AWS Lab 4: Working with EBS

Joshua Widjaja

P0-2 Cisco CCNP

A close-up of a logo

Description automatically generated

**Purpose:**

The main concept of this lab was centered around the basics of Amazon Elastic Block Store (Amazon EBS), an important storage tool for Amazon EC2 instances. The sole purpose of this lab is to gain skills on creating an Amazon EBS volume, attaching and mounting your volume to an EC2 instance, creating a snapshot of your volume, creating a new volume from your snapshot, and attaching and mounting the new volume to my EC2 instance.

**Background Information:**

This lab is mainly about AWS’s attribute called Amazon Elastic Block Store (Amazon EBS). Amazon EBS offers persistent storage for Amazon EC2 instances. The Amazon EBS volumes are highly available, reliable, volumes that can be used as Amazon EC2 instances boot partition or attached to a running Amazon EC2 instance as a standard block device. Amazon EBS enables you to create individual storage volumes and attach them to an Amazon EC2 instance. It also offers block-level storage in which volumes are automatically replicated within its Availability Zone as well as being able to back up automatically to Amazon S3 through snapshots. Examples of its uses include boot volumes and storage for Amazon EC2 instances, data storage with a file system, database hosts, and enterprise applications.

Amazon EBS consists of several different volume types, each having different use cases. The different volume types have the maximum volume size. Here is a table of the different Amazon EBS volume types and their characteristics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | General Purpose | Provisioned IOPS | Throughput-Optimized | | Cold |
| Use cases | * Recommended for most workloads. * System boot volumes * Virtual desktops * Low-latency interactive applications * Development and test environments | * Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/second of throughput per volume * Large database workloads | * Streaming workloads that require consistent, fast throughput at a low price * Big data * Data warehouses * Log processing * It cannot be a boot volume | | * Throughput-oriented storage for large volumes of data that is infrequently accessed. * Scenarios where the lowest storage cost is important. * It cannot be a boot volume |
| Max Volume Size | 16 TiB | 16 TiB | | 16 TiB | 16 TiB |
| Max IOPS/volume | 16,000 | 64,000 | | 500 | 250 |
| Max Throughput/Volume | 250 MiB/s | 1,000 MiB/s | | 500 MiB/s | 250 MiB/s |

Amazon EBS Volume Features:

Persistent storage: Volume lifetime is independent of any 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.

Amazon EBS features:

Snapshots: Point-in-time snapshots. Able to recreate a new volume at any time.

Encryption: Encrypted Amazon EBS volumes at no additional cost.

Elasticity: Able to increase capacity and change to different types.

**Lab Summary:**

1. Accessing the AWS Management Console
2. Create and attach an Amazon EBS volume to a new Amazon EC2 instance.
3. Attach the Volume to your Amazon EC2 Instance.
4. Connect to Your Amazon EC2 Instance.
5. Create and Configure Your File System with PuTTY.
6. Create an Amazon EBS Snapshot.
7. Restore the Amazon EBS Snapshot you created.

