progress of the installation, including the fetching of the package (434 kB) and the unpacking of the package (1:8.9p1-3). The terminal also shows the service being started: `sudo service ssh start`. The terminal output shows the service starting successfully.

```
kaiqiliang@kaiqiliang:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  openssh-server
0 to upgrade, 1 to newly install, 0 to remove and 0 not to upgrade.
Need to get 434 kB of archives.
After this operation, 1,537 kB of additional disk space will be used.
Get:1 http://au.archive.ubuntu.com/ubuntu jammy/main amd64 openssh-server amd64
1:8.9p1-3 [434 kB]
Fetched 434 kB in 0s (1,168 kB/s)
Preconfiguring packages ...
Selecting previously unselected package openssh-server.
(Reading database ... 199660 files and directories currently installed.)
Preparing to unpack .../openssh-server_1%3a8.9p1-3_amd64.deb ...
Unpacking openssh-server (1:8.9p1-3) ...
Setting up openssh-server (1:8.9p1-3) ...
rescue-ssh.target is a disabled or a static unit not running, not starting it.
ssh.socket is a disabled or a static unit not running, not starting it.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4build1) ...
kaiqiliang@kaiqiliang:~$ sudo service ssh start
kaiqiliang@kaiqiliang:~$
```

Ssh into the VM from a terminal.

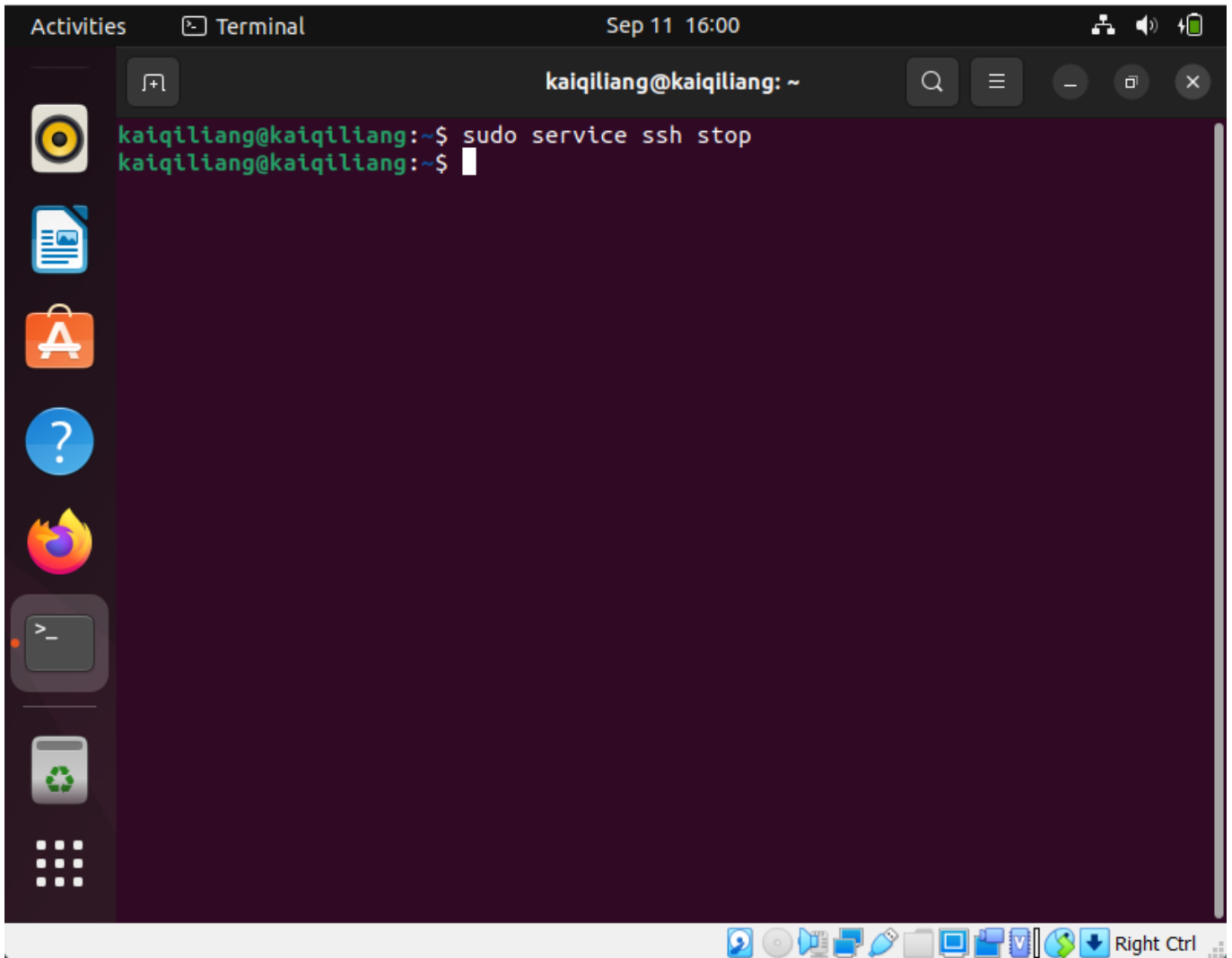
```
C:\>ssh -p 2222 kaiqiliang@127.0.0.1
kaiqiliang@127.0.0.1's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 updates can be applied immediately.

Last login: Sun Sep 11 15:56:55 2022 from 10.0.2.2
kaiqiliang@kaiqiliang:~$ exit
logout
Connection to 127.0.0.1 closed.
```

