

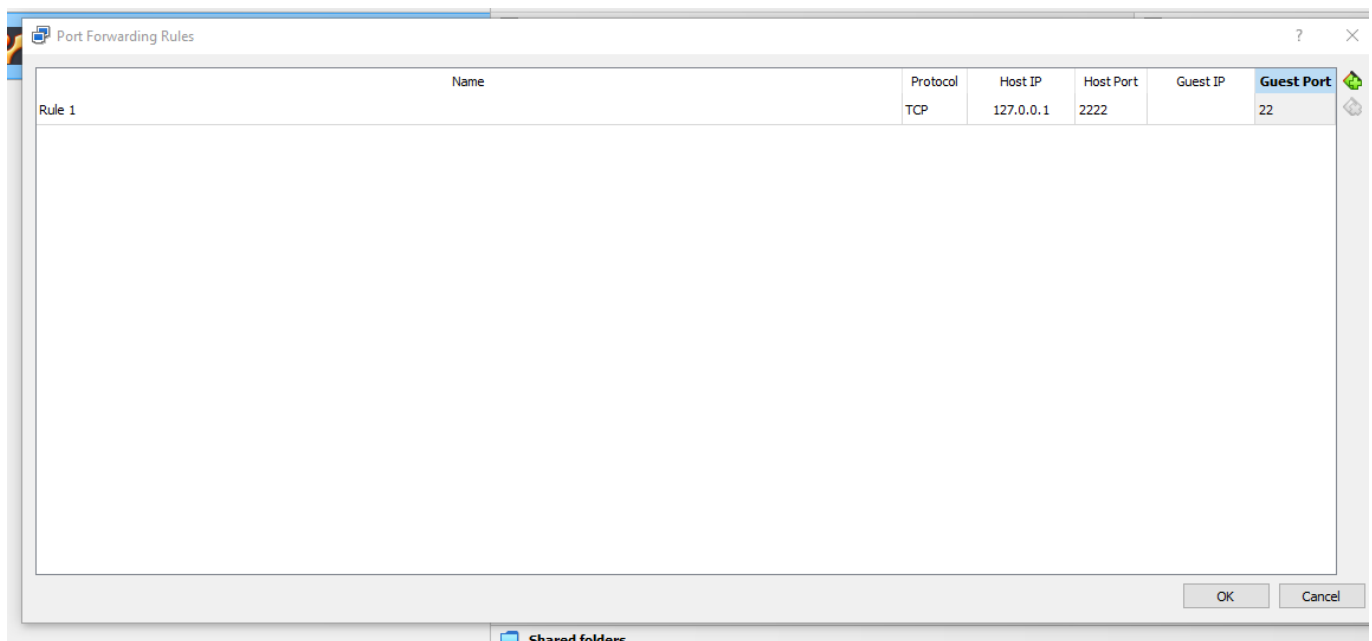
Lab Report 5 - 9

Author: Joo Kai Tay (22489437)

Lab 5: Networking

Section 1: Configure inbound IP on VM

1. Configure the network adapted in VirtualBox Manager using the rule: host IP 127.0.0.1 and host port 2222 mapped to Guest Port 22



