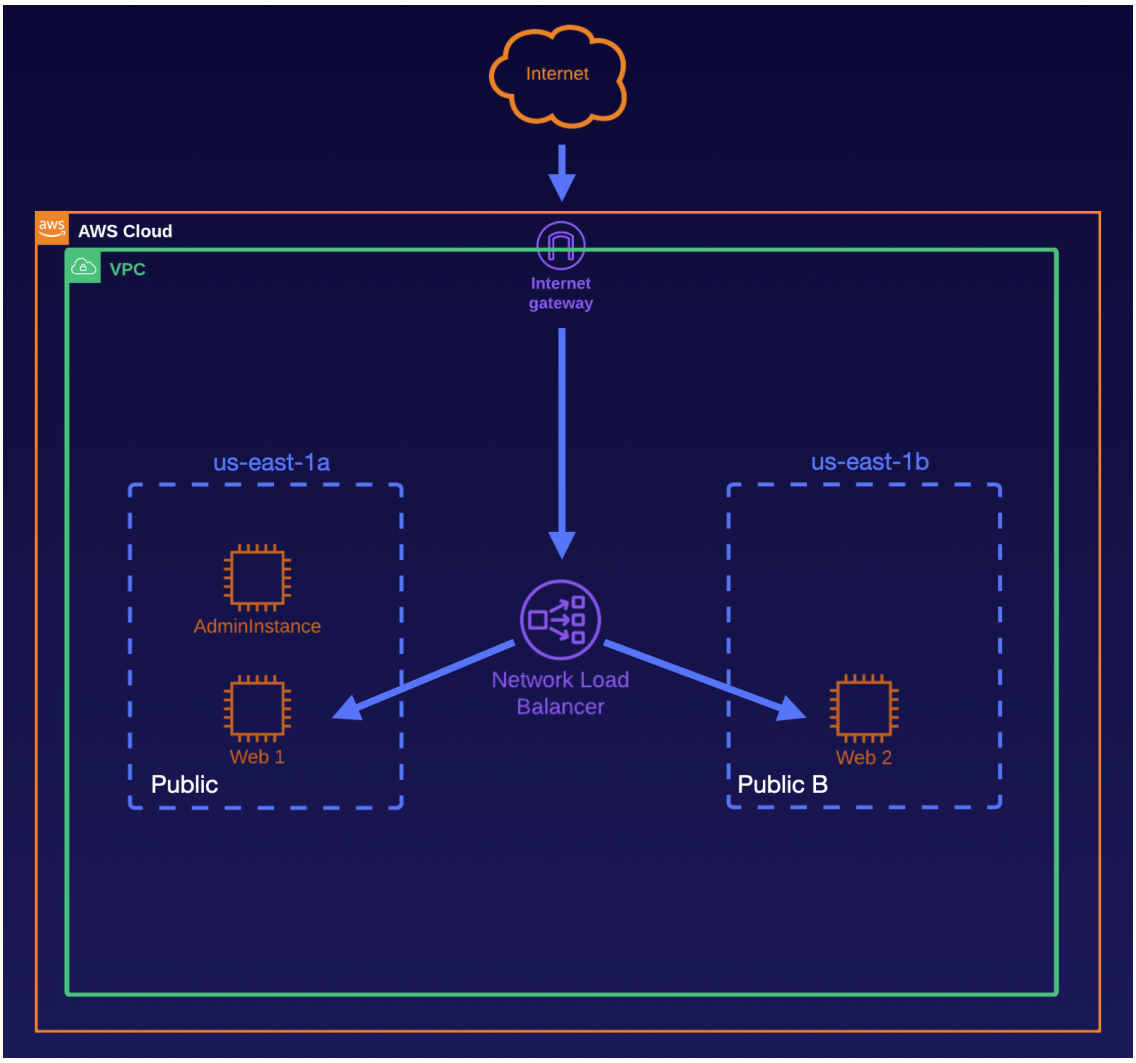
Creating and Configuring a Network Load Balancer in AWS

Introduction

In this hands-on lab, you will prepare the AWS environment for the Network Load Balancer (configuring subnets, network ACL, and EC2 instances). When the preparation is complete, you will create and configure a Network Load Balancer. After configuration of the load balancer, you will work from the CLI to run a small test on the load balancer and view the results in the CloudWatch service.



