AWS Lab 1: Introduction to AWS IAM

Emma Matsuda

Advanced Cisco CCNP

**Purpose**

The purpose of this lab is to give the user a brief and introductory level of understanding of the AWS IAM. In this lab, we will be exploring pre-created IAM Users and Groups, inspecting IAM policies as they’re applied to the pre-created groups from before, following a real-world scenario that requires us to add users to groups with specific capabilities enabled, locating and using IAM sign-in URL, and experimenting with effects of the policies on service access.

**Background Info**

This lab focuses on the basic aspects of AWS Identity and Access Management. This is a feature that is used to manage access to the AWS resources. For example, it can be used to control who can terminate Amazon EC2 instances. It can also be used to define fine-graded access rights, like who can access the resources, which resources can be accessed and what can the user do to the resource, and how these resources can be accessed. Now, it is important to not the IAM is a no-cost AWS account feature. IAM has four essential components, the IAM user, group, policy, and role. Below is a quick description of each component:

IAM user – a person or application that can authenticate with an AWS account

IAM group – a collection of IAM users that are granted identical authorization, a user can belong to multiple groups, and this cannot be nested

IAM policy – the document that defines which resources can be accessed and the level of access to each resource

IAM role – useful mechanism to grant a set of permissions for making AWS service request, and provides temporary security credentials

Along with this, an important aspect to keep in mind about IAM is the IAM policies. IAM policies are documents that define permissions, and IAM has two of these – the identity-based and the resource-based. The identity-based policies attach a policy to any IAM entity, which specify actions that may be performed by the entity, or actions that may not be performed by the entity. And with identity-based policies, a single policy can be attached to multiple entities, and a single entity can have multiple policies attached. On the other hand, a resource-based policy is attached to a resource and specifies who has access to the resource and what actions they can perform on it. These policies are inline only, and not managed. Keep in mind that these resource-based policies are support only by some Amazon services.

**Lab Summary**

1. Access the AWS Management Console
2. Explore the Users and Groups through a provided “user-1” User.
3. Give and grant different users different permissions
4. Run through a business scenario
5. Add Users to Groups
6. Add different users to different groups pre-setup by the lab
7. Sign-in and test the users