2. Install tasksel and openssh-server

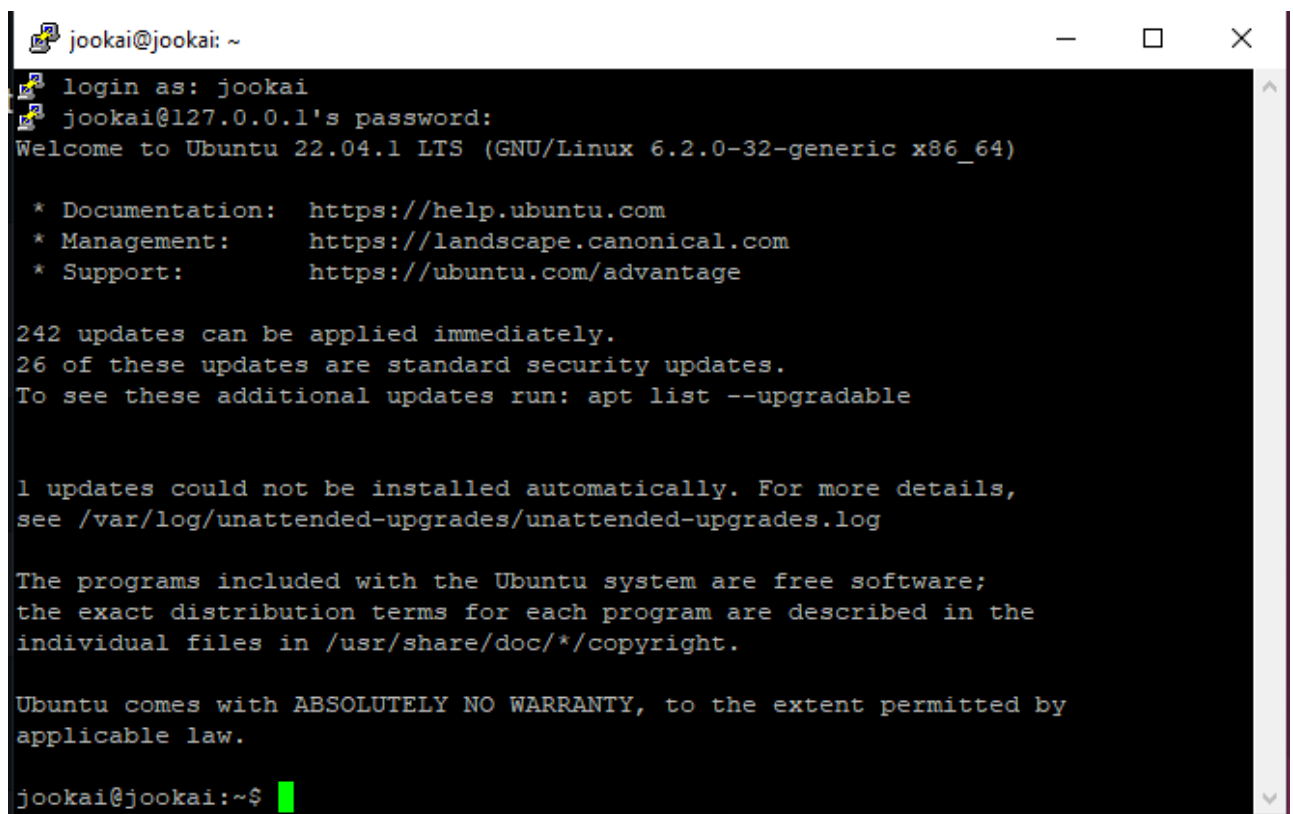
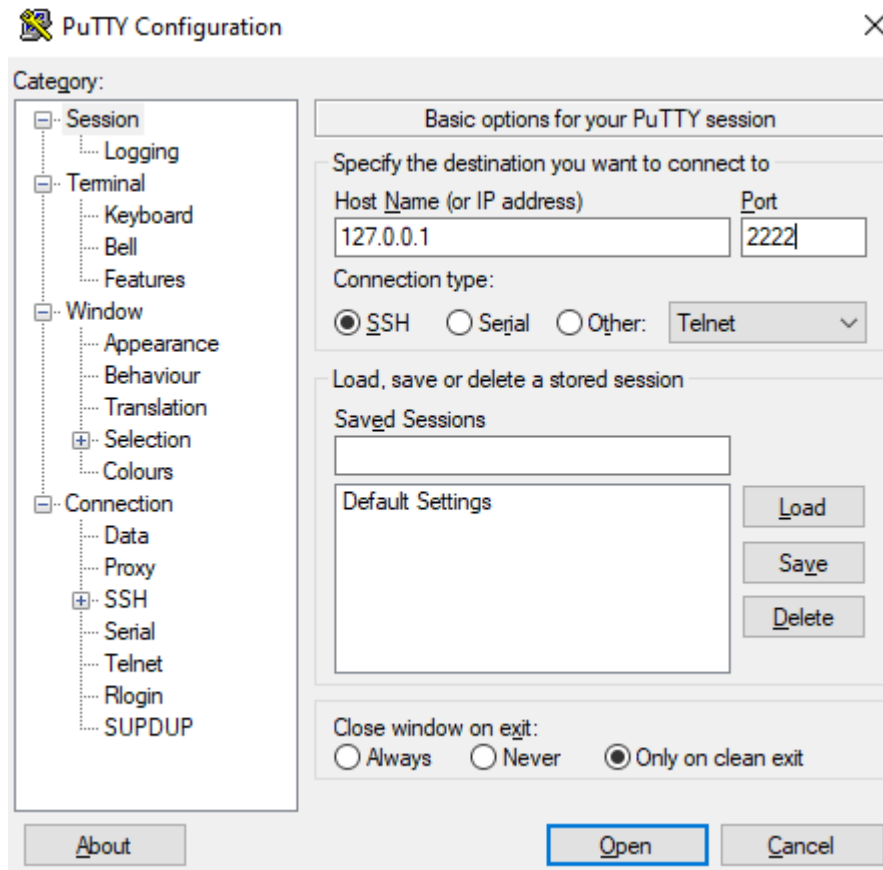
```
jookai@jookai:~$ sudo apt install tasksel
```

```
Processing triggers for man-db (2.10.2-1) ...  
jookai@jookai:~$ sudo tasksel install openssh-server
```