**Configurations:**

|  |  |  |
| --- | --- | --- |
| In the AWS management console, search EC2 on Services menu and select it. | |  |
| On the left navigation pane, click instances. Availability zone should be “us-east-1a” and Lab should be an instance. | |  |
| In the left navigation pane, choose Volumes. Then select create volume. | |  |
| In volume settings, configure the following. In tags, select add tag and make the key “Name” and value as “My Volume. Then create volume. | |  |
| Check the box next to My Volume. Under the Actions menu select Attach volume. | |  |
| In Basic details, choose instance field, and select the (lab) instance. Next select Attach volume. | |  |
| Return to AWS lab instructions. Under details, select show. A credentials popup will show up. Click Download PPK. | |  |
| Click the link saying “download it here” to download PuTTY. | |  |
| Open PuTTY. Click onto Connection and set the seconds between keepalives to 30. Click onto the Session Category. | |  |
| Select Instances on the left navigation pane. With Lab instance selected, copy the public IPv4 Address. Paste the copied address under Host Name. | |  |
| In Category, expand +SSH and then expand +SSH. Select Credentials. Under private key file for authentication, select Browse. In your downloads folder, select labsuser.ppk. Then choose “open” in the PuTTY session. Choose accept to connect to host. |  | |
| When prompted login as: enter ec2-user. This will connect you to the EC2 Instance. | |  |
| In PuTTY, to view the storage available on your instance, paste df -h by right clicking. You will receive the following output. | |  |
| To create an ext3 file system on the new volume, enter “sudo mkfs -t ext3 /dev/sdf” into PuTTY. You will receive the following output. | |  |
| To create a directory for mounting the new storage volume, enter “sudo mkdir /mnt/data-store.” | |  |
| Mount the new volume by entering “sudo mount /dev/sdf /mnt/data-store.” To configure the Linux instance to mount this volume whenever the instance is started, enter “echo "/dev/sdf /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab.” | |  |
| View the configuration file to see the setting on the last line by entering “cat /etc/fstab.” The following output is shown. | |  |
| To view the available storage again, enter “df -h.” The output now contains an additional line /dev/xvdf. | |  |
| On your mounted volume, create a file and add text to it by entering “sudo sh -c "echo some text has been written > /mnt/data-store/file.txt".” Verify the text has been written to your volume by entering “cat /mnt/data-store/file.txt” | |  |
| In AWS management console, choose volumes and check my Volume. In the Actions menu, select Create snapshot. | |  |
| Choose add tag and then configure Key as “Name” and Value as “My Snapshot.” Choose Create snapshot. | |  |
| In the left navigation pane, choose Snapshots. Wait until snapshot status is completed. | |  |
| In your remote SSH session delete the file that you created on your volume by entering “sudo rm /mnt/data-store/file.txt.” Verify that the file has been deleted by entering “ls /mnt/data-store/” | |  |
| In the AWS Management Console, select My Snapshot. Under the Actions menu, select Create volume from snapshot. | |  |
| Scroll to the bottom and select the same availability zone that you used earlier (us-east-1a). Choose Add tag then configure Key as “Name” and Value as “Restored Volume.” Then choose Create volume. | |  |
| In the left navigation pane, choose Volumes and select the box next to Restored Volume. Under the Actions menu, select Attach volume. | |  |
| Choose the Instance field, then select the (Lab) instance that appears. The Device field is set to /dev/sdg. Then choose Attach volume. The volume state will be in-use. | |  |
| Create a directory for mounting the new storage volume by entering “sudo mkdir /mnt/data-store2.” Mount the new volume by entering “sudo mount /dev/sdg /mnt/data-store2.” Verify that volume you mounted has the file you created earlier by entering “ls /mnt/data-store2/.” You should see file.txt. You are now finished with the lab. | |  |

**Conclusion:**

Throughout the completion of this lab, I was able to gain skills such as learning to create an Amazon EBS volume, attach and mount the volume to an EC2 instance, create a snapshot of the volume, create a new volume from the snapshot, and attach and mount the new volume to your EC2 instance. With my previous experience with PuTTY, it was fun and applicable to real life in which I was able to Connect to my Amazon EC2 instance using PuTTY.

AWS Lab 5: Build your DB Server and Interact with Your DB Using an App

Joshua Widjaja

P0-2 Cisco CCNP

A close-up of a logo

Description automatically generated

**Purpose:**

The lab was centered around reinforcing the concept of leveraging an AWS-managed database instance for solving relational database needs. You will gain skills from this lab such as launching an Amazon RDS DB instance with high availability, configuring the DB instance to permit connections with your web server, and opening a web application and interacting with your database.

**Background information:**

The major concept of this lab was about AWS databases but particularly focused on Amazon Relational Database Service (Amazon RDS). Amazon RDS is a managed service that sets up and operates a relational database in the cloud. Amazon RDS provides cost-efficient and resizable capacity, while automating time-consuming administrative tasks. It also enables you to focus on your application, so you can give applications the performance, high availability, security, and compatibility that they need. With Amazon RDS, your primary focus is your data and optimizing your application. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL, and Maria DB.

The basic building block of Amazon RDS is a database instance. A database instance is an isolated database environment that can contain multiple user-created databases. It can be accessed by using the same tools and applications that you use with a standalone database instance. The resources in a database instance are determined by its database instance class, and the type of storage is dictated by the type of disks.

DB Instance Class:

* CPU
* Memory
* Network Performance

DB Instance Storage:

* Magnetic
* General Purpose (Solid state drive, or SSD)
* Provisioned IOPS

A blue square with white letter m

Description automatically generated

Use Amazon RDS when your application requires:

* Complex transactions or complex queries.
* A medium to high query or write rate – up to 30,000 IOPS.
* No more than a single worker node or shard.
* High durability

Do not use Amazon RDS when your application requires:

* Massive read/write rates (150,000 writes per second for example)
* Sharding due to high data size or throughput demands.
* Simple GET or PUT requests and queries that a NoSQL database can handle.
* Relational database management system customization.

