aws Essentials for Dummies

A complete guide for beginners to master Amazon Web Services

Abound Academy

AWS® Essentials for Dummies

A complete guide for beginners to master Amazon Web Services

Abound Academy

Amazon Edition

For any other information, visit our website: www.aboundacademy.com, or contact our email helpdesk contact@aboundacademy.com.

Table of Content

About the Author

Abound Academy is a Professional Certification Provider Institution which provides content for major professional certification exams such as PMP®, Agile®, Disciplined Agile®, Scrum®, AWS®, Azure®, PSM®, and many other such high-demand certifications. We offer our candidates with exam study materials like online courses, training books, realistic mock questions, and downloadable pdf for all the resources that are featured in our Academy. We help you to boost your professional career by providing a definitive way of getting you certified on your respective certification on your very 1st attempt.

As an academy, we have enabled more than 100,000 individuals with their certification requirements and delivered successful results for more than 50,000 students. Our mission is to act as a stimulant to bring a positive boost in career change for everyone. Our study material and exam simulators are made to help the professionals to get certified, and thus achieve their goals in their respective fields.

We believe that skills and their certification has the power to transform lives and the whole world. We are dedicated to providing best-in-industry training and mock tests that are delivered by highly experienced and competent industry experts. We thrive to work in partnership with communities over the boundaries. Our focus is to become the leading provider of high-quality online certification training to professionals over the boundaries.

1. Introduction

1.1 About this book

Welcome to the AWS Essential content, we are dedicated to getting started on your AWS journey. This AWS essential book is going to introduce you to various AWS products, services and common solutions. This book will also teach you how to recognize the terminology and key concepts as they relate to the AWS platform and also help you navigate the AWS management console. It is going to provide you with the fundamentals to become more prepared in identifying which AWS service to use, so that you can make better decisions about how to build IT solutions based upon your business requirements.

This is the best way, how you can get started using the AWS platform. We are going to cover our foundational AWS services and these include services like:

1. Amazon Elastic Cloud Compute (EC2)
2. Amazon Virtual Private Cloud (VPC)
3. Amazon Simple Storage Service (S3) and
4. Elastic Block Store (EBS)

You will also learn about the key security measures that we provide to you as a customer and a very important service called Identity Access Management (IAM). This book will then dive into some of our AWS database services that we have including things like: Amazon's database: DynamoDB and also the Managed Service Database Environment which we refer to as Amazon Relational Database Services (RDS).

And finally, this book will cover some of the many AWS Management tools that are available to you including:

1. Auto Scaling
2. CloudWatch
3. Elastic Load Balancer and

4. Trusted Advisor

This book has laid it out across these five modules and a couple of those key topics previously discussed.

- Module 1: **Introduction to AWS Cloud Computing**
- Module 2: **Foundational Services** - Amazon EC2, Amazon VPC, Amazon S3, Amazon EBS
- Module 3: **Security, Identity, and Access Management** - IAM
- Module 4: **Databases** - Amazon DynamoDB and Amazon RDS
- Module 5: **AWS Elasticity and Management Tools** - Auto Scaling, Elastic Load Balancing, Amazon CloudWatch, and AWS Trusted Advisor

In this book, we will continue with demos that you can follow along to help you get visual learning cues as to how you build key components inside of AWS.

2. AWS Essentials

This book is geared towards technical end users that are new to Amazon Web Services. Whether you are a developer, an architect, a technical project manager or in any other technical role, this is the place for you to start. AWS essentials will provide you with the foundational knowledge required for you to put AWS to use in your environment and on your projects. This book will provide you with an introduction to AWS products, services and use cases.

After you complete reading this book, you will be able to understand the core benefits of the AWS cloud and you will be better positioned to identify the AWS services and products that best fit your requirements.

In 2006, AWS began offering IT infrastructure services to businesses in the form of web services, now commonly known as **Cloud Computing**. One of the key benefits of Cloud Computing is an opportunity to replace Upfront Capital Infrastructure Expenses with low variable costs and scale your business.

With the cloud, businesses no longer need to plan for epicures servers and other IT infrastructure weeks or months in advance. Instead, they can easily spin up hundreds or thousands of servers in minutes and deliver results faster and more efficiently.

Today, Amazon Web Services provides a highly reliable, scalable, secure and low cost infrastructure platform in the cloud with over a million customers around the world, with data centers located in the U.S, Europe, Brazil, Singapore, Japan, Australia and India, businesses across all industries are taking advantage of the benefits of the cloud via AWS.

As you are about to read, our teams have built some truly amazing services.


aws

3. Module 1: Introduction to AWS Cloud Computing

3.1 AWS Cloud Computing

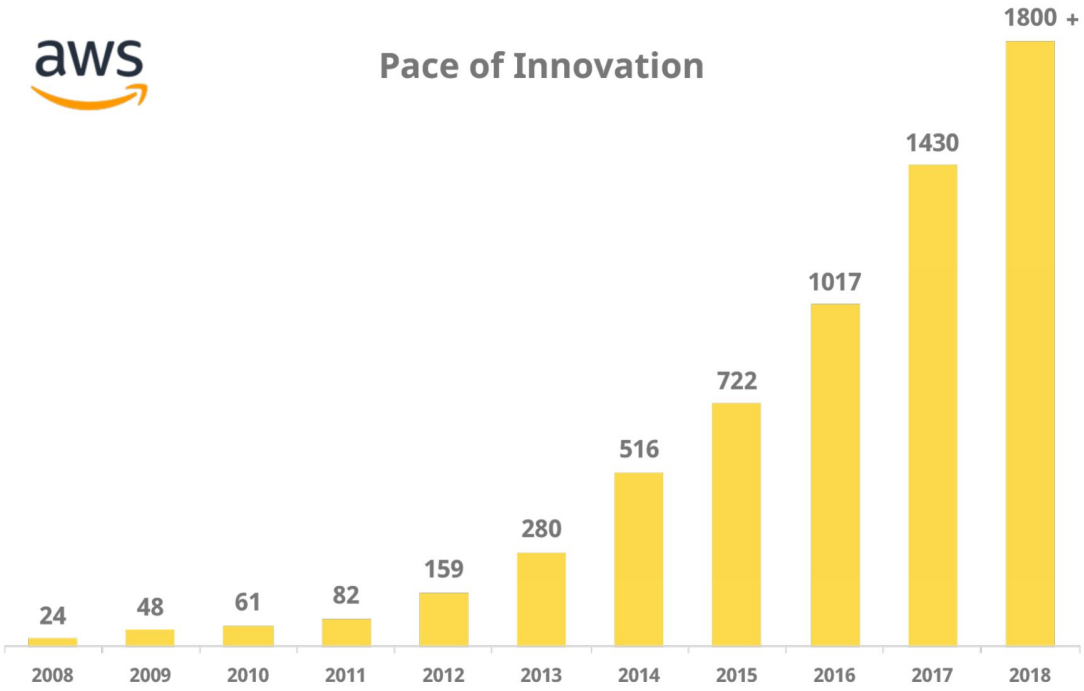
Amazon Web Services is over 10 years in the making and it is a collection of remote computing services that we call **web services**. These web services make up a cloud computing platform offered via the Internet and these cloud services cover a number of different categories that includes:

- Storage,
- Computing,
- Networking and
- Databases.

The AWS mission is to enable businesses and developers to use web services to build scalable, and sophisticated applications on the Internet. Web services is another name for what people now call the **cloud**.

So one thing that drives AWS is the rapid pace of innovation and AWS have been continually expanding their services to support virtually any cloud

workloads. We now have over 50 services available that range from compute, storage, networking, database analytics, application services, deployment, management and also mobile services.



In 2015, AWS launched 722 new features for a total of over 2000 new features available on the platform today since its inception in 2006. Innovation is in their DNA and their structure and approach to product development and delivery is fundamentally different from any other IT vendor out there.

They have decentralized autonomous development teams who are working directly with customers.These development teams are empowered to develop and launch based upon what they learn from those customer interactions and their team iterates products continuously and the newest latest is instantly available to customers.There is no need to upgrade, deploy or migrate.

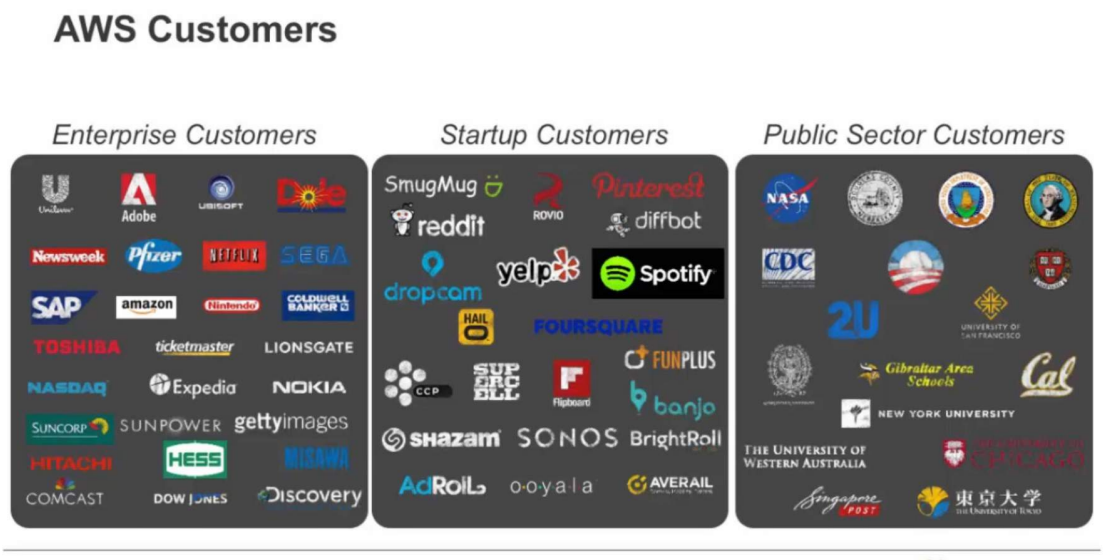
When a feature announcement is ready, the AWS team pushes it out and it is instantly available to any customer that uses that service.This approach also

enables AWS to introduce new services very rapidly and iterate all of them.

We see this today and already very typical of AWS that the number of features are always changing. The previous number is always out of date.We are all in excess of 2000 features and functions that we have across all of the various services that we support on top of the platform.

Some of the AWS customers across really three key categories are:

- Enterprise
- Startup and
- Public Sector



First thing: Starting at **enterprise**, this is a really good demonstration of the maturity of the cloud computing environment and you can refer to the images with some well-known companies/brands and we strongly recommend that you go and have a look at the AWS website.

One of the fantastic things about AWS is that the customers love to talk about what they are doing on top of the AWS platform. It is very difficult to come up with maybe an absolutely unique case for your business or an underlying technology that supports a business application.So learn from what other people are doing and we try and help with that with some fantastic case studies behind lots of these leading enterprise customers.

Secondly, when we look at **startup** customers, it really demonstrates that the truth behind the phrase Cloud is the new norm and you can see here some very interesting companies that are producing fantastic innovative solutions particularly - Super Cell the gaming company. Companies like Reddit, Spotify and Yelp are in our belt. The most amazing thing to see what is behind the game is really a big data analytics solution. It is a very interesting product.

And finally, one is the **public sector** customers. One of the favorite things that public sector
customers demonstrate belief and trust in the platform, or else NASA would have not been in our customer list.

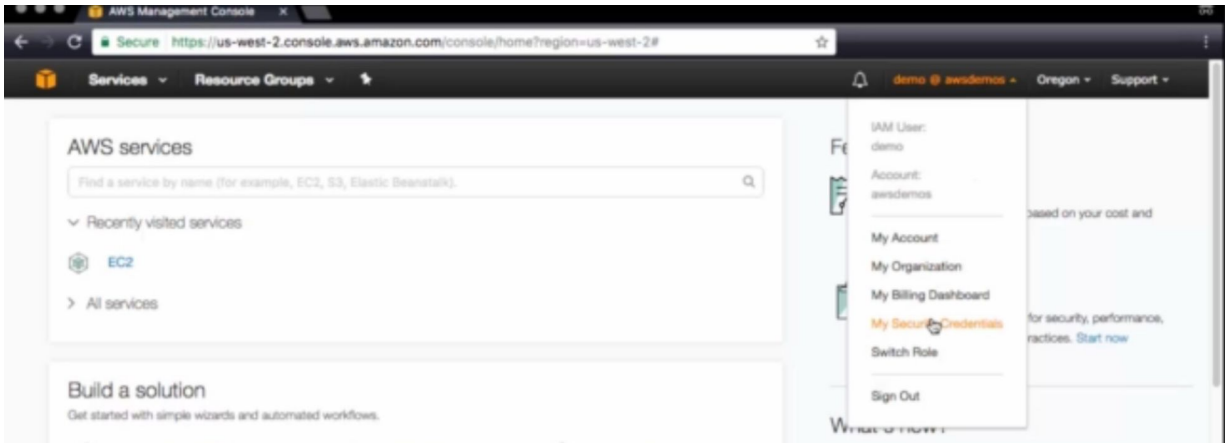
More than that, AWS always tries to enable all of the customer sectors, be it public sector customers, or enterprise, or be a start up, with a number of regulatory standards that we help them achieve on our cloud platform.