3. Starting the ssh service on the ubuntu VM

```
jookai@jookai:~$ sudo service ssh start
```

4. SSH into the Ubuntu VM from the hostOS using Putty:



5. Terminate the SSH service:

```
jookai@jookai:~$ sudo service ssh stop
```

Section 2: Setting up an Application Load Balancer

1. The following function is used to create 2 EC2 instances in two different availability zones of ap-southeast-1. The reason ap-southeast-1 was used instead of ap-southeast-2 was due to the limit in VPCUs on ap-southeast-2 which did not allow for any new EC2 instances to be created on the region at the time of attempting this lab.

```
def launch_ec2_instances():
    # Create a security group
    response = ec2.create_security_group(
        GroupName=f"{student_number}-sg",
        Description="security group for development environment"
    )
    security_group_id = response['GroupId']

    # Authorize inbound SSH traffic for the security group
    ec2.authorize_security_group_ingress(
        GroupId=security_group_id,
        IpProtocol="tcp",
        FromPort=22,
        ToPort=22,
        CidrIp="0.0.0.0/0"
    )

    # Create a key pair and save the private key to a file
    response = ec2.create_key_pair(KeyName=f"{student_number}-key")
    private_key = response['KeyMaterial']
    private_key_file = f"{student_number}-key.pem"

    # Allow writing to the private key file
    os.chmod(private_key_file, 0o666)
    with open(private_key_file, 'w') as key_file:
        key_file.write(private_key)
    # Set the correct permissions for the private key file
    os.chmod(private_key_file, 0o400)
    # Copy the private key file to ~/.ssh directory
    ssh_directory = os.path.expanduser("~/ssh")
    if not os.path.exists(ssh_directory):
        os.makedirs(ssh_directory)

    shutil.copy(private_key_file, ssh_directory)

    availability_zones = ["ap-southeast-1a", "ap-southeast-1b"]

    for i, az in enumerate(availability_zones):
        instance_name = f"{student_number}-{az}"

        instance_params = {
            'ImageId': 'ami-0df7a207adb9748c7',
            'InstanceType': 't2.micro',
            'KeyName': f"{student_number}-key",
            'SecurityGroupIds': [security_group_id],
            'MinCount': 1,
            'MaxCount': 1,
```

```

        'Placement': {'AvailabilityZone': az},
        'TagSpecifications': [
            {
                'ResourceType': 'instance',
                'Tags': [{'Key': 'Name', 'Value': instance_name}]
            }
        ]
    }

# Launch an EC2 instance
response = ec2.run_instances(**instance_params)

instance_id = response['Instances'][0]['InstanceId']

# Wait for the instance to be up and running
ec2.get_waiter('instance_running').wait(InstanceIds=[instance_id])

# Describe the instance to get its public IP address
response = ec2.describe_instances(InstanceIds=[instance_id])
public_ip_address = response['Reservations'][0]['Instances'][0]
['PublicIpAddress']

print(f"Instance {i+1} created successfully in Availability Zone {az} with
Public IP: {public_ip_address}")

```

The created EC2 instances can be observed below. Note that the highlighted public IP addresses and availability zones in the AWS console correspond to the terminal output.

```

jookai@jookai:~/Desktop/cits5503/lab5$ python3 lab5.py -i
Instance 1 created successfully in Availability Zone ap-southeast-1a with Public
IP: 52.221.213.96
Instance 2 created successfully in Availability Zone ap-southeast-1b with Public
IP: 13.213.33.230

```

Instances (2) Info								
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/> < 1 >								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	22489437-ap-southeast-1a	i-01624737c61ac9b4d	Running	t2.micro	2/2 checks passed	No alarms	ap-southeast-1a	ec2-52-221-213-96.e
<input type="checkbox"/>	22489437-ap-southeast-1b	i-0e105acc6d5603f70	Running	t2.micro	2/2 checks passed	No alarms	ap-southeast-1b	ec2-13-213-33-230.e