**Lab Summary:**

1. Create a security group for the RDS DB instance which allows your web server to access your RDS DB instance.
2. Create a DB subnet group which will be used to tell RDS which subnets can be used for the database.
3. Configure and launch a Multi-AZ Amazon RDS for MySQL database instance.
4. Interact with your database by opening a web application running on your web server and configure it to use the database.

**Configurations:**

|  |  |
| --- | --- |
| In the Services menu, search VPC and select it. | **A screenshot of a computer  Description automatically generated** |
| On the left navigation pane, select security groups and choose create security group. | **A screenshot of a computer  Description automatically generated** |
| Under Basic details, configure the following. Under Inbound rules, choose Add rule. | **A screenshot of a computer  Description automatically generated** |
| In Inbound rules, make the Type MYSQL/Aurora and type sg under source and select Web Security Group. | **A screenshot of a computer  Description automatically generated** |
| Click create security group. You will use this security group when launching Amazon RDS database. | **A screenshot of a computer  Description automatically generated** |
| In the services menu, search RDS and select it. | **A screenshot of a computer  Description automatically generated** |
| On the left navigation pane, choose subnet groups and select create DB subnet group. | **A screenshot of a computer  Description automatically generated** |
| Configure the following under Subnet group details. In add subnets and under the availability zones, check us-east-1a and us-east-1b. | **A screenshot of a computer  Description automatically generated** |
| Expand the list of values under Subnets and select the subnets associated with CIDR ranges 10.0.1.0/24 and 10.0.3.0/24. These subnets will show up in the Subnets selected table. Choose create. | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose Databases. Then choose create database. | **A screenshot of a computer  Description automatically generated** |
| Select MySQL under Engine Options. | **A screenshot of a computer  Description automatically generated** |
| Under Templates choose Dev/Test. Under Availability and durability, choose Multi-AZ DB instance. | **A screenshot of a computer  Description automatically generated** |
| Configure the following. DB instance identifier is “lab-db.” Master username is “main.” Master password is “lab-password.” Confirm password “lab-password.” | **A screenshot of a computer  Description automatically generated** |
| Under DB instance class, select Burstable classes (includes t classes). Under storage, make storage type General Purpose (SSD). Make allocated storage as 20. | **A screenshot of a computer  Description automatically generated** |
| Under connectivity, select Lab VPC under Virtual private cloud (VPC). | **A screenshot of a computer  Description automatically generated** |
| Under Existing VPC security groups, choose DB Security Group. Then deselect default. | **A screenshot of a computer  Description automatically generated** |
| Expand additional configuration then make initial database name “lab.” Uncheck enable automatic backups, enable encryption, and enable enhanced monitoring. | **A screenshot of a chat  Description automatically generated**A screenshot of a computer  Description automatically generated**A screenshot of a computer  Description automatically generated** |
| Wait until the status of lab-db goes from modifying to available. Then click lab-db. | **A screenshot of a computer  Description automatically generated** |
| In lab-db1, copy the link under endpoint. Paste it into a text editor to use later. | **A screenshot of a computer  Description automatically generated** |
| In the AWS lab instructions, choose details, and choose show. Copy the WebServer address. | A screenshot of a computer  Description automatically generated**A screenshot of a computer  Description automatically generated** |
| Paste the WebServer address into a new tab. Select RDS at the top of the page. | **A screenshot of a computer  Description automatically generated** |
| Enter the following information and the password is “lab-password.” Then choose submit. | **A screenshot of a computer  Description automatically generated** |
| After a few seconds the application will display an Address Book. It is using RDS database to store information. Test the web application by adding, editing, and removing contacts. You have successfully completed the lab. | **A screenshot of a computer  Description automatically generated** |

**Conclusion:**

Throughout the completion of this lab, I was able to gain important skills such as knowing how to launch an Amazon RDS DB instance with high availability, configuring the DB instance to permit connections from my web server, and opening a web application and interacting with the database I created. The interaction with the web in this lab was very intriguing to me. I felt immersed into a real-life situation after I ran the application to copy information to the database and the application displayed an Address Book. The interaction I had testing the web application by adding, editing, and removing contacts was very similar to a real situation, in which I have a good chance of encountering when using Amazon Web Services.