3.2 Demo: AWS Management Console

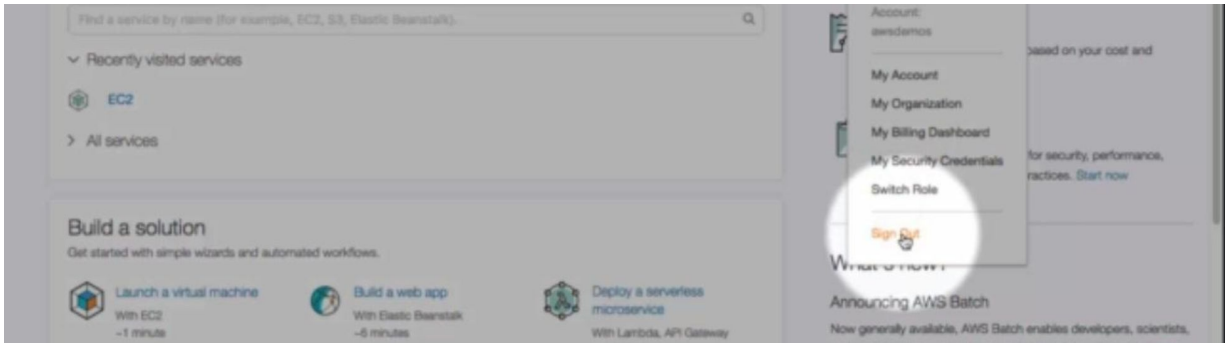
There are a couple of things we can take a look at on the landing page. Let's start by taking a look in the top right hand corner.

We can see that I'm logged in with the demo user identity in the AWS demo's account.

If we click here, we get an additional dropdown that allows us to get to Candy links like our billing dashboard or if we are interested in changing our login password we can click on my security credentials to be taken to that page (below image).



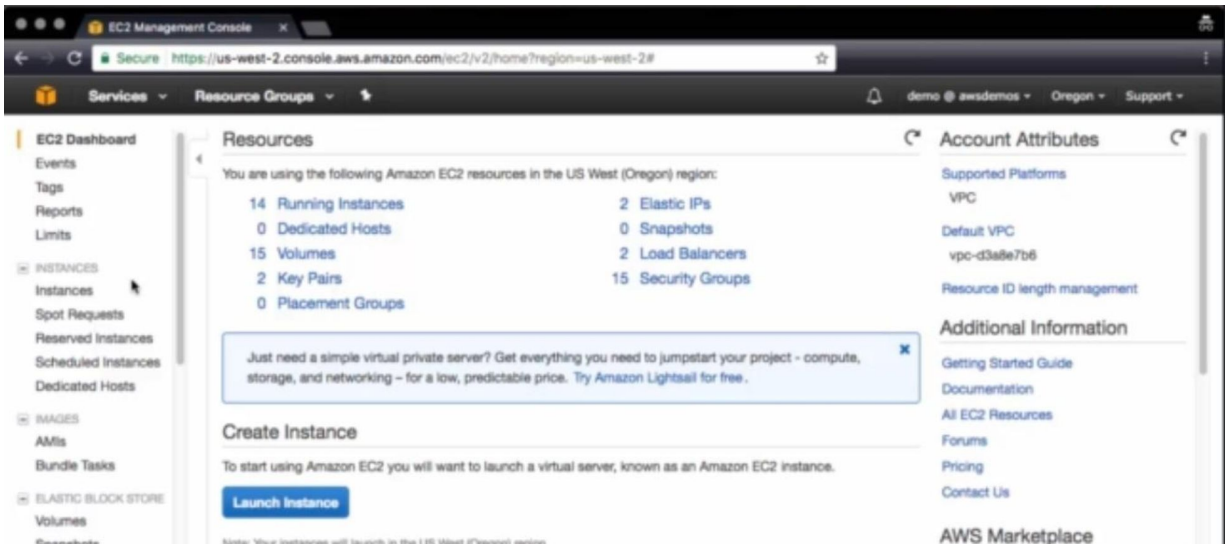
This is also where we will go (below image) if we are interested in signing out of our session.



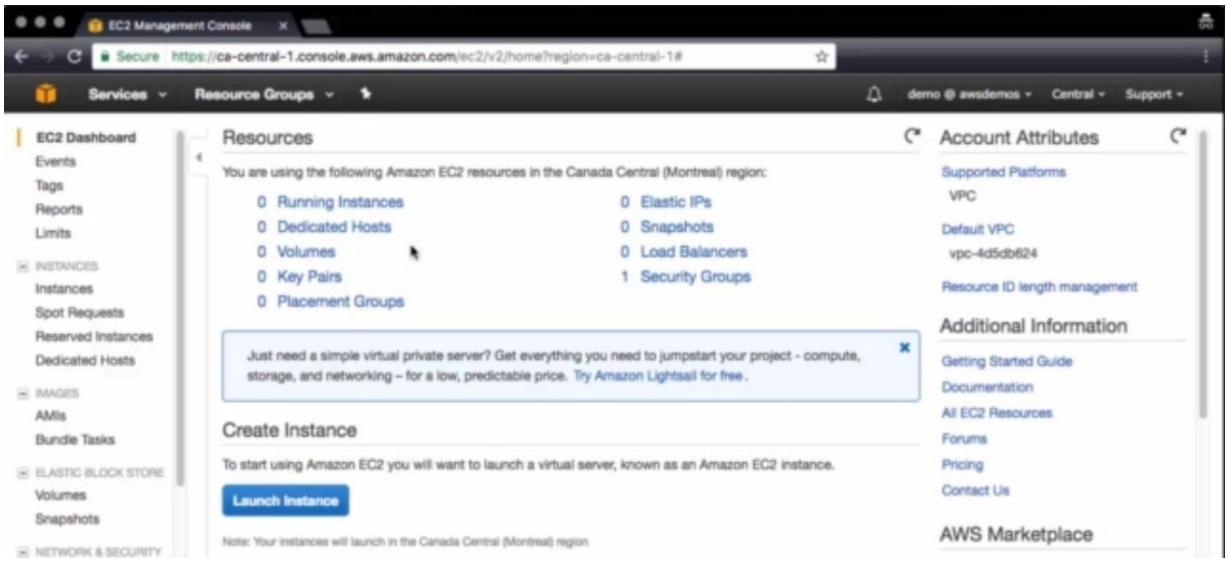
Another item in the top right corner is the region. We can see that I am currently in the Oregon region. By clicking on it, I get a dropdown with all of the available regions that I can change my management console to.

Depending on which service we are operating in, we will only see the resources that exist for the region that the console page is currently set to.

It is important to always take a look at which region you're currently operating inside of because if you load in something like EC2, we are only going to be able to see the resources that exist in our Oregon region.



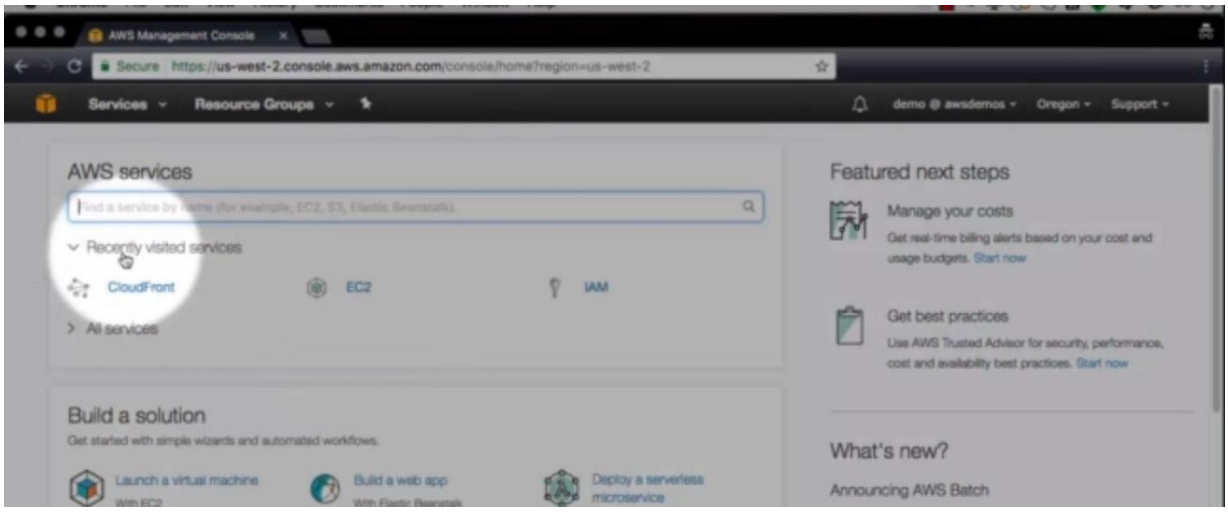
If I were to load this up and switch to our Canada region, we can see that I currently have no running instances. This can be a little bit of a shock when you first load into the console expecting to see something there.



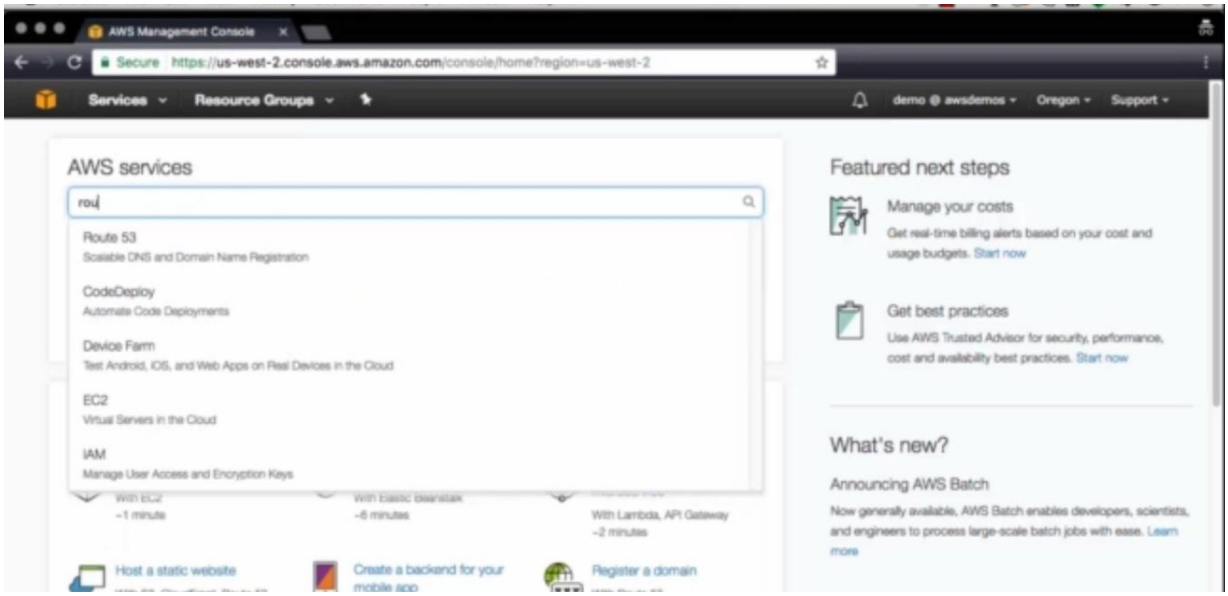
Note: Always check which region you are operating in.

If we want to get to a specific AWS service, we have a couple of options on the left hand side. We can see a listing of all my recently visited

services.This list will be constantly updated as you move around within the management console.



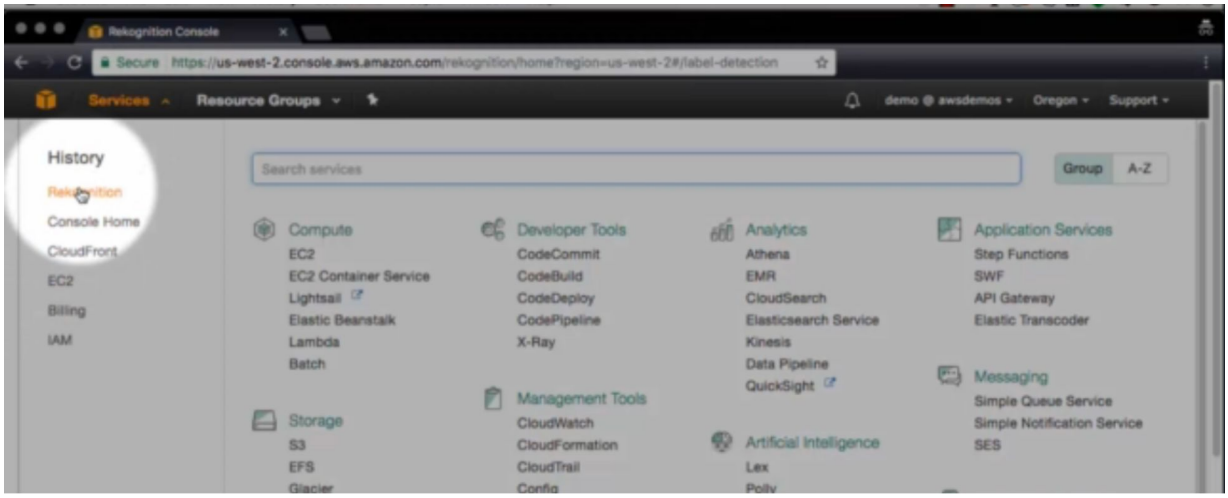
If you know the name of the service you would like to load, you can simply type it in the search bar and the list will be populated based on that search term.