Instance summary for i-01624737c61ac9b4d (22489437-ap-southeast-1a) Info

Updated less than a minute ago

Refresh

Connect

Instance state ▼

Actions ▼

<div>Instance ID</div> <div>i-01624737c61ac9b4d (22489437-ap-southeast-1a)</div>	<div>Public IPv4 address</div> <div>52.221.213.96 open address</div>	<div>Private IPv4 addresses</div> <div>172.31.34.17</div>
<div>IPv6 address</div> <div>–</div>	<div>Instance state</div> <div>Running</div>	<div>Public IPv4 DNS</div> <div>ec2-52-221-213-96.ap-southeast-1.compute.amazonaws.com open address</div>
<div>Hostname type</div> <div>IP name: ip-172-31-34-17.ap-southeast-1.compute.internal</div>	<div>Private IP DNS name (IPv4 only)</div> <div>ip-172-31-34-17.ap-southeast-1.compute.internal</div>	<div>Elastic IP addresses</div> <div>–</div>
<div>Answer private resource DNS name</div> <div>–</div>	<div>Instance type</div> <div>t2.micro</div>	<div>AWS Compute Optimizer finding</div> <div>Opt-in to AWS Compute Optimizer for recommendations. Learn more</div>
<div>Auto-assigned IP address</div> <div>52.221.213.96 [Public IP]</div>	<div>VPC ID</div> <div>vpc-02806703abdc316d0</div>	<div>Auto Scaling Group name</div> <div>–</div>
<div>IAM Role</div> <div>–</div>	<div>Subnet ID</div> <div>subnet-080783bde78702ba9</div>	
<div>IMDSv2</div> <div>Optional</div>		

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Networking details Info

<div>Public IPv4 address</div> <div>52.221.213.96 open address</div>	<div>Private IPv4 addresses</div> <div>172.31.34.17</div>	<div>VPC ID</div> <div>vpc-02806703abdc316d0</div>
<div>Public IPv4 DNS</div> <div>ec2-52-221-213-96.ap-southeast-1.compute.amazonaws.com open address</div>	<div>Private IP DNS name (IPv4 only)</div> <div>ip-172-31-34-17.ap-southeast-1.compute.internal</div>	
<div>Subnet ID</div> <div>subnet-080783bde78702ba9</div>	<div>IPv6 addresses</div> <div>–</div>	<div>Secondary private IPv4 addresses</div> <div>–</div>
<div>Availability zone</div> <div>ap-southeast-1a</div>	<div>Carrier IP addresses (ephemeral)</div> <div>–</div>	<div>Outpost ID</div> <div>–</div>

Instance summary for i-0e105acc6d5603f70 (22489437-ap-southeast-1b) Info

Updated less than a minute ago

Refresh

Connect

Instance state ▼

Actions ▼

<div>Instance ID</div> <div>i-0e105acc6d5603f70 (22489437-ap-southeast-1b)</div>	<div>Public IPv4 address</div> <div>13.213.33.230 open address</div>	<div>Private IPv4 addresses</div> <div>172.31.19.253</div>
<div>IPv6 address</div> <div>–</div>	<div>Instance state</div> <div>Running</div>	<div>Public IPv4 DNS</div> <div>ec2-13-213-33-230.ap-southeast-1.compute.amazonaws.com open address</div>
<div>Hostname type</div> <div>IP name: ip-172-31-19-253.ap-southeast-1.compute.internal</div>	<div>Private IP DNS name (IPv4 only)</div> <div>ip-172-31-19-253.ap-southeast-1.compute.internal</div>	<div>Elastic IP addresses</div> <div>–</div>
<div>Answer private resource DNS name</div> <div>–</div>	<div>Instance type</div> <div>t2.micro</div>	<div>AWS Compute Optimizer finding</div> <div>Opt-in to AWS Compute Optimizer for recommendations. Learn more</div>
<div>Auto-assigned IP address</div> <div>13.213.33.230 [Public IP]</div>	<div>VPC ID</div> <div>vpc-02806703abdc316d0</div>	<div>Auto Scaling Group name</div> <div>–</div>
<div>IAM Role</div> <div>–</div>	<div>Subnet ID</div> <div>subnet-0da033b36a320696f</div>	
<div>IMDSv2</div> <div>Optional</div>		