**Configurations** – step by step screenshots

|  |  |
| --- | --- |
| Click on “Start Lab”  Wait until the message: “Lab status: ready” |  |
| Click on “AWS” |  |
| Click on the AWS Management Console |  |
| on the Services menu, select IAM |  |
| Click the “Users” on the navigation pane on the left |  |
| Click on “user-1” |  |
| Look under the “Permissions” |  |
| Notice that user-1 doesn’t have any permissions |  |
| Click on the “Groups” |  |
| Notice that user-1 doesn’t belong to any groups |  |
| Look under the “Security credentials” tab |  |
| Notice that user-1 is assigned a Console password |  |
| Click on “User groups” on the left pane |  |
| Out of the 3 groups, click on “EC2 Support” |  |
| Look under the “permission” tab |  |
| This policy is a Managed Policy. This is usually assigned by the administrator or AWS, all users and groups get all modifications made with this policy immediately applied. |  |
| Click on the “+” next to the AmazonEC2ReadOnlyAccess |  |
| Notice that this allows you to view the details of the policy |  |
| Go back to the User groups on the left pane, and click on the “S3-Support” |  |
| Notice that under the ‘Permissions” tab, the AmazonS3ReadOnlyAccess policy is attached |  |
| Again, clicking on the “+’ icon allows you to view the details of the policy |  |
| Click on the User groups on the left pane and click on the EC2-Admin |  |
| In the user group, look under the permission tab, and under the EC2 Admin Policy details, the information gets displayed. |  |
| Go back to the “User groups” and click on the S3-Support” group |  |
| Under the “Users” tab, click on the “Add user” option to create a user. |  |
| Click on “user-1” to add user-1 to the s3-Support group |  |
| Click on “Add users” to add user-1. |  |
| With a green message banner at the top, you will see that “user-1” has been added to the group. |  |
| Go back to the user groups menu to go into the EC2-Support group |  |
| Under the “Users” tab, click on the “add users” again |  |
| Click on “user-2” and “add users” to make sure that user-3 gets added into the EC2-Support group |  |
| With the green banner message, you will see that user-2 is now added into the group. |  |
| Go back to the user groups menu to go into the EC2-Admin group |  |
| Under the “Users” tab, click on the “add users” again |  |
| Click on “user-3” and “add users” to make sure that user-3 gets added into the EC2-Admin group |  |
| With the green banner message, you will see that user-3 is now added into the group. |  |
| Click on the Dashboard on the left pane and copy the IAM user sign-in link under “AWS Account” |  |
| In a Private tab on a browser, paste the link and click enter |  |
| Put “user-1” as the username and “Lab-Password1” to sign-in to the lab as user-1 |  |
| Under the Services on the left top corner, click on the S3 option. |  |
| Click on the only option available |  |
| Browse what is visible to the user |  |
| One the Services option on the left top corner, click the EC2 option |  |
| Click on the Instances on the left pane and see that there is a message saying that the user is not authorizes to access this. |  |
| To sign out of user-1, click on the upper right corner where it says the username and then click on the sign out option |  |
| Put “user-2” as the username and “Lab-Password2” to sign-in to the lab as user-2 |  |
| One the Services option on the left top corner, click the EC2 option |  |
| Click on the instances on the left and the LabHost that comes up |  |
| Under the Instance State menu, choose the “stop instance” |  |
| Click “Stop” |  |
| Check that you get an error message saying that you cannot perform this action. |  |
| Under the Services tab on the left top corner, click on the “S3” service. |  |
| Under the buckets option on the left, you will see that user-2 doesn’t have permission to list buckets. |  |
| Sign user-2 out by clicking on the sign out option on the top left corner. |  |
| After signing out, sign back in with the username “user-3” and password “Lab-Password3” to sign in as user 3 |  |
| On the services menu on the left top corner, click on EC2 |  |
| On the instances on the left pane, click on LabHost |  |
| Under the instance state option, click on the “stop instance” to stop it. |  |
| Confirm by clicking “stop” |  |
| Click on End Lab to confirm completion of the lab. |  |

**Conclusion**

Through this lab, I was able to first complete the basic steps of the lab and understand the base structure of the AWS labs. Along with this, I was able to complete the original goals of exploring pre-created IAM users and groups, inspecting IAM policies as applied to the pre-created groups, following a real-world scenario, adding users to groups with specific capabilities enabled, locating and using the IAM sign-in URL, and experimenting with the effects of policies on service access. Through this, I learned how the IAM policies that we learned about through this module is applied in real life, and learned some real-world scenarios of how I would use this.

AWS Lab 2: Build your VPA and Launch a Web Server

Emma Matsuda

Advanced Cisco CCNP

**Purpose**

The purpose of this lab is to use Amazon Virtual Private Cloud (VPC) to create an original VPA and add additional components to produce a customized network. This lab will also teach how to create a security group. Then, the configuration and customization for an EC2 instance to run a web server, and the launch of the EC2 instance to run in a subnet in the VPC will be done.

**Background Info**

This lab focuses on the basic aspect of Amazon Virtual Private cloud (Amazon VPC). Amazon VPC enables you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. It gives you control over your virtual networking resources, such as the selection of IP address range, creation of subnets, and configuration of route tables and network gateways. Along with this, it enables you to customize the network gateways, and to use multiple layers of security.

Now, let’s talk about the division of VPCs and subnets. VPCs are logically isolated from other VPCs and is dedicated to your AWS account. It also belongs to a single AWS Region and can spam multiple Availability Zones. Subnets are a range of IP addresses that divide a VPC, and they belong to a single Availability Zone. They can be classified as public or private.