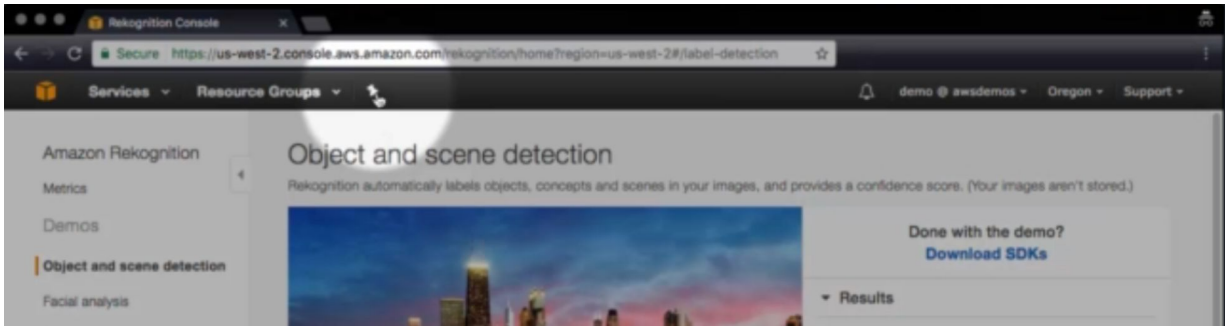
If I am uncertain what the name of the services is but I know the category would fit in, I can click on services in the top left and I get a sorted list of all of the AWS services based on their group.

So if I am interested in the brand new artificial intelligence services, I can very easily find those and then jump over to the service page.

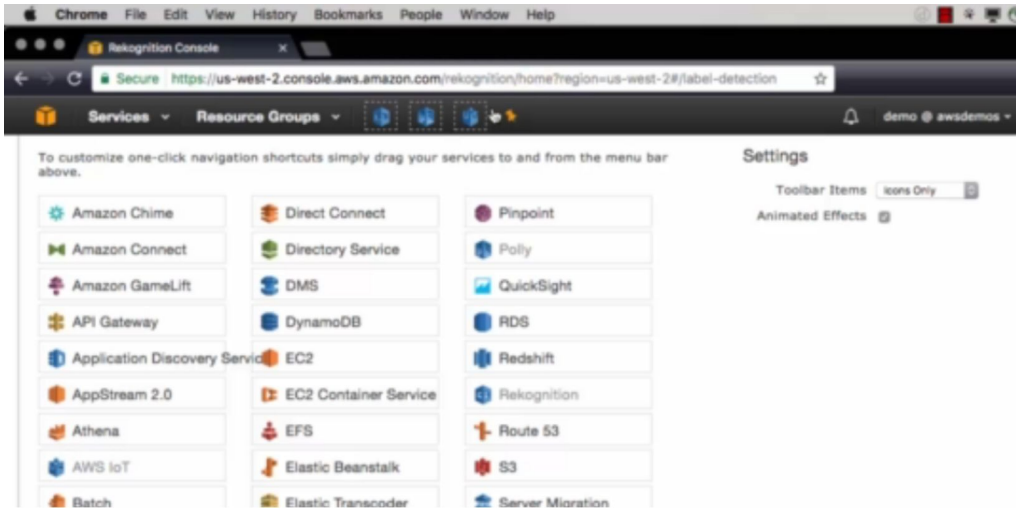
As you continue to operate in the AWS Management Console, you will likely find yourself returning to the same services over and over. One way we can get to those is by clicking services in the top left and taking a look at our history which will have our most recently visited AWS services.



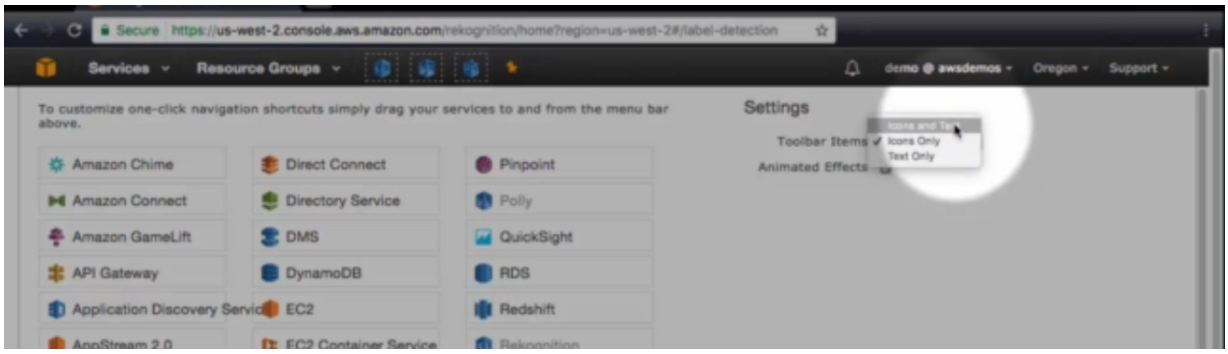
However there's an even easier way built into the console that will allow us to easily get back to frequently visited services and that's going to be with this pin icon in the top.



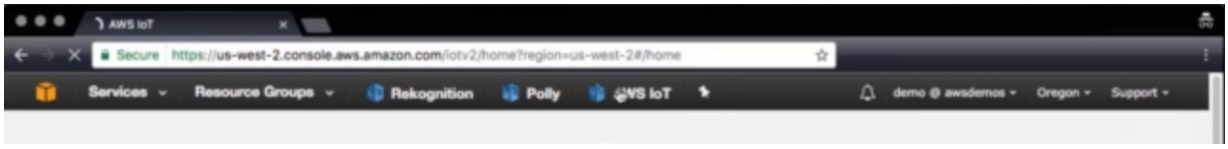
The pin icon allows us to pin services directly to the toolbar at the top of the page. So if I know that I am going to be working with “Rekognition” and “Polly” as well as “AWS IoT”, I can pin those icons to the top.



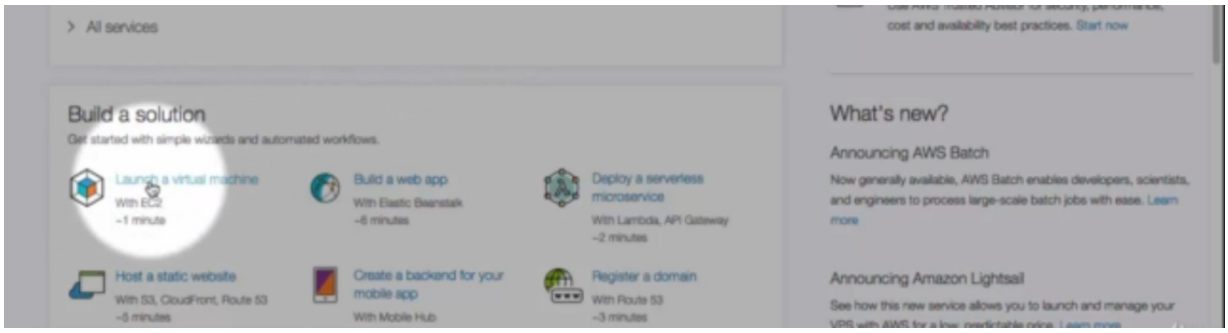
I also have an option on the left hand side of using icons and text or text only for what I pin.



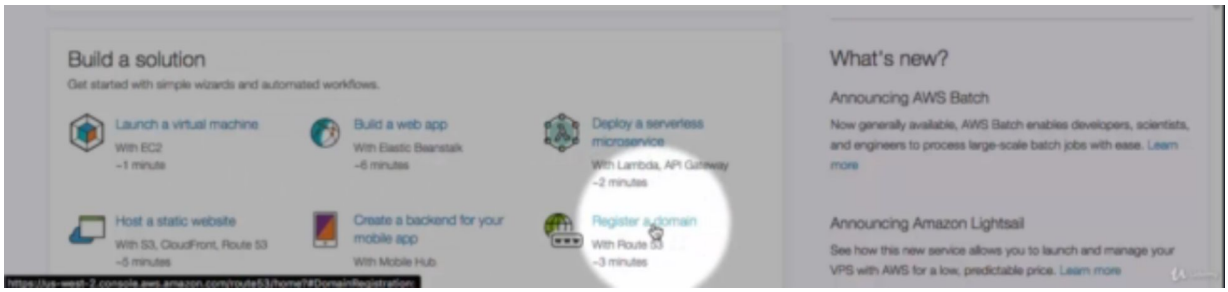
Once services are pinned to the top no matter what page I am in, that will always be available for me to click on up (at the top of the screen).



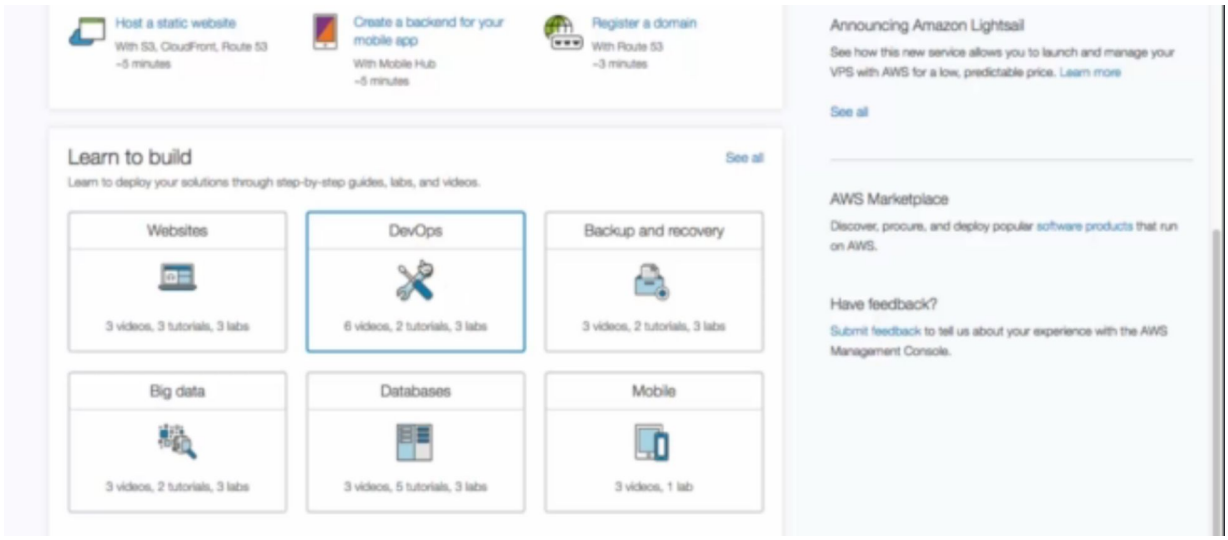
It makes it very easy to navigate between the services that we use most frequently. Also available on the landing page are a couple of learning resources.



The first is a set of QuickStart guides and wizards that allow us to do simple solutions like launch a virtual machine or register a domain with route 53.

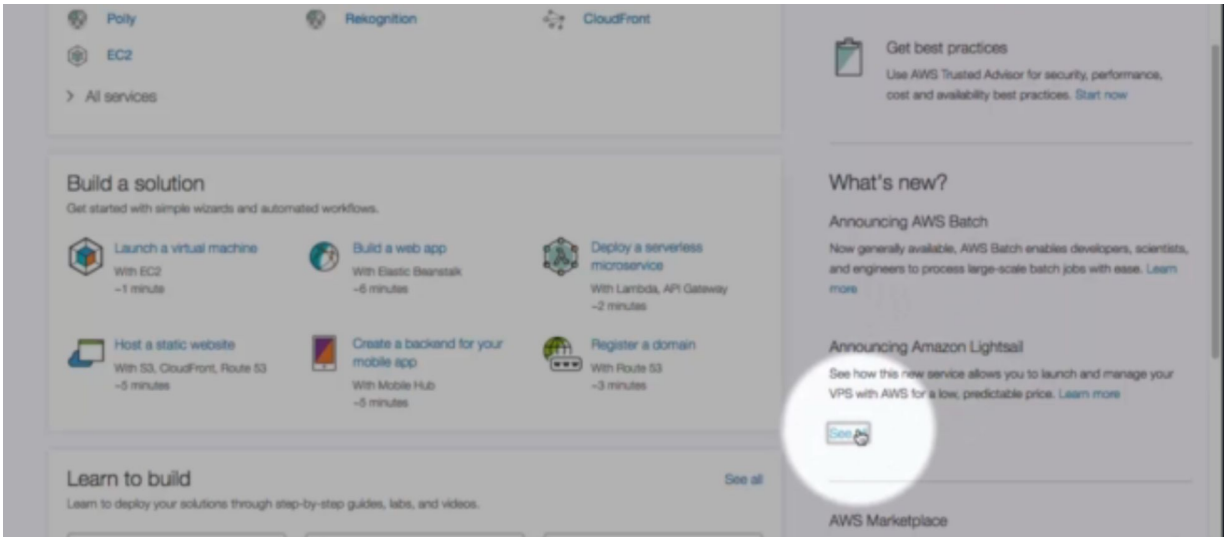


If you want to learn even more by scrolling down on the site, we can see that we have additional groupings of knowledge bases to allow us to do more complex projects.