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Networking details Info

<div>Public IPv4 address</div> <div>13.213.33.230 open address</div>	<div>Private IPv4 addresses</div> <div>172.31.19.253</div>	<div>VPC ID</div> <div>vpc-02806703abdc316d0</div>
<div>Public IPv4 DNS</div> <div>ec2-13-213-33-230.ap-southeast-1.compute.amazonaws.com open address</div>	<div>Private IP DNS name (IPv4 only)</div> <div>ip-172-31-19-253.ap-southeast-1.compute.internal</div>	
<div>Subnet ID</div> <div>subnet-0da033b36a320696f</div>	<div>IPv6 addresses</div> <div>–</div>	<div>Secondary private IPv4 addresses</div> <div>–</div>
<div>Availability zone</div> <div>ap-southeast-1b</div>	<div>Carrier IP addresses (ephemeral)</div> <div>–</div>	<div>Outpost ID</div> <div>–</div>

2. The code below creates an application load balancer. a. The code creates the load balancer and specifies the two region subnets retrieved from step 1. b. The code creates a listener with a default rule

Protocol: HTTP and Port 80 forwarding on to the target group c. The code creates a target group using the VPC from step 1 d. The code registers the two EC2 instances from step 1 as targets

```
def create_load_balancer():
    vpc_id = 'vpc-02806703abdc316d0'
    security_group_id = 'sg-0021774194b407020'
    subnet_ids = ['subnet-080783bde78702ba9', 'subnet-0da033b36a320696f']

    response = elb.create_load_balancer(
        Name='22489437-LoadBalancer',
        Subnets=subnet_ids,
        SecurityGroups=[security_group_id],
        Scheme='internet-facing',
        Tags=[
            {
                'Key': 'Name',
                'Value': '22489437-LoadBalancer'
            },
        ]
    )

    load_balancer_arn = response['LoadBalancers'][0]['LoadBalancerArn']
    print(f"Load Balancer ARN: {load_balancer_arn}")

    # Create a listener for HTTP traffic (Port 80)
    response = elb.create_listener(
        DefaultActions=[
            {
                'Type': 'fixed-response',
                'FixedResponseConfig': {
                    'ContentType': 'text/plain',
                    'StatusCode': '200',
                    'MessageBody': 'OK',
                },
            },
        ],
        LoadBalancerArn=load_balancer_arn,
        Port=80,
        Protocol='HTTP',
    )

    listener_arn = response['Listeners'][0]['ListenerArn']
    print(f"Listener ARN: {listener_arn}")

    # Create a target group
    response = elb.create_target_group(
        Name='22489437-target-group',
        Protocol='HTTP',
        Port=80,
        VpcId=vpc_id,
        TargetType='instance',
    )
```

```

# Get the ARN of the target group
target_group_arn = response['TargetGroups'][0]['TargetGroupArn']

# Print the target group ARN
print(f"Target Group ARN: {target_group_arn}")

instance_1_id = 'i-01624737c61ac9b4d'
instance_2_id = 'i-0e105acc6d5603f70'

# Register the instances in the target group
elb.register_targets(
    TargetGroupArn=target_group_arn,
    Targets=[
        {'Id': instance_1_id},
        {'Id': instance_2_id},
    ]
)

# Print registration status
print("Targets registered successfully.")

```

The following screenshots show the output of running the code as well as the results in the AWS terminal.

```

jookai@jookai:~/Desktop/cits5503/labs$ python3 lab5.py -lb
Load Balancer ARN: arn:aws:elasticloadbalancing:ap-southeast-1:489389878001:load
balancer/app/22489437-LoadBalancer/11c6918aab0606fd
Listener ARN: arn:aws:elasticloadbalancing:ap-southeast-1:489389878001:listener/
app/22489437-LoadBalancer/11c6918aab0606fd/913934794eb2c296
Target Group ARN: arn:aws:elasticloadbalancing:ap-southeast-1:489389878001:targe
tgroup/22489437-target-group/3d701f2f5fefc404
Targets registered successfully.

```

Load balancers (2)								<div> <div>↻</div> <div>Actions ▾</div> <div>Create load balancer ▾</div> </div>	
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.								<div> <div>Filter by property or value</div> <div>< 1 > ⚙</div> </div>	
<input type="checkbox"/>	Name ▾	DNS name ▾	State ▾	VPC ID ▾	Availability Zones ▾	Type ▾	Date created		
<input type="checkbox"/>	22489437-LoadBalancer	22489437-LoadBalancer-1...	Active	vpc-02806703abdc316d0	2 Availability Zones	application	September 17, 2023, 11:35 (UTC+08:00)		

