Mike Azure

Advanced Cisco CCNP Networking

AWS User Permissions Configuration

Lab 1

Background Information:

This lab contains information on how to setup AWS Identity and Access Management (IAM). IAM is a webservice that enables the management of users and permissions on AWS. It allows you to control users, security credentials, access keys, and permissions.

You can create users and assign them individual security credentials. You can also create and manage IAM roles. Roles are groups of users with unique permissions for accessing AWS. Identity federation also allows existing users to access the AWS Management Console without an IAM user for every person.

Setup Steps:

Adding Permissions to a Group:

1. Open to AWS management Console.
2. In the services menu, select IAM.
3. Choose User Groups on the left-hand panel.
4. Create a group or select an existing one.
5. Choose the permissions tab.
6. Add or remove permissions to this user group as desired using the add permissions tab.
    a. The plus icon can be used to view the policy details. A policy defines what actions are allowed or denied for specific AWS resources.
        i. Policies follow the structure of effect and action. Effect determines whether to allow or deny a permission, and action specifies the API that is called.
    b. Inline policies are policies that you create and manage and are embed directly into a single user, group, or role.

Adding Users to a Group:

1. Open to AWS management Console.
2. In the services menu, select IAM
3. Choose User Groups on the left-hand panel.
4. Create a group or select an existing one.
5. Choose the Users tab.
6. Choose Add users.
7. Select the user you would like to add to the chosen group.

Sign-in and Test Permissions of IAM Users:

1. In the navigation pane, choose the Dashboard.
2. IAM users sign-in link will be displayed.
    a. It will look similar to https://123456789012.signin.aws.amazon.com/console
    b. This link can be used to sign-in to the desired AWS account.
3. Paste this link into a new browser window.
4. Sign-in to the user using the username and password previously configured when creating the user.
5. Test the capabilities of the user to ensure its scope is within what is intended.

Conclusion:

This lab outlined the creation of IAM users and groups and inspecting IAM policies in those groups. It outlined how to assign users to groups with specific capabilities enabled. It explained how to login to IAM users using the sign-in URL.

Mike Azure

Advanced Cisco CCNP Networking

Build a VPC and Launch a Web Server

Lab 2

Background Information:

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can create a VPC that spans multiple Availability Zones. An Internet gateway (IGW) is a useful VPC component that allows communication between instances in your VPC and the Internet. After creating a VPC, you can add subnets. Each subnet resides entirely within one Availability Zone and cannot span zones. If a subnet's traffic is routed to an Internet Gateway, the subnet is known as a public subnet. If a subnet does not have a route to the Internet gateway, the subnet is known as a private subnet. The wizard will also create a NAT Gateway, which is used to provide internet connectivity to EC2 instances in the private subnets

Setup Steps:

Create the VPC:

1. In the AWS Management Console, choose the service menu, then choose VPC.
2. Choose the Launch VPC Wizard button.
3. Choose VPC with Public and Private Subnets.
4. Configure the settings as desired and choose Create VPC.
5. When the VPC is finished creating, choose OK.

Creating Additional Subnets:

The following steps will configure the Private Subnets to route internet-bound traffic to the NAT Gateway so that resources in the Private Subnet are able to connect to the Internet, while still keeping the resources private.

1. From the VPC resource choose Subnets.
2. Choose create Subnet and configure desired settings.

3. In the left panel, choose Route Tables.
4. Select desired values. Rename this route to Private Route Table
5. Choose subnet associations.
6. Choose edit associations.
7. Select desired subnets.
8. Choose save associations.
9. Select the desired routes in the table.
10. Rename the route to Public Route Table.
11. Choose subnet associations.
12. Choose edit subnet associations.
13. Select the public subnets.
14. Choose save associations.

The VPC now has public and private subnets configured in two Availability Zones:

Create a VPC Security Group:

1. Choose security groups and configure to desired settings.
2. In the inbound rules, choose add rule.
3. Configure desired settings.
4. Choose create security group.