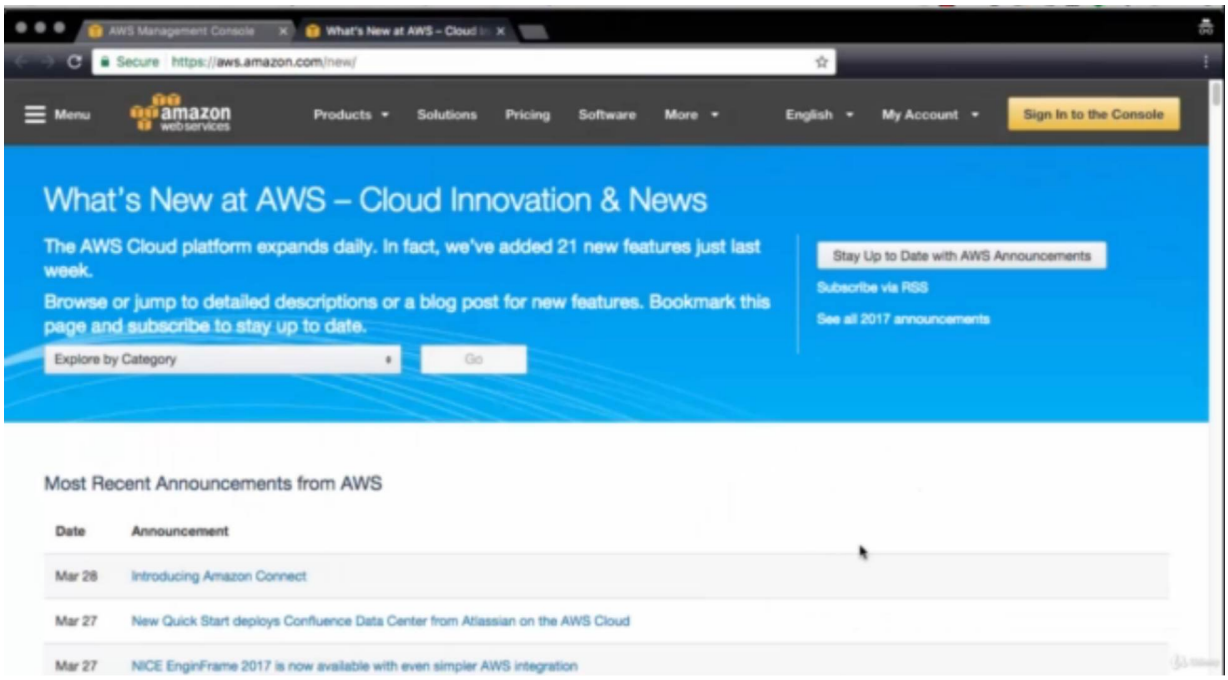


By clicking on one of the topics, we are taken to a curated list of project guides, tutorials and labs as well as additional video resources that can be referenced for that topic. This is a great way to quickly gain the knowledge that you need to get started in AWS.

AWS is continually rolling out new features and services. The best way to keep up with this is actually found right on the right hand side of the console page.



By clicking on the “See All” under the What's New section, we are brought over to the new blog posting which will show us the launch release of each one of these services and features.



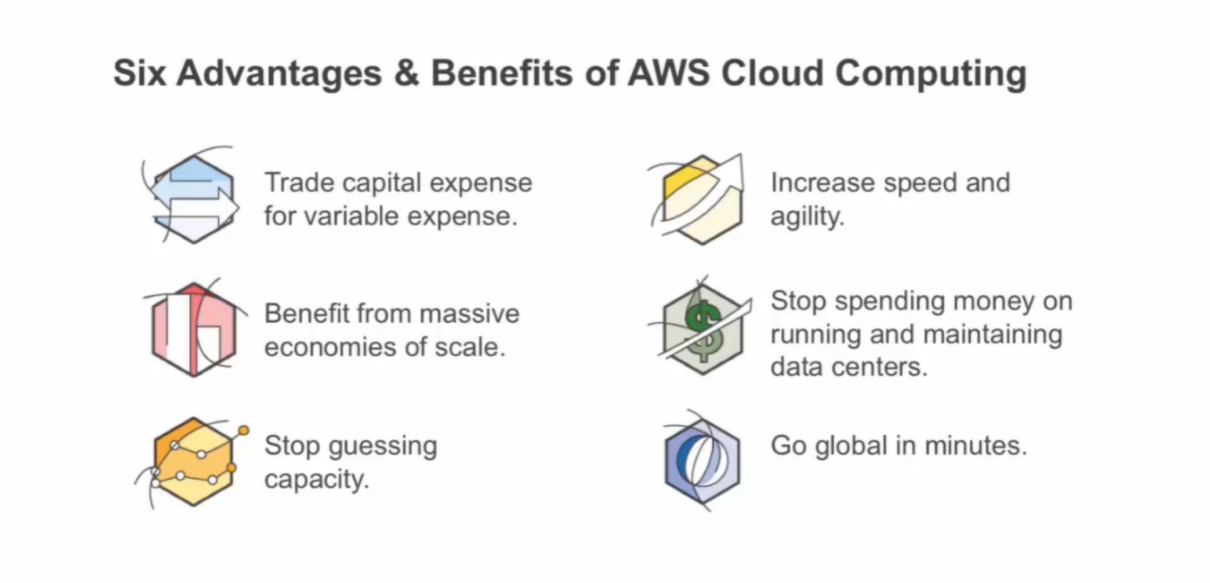
This is the easiest way to keep on top of all of the latest releases.

Now that we've taken it or the AWS management console, the best thing to do is explore just like our AWS services data management console is being continually updated and enhanced.

You never know when you're going to stumble across the cool new feature or release.
Since we are now finished with our session, we can go ahead and sign out.

4. Benefits of Cloud Computing

Here are the Six Significant Advantages and benefits of AWS Cloud Computing.



1. Trade Capital Expense for variable expense.
2. Benefit from massive economies of scale.
3. Stop guessing capacity
4. Increase speed and agility
5. Stop spending money on running and maintaining data centers
6. Ability to go global in minutes

4.1 Trade Capital Expense for variable expense

Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can pay only when you consume computing resources and pay only for how much of that resource you actually need.

4.2 Massive economies of scale

By using cloud computing you can achieve a lower variable cost than you could probably get on your own because you have hundreds of thousands of customers aggregated in the cloud, we have the ability to achieve higher economies of scale.

This translates into a benefit for you with lower pay as you go prices and a very good example of that is the S3 as a service where we've had several significant price reductions in the cost of storage over the 10 years the store has been operating.

4.3 Stop guessing capacity

And then the third one, which is the ability to stop guessing capacity. Eliminating guessing on your infrastructure capacity needs. This is really important because one of the problems that we used to have was the decision on how much capacity do we engineer for an application. And we used to have to look into the future into our crystal ball to decide what was going to be our high watermark of utilization for a particular workload.

And then we would have to pervasion enough IT resource to support that high watermark was a critical benefit nowadays and if I equate this again to AWS services is instead of this pre provisioning of IT whistles we can mix a number of services such as Auto Scaling to look at current utilization of an application and ask that utilization goes up to dynamically add EC2 servers to support that utilization on demand.

And very importantly here, as well as part of a good cost optimization exercise the ability to remove those servers when that demand goes away and most importantly here is that all of this can be automated and left to run independently.

4.4 Increase speed and agility

We are finding most companies today are really focused on the ability to iterate quickly and that ability to iterate fast and implement changes to

adapt to the market they are operating in as a business with speed and agility is a very critical component of them being competitive.

We have lots of AWS tools for developers that give them the ability to spin up environments and go through that development test production release of applications quickly and we tend to use this phrase in AWS which is "**fail early and fail fast**". It is the ability to find problems quickly in early in the development lifecycle.

4.5 Stop spending money on running and maintaining data centers

We say AWS friends do not let friends build data centers. Focus on metrics that differentiate your business and do not focus on infrastructure unless you are an IT company. Cloud computing lets you focus on your customers, on the applications that deliver benefits to those customers rather than all the heavy lifting of racking and stacking and powering servers.

4.6 Ability to go global in minutes

This is the ability to easily deploy your application in multiple regions around the world with just a few clicks. This means that you can provide a low latency and better experience for your customers simply and at minimal cost.

We will cover this in the book ahead when we go through and review the global infrastructure that AWS gives you the ability to run your applications on.

Gartner Magic Quadrant for Cloud Infrastructure as a Service, Worldwide.

When we look at the Gartner Magic Quadrant for cloud infrastructure as a service, there are two significant statements that we can make.

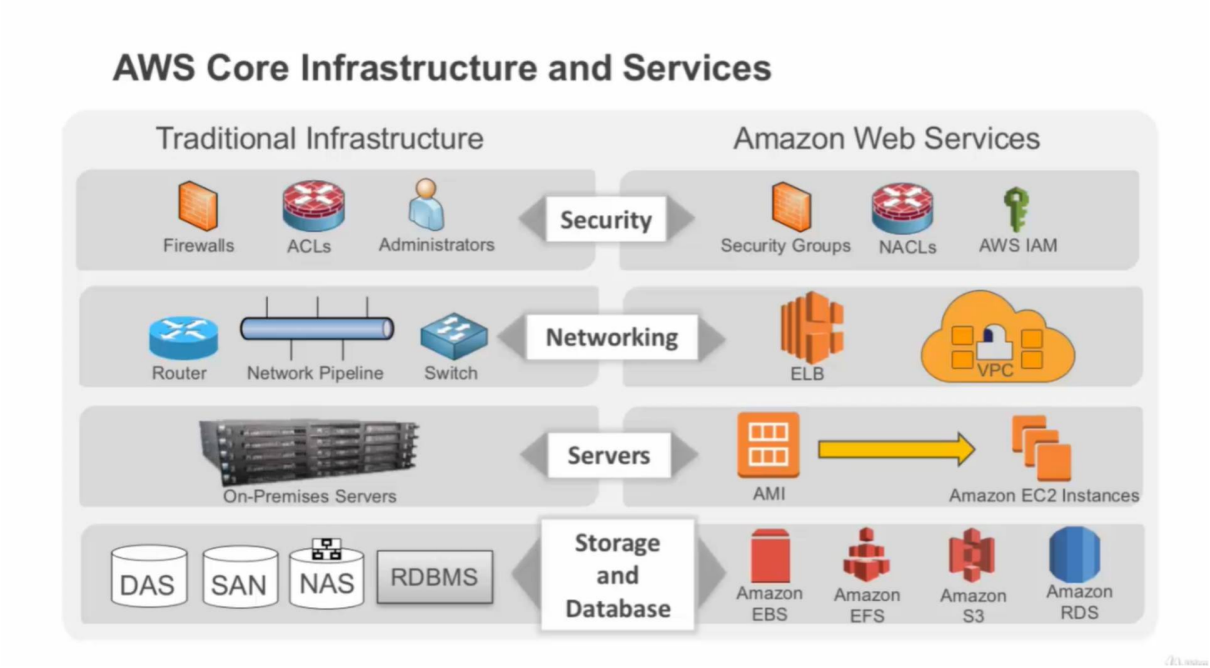
- The first one is that AWS has been operating since 2006 and that has enabled us to grow our global infrastructure that supports the services across the world at significant scale.

- And the second thing that is more exciting is how far to the right of this graph/quadrant AWS is. Because being so far to the right describes that we have a very wide breadth of vision and what that

means to you as a customer is that we have a lot of services that you can use to build exciting applications of workloads within the AWS platform.

5. AWS Platform

The below illustration shows the core AWS infrastructure and services. We may need a dictionary to do some translation of names that the building blocks that we can use to build applications and workloads within AWS should be familiar.



5.0.1 Security layer

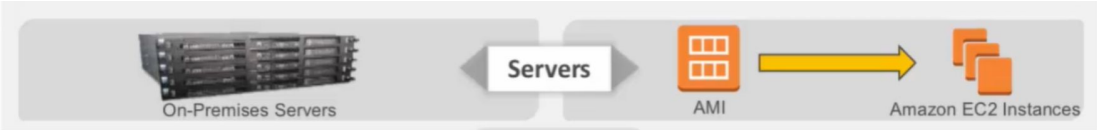
Traditionally, we would refer to things such as Firewalls, Access Control Lists (ACLs) and Administrators but when we come to AWS we reference them by the names of Security Groups, Network Access Control Lists (NACLs) and then we have a service called AWS Identity Access Management (IAM) which allows us to control use of based access to the platform and what those users can do within the AWS environment.

5.0.2 Networking Layer

In the networking layer, a very important component on the right hand side is a VPC - Virtual Private Cloud which we create. We still have to assign IP addresses and all those things that were traditional and are familiar with. We also create subnets within those environments and it is within the PC that we launch our application services that we can use.

There are lots of other components and services that we interact with as part of that network plan including things like Elastic Load Balancers.

5.0.3 Server Layer



When we are looking at AWS, you will see that we have two constructs being shown.
The first one is an Amazon machine image. This is like an image of an operating system that we can use to create an EC2 instance or a Virtual Server running in the cloud.