22489437-LoadBalancer

Actions

▼ Details

Load balancer type

Application

Scheme

Internet-facing

Status

Active

Hosted zone

Z1LMS91P8CMLE5

VPC

vpc-02806703abdc316d0

Availability Zones

subnet-0da033b36a320696f ap-southeast-1b (apse1-az1)
subnet-080783bde78702ba9 ap-southeast-1a (apse1-az2)

IP address type

IPv4

Date created

September 17, 2023, 11:35 (UTC+08:00)

Load balancer ARN

arn:aws:elasticloadbalancing:ap-southeast-1:489389878001:loadbalancer/app/22489437-LoadBalancer/11c6918aab0606fd

DNS name

Info

22489437-LoadBalancer-1775619243.ap-southeast-1.elb.amazonaws.com (A Record)

Listeners and rules

Network mapping

Security

Monitoring

Integrations

Attributes

Tags

Listeners and rules (1) Info

Manage rules

Manage listener

Add listener

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Filter listeners by property or value

< 1 > ⚙

<input type="checkbox"/>	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL cert
<input type="checkbox"/>	HTTP:80	<div>Return fixed response<ul style="list-style-type: none">Response code: 200Response body: OKResponse content type: text/plain</div>	1 rule	<div>ARN</div>	Not applicable	Not applicable

Network mapping Info

Edit IP address type

Edit subnets

Targets in the listed zones and subnets are available for traffic from the load balancer using the IP addresses shown.

VPC

vpc-02806703abdc316d0

IPv4: 172.31.0.0/16

IPv6 : -

IP address type

IPv4

Mappings

Including two or more Availability Zones, and corresponding subnets, increases the fault tolerance of your applications.

Zone	Subnet	IPv4 address	Private IPv4 address	IPv6 address
ap-southeast-1b (apse1-az1)	subnet-0da033b36a320696f	Assigned by AWS	Assigned from CIDR 172.31.16.0/20	Not applicable
ap-southeast-1a (apse1-az2)	subnet-080783bde78702ba9	Assigned by AWS	Assigned from CIDR 172.31.32.0/20	Not applicable

8 / 11

22489437-target-group

Actions

Details

arn:aws:elasticloadbalancing:ap-southeast-1:489389878001:targetgroup/22489437-target-group/3d701f2f5fec404

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-02806703abdc316d0
IP address type	Load balancer		
IPv4	None associated		

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

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (2)

Filter resources by property or value

< 1 >

	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0e105acc6d5603f70	22489437-ap-southeast...	80	ap-southeast-1b	unused	Target group is not con...
<input type="checkbox"/>	i-01624737c61ac9b4d	22489437-ap-southeast...	80	ap-southeast-1a	unused	Target group is not con...

3. In this step, we will SSH into each of the instances created in step 1 and install Apache2. Screenshots showing this process for one of the EC2 instances have been attached:

```
jookai@jookai: ~/.ssh$ ssh -i 22489437-key.pem ubuntu@52.221.213.96
The authenticity of host '52.221.213.96 (52.221.213.96)' can't be established.
ED25519 key fingerprint is SHA256:PHQL/z7oZ1klTmFoiao+r0jS708r4bLdfADQwiAlBIg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.221.213.96' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

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

System information as of Sun Sep 17 03:50:30 UTC 2023

System load:  0.0                Processes:           96
Usage of /:   20.6% of 7.57GB    Users logged in:    0
Memory usage: 24%                IPv4 address for eth0: 172.31.34.17
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-172-31-34-17: ~$ S
```

```
ubuntu@ip-172-31-34-17: ~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support
  ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
  bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support
  ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 127 not upgraded.
Need to get 2137 kB of archives.
After this operation, 8505 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

22489437-target-group

Actions

Details

arn:aws:elasticloadbalancing:ap-southeast-1:489389878001:targetgroup/22489437-target-group/3d701f2f5fec404

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-02806703abdc316d0
IP address type IPv4	Load balancer 22489437-LoadBalancer		

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

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (2)

Filter resources by property or value

< 1 >

	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0e105acc6d5603f70	22489437-ap-southeast...	80	ap-southeast-1b	healthy	
<input type="checkbox"/>	i-01624737c61ac9b4d	22489437-ap-southeast...	80	ap-southeast-1a	healthy	

4. In this step we will edit the `/var/www/html/index.html` file to report the instance name and availability zone.

```
<!DOCTYPE html>
<html>
<body>
<h1>This is Instance 1 from availability zone ap-southeast-1a</h1>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<h1>This is instance 2 from availability zone ap-southeast-1b</h1>

</body>
</html>
```

5. By refreshing the page repeatedly, we can access both EC2 instances