Stop the ssh service.



Ssh will not connect after it is stopped.

```
C:\>ssh -p 2222 kaiqiliang@127.0.0.1
kex_exchange_identification: read: Connection aborted
```

Setting up an Application Load Balancer

```
'''lab5.py'''
import boto3

ec2 = boto3.client('ec2')
elb = boto3.client('elbv2')

# get the availability zones from the region and get the first two
availability_zones = sorted(
    [
        (
            availability_zone['SubnetId'],
            availability_zone['AvailabilityZone']
        )
        for availability_zone in ec2.describe_subnets()['Subnets']
    ]
)
```

```

    ], key=lambda availability_zone: availability_zone[1]
)[:2]

# create security group for the instances and load balancer
security_group_id = ec2.create_security_group(
    GroupName='23344153-sg',
    Description='Security Group'
)['GroupId']

# add inbound rule for the given security group ID
data = ec2.authorize_security_group_ingress(
    GroupId=security_group_id,
    IpPermissions=[
        {
            'IpProtocol': 'tcp',
            'FromPort': 22,
            'ToPort': 22,
            'IpRanges': [{'CidrIp': '0.0.0.0/0'}]
        },
        {
            'IpProtocol': 'tcp',
            'FromPort': 80,
            'ToPort': 80,
            'IpRanges': [{'CidrIp': '0.0.0.0/0'}]
        }
    ]
)

# create two EC2 instances in two different availability zones of a
specific region
instances = []
for _, availability_zone in availability_zones:
    instance = boto3.resource('ec2').create_instances(
        ImageId='ami-d38a4ab1',
        MaxCount=1,
        MinCount=1,
        InstanceType='t2.micro',
        KeyName='23344153-key',
        SecurityGroupIds=[security_group_id],
        Placement={
            'AvailabilityZone': availability_zone
        }
    )[0]
    instances.append(instance.id)
    instance.wait_until_running()
    print(ec2.describe_instances(InstanceIds=[instance.id])
['Reservations'][0]['Instances'][0]['PublicIpAddress'])

# create the Application Load Balancer
load_balancer = elb.create_load_balancer(
    Name='23344153-lb',
    SecurityGroups=[security_group_id],
    Subnets=[subnet for subnet, _ in availability_zones]
)['LoadBalancers'][0]['LoadBalancerArn']

```

```
# create a target group using the same VPC that was used to create the
instances
target_group = elb.create_target_group(
    Name='23344153-tg',
    Protocol='HTTP',
    Port=80,
    VpcId=ec2.describe_vpcs()[0]['VpcId']
)['TargetGroups'][0]['TargetGroupArn']

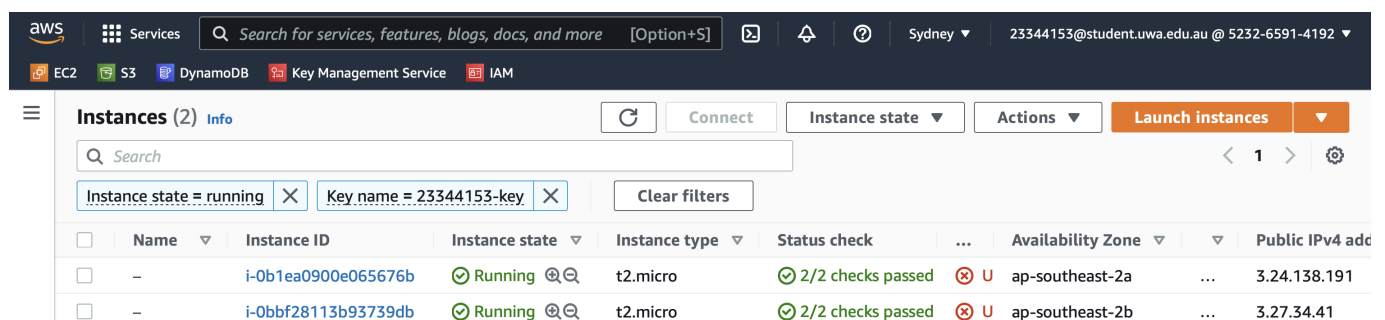
# register targets in the target group
elb.register_targets(
    TargetGroupArn=target_group,
    Targets=[{ 'Id': instance } for instance in instances]
)

# create a listener with a default rule Protocol: HTTP and Port 80
forwarding on to the target group
elb.create_listener(
    LoadBalancerArn=load_balancer,
    DefaultActions=[
        {
            'TargetGroupArn': target_group,
            'Type': 'forward',
        },
    ],
    Protocol='HTTP',
    Port=80,
)
```