Launch a Web Server Instance:

1. On the services menu, choose EC2.
2. Choose Launch Instance twice.

3. In the Amazon Linux 2 row choose select.
4. Select t2.micro.
5. Choose “Next: Configure Instance Details”
6. Configure desired settings.
    a. After expanding the advanced details, you must include these settings in the user data box. This script will be run automatically when the instance launches for the first time. The script loads and configures a PHP web application. #!/bin/bash
    # Install Apache Web Server and PHP
    yum install -y httpd mysql php
    # Download Lab files
    wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-ACCLFO-2/2-lab2-vpc/s3/lab-app.zip
    unzip lab-app.zip -d /var/www/html/ #
    Turn on web server
    chkconfig httpd on
    service httpd start
7. Choose Next: Add Storage
8. Choose Next: Add Tags
9. Choose Add Tag and set desired configuration.
10. Choose Next: Configure Security Group.
11. Choose Select an existing security group.
12. Select Web Security Group.
13. Choose Review and Launch
14. Choose continue.
15. Review the instance information for errors and choose Launch.
16. In the Select an existing keypair dialog, select I acknowledge.
17. Choose Launch Instances and then choose View Instances.
18. Select Web Server 1.
19. Copy the Public DNS (IPv4) value shown in the Description tab at the bottom of the page.
20. Open a new web browser tab, paste the Public DNS value and press Enter.
21. This should display the desired information.

Conclusion:

This lab detailed how to use Amazon Virtual Private Cloud (VPC) to create your own VPC and add additional components to produce a customized network. It also includes an optional method to create security groups for the EC2 instance. It also explained how to configure and customize an EC2 instance to run a web server and launch it into the VPC.

Mike Azure

Advanced Cisco CCNP Networking

Introduction to Elastic Compute Cloud (Amazon EC2)

Lab 3

Background Information:

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. It provides you with complete control of your computing resources. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Economically, Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

Setup Steps:

Launch the EC2 Instance with Termination Protection:

1. In the AWS Management Console on the Services menu, choose EC2.
2. Launch the instance.
3. Choose an Amazon Machine Image (AMI).
4. Choose Next: Configure Instance Details.
5. For Network, select Lab VPC.
6. For Enable termination protection, select Protect against accidental termination
7. In the advanced details section, open the field for User data. Paste the following into User data:
   ```
   #!/bin/bash
   yum -y install httpd
   systemctl enable httpd
   systemctl start httpd
   echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
   ```
8. Choose Next: Add Storage.
9. Select desired storage.
10. Choose Next: Add Tags
11. Add tags as desired.
12. Choose Next: Configure Security Group.
13. Configure security group as desired.
14. Choose Review and Launch.
15. Launch instance.

Update the Security Group and Access the Web Server:

1. Choose the Details tab.
2. Copy the IPv4 Public IP of your instance to your clipboard. Paste this into the web address bar.
3. In the EC2 Management Console tab, choose Security Groups.
4. Select Web Server security group.
5. Choose the Inbound rules tab.
6. Choose Edit inboud rules then configure:
   Type: HTTP
   Source: Anywhere-IPv4
7. Save the rules.
8. Return to the web server tab that you previously opened and refresh the page. You should see the message Hello From Your Web Server!

Resize Your Instance: Instance Type and EBS Volume:

1. Stop the instance in order to resize it.
2. In the Actions menu, select Instance Settings Change Instance Type, then configure:
   Instance Type: t2.small
3. Choose Apply
4. In the left navigation menu, choose Volumes.
5. In the Actions menu, select Modify Volume.
6. Change the size to 10.
7. Choose Modify.
8. Choose Yes to confirm and increase the size of the volume.
9. Choose Close.
10. Start the instance again.

Conclusion:

This lab detailed how to launch a web server with termination protection enabled, modify the security group that your web server is using to allow HTTP access, resize your Amazon EC2 instance to scale, and terminate the EC2 instance.