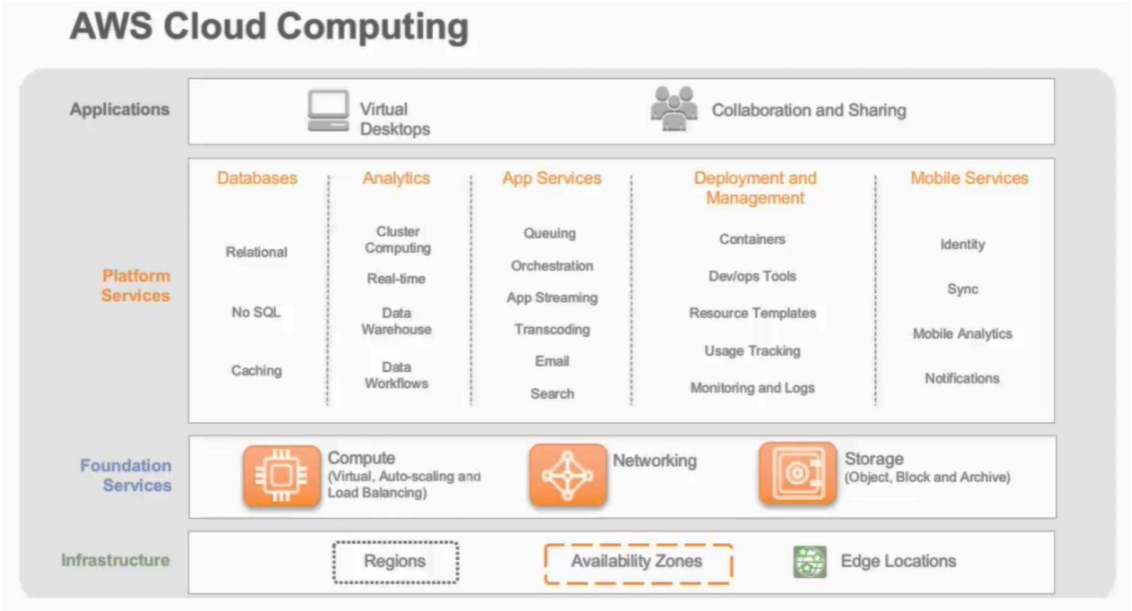
5.0.4 Storage and Databases

Whether it is an Amazon Elastic Block Store (EBS), Elastic Fall Service (EFS), Simple Storage Service (S3) or you're storing data in a fully managed Amazon Relational Database Service (RDS). There are lots of different choices and it is very important that we choose the right one based on cost and performance for our applications.

5.1 AWS Cloud Computing

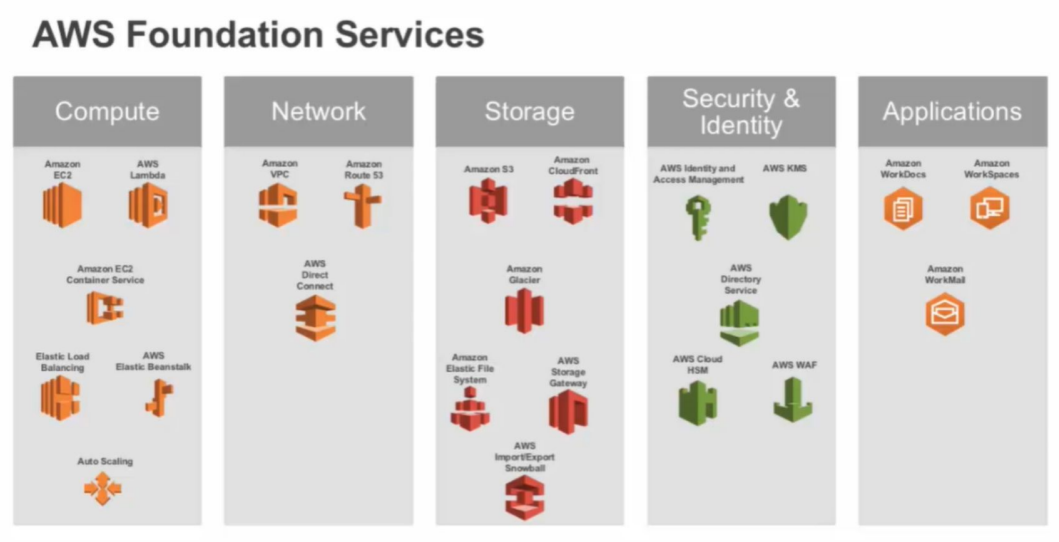
Next, we are going to look at how we break down AWS Cloud Computing into the three critical layers which are:

1. Platform Services
2. Foundation Services and
3. Infrastructure.



When we look at the Foundation Services within AWS, we are breaking those down into five primary categories which are:

1. Compute
2. Network
3. Storage
4. Security & identity and
5. Applications

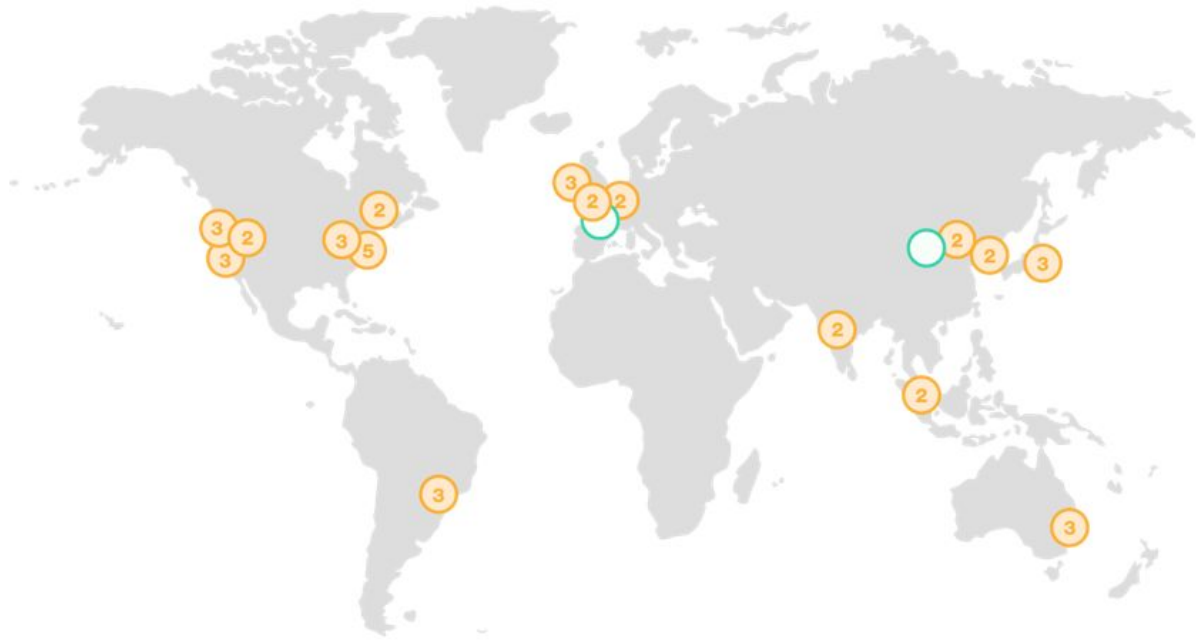


Throughout the remaining book we are going to look at services that fall into each of these Foundation Categories.

We also have that platform services so platform services fit across a large number of domains including databases, analytics, apps services, management tools, development tools, mobile services and Internet of Things (IoT).

5.2 Global Infrastructure

AWS operates its global infrastructure across multiple regions around the world. And there are really significant decisions that you have to go through when you choose to run your application workload in a particular region.



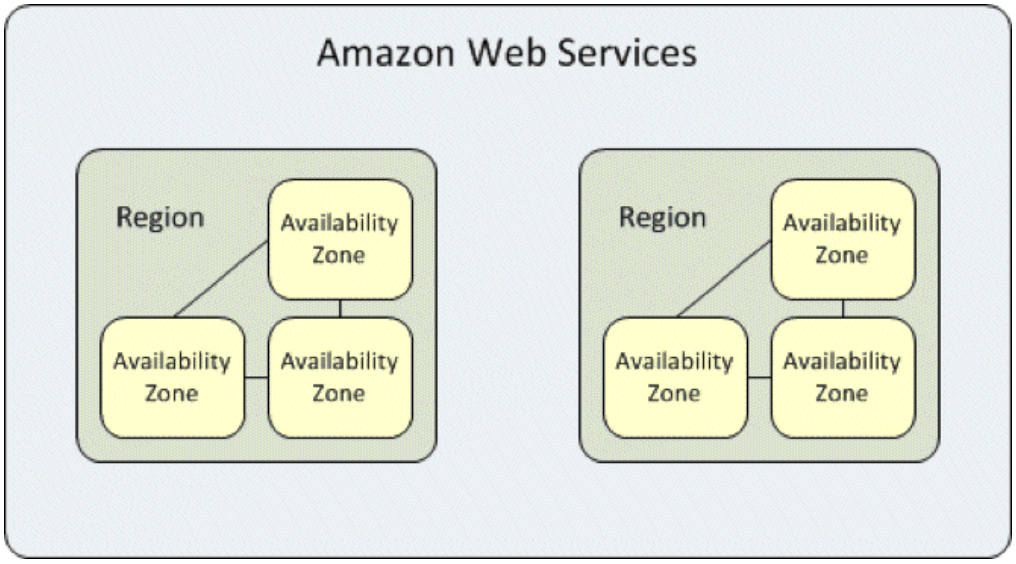
And primarily there are four decisions that you have to go through before you make the correct choice of a region. The four decisions revolve around:

1. Latency,
2. Regulator & Client Compliance Control
3. Cost and
4. Service Availability.

5.2.1 AWS Region

An AWS Region is a geographical location with a collection of Availability Zones (AZs) mapped to physical data centers in that region. Availability Zones consist of multiple data centers clustered in a region. Every region is physically isolated from and independent of every other region in terms of location, power, water supply, etc. This level of isolation is critical for workloads with compliance and data sovereignty requirements where guarantees must be made that user data does not leave a particular geographic region. The presence of AWS regions worldwide is also important for workloads that are latency-sensitive and need to be located near users in a particular geographic area.

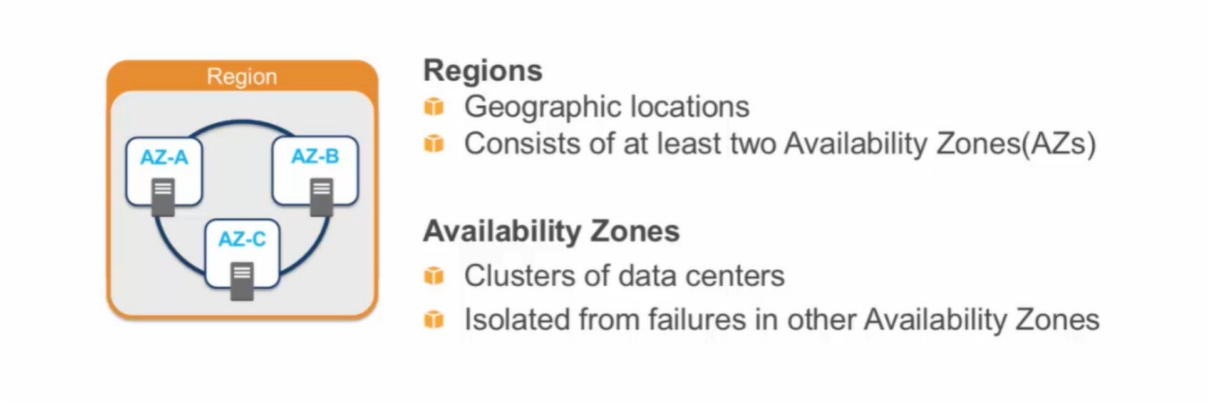
Inside each region, you will find two or more availability zones with each zone hosted in separate data centers from another zone.



5.2.2 Availability Zone

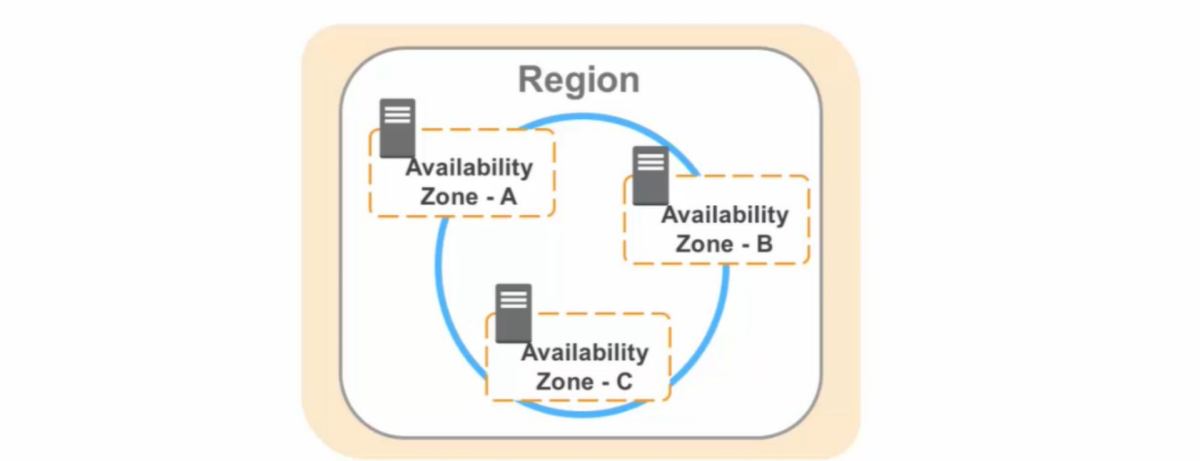
An Availability Zone is a logical data center in a region available for use by any AWS customer. Each zone in a region has redundant and separate power, networking and connectivity to reduce the likelihood of two zones failing simultaneously. A common misconception is that a single zone equals a single data center.

In fact, each zone is backed by one or more physical data centers, with the largest backed by five. While a single availability zone can span multiple data centers, no two zones share a data center. Abstracting things further, to distribute resources evenly across the zones in a given region, Amazon independently maps zones to identifiers for each account.