Run `lab5.py`.

```
(venv) >_ Labs/lab5 $ python lab5.py
3.24.138.191
3.27.34.41
```

Check the instances on AWS console.



	Name	Instance ID	Instance state	Instance type	Status check	...	Availability Zone	Public IPv4 add
<input type="checkbox"/>	-	i-0b1ea090e065676b	Running	t2.micro	2/2 checks passed	U	ap-southeast-2a	3.24.138.191
<input type="checkbox"/>	-	i-0bbf28113b93739db	Running	t2.micro	2/2 checks passed	U	ap-southeast-2b	3.27.34.41

Install `apache2` on each instance.

```
(venv) > _ Labs/lab5 $ ssh ubuntu@3.24.138.191
The authenticity of host '3.24.138.191 (3.24.138.191)' can't be established.
ED25519 key fingerprint is SHA256:1xQKmsIUdUB1IrFY9th/uAdoMTWA+pj5vkxs8SaHFsY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.24.138.191' (ED25519) to the list of known hosts.
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1052-aws x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud
```

```
0 packages can be updated.
0 updates are security updates.
```

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

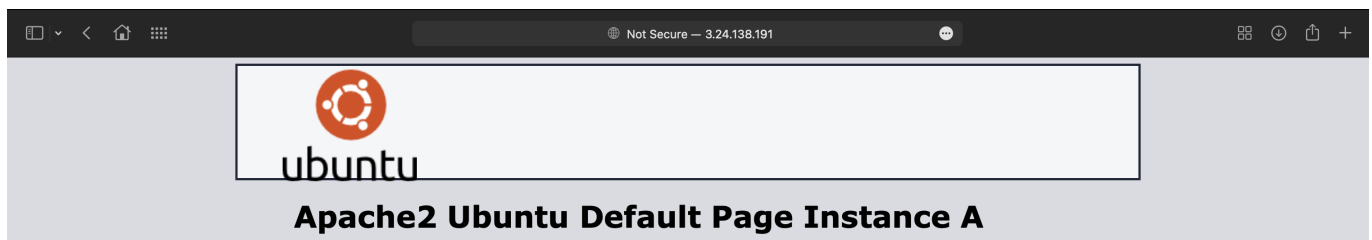
```
ubuntu@ip-172-31-6-243:~$ sudo apt install apache2
```

```
ubuntu@ip-172-31-6-243:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.1-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.1-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 221 not upgraded.
Need to get 1,257 kB/1,559 kB of archives.
After this operation, 6,448 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-bin amd64 2.4.18-2ubuntu3.17 [927 kB]
Get:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-utils amd64 2.4.18-2ubuntu3.17 [81.9 kB]
Get:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-data all 2.4.18-2ubuntu3.17 [162 kB]
Get:4 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2 amd64 2.4.18-2ubuntu3.17 [86.8 kB]
Fetched 1,257 kB in 0s (35.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1:amd64.
(Reading database ... 51142 files and directories currently installed.)
Preparing to unpack .../libapr1_1.5.2-3_amd64.deb ...
Unpacking libapr1:amd64 (1.5.2-3) ...
Selecting previously unselected package libaprutil1:amd64.
```