Mike Azure

Advanced Cisco CCNP Networking

Working with EBS (Amazon Elastic Block Store)

Lab 4

Background Information:

Amazon Elastic Block Store (Amazon EBS) offers persistent storage for Amazon EC2 instances. Amazon EBS volumes are network-attached and persist independently from the life of an instance. When used as a boot partition, Amazon EC2 instances can be stopped and subsequently restarted, enabling you to pay only for the storage resources used while maintaining your instance's state. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone). For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon Simple Storage Service (Amazon S3) and automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes and can protect your data for long-term durability. You can also easily share these snapshots with co-workers and other AWS developers.

Setup Steps:

Create a New EBS Volume:

1. In the left navigation pane, choose **Volumes**.
2. Choose "Create Volume" and configure desired settings.
3. In order to attach the volume to an instance, select my volume.
4. In the actions menu, choose attach volume.
5. Choose the instance field and select the instance that appears. Note the device field indentifier.
6. Choose attach volume.

Connect to Your Amazon EC2 Instance:

1. For windows, choose the details drop down menu and select show. Choose the Download PPK button and save the labsuser.ppk file.
2. Download PuTTY from here: https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe
3. Install PuTTY.
4. Configure PuTTY to not timeout by choosing Connection and setting Seconds between keepalives to 30.
5. Configure your PuTTY session:
   - Choose Session
   - Host Name (or IP address): Copy and paste the IPv4 Public IP address for the instance. To find it, return to the EC2 Console and choose Instances. Check the box next to the instance and in the *Description* tab copy the IPv4 Public IP value.
   - Back in PuTTy, in the Connection list, expand SSH
   - Choose Auth (don't expand it)
   - Choose Browse
   - Browse to and select the labsuser.ppk file that you downloaded
   - Choose Open to select it
   - Choose Open
6. Choose Yes, to trust the host and connect to it. When prompted, login.

Create and Configure Your File System:

1. View the storage available:
   a. df – h
2. Create an ext3 file system on the new volume:
   a. sudo mkfs -t ext3 /dev/sdf
3. Create a directory for mounting the new storage volume:
   a. sudo mkdir /mnt/data-store
4. Mount the new volume:
   a. sudo mount /dev/sdf /mnt/data-store
5. View the configuration file to see the setting on the last line:
   a. cat /etc/fstab

6. On your mounted volume, create a file and add some text to it.
    a. sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
7. Verify that the text has been written to your volume.
    a. cat /mnt/data-store/file.txt

Create an Amazon EBS Snapshot:

1. In the AWS Management Console, choose Volumes and select My Volume.
2. In the Actions menu, select Create snapshot.
3. In your remote SSH session, delete the file that you created on your volume.
    a. sudo rm /mnt/data-store/file.txt
4. Verify that the file has been deleted.
    a. ls /mnt/data-store/

Restore the Amazon EBS Snapshot:

1. In the **AWS Management Console**, select **My Snapshot**.
2. In the **Actions** menu, select **Create volume from snapshot**.
3. For **Availability Zone** Select the same availability zone that you used earlier.
4. Choose Add tag then configure:
    a. Key: Name
    b. Value: Restored Volume
    c. Choose Create volume
5. In the left navigation pane, choose **Volumes**.
6. Select **Restored Volume**.
7. In the **Actions** menu, select **Attach volume**.
8. Choose the **Instance** field, then select the (Lab) instance that appears.
9. Choose **Attach volume**
10. Create a directory for mounting the new storage volume:
    a. sudo mkdir /mnt/data-store2
11. Mount the new volume:
    a. sudo mount /dev/sdg /mnt/data-store2
12. Verify that volume you mounted has the file that you created earlier
    a. ls /mnt/data-store2/

Conclusion:

This lab contained details on how to:

- Create an Amazon EBS volume
- Attach and mount your volume to an EC2 instance
- Create a snapshot of your volume
- Create a new volume from your snapshot
- Attach and mount the new volume to your EC2 instance

Amazon EBS volumes deliver the following features:

- Persistent storage: Volume lifetime is independent of any particular Amazon EC2 instance.
- General purpose: Amazon EBS volumes are raw, unformatted block devices that can be used from any operating system.
- High performance: Amazon EBS volumes are equal to or better than local Amazon EC2 drives.
- High reliability: Amazon EBS volumes have built-in redundancy within an Availability Zone.
- Designed for resiliency: The AFR (Annual Failure Rate) of Amazon EBS is between 0.1% and 1%.
- Variable size: Volume sizes range from 1 GB to 16 TB.

- Easy to use: Amazon EBS volumes can be easily created, attached, backed up, restored, and deleted.

Mike Azure

Advanced Cisco CCNP Networking

Build a DB Server and Interact With the DB Using an App

Lab 5

Background Information:

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six commonly used database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

Setup Steps:

Create a Security Group for the RDS DB Instance:

1. In the AWS Management Console, on the Services menu, choose VPC.
2. In the left navigation pane, choose Security Groups.
3. Choose Create security group and then configure as desired.
4. In the Inbound rules pane, choose Add rule.
5. Configure the following settings:
    a. Type: *MySQL/Aurora (3306)*
    b. CIDR, IP, Security Group or Prefix List: Type sg and then select *Web Security Group*.
6. Choose Create security group

Create a DB Subnet Group:

1. On the Services menu, choose RDS.
2. In the left navigation pane, choose Subnet groups.
3. Choose Create DB Subnet Group then configure:
4. Scroll down to the Add Subnets section.
5. Expand the list of values under Availability Zones and select the first two zones: us-east-1a and us-east-1b.
6. Expand the list of values under Subnets and select the subnets associated with the CIDR ranges 10.0.1.0/24 and 10.0.3.0/24.
7. Choose Create

Create an Amazon RDS DB Instance:

1. In the left navigation pane, choose Databases.
2. Choose Create database.
3. Under Settings, configure as desired.
4. Under DB instance class, configure as desired.
5. Under Storage, configure as desired.
6. Under Connectivity, configure as desired.
7. Under Existing VPC security groups, from the dropdown list configure as desired.
8. Expand Additional configuration, then configure as desired.
    a. Uncheck Enable automatic backups.
    b. Uncheck Enable encryption
    c. Uncheck Enable Enhanced monitoring.
9. Choose Create database
10. Choose the link itself.
11. Wait until Info changes to Modifying or Available.
12. Scroll down to the Connectivity & security section and copy the Endpoint field.
13. Paste the Endpoint value into a text editor. You will use it later in the lab.

Interact with the Database:

1. To copy the WebServer IP address, choose on the Details drop down menu above these instructions, and then choose Show.
2. Open a new web browser tab, paste the *WebServer* IP address and press Enter
3. Choose the RDS link at the top of the page.
    a. Paste the Endpoint you copied to a text editor earlier

4. Test the web application by adding, editing and removing contacts.

Conclusion:

This lab outlined how to launch an Amazon RDS DB instance with high availability, configure the DB instance to permit connections from your web server, and open a web application and interact with your database.

Mike Azure

Advanced Cisco CCNP Networking

Scale and Load Balance Your Architecture

Lab 6

Background Information:

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity out or in automatically according to conditions you define.