AWS Lab 6: Scale & Load Balance your Architecture

Joshua Widjaja

P0-2 Cisco CCNP

A close-up of a logo

Description automatically generated

**Purpose:**

The main concept of this lab was primarily centered around using the Elastic Load Balancing (ELB) and Auto Scaling services to load balance and automatically scale an infrastructure. The skills that we should be able to gain after the completion of the lab is creating an Amazon Machine Image (AMI) from a running instance, creating a load balancer, creating a launch template and an Auto Scaling group, automatically scaling new instances, and creating Amazon CloudWatch alarms and monitoring performance of your infrastructure.

**Background Information:**

The two main concepts covered in this lab were Elastic Load Balancing and Auto Scaling.

Elastic Load Balancing automatically distributes incoming application or network traffic traffic across multiple targets such as Amazon EC2 instances in a single Availability Zone or across multiple Availability Zones. It enables you to achieve fault tolerance in your applications by smoothly 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. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce cost. Auto Scaling is suitable to applications that have stable demand patters or those that experience hourly, daily, or weekly variability in usage.

Elastic Load Balancing is available in three types:

|  |  |  |
| --- | --- | --- |
| Application Load Balancer | Network Load Balancer | Classic Load Balancer |
| * Load balancing of HTTP and HTTPS traffic * Routes traffic to targets based on content of request. * Provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. * Operates at the application layer (OSI model layer 7) | * Load balancing of TCP, UDP, and TLS traffic where extreme performance is required. * Routes traffic to targets based on IP protocol data. * Can handle millions of requests per second while maintaining ultra-low latencies. * Is optimized to handle sudden and volatile traffic patterns. * Operates at the transport layer (OSI model layer 4) | * Load balancing of HTTP, HTTPS, TCP, and SSL traffic. * Load balancing across multiple EC2 instances. * Operates at both the application and transport layers. |

Elastic Load Balancing use cases include highly available and fault-tolerant applications, containerized applications, elasticity and scalability, virtual private cloud (VPC), hybrid environments, and invoke lambda function over HTTP(S).

**Lab Summary:**

1. Accessing the AWS Management Console.
2. Create an AMI for Auto Scaling from the existing Web Server 1 which saves the contents of the boot disk so that new instances can be launched with identical content.
3. Create a Load Balancer that balances traffic across multiple EC2 instances and Availability Zones.
4. Create a Launch Template and an Auto Scaling Group.
5. Verify that Load Balancing is Working
6. Test Auto Scaling: Increase the load to cause Auto Scaling to add additional instances.
7. Terminate Web Server 1 because it is no longer needed.