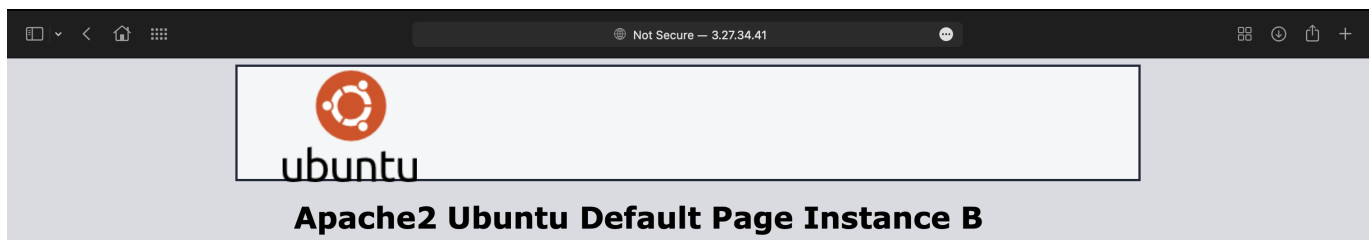
Edit the `/var/www/html/index.html` file to report the instance name.


```
ubuntu@ip-172-31-6-243:~$ sudo vi /var/www/html/index.html
ubuntu@ip-172-31-6-243:~$ cat /var/www/html/index.html | tail -13
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
      <span class="floating_element">
        Apache2 Ubuntu Default Page Instance A
      </span>
    </div>
  </div>
  <div class="validator">
  </div>
</body>
</html>
```

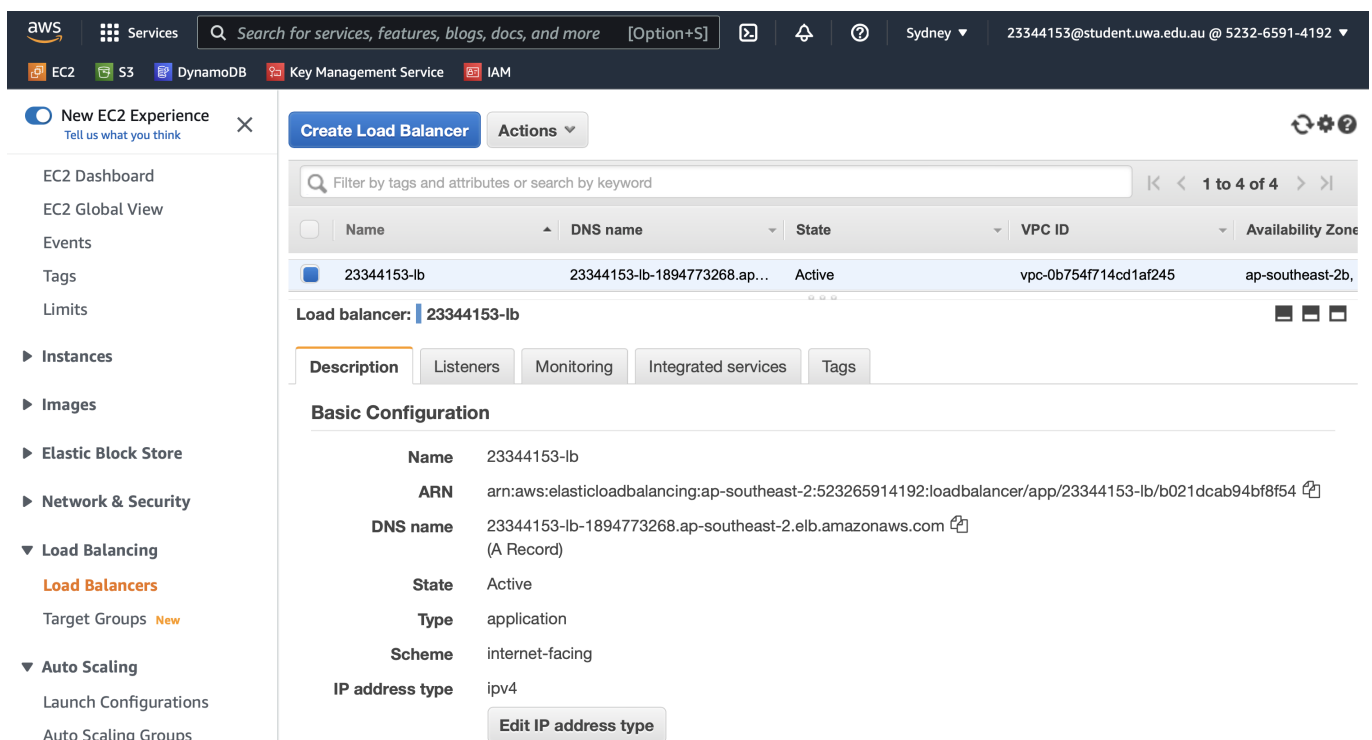
Accessing instance A using its public IP address.