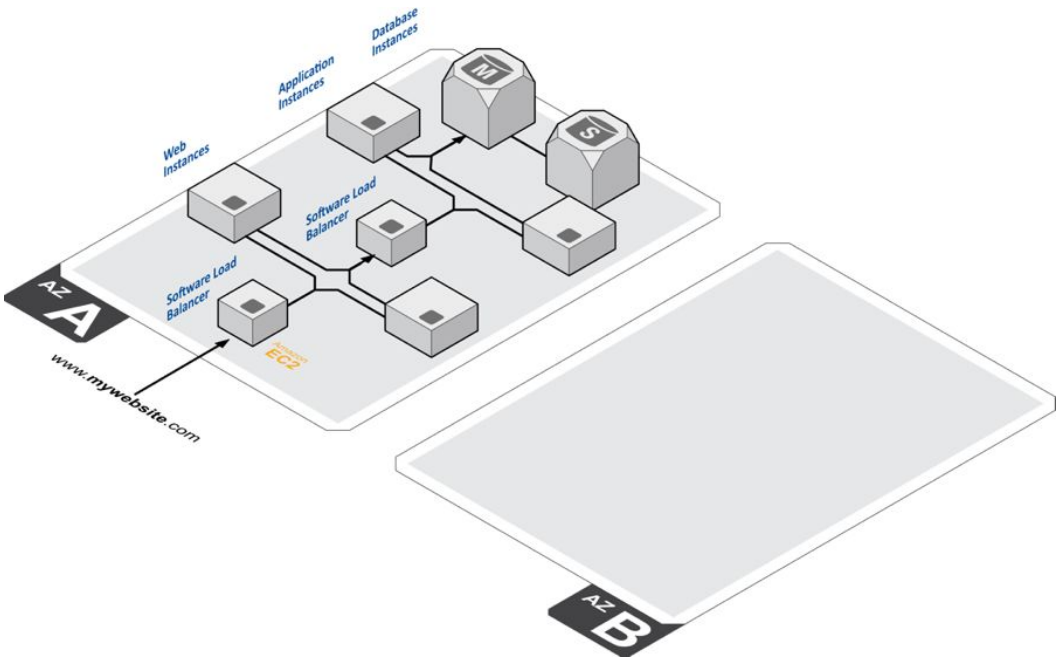


It is very important that you understand how we build each region with multiple Availability Zones so that you can architect highly available and fault tolerant application workloads when building an AWS environment.

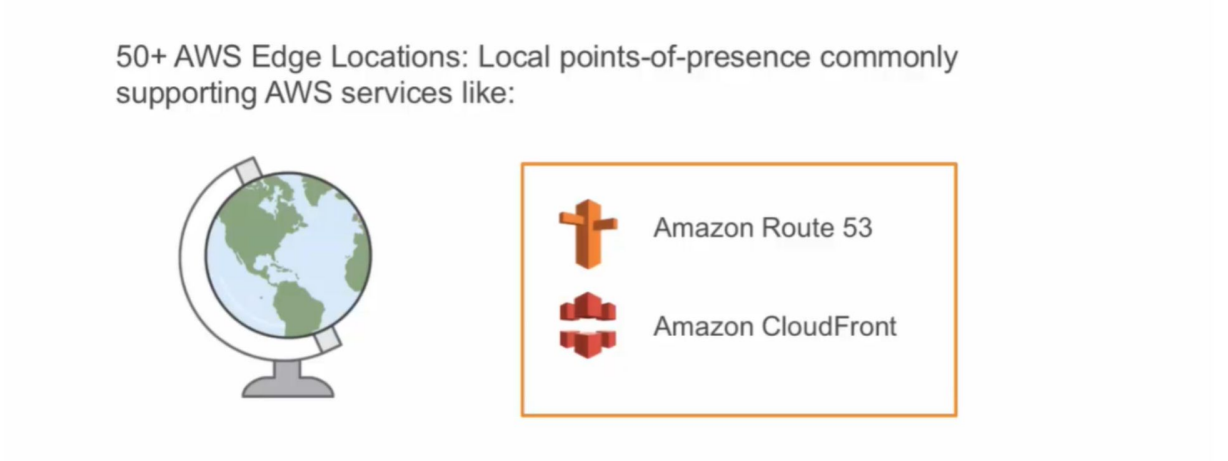
So we highly recommend the provision of compute resources across multiple Availability Zones. If you have multiple instances you can run them across more than one AZ and get added redundancy. If a single AZ has a problem, all assets in your second AZ will be unaffected.



The next diagram illustrates a region with two zones where only one is being utilized. The architecture mirrors what a typical three-tier application running in a user's single on-premises data center may look like. While there are redundant servers running in each tier, the data center itself is a single point of failure.



Another part of the AWS global infrastructure are the 50 plus AWS Edge Locations that we maintain. These are local points of presence commonly supporting AWS services like Amazon Route 53 which is our DNS service and also Amazon CloudFront with our content distribution network.



Edge Locations help reduce latency and improve performance for end users and AWS Edge Location hosts a robust content delivery network that can be used to deliver websites, dynamic & static and streamlining content. Requests for content are automatically routed to the near central location, so the content is delivered with the best possible performance.

6. Module 2: Introduction to AWS Foundational Services

The core AWS Foundational Services are :

1. Compute
2. Storage and
3. Networking

AWS provides a variety of computing services which allow you to obtain and configure a capacity with minimal friction. You get complete control of the computing resources that you run on AWS's proven computing environments.

Amazon EC2 reduces the time required to obtain and boot a new server instance system in minutes. It also allows you to quickly scale capacity both up and down as your computing requirements change.

AWS networking products enable you to:

- Isolate and protect your cloud infrastructure.
- Scale your request handling capacity and
- Connect your existing physical network to your private virtual network in the cloud.

You can use the computer networking services with storage, database and application services to create solutions for Computing, Query processing and Storage that span a wide range of applications.

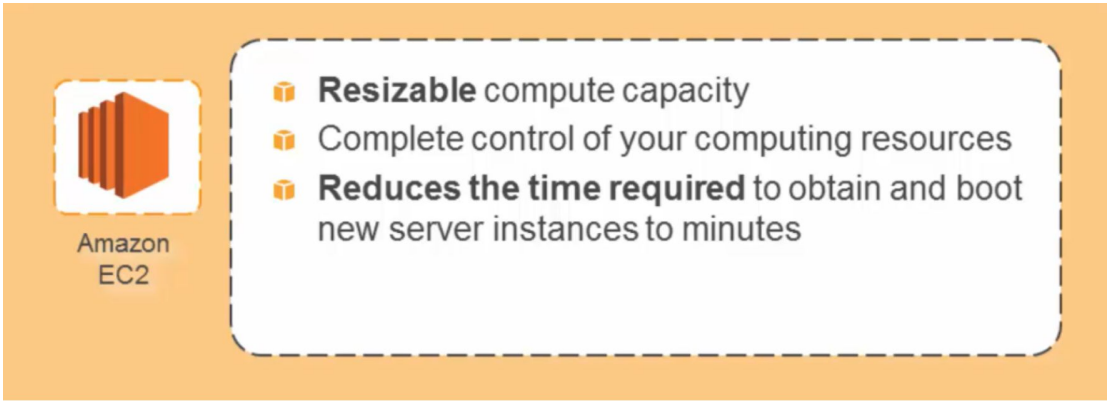
A complete range of cloud storage services are offered by us to support both application and archival compliance requirements. From low cost archival storage, to persistent flexible high performance block storage.

6.1 Amazon Compute Services

The Amazon Elastic Compute Cloud is most popularly known as EC2.

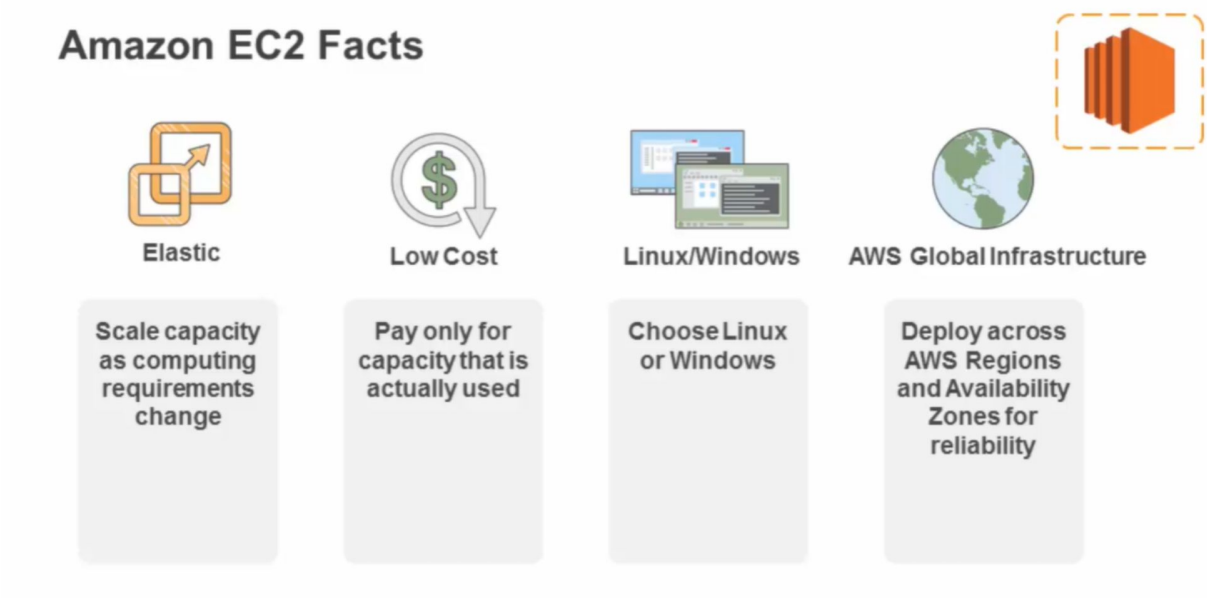
6.1.1 What is EC2 ?

Amazon's EC2 instance is a virtual server that runs in the AWS data centers. It is designed to make web scale computing very easy for developers and IT professionals.EC2's simple web interface allows you to obtain and configure capacity with minimal friction.



It provides you with complete control of your computing resources and even better, it allows you to run on Amazon's proven computing environment.

Amazon EC2 reduces the time required to obtain and boot new server instances. Usually in minutes, allowing to quickly scale capacity as your computing requirements change.



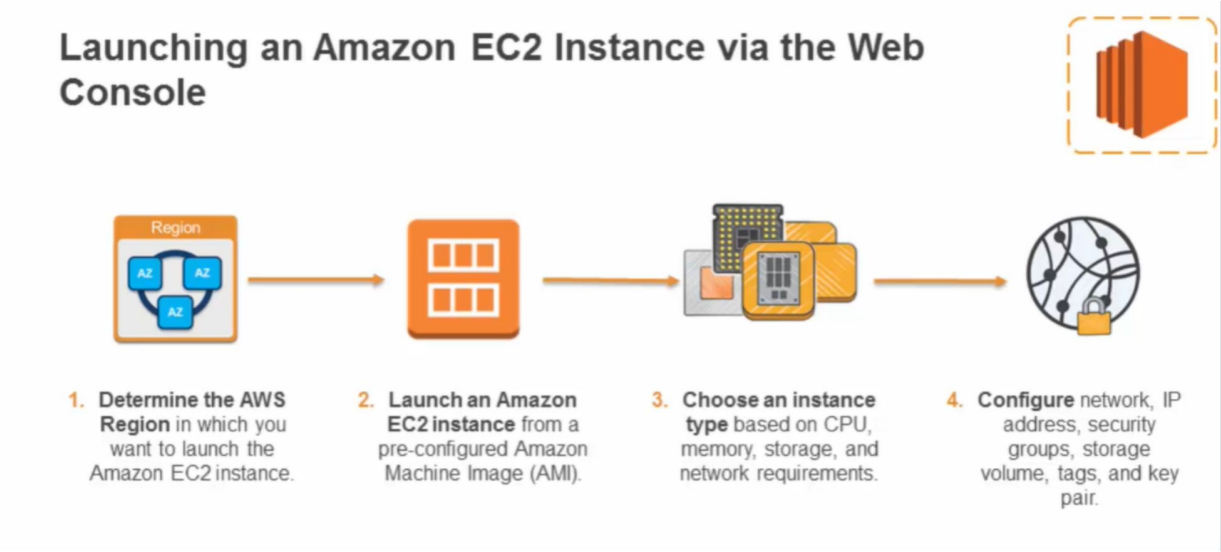
EC2 presents a true virtual computing environment allowing you to use the web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage network access permissions and run your image using as many or few systems that you may need.

You have the ability to program it at the scale of your computing capacity as your requirements change. You pay only for the capacity that you actually use and can choose either Linux or Windows operating systems. You can leverage the global infrastructure to deploy across regions and Availability Zones for reliability.

Now before you create your first Amazon EC2 instance, you need to think about the region in which you want to launch your instances in. Next, you choose an Amazon Machine Image (AMI).

AMI is like the building blocks of an EC2 instance. They are the templates of a computer's volumes. AMI can either have a public or private access.

You can also create gold master images of your Amazon EC2 infrastructure which allow you to decrease your boot times. Once you have your AMI selected, you are then prompted to select an instance type.