**Configurations:**

|  |  |
| --- | --- |
| In the AWS Management Console, click on Services and search and select EC2. | **A screenshot of a computer  Description automatically generated** |
| On the left navigation pane click instances. Make sure that status check says 2/2 and check Web Server 1. Under Actions menu, click image and templates and click create image. | **A screenshot of a computer  Description automatically generated** |
| Configure image name as “WebServerAMI” and Image description as “Lab AMI for Web Server.” | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, select Target Groups and choose Create target group. | **A screenshot of a computer  Description automatically generated** |
| Choose instances as target type and enter “LabGroup” as the Target group name. | **A screenshot of a computer  Description automatically generated** |
| Select Lab VPC from VPC drop-down menu. Choose next and the Register targets screen will appear. | **A screenshot of a computer  Description automatically generated** |
| After reviewing the settings, choose Create target group. | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose Load Balancers and select Create load balancer. | **A screenshot of a computer  Description automatically generated** |
| Under Application Load Balancer, choose create. | **A screenshot of a computer  Description automatically generated** |
| Under Load balancer name, enter “LabELB.” | **A screenshot of a computer  Description automatically generated** |
| Scroll to the Network mapping section. For VPC, choose Lab VPC. Check both availability zones. Select Public Subnet 1 from subnet dropdown and Public Subnet 2 from subnet dropdown. | **A screenshot of a computer  Description automatically generated** |
| In the Security groups section, check Web Security Group and uncheck default. For the Listener HTTP:80 row, set Default action to forward to LabGroup. | **A screenshot of a computer  Description automatically generated** |
| At the bottom, choose Create load balancer. | **A screenshot of a computer  Description automatically generated** |
| Choose View load balancer. The load balancer will show a state of provisioning. No need to wait until it is ready. | A screenshot of a computer  Description automatically generated**A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose launch templates. Select Create launch template. | **A screenshot of a computer  Description automatically generated** |
| Configure launch template name as “LabConfig.” Check the box next to “Provide guidance to help me set up a template that I can use with EC2 Auto Scaling. | **A screenshot of a computer  Description automatically generated** |
| In the Application and OS Images area, choose the My AMIs tab and for Amazon Machine Image, choose WebServerAMI. | **A screenshot of a computer  Description automatically generated** |
| For instance type, choose t2.micro. For key pair name, choose vockey. | **A screenshot of a computer  Description automatically generated** |
| Under Firewall (security groups), choose select existing security group. Under the Security group dropdowns, check Web Security Group. | **A screenshot of a computer  Description automatically generated** |
| Scroll down to Advanced details and expand it. Under the Detailed CloudWatch monitoring dropdown, select enable. Next, choose Create launch template. | **A screenshot of a computer  Description automatically generated** |
| In the success dialog, choose the LabConfig launch template. Under the Actions dropdown, choose Create Auto Scaling group. | **A screenshot of a computer  Description automatically generated** |
| Configure Auto Scaling group name as “Lab Auto Scaling Group.” Confirm LabConfig template you just created is selected and click next. | **A screenshot of a computer  Description automatically generated** |
| Under VPC, choose Lab VPC. Under the Availability Zones and subnets dropdown, choose Private Subnet 1 and Private Subnet 2. Choose next. | **A screenshot of a computer  Description automatically generated** |
| Choose Attach to an existing load balancer. Under the Existing load balancer target groups dropdown, check LabGroup. | **A screenshot of a computer  Description automatically generated** |
| Scroll down to Additional settings and check “Enable group metrics collection within CloudWatch.” Choose next. | **A screenshot of a computer  Description automatically generated** |
| For group size, set Desired capacity to 2. For scaling limits, set Min desired capacity to 2 and Max desired Capacity to 6. Under scaling policies, choose “Target tracking scaling policy” and configure Scaling policy name as “LabScalingPolicy”, Metric Type as “Average CPU Utilization”, and Target value as 60. | **A screenshot of a computer  Description automatically generated** |
| Scroll to the bottom and choose next. | **A screenshot of a computer  Description automatically generated** |
| You will use the default settings so just choose next. | **A screenshot of a computer  Description automatically generated** |
| Choose Add tag and configure key as “Name” and Value as “Lab Instance.” Choose next. | **A screenshot of a computer  Description automatically generated** |
| Choose Create Auto Scaling group. | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose Instances. You should see two new instances named Lab Instance. | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose Target Groups. Select LabGroup and choose the Targets tab. Wait until the status of both instances transitions to healthy. | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose Load Balancers. Check “LabELB” Load Balancer. In the details pane, copy the DNS name of the load balancer. | **A screenshot of a computer  Description automatically generated** |
| Open a new web browser tab and paste the DNS Name you just copied. The application will appear in your browser. | **A screenshot of a computer  Description automatically generated** |
| On the Services menu, search and choose CloudWatch. | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose All alarms. Check the “OK” alarm. | **A screenshot of a computer  Description automatically generated** |
| Choose Load Test next to the AWS logo. Application will generate high loads. | **A screenshot of a computer  Description automatically generated** |
| Return to the CloudWatch console. AlarmLow should change to “OK” and AlarmHigh status should change to “In alarm.” | **A screenshot of a computer  Description automatically generated** |
| In the Services search box, search and select EC2. | **A screenshot of a computer  Description automatically generated** |
| In the left navigation pane, choose instances. Check Web Server 1. Under the instance state menu, choose Terminate instance. Select Terminate instance to confirm. | A screenshot of a computer error  Description automatically generated**A screenshot of a computer  Description automatically generated** |

**Conclusion:**

Throughout the completion of this lab, I was able to learn how to create an Amazon Machine Image (AMI) from a running instance, creating a load balancer, creating a launch template and an Auto Scaling group, automatically scaling new instances, creating Amazon CloudWatch alarms and monitoring performance of my infrastructure. I learned that I had to create a launch template first and couldn’t go straight to creating the Auto Scaling group. The launch template helped tremendously as an outline for the creation of the Auto Scaling group.