Accessing instance B using its public IP address after the going through the same process as instance A.



Check the Application Load Balancer on AWS console.



Check the target group on AWS console.

aws Services Search for services, features, blogs, docs, and more [Option+S] Sydney 23344153@student.uwa.edu.au @ 5232-6591-4192

EC2 S3 DynamoDB Key Management Service IAM

New EC2 Experience Tell us what you think

- EC2 Dashboard
- EC2 Global View
- Events
- Tags
- Limits
- Instances
- Images
- Elastic Block Store
- Network & Security
- Load Balancing
 - Load Balancers
 - Target Groups **New**
- Auto Scaling
 - Launch Configurations
 - Auto Scaling Groups

Target groups (1/4) Info

Search or filter target groups

	Name	ARN	Port	Protocol	Target type
<input checked="" type="checkbox"/>	23344153-tg	arn:aws:elasticloadbalancing:ap-southeast-2:523265914192:targetgroup/23344153-tg/693c698d89427bf4	80	HTTP	Instance

Target group: 23344153-tg

Details Targets Monitoring Health checks Attributes Tags

Details

arn:aws:elasticloadbalancing:ap-southeast-2:523265914192:targetgroup/23344153-tg/693c698d89427bf4

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0b754f714cd1af245
IP address type IPv4	Load balancer 23344153-lb		

Check the registered targets on AWS console.

aws Services Search for services, features, blogs, docs, and more [Option+S] Sydney 23344153@student.uwa.edu.au @ 5232-6591-4192

EC2 S3 DynamoDB Key Management Service IAM

arn:aws:elasticloadbalancing:ap-southeast-2:523265914192:targetgroup/23344153-tg/693c698d89427bf4

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0b754f714cd1af245
IP address type IPv4	Load balancer 23344153-lb		

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

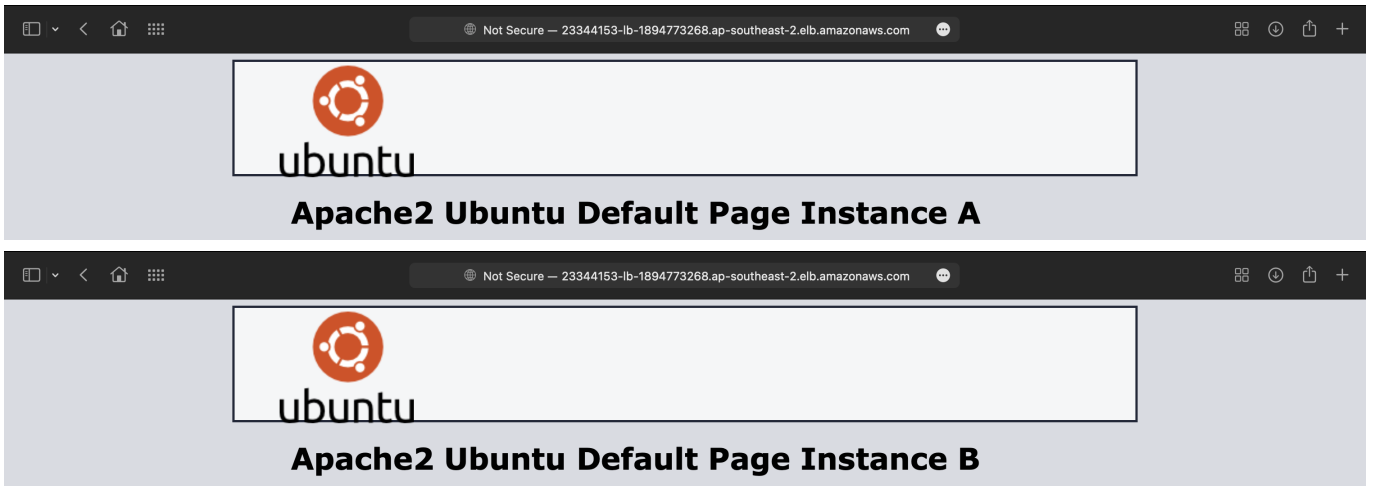
Targets Monitoring Health checks Attributes Tags

Registered targets (2)

Filter resources by property or value

	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0b1ea0900e065676b		80	ap-southeast-2a	healthy	
<input type="checkbox"/>	i-0bbf28113b93739db		80	ap-southeast-2b	healthy	

Verify that the load balancer is working by using the same DNS name to access both of the EC2 instances by refreshing the page.



Delete the load balancer.