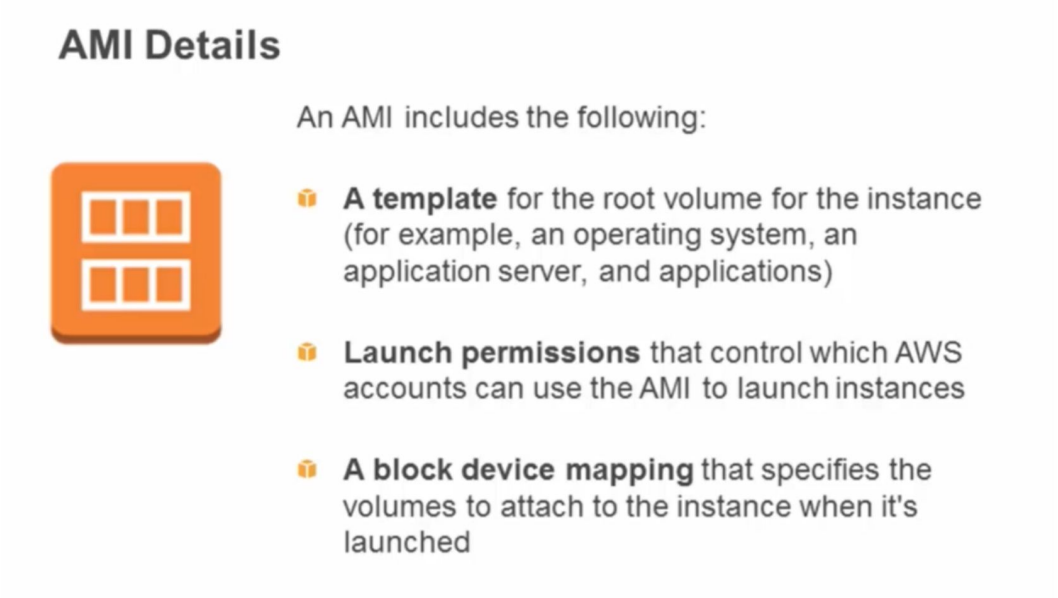
Unlike in a typical virtualized environment like that a VM ware or Hyper-V, where you get the tools that I need to have 10 GB of RAM or 4 virtual CPUs. In EC2, you need to select an instance type.

7. AMIs and Instances

7.1 What is an instance type ?

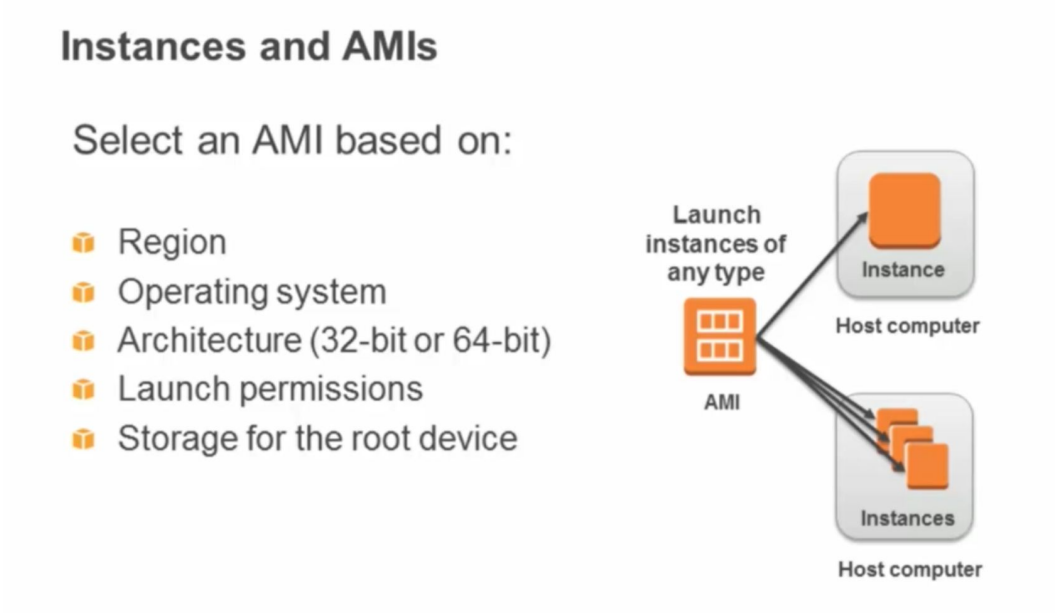
An Instance type is a pre-configured hardware specification of your instances. We have studied the building blocks of EC2 instances that are AMI. An AMI is a template that contains a soft configuration such as an operating system, application server and applications.

You use an AMI to launch an instance which is a copy of the AMI running as a virtual server
on a host computer in the AWS data center. You can launch as many instances as you want from an AMI



You can also launch instances from as many AMIs as you need. You can create your own AMI by customizing the instance that you launch from a public AMI and then saving the conflagration as a custom AMI for your own use.

You can also buy, share and even sell AMIs. AMIs also include a block device mapping that specifies the volumes to attach to an instance when it is launched.

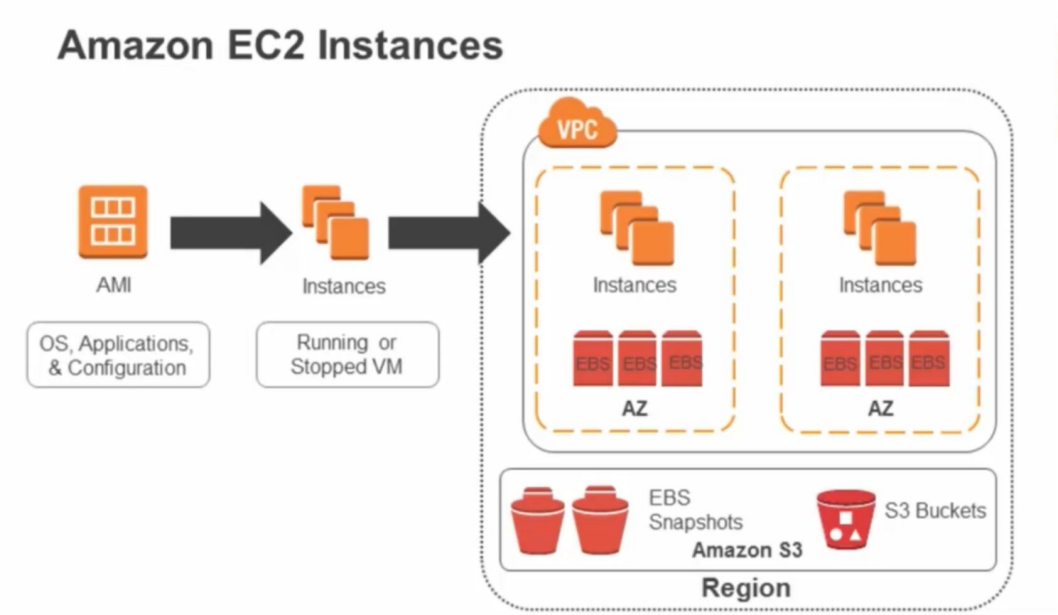


You select an AMI based on the region, the operating system, architecture, launch permissions and storage for the route device. Launch permissions determine the availability of an AMI and are either public ie..the owner grants to launch permissions to all AWS accounts explicit, where the owner grants launch permissions to a specific AWS account or implicit, when the Owner has an implicit launch permissions for any AMI.

You can launch multiple instances of different types from a single AMI. When launching an EC2 instance, an instance type essentially mines the hardware of the host computer used for instance. Each instance type offers different compute and memory capabilities.

Select an instance type based on the amount of memory and computing power that you need for your application or software that you plan to run on the instance. Always remember, Instance keeps running until you stop or terminate it or until it fails.

Instances are deployed in Amazon's EC2 Public Cloud and on the Amazon Virtual Private Cloud in an Availability Zone within a region. You can configure security and network access on your Amazon EC2 instance.



Customers can then deploy to multiple Availability Zones within a region. You choose which instance types you want and then start to terminate, monitor as many instances of your AMI that you need using the web service APIs or the variety of Management tools provided.

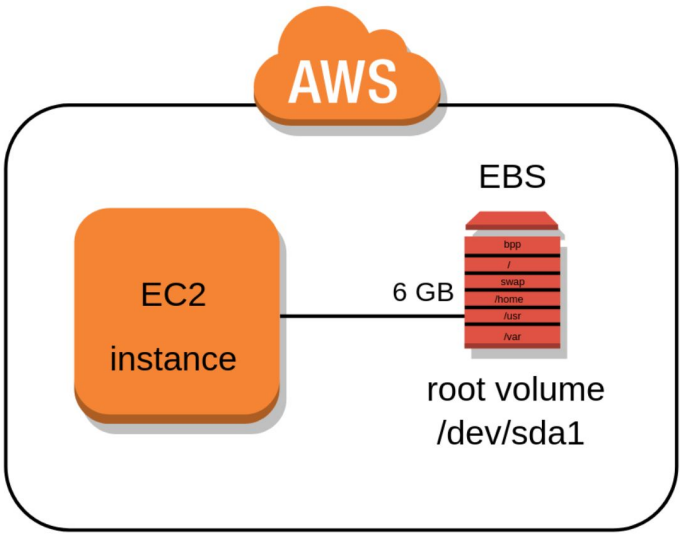
EC2 instances leverage elastic blocks to volumes in each Availability Zone. Determine whether you want to run in multiple locations, utilize static IP in points or attach persistent block stores to your instances.

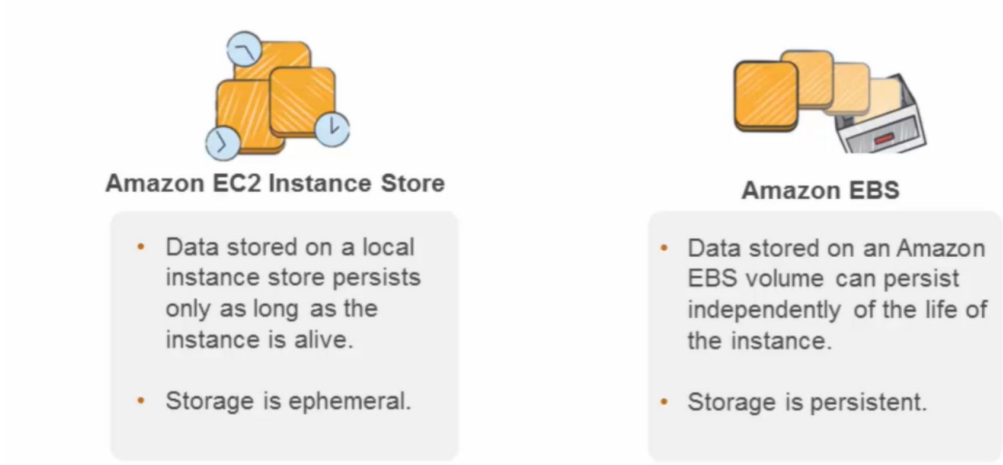
Amazon EBS volumes can be saved using snapshot. Additionally, Amazon S3 buckets can be used to store data/objects that are required by EC2 instances. Pay only for what the resources that you're going to actually consume.

7.2 Instance Store and EBS

Use the local instance store volumes ie.. nothing but the EC2 instance store or the EC2 ephemeral volumes only for storing **temporary data**. When you typically stop and start an EC2 instance, the underlying host is changed. The EC2 instance store volumes are presented to the EC2 instance via the underlying host.

So naturally when you stop and start, the volumes are lost and the data is lost. For data requiring a higher level of durability, use the Amazon Elastic Block Storage (EBS) volumes or backup the data to Amazon S3.





If you are using Amazon EBS volume as your route partition, you need to ensure that you set the need on termination flag to “no”. If you want your Amazon EBS volume to persist outside of the life of theEC2 instance.

AMIs are either Amazon Elastic Blockstore/EBS-backed or backed by the Instance Store Volumes. When an Amazon AMI is EBS-backed, it means that the route device for instance is an EBS volume created from an EBS snapshot.

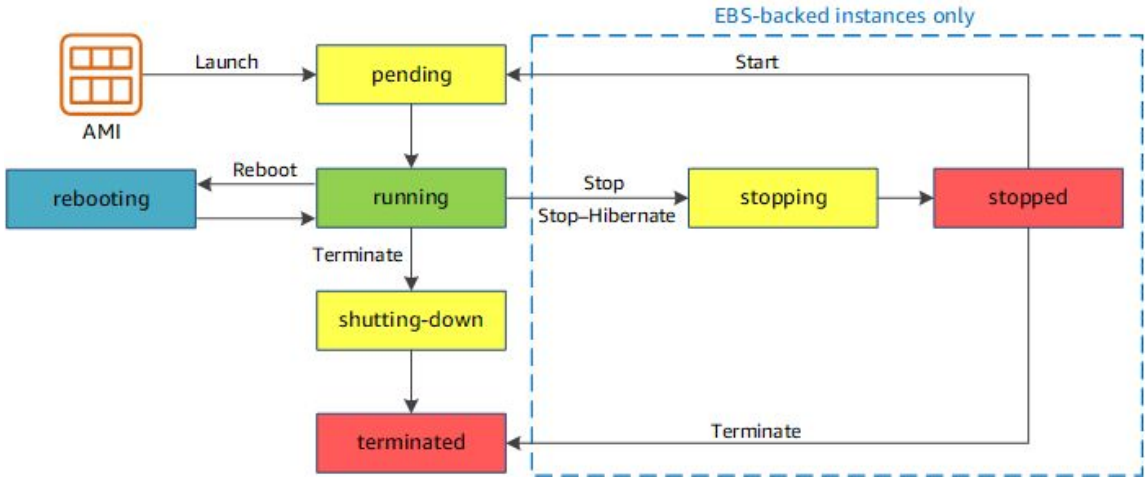
Characteristics	**Amazon EBS-Backed**	**Amazon Instance Store-Backed**
Boot time	Usually < 1 minute	Usually < 5 minutes

Size limit	16 TiB	10 GiB
Data persistence	The root volume is deleted when the instance terminates. Data on any other amazon EBS volumes persists after instance termination.	Data on any instance store volumes persists only during the life of the instance.
Charges	Instance usage, Amazon EBS volume usage, and storing your AMI as an amazon EBS snapshot.	Instance usage and storing your AMI in amazon S3.
Stopped state	Can be stopped	Cannot be stopped

When an AMI is Instance Store-Backed, it means that the route device of the instance was created from a template stored in Amazon S3.