1- Prepare the Environment

1.1- Create and Configure a Subnet

1. Navigate to the VPC Management Console by searching for VPC in the search bar at the top of the AWS Console and selecting VPC from the list of services.

2. Click Subnets in the left-hand menu, and then click Create subnet.

- VPC ID: Select the listed VPC
- Subnet name: Public B
- Availability Zone: us-east-1b
- IPv4 CIDR block: 10.0.2.0/24 (you might want to adapt this based on the CIDR block of the VPC)

3. Click Create subnet.

4. On the Subnets page, with the Public B subnet selected, click the Route table tab below.

5. Click the Route table link.

6. With the route table selected, click the Routes tab below.

7. Click Edit routes, and then Add route.

8. Set the following values for the new route:

- Destination: 0.0.0.0/0
- Target: Select Internet Gateway, and then select the listed internet gateway

9. Click Save changes.

10. Click the Subnet associations tab.

11. For the Explicit subnet associations section of the page, click Edit subnet associations.

12. Select the Public B subnet and click Save associations.

1.2- Edit the Network ACL

1. Click Subnets in the left-hand menu.
2. Select the Public B subnet.
3. Click the Network ACL tab below, and then click the Network ACL link.
4. With the NACL selected, click on the Actions button and select Edit inbound rules from the dropdown list.
5. Update Rule 100 with the following:
    - Rule number: 100
    - Type: HTTP (80)
    - Protocol: TCP (6)
    - Port range: 80
    - Source: 0.0.0.0/0
    - Allow/Deny: Allow

6. Click Add new rule.
7. Set the following values:
    - Rule number: 101
    - Type: HTTPS (443)
    - Protocol: TCP (6)
    - Port range: 443
    - Source: 0.0.0.0/0
    - Allow/Deny: Allow

8.Click Add new rule.
9. Set the following values:
    - Rule number: 102

- Type: SSH (22)
- Protocol: TCP (6)
- Port Range: 22
- Source: 0.0.0.0/0
- Allow/Deny: Allow

10. Set the following values:
    - Rule number: 103
    - Type: Custom TCP
    - Protocol: TCP (6)
    - Port Range: 1024-65535
    - Source: 0.0.0.0/0
    - Allow/Deny: Allow
11. Click Save changes.

2- Create EC2 Instances

2.1- Create the Web-A Instance

1. Navigate to the EC2 Management Console and click on Instances in the left-hand menu.

2. Click Launch instances.

3. On the AMI page, select the latest Amazon Linux 2 AMI.

4. On the next page, leave t2.micro selected, and click Next: Configure Instance Details.

5. On the Configure Instance Details page, set the following values:

    - Network: Leave default

    - Subnet: us-east-1a

    - Auto-assign Public IP: Enable

6. Scroll down to the User data section and paste in the following script:

```
#!/bin/bash

yum update -y && yum -y install httpd && systemctl enable httpd
&& systemctl start httpd

usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "Request Handled by: Web-A" >> /var/www/html/index.html
```

7. Click Next: Add Storage and leave the volume at the default 8 GB size.

8. Click Next: Add Tags, click Add Tag, and add the following tag:

- Key: Name

- Value: Web-A

9. Click Next: Configure Security Group.

10. Click Select an existing security group.

11. Select the provided security group (not the default security group) from the table.

12. Click Review and Launch, and then Launch.

13. In the key pair dialog box, from the first dropdown, select Create a new key pair.

14. Give it a Key pair name of nlb-lab.

15. Click Download Key Pair, and then click Launch Instances.

16. Click View Instances.

2.2- Create the Web-B Instance

1. On the Instances page, click Launch instances.

2. On the AMI page, select the Amazon Linux 2 AMI.

3. On the next page, leave t2.micro selected, and click Next: Configure Instance Details.

4. On the Configure Instance Details page, set the following values:

    - Network: Leave default

    - Subnet: us-east-1b

    - Auto-assign Public IP: Enable

5. Scroll down to the User data section and paste in the following scrip:

```
#!/bin/bash

yum update -y && yum -y install httpd && systemctl enable httpd &&
systemctl start httpd

usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "Request Handled by: Web-B" >> /var/www/html/index.html
```