Within VPC, there are several networking options. Below is a list of the different networking options and functionality.

|  |  |
| --- | --- |
| Name | Functionality |
| Internet Gateway | * Scalable, redundant, highly available VPA component * Provide a target in VPA route tables for inter-routable traffic. * Perform network-translation for instances assigned for IPv4 |
| NAT Gateway | * Enables instances in a private subnet to connect to the internet or other AWS services, but prevents the internet from initiating a conversation with those instances |
| VPC Endpoint | * Virtual device that enables you to privately connect your VPC to supported AWS services and VPA endpoint services that are powered by AWS Private Link |
| VPC Peering | * Networking connection between 2 VPCs that enables you to route traffic between them privately |
| VPA Sharing | * Enables customers to share subnets with other AWS accounts in the same organizations. * Enables multiple AWS accounts to create their application resources |
| AWS site-to-site VPN | * Links the 2 systems (VPN gateway and VPC network) |
| AWS Direct Connect | * Enables you to establish a dedicated, private network connection between your network and DX location |
| AWS Transit Gateway | * Acts as a hub that controls how traffic is routed among all the connected networks, which acts like spokes |

\*You can use the VPC Wizard to implement your design.

**Lab Summary**

1. Access the AWS Management Console
2. Create a VPC using the *VPC and more* option in the VPA console
3. Create 2 additional subnets for the VPA in a second Availability Zone
4. Create a VPC security group that will act as a virtual firewall
5. Launch an Amazon EC2 instance into the new VPC

**Configurations** – step by step screenshots

|  |  |
| --- | --- |
| Click on “Start Lab”  Wait until the message: “Lab status: ready” |  |
| Click on “AWS” |  |
| Under the Services menu on the top left, click on VPC. This opens the VPC console. |  |
| After confirming that the region says N. Virginia, click on the Create VPC button in the VPC dashboard pane. |  |
| Under the VPA settings, click on the “VPC and more” but change the name from “project” to “lab.” Keep the “Auto-generate” option clicked. |  |
| Keep the IPv4 CIDR block but change the Availability Zone to 1. |  |
| Keep both the public subnets and private subnets to 1. |  |
| Under the “Customize subnets CIDR blocks” option, set the Public subnet to “10.0.0.0/24” and the Private subnet to “10.0.1.0/24” |  |
| Set the Nat gateways to “In 1 AZ,” then choose the VPC endpoints to “None,” and keep both DNS options clicked |  |
| Under the Preview tab, make sure that all the information is correct |  |
| Once everything is confirmed and is correct, click on “Create VPC” to confirm the creation. |  |
| This screen will show a bar that will complete to 100% to make the VPC. |  |
| After confirming the “Success,” click on the “View VPC” button. |  |
| Under the Subnets, click on the “Create subnet” to create a new subnet. |  |
| Choose the one labeled “lab-vpc” for the VPC ID. |  |
| Under the subnet settings, name the subnet, choose the Availability zone and IPv4 subnet CIDR block. |  |
| Click on “Create subnet” |  |
| Once you see the green banner confirming the subnet creation, click on “Create subnet” to make another subnet. |  |
| Choose the “lab-vpc” for the VPC ID. |  |
| Choose the appropriate subnet name, availability zone, and the IPv4 subnet CIDR block. |  |
| Click on “create subnet” |  |
| Check the green banner to make sure that a subnet has been successfully created. |  |
| Under the “route tables” option, click on the very last option. |  |
| Under the Routes tab, make sure that the Target starts with “nat-….” so that the traffic destined for the internet will be sent to the NAT gateway. |  |
| Under the “subnet associations” tab, click on the “Edit subnet associations” |  |
| Make sure to select the “lab-subnet-private2” and make sure to “Save associations” |  |
| Under the “lab-rtb-public” route and its route table tab, make sure that the target for 0.0.0.0/0 is “igw-….” so that all internet bound traffic will be sent through Internet Gateway. |  |
| Under the subnet associations tab, click on the “Edit subnet associations” |  |
| Click on the “lab-subnet-public2” and click the “save associations” to make sure that both subnet associations are selected and confirmed. |  |
| Under the “Security groups” on the left pane, click on “Create security group” |  |
| Make the name security group name to “Web Security Group” and description to “Enable HTTP access,” the make the VPC “lab-vpc” |  |
| After clicking on the “add rule” button, configure Inbound rule 1 and its type, source type, and description. |  |
| Click on the “create security group” button to create the security group. |  |
| Under the Services option on the upper left, click on “EC2” to open the EC2 console. |  |
| Click on launch instance to launch a new instance to modify. |  |
| Name it “Web Server 1” and make sure that the AMI is chosen as amazon Linux and Amazon Linux 2023. |  |
| Keep the default instance type “t2.micro” and set the key pair name as “vockey” |  |
| When editing the network settings, click on the “lab-vpc” for the VPC, the “lab-subnet-public2” for the Subnet, and enable the auto-assign public IP. |  |
| Select the option that says “Select existing security group” for the Firewall and click on the “Web Security Group” option for the common security groups. |  |
| Keep basic settings for the “configure storage” setting |  |
| Expand the advanced details tab to edit it. |  |
| Under the User data, copy and paste what’s provided by the instructions as shown in the picture on the left. |  |
| Click on the “Launch instance” button to launch the instance. |  |
| Click on “view all instances” to view the instances. |  |
| Wait until 2/2 checks passed on Web Server 1. |  |
| After selecting the Web Server 1, copy the public IPv4 address |  |
| After pasting the IP address to a new tab, you will see a page with the AWS logo and the Meta-Data posted. |  |