Setup Steps:

Create an AMI for Auto Scaling:

1. In the AWS Management Console, on the Services menu, click EC2.
2. In the left navigation pane, click Instances.
3. Wait until the Status Checks for Web Server 1 displays 2/2 checks passed. Click refresh to update.
4. Select Web Server 1.
5. In the Actions menu, click Image and templates > Create image, then configure as desired.
6. Click Create image

Create a Load Balancer:

1. In the left navigation pane, choose Target Groups.
2. Choose Next. The Register targets screen appears.
3. Review the settings and choose Create target group
4. In the left navigation pane, click Load Balancers.
5. At the top of the screen, choose Create Load Balancer.
6. Under Application Load Balancer, choose Create.
7. Under Load balancer name, enter your desired name.
8. Scroll down to the Network mapping section, then:
9. In the Security groups section:
10. For the Listener HTTP:80 row, set the Default action to forward to LabGroup.
11. Scroll to the bottom and choose Create load balancer

Create a Launch Configuration and an Auto Scaling Group:

1. In the left navigation pane, click Launch Configurations.
2. Click Create launch configuration.
3. Configure these settings:
    a. Launch configuration name as desired.
    b. Amazon Machine Image (AMI)
        i. Choose Web Server AMI
    c. Instance type:
        i. Choose Choose instance type
        ii. Select t3.micro
        iii. Choose Choose
4. Under Security groups, you will configure the launch configuration to use the Web Security Group that has already been created. Choose Select an existing security group. Select desired Security Group.
5. Under Key pair configure:
    a. Key pair options: Choose an existing key pair
    b. Existing key pair: vockey
    c. Select I acknowledge...
    d. Click Create launch configuration
6. Select the checkbox for the LabConfig Launch Configuration.
7. From the Actions menu, choose Create Auto Scaling group.
8. Enter Auto Scaling group name.

9. Choose Next.
10. On the Network page configure as desired.
11. Choose Next
12. In the Load balancing - optional pane, choose Attach to an existing load balancer
13. In the Attach to an existing load balancer pane, use the dropdown list to select your load balancer.
14. In the Additional settings - optional pane, select Enable group metrics collection within CloudWatch
15. Choose Next
16. Under Group size, configure:
    a. Desired capacity: 2
    b. Minimum capacity: 2
    c. Maximum capacity: 6
17. Under Scaling policies, choose Target tracking scaling policy and configure:
    a. Metric type: Average CPU Utilization
    b. Target value: 60
18. Choose Next
19. Choose Next
20. Choose Add tag and Configure as desired.
21. Click Next
22. Review the details of your Auto Scaling group, then click Create Auto Scaling group.

Verify that Load Balancing is Working:

1. In the left navigation pane, click Instances.
2. In the left navigation pane, click Target Groups and choose desired group.
3. Click the Targets tab.
4. Wait until the Status of both instances transitions to healthy. Click Refresh in the upper-right to check for updates.
5. In the left navigation pane, click Load Balancers.
6. In the lower pane, copy the DNS name of the load balancer, making sure to omit "(A Record)".
7. Open a new web browser tab, paste the DNS Name you just copied, and press Enter.

Test Auto Scaling:

1. Return to the AWS management console, but do not close the application tab — you will return to it soon.
2. On the Services menu, click CloudWatch.
3. In the left navigation pane, choose All alarms.
4. On the Services menu, click EC2.
5. In the left navigation pane, choose Auto Scaling Groups.
6. Select Lab Auto Scaling Group.
7. In the bottom half of the page, choose the Automatic Scaling tab.
8. Select LabScalingPolicy.
9. Click Actions and Edit.
10. Change the Target Value to 50.
11. Click Update
12. On the Services menu, click CloudWatch.
13. In the left navigation pane, click All alarms and verify you see two alarms.
14. Click the OK alarm, which has AlarmHigh in its name.
15. Return to the browser tab with the web application.
16. Click Load Test beside the AWS logo.
17. Return to browser tab with the CloudWatch console.
18. Wait until the AlarmHigh alarm enters the In alarm state.
19. On the Services menu, click EC2.
20. In the left navigation pane, click Instances.

Terminate Web Server 1:

1. Select Web Server 1 (and ensure it is the only instance selected).
2. In the Instance state menu, click Instance State > Terminate Instance.
3. Choose Terminate

Conclusion:

After completing this lab, you can:

- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.