6. Click Next: Add Storage and leave the volume at the default 8 GB size.

7. Click Next: Add Tags, click Add Tag, and add the following tag:

    - Key: Name

    - Value: Web-B

8. Click Next: Configure Security Group.

9. Click Select an existing security group.

10. Select the provided security group (not the default security group) from the table.

11. Click Review and Launch, and then Launch.

12. In the key pair dialog box, from the first dropdown, select Choose an existing key pair.

13. Select nlb-lab, check the I acknowledge that I have access... checkbox, and then click Launch Instances.

14. Click View Instances, and give the Web-A and Web-B instances a few minutes to enter the Running state.

3- Create and Configure a Network Load Balancer

1. Click on Load Balancers in the Load Balancing section of the left-hand menu.

2. Click Create Load Balancer.

3. In the Network Load Balancer card, click Create.

4. In the Basic Configuration section, set the following values:

- Load Balancer name: NLB4LAB

- Scheme: Internet-facing

5. In the Network mapping section, from the Select a VPC dropdown, select the listed VPC and then, under Mappings, check both the us-east-1a and us-east-1b checkboxes.

6. In the Listeners and routing section, click Create target group. It will open a new browser tab.

7. On the Specify group details page, in the Basic configuration section, set the following values:

- Choose a target type: Instances

- Target group name: nlbTargets

- Protocol: TCP

- Port: 80

8. In the Health checks section, set the following values:

- Health check protocol: TCP

9. Leave the settings in the Advanced health check settings section as-is.

10. Click Next.

11. On the Register targets page, select the Web-A and Web-B instances you created, and click Include as pending below.

12. Click Create target group.

13. Close the Target groups tab and navigate back to the Load balancers tab.

14. On the Create Network Load Balancer page, under Listeners and routing, click on the refresh button next to the Default action | Forward to field.

15. From the Forward to dropdown, select the nlbTargets target group we just created.

16. Scroll down and click Create load balancer.

17. Click View load balancer.

18. Scroll down to the Load Balancers section in the left-hand menu, right-click on Target Groups, and open it in a new tab.

19. In the new Target groups tab, select the nlbTargets target group.

20. Click the Targets tab. After a few minutes and once your NLB is active, you should see both the Web-A and Web-B instances display a healthy status.

4- Test and Monitor the Network Load Balancer

1. Click Load Balancers in the left-hand menu.

2. Select the NLB4LAB Network Load Balancer and, in the Description tab below, copy the contents of the DNS name field.

3. Paste the DNS name you copied into a new browser tab and press Enter. It should result in a web page that says Request handled by: Web-A or Request handled by: Web-B, depending on which instance the request is routed to.

4. Click the shell icon in the top navigation bar opens a CloudShell environment in a new browser tab.

5. In the first tab with the Load Balancers page open, with NLB4LAB still selected, from the Description tab, copy the DNS name again.

6. In the terminal, bombard your load balancer with requests with the following command, using the DNS name you just copied:

```
while true; do curl <LOAD BALANCER DNS NAME>; done
```

7. Hit Enter. Your terminal will most likely be flooded by a lot of scrolling text telling you which web instance — Web-A or Web-B — the request was handled by.

8. Hit Ctrl+C to break out of the loop.

9. In the first tab with the Load Balancers page open, click the Monitoring tab to keep an eye on the CloudWatch metrics. It may take a few minutes, but you should see the spikes in the different charts representing the simulated traffic.