**Conclusion**

Through this lab, I was able to complete the basic steps of the lab and understand how to build a VPC and launch a web server. There were a few objectives that I was to complete through the lab, in which I did, which were: Create a VPC, create subnets, configure a security group, and launch an EC2 instance into a VPC. Through this, I learned how the Amazon VPC enables me to launch amazon Web Services resources into a virtual network that I defined. Along with that, I learned how VPC is applied in real life, and learned some real-world scenarios of how I would use this.

AWS Lab 3: Introduction to Amazon EC2

Emma Matsuda

Advanced Cisco CCNP

**Purpose**

The purpose of this lab is to give the user a brief and introductory level of understanding of the Amazon EC2. This lab will provide the user with an overview of launching, resizing, managing, and monitoring an EC2 interface. In this lab, the user will launch a web server with termination protection activated, monitor EC2 instances, modify security groups that the web server is using to allow HTTP access, resize Amazon EC2 instances to scale, explore EC2 limits, test termination protection, and terminate the EC2 instance in the end.

**Background Info**

This lab focuses on the basic aspects of Amazon EC2. Amazon EC2 provides resizable virtual machines and enables you to run Windows & Linux virtual machines in the cloud.

AWS provides a number of different compute services in the cloud. Just to name a few of the most popular ones:

Amazon EC2 Auto-Scaling supports application availability by allowing you to define conditions that will automatically launch or terminate EC2 instances, so the user doesn’t have to. Amazon Elastic Container Registry (Amazon ECR) is used to store and retrieve docker images. Amazon Elastic Container Service (Amazon ECS) is a container orchestration service that supports Docker/ AWS Elastic Beanstalk provides a simple way to run and manage web applications. AWS Lambda is a serverless compute solution, and the user pays for only the compute time that they use. Amazon Elastic Kubernetes Service (Amazon EKS) enables the user to run managed Kubernetes on AWS. AWS Fargate provides a way to run containers that reduce the need for you to manage servers or clusters.

When choosing the optimal compute service, it is important to evaluate the available compute options, understand the available compute config options, collect computer-related metrics, use the available elasticity of resources, and then re-evaluate compute needs based on metrics.

When understanding the Amazon EC2, it is crucial that we understand what an EC2 instance is. An EC2 instance is simply a virtual server in AWS terminology. Common uses for this include: application, web, database, game, mail, media, data log, file, computing, and proxy servers.

Amazon EC2 provides virtual machines – referred to as EC2 instances – in the cloud. This gives you full control over the guest operating system on each instance. The user can launch instances of any size into an Availability Zone anywhere in the world, and you can control traffic to and from instances. The user launches EC2 instances from an AMI template into a VPC in the account. There are many instance types to choose from, and each instance type offers different combinations of CPU, RAM, storage, and networking capabilities, it is also possible to configure security groups to control access to instances. Only instances that are backed up by Amazon EBS can be stopped, and the user can use Amazon CloudWatch to capture and review metrics on EC2 instances.

**Lab Summary**

1. Access the AWS Management Console
2. Launch an amazon EC2 instance
3. Monitor the instance that was created in the last step
4. Update security group settings, and access the web server
5. Resize the instance by modifying instance type and EBS volume
6. Explore the different EC2 limits that are provided, including the default limits with the AWS account
7. Test the termination protection on the instance.

**Configurations** – step by step screenshots

|  |  |
| --- | --- |
| Click on “Start Lab”  Wait until the message: “Lab status: ready” |  |
| Click on “AWS” |  |
| Click on the AWS Management Console |  |
| Under “Compute,” click on the EC2. |  |
| Click on the ‘Launch instances’ option |  |
| Name the instance “Web server” |  |
| Keep the AMI the same as the default. |  |
| Chose the t2.micro for the instance type, and then choose the “vockey” as the key pair. |  |
| Click on the “Edit” button edit the network settings |  |
| Choose the VPC that says “Lab VPC,” then create a security with the name Web Server security group” and the Description “Security group for my web server.” |  |
| Keep the default configuration settings for the “configure storage” |  |
| Then, under “Advanced Details” enable the termination protection |  |
| Under user data, paste the text. |  |
| Launch the instance |  |
| Make sure that the green banner notifying that the instance was crated successfully appears on the screen. |  |
| CLICK ON THE “View all instances” |  |
| After selecting the Web Server, notice that there is a public cIPv4 address attached to it. |  |
| Make sur that the instance state says “Running” and 2/2 system checks are passed. |  |
| Under the “statuses and alarms” notice that both the system statue check and instance status check in passed. |  |
| Click on:  Actions  Monitor and troubleshoot  Get system log |  |
| Check the system log and that it has the user data that you put in beforehand |  |
| Click on:  Actions  Monitor and troubleshoot  Get instance screenshot |  |
| This give s preview on how the instance will look |  |
| Under the details tab of the Web Server, copy the pubic IPv6 address that we recognized before. |  |
| After pasting and trying to connect to the IPv4 address page, you will notice that you cannot reach the site. |  |
| Under the Security groups tab on the left, select the Web Server security group |  |
| Under the Inbound rules tab, click on the “edit inbound rules” |  |
| Create an inbound rule with the shown options – make sure to save rules at the end |  |
| Back at the original web page from earlier with the public IPv4 address, you should see the message “Hello From Your Web Server!” |  |
| Under the instances, select the web server and stop the instance under the “instance status” option. |  |
| After seeing the success green banner, make sure that the instance is really stopped. |  |
| Then click:  Actions  Instance settings  Change instance type |  |
| Change the instance type to t2.small, then apply the change. |  |
| Under the storage tab, click on the Volume name. |  |
| Under actions, choose the modify volume option. |  |
| Change the size to 10 from 8, then click on the modify button to appl the change. |  |
| On the left navigation pane, choose the Instances, then under the Instance state, chose the “start instance” to restart the resized instance |  |
| Under the management console on the top left, choose the “service Quotas” |  |
| Choose the AWS services on the left, then choose Amazon EC2 |  |
| Search “running on-demand” to see that there is a list of service quotas that match this criteria |  |
| Under the management console, choose EC2 to go back to Amazon EC2 |  |
| Select instances on the left, then under instance status, choose “terminate data” |  |
| Notice that with the red banner message, you are not able to terminate the instance. |  |
| Under actions, instance settings, choose the option “change termination protection” |  |
| Remove the check next to “Enable,” and save the change |  |
| Then using the same steps, try termination the instance again. |  |
| You will see that after clicking o “Terminate,’ you are able terminate this time. |  |

**Conclusion**

Through this lab, I was able to first complete the basic steps of the lab and understand the basis of EC2. With the completion of this lab, I went over things such as launching a web server with termination protection on, monitoring the instance, modifying security groups that the web server is using to allow HTTP access, resizing EC2 instances to scale, and exploring EC2 limits. In addition, towards the end, I learned how to text termination protection, and then turning that off to terminate my EC2 instance. I picked up what EC2 is able to do through simple steps.