7.3 Instance Lifecycle

An Amazon EC2 instance transitions through different states from the moment you launch it through to its termination. The following illustration represents the transitions between instance states. Note that you can only stop and start instances that are Amazon EBS-Backed or Elastic Blocks Store-Backed.



Amazon EC2 Instance Lifecycle

An EC2 instance can be in one of the following states.

7.3.1 Pending.

When you launch an instance, it enters the pending state and the instance moves from the new host computer. The instance time specified at the launch date to mind the hardware of the host computer or for instance.

7.3.2 Running.

AWS uses the AMI specified at launch to boot the instance. Once your instance is ready for you, it enters the running state. You can connect your running instance and use it as you would do in a typical data center where a computer is sitting in front of you. As soon as your instance is in the running state, you are built for each hour or partial hour that you keep the instance running. You are built for all running instances even if they are idle and not being connected to.

7.3.3 Rebooting.

You can reboot your instance to the Amazon EC2 Console,through the EC2 CLI or by making use of an Amazon EC2 API.
It is recommended that you reboot your EC2 instance rather than running the operating system reboot from the instance. When an instance is rebooted, it remains on the same host computer and it maintains its public DNS name, its private IP address and any data on its instance to a volume. Rebooting an instance does not start a new instance billing hour.

So remember that when you reboot an instance, you are still on the same physical host. But when you stop an instance and start it, you move from one host to another host.

7.3.4 Shutting down.

When you decide that you no longer need an instance, you can go ahead and terminate the instance. The instance will enter the shutting down state. You will stop incurring charges as soon as the instance does the shutting down or terminates its state.

7.3.5 Terminated.

A terminated instance remains visible in the console for a while before it is deleted. You cannot connect or recover a terminal instance.

7.3.6 Stopping.

Amazon EBS-backed instances can be stopped. When you stop an instance, it basically enters into the stopping state. And finally stopped. Amazon EBS-backed instances in the “stopped” state are no longer eligible for early usage or data transfer fees. AWS thus charges for the storage of the EBS volumes on stop instances.

You can modify certain attributes of the stop instances including the instance time. When you start a “Stopped” Instance, it basically puts it into a pending state and then moves the instance to a new host machine.

When you stop and start an instance, you lose any data that you have stored on the instance store or the ephemeral store devices.

7.3.7 Instance reboot

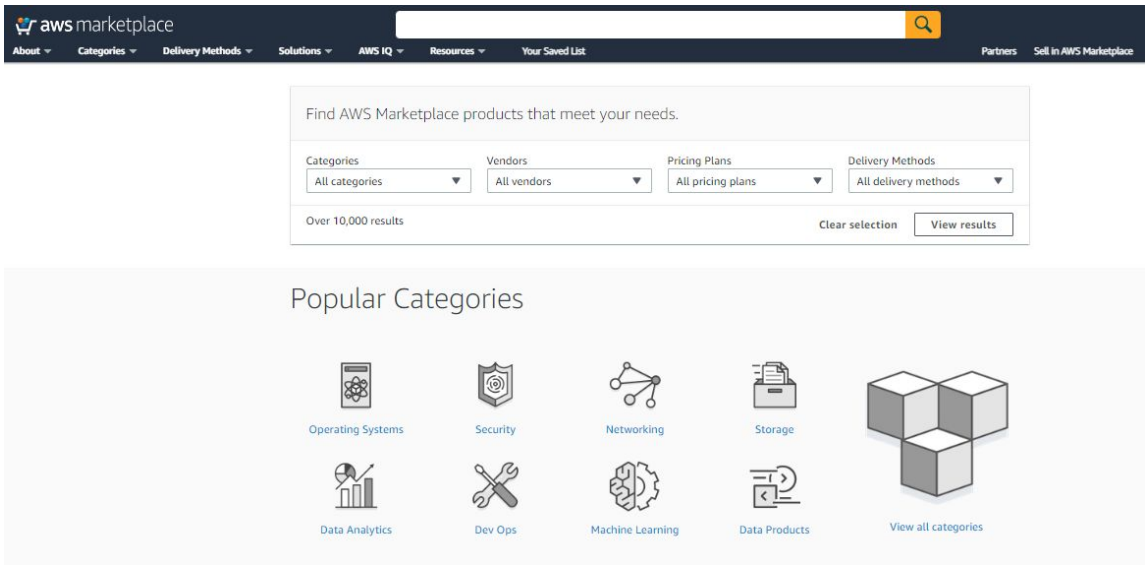
You can reboot your instance using the Amazon EC2 console, a command line tool, and the Amazon EC2 API. We recommend that you use Amazon EC2 to reboot your instance instead of running the operating system reboot command from your instance.

Rebooting an instance is equivalent to rebooting an operating system. The instance remains on the same host computer and maintains its public DNS name, private IP address, and any data on its instance store volumes. It typically takes a few minutes for the reboot to complete, but the time it takes to reboot depends on the instance configuration.

Rebooting an instance doesn't start a new instance billing period; per second billing continues without a further one-minute minimum charge.

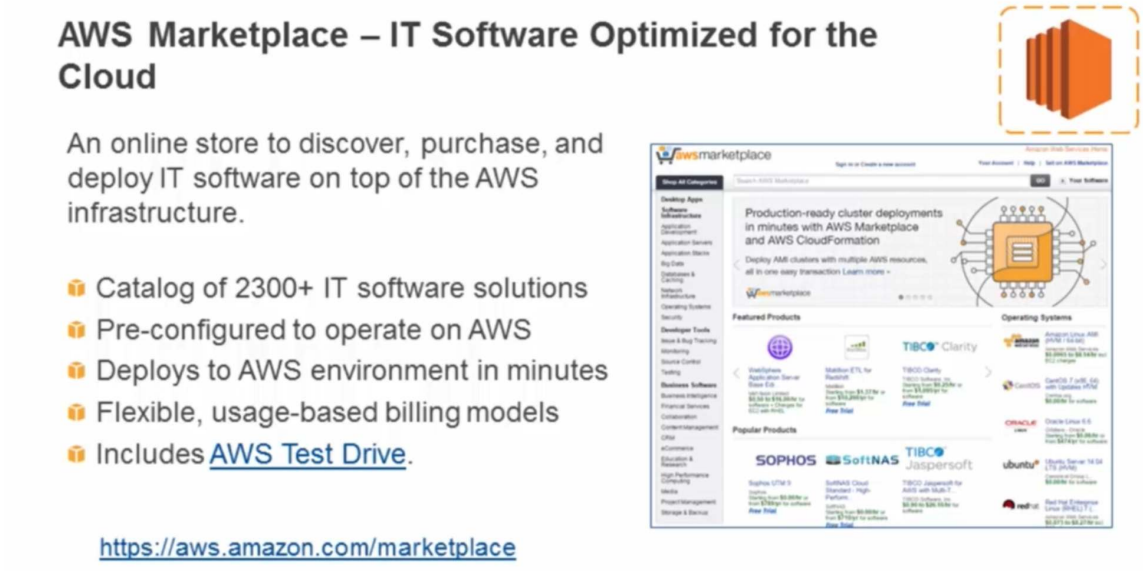
7.4 AWS Marketplace

The AWS marketplace is just like the App Store for Apple iPhones, or the Windows Store for the Windows phones where you go ahead and shop for images that are provided by Amazon or by its partners.



For example, I need a Trend Micro instance. Instead of configuring an EC2 instance and then installing Trend Micro, I can just hop on to the AWS marketplace and search for Trend Micro instances.

There are several benefits that the AWS marketplace provides. For example, easy perk discovery, streamline buying experience, simplified building, a speedier deployment cycle and optimized software capacity.



AWS test drives provide a private IT sandbox environment containing a preconfigured server based solution. In under an hour,and using step by step lab

manuals and videos, launch log in and learn about the popular third party ID solutions powered by AWS and the AWS Cloud Formation.

7.5 Choosing the Right Amazon EC2 Instance

Choosing the right EC2 instance type matters. Selecting an appropriate instance type for your workload can save time and money. AWS has a wide variety of EC2 compute instances to choose from.

Each instance type or family, for example, t2, m3, c4, c3, g2, r3 and so on is optimized for different workloads or use cases. Within an EC2 family, you can choose from different sizes. For example: Micro, Small, Medium, Large, XL and 2XL.

- AWS uses Intel Xeon processors for EC2 instances to provide customers with high performance and value for their computing needs. When you choose your “instance type”.
- You should consider several different attributes of each family. Such as the number of goals, amount of memory, the amount and type of storage, network performance and processor technologies.

Another important consideration is the Total Cost of Ownership (TCO). A lower price per hour instance is not necessarily a money saver. A larger compute instance can sometimes save both money as well as time. It is important to evaluate all the options that you have to see what is the best fit for your workloads.

- AWS recently launched the C4 compute optimized instances that utilize Intel's latest 22nm Haswell microarchitecture. C4 instances use the custom Intel Xeon V3 processors designed and built especially for AWS.

Through its relationship with Intel, AWS provides its customers with the latest and greatest Intel Xeon processors that help in delivering the highest level of processor performance in Amazon EC2. Intel Xeon processors have several other important technology features that can be leveraged by the EC2 instances.

- Intel AVX is perfect for highly parallel HPC workloads such as life science engineering, data mining, financial analysis or other technical computing applications.

- Intel AES-NI enhances your security with new encryption/decryption of data and therefore reduces the performance penalty that usually comes with encryption.

- Intel Turbo Boost Technology automatically gives you more computing power when your workloads are not fully utilizing all the CPU cores. Think of it as an automatic overclocking when you have thermal headroom.

The matrix on this image highlights the individual Intel technologies that we will discuss previously and the EC2 instance family that can leverage each of these technologies.

Let's look at the current generation instances.

Instance Family	**Some Use Cases**
General purpose (t2, m4, m3)	● Low traffic websites and web applications ● Small databases and mid-size databases
Compute optimized (c4, c3)	● High performance front-end fleets ● Video-encoding
Memory optimized (r3)	● High performance databases ● Distributed memory caches
Storage optimized (i2, d2)	● Data warehousing ● Log or data-processing applications
GPU instances (g2)	● 3D application streaming ● Machine learning

General Purpose	Compute Optimised	Memory Optimised	Accelerated Computing	Storage Optimised
A1 ARM based core and custom silicon	C4 Compute - CPU intensive apps and DBs	R4 RAM - Memory intensive apps and DB's	P2 Processing optimised- Machine Learning	H1 High Disk Throughput - Big data clusters
T2 Tiny - Web servers and small DBs		X1 Xtreme RAM - For SAP/Spark	G3 Graphics Intensive - Video and streaming	I3 IOPS - NoSQL DBs
M4 Main - App servers and general purpose		z1d High Compute and High Memory - Gaming	F1 Field Programmable - Hardware acceleration	D2 Dense Storage - Data Warehousing

7.5.1 General Purpose Instances:

T2 instances are low cost burstable performance instance times that provide a baseline level of CPU performance with the ability to burst about the baseline. They offer a balance of compute, memory and network resources for workloads that are occasionally needing to burst such as web servers, built servers and development environments.

The entries in the m4 instances provide a balance of compute, memory and network resources. These instances are ideal for applications that require a high CPU and memory performance such as encoding applications, high traffic content management systems and the memcache applications.

7.5.2 Compute Optimized Instances

Compute optimized instances like c3 and c4 are optimized for compute intensive workloads. These instances have proportionally more CPU than the memory ie..RAM. They are well-suited to applications such as high performance web servers, batch processing and high performance scientific and engineering applications.

7.5.3 Memory Optimized Instances

These instances are r3 instances which are optimized for memory intensive workloads. These instances offer the large memory sizes for high throughput

applications such as high performance databases, distributed memory caches, in-memory analytics and large enterprise deployments of software such as SAP.

7.5.4 Storage optimized instances

The i2 instances are optimized for storage and high random IO performance such as non-sequel databases, scaleout transactional databases, data warehousing, Hadoop and cluster file systems.

The d2 instances are optimized for storage and delivering high disk throughput. d2 instances are exceptionally suitable for the MPP (Massively Parallel Processing) which is MPP data warehousing, map reduce, Hadoop distributed computing, distributed file systems and data processing applications.

7.5.5 GPU Instances

GPU instances or g2 instances are optimized for graphics and graphic processing units that are GPU to compute applications such as machine learning, video encoding, interactive streaming applications.

7.6 Instance Metadata and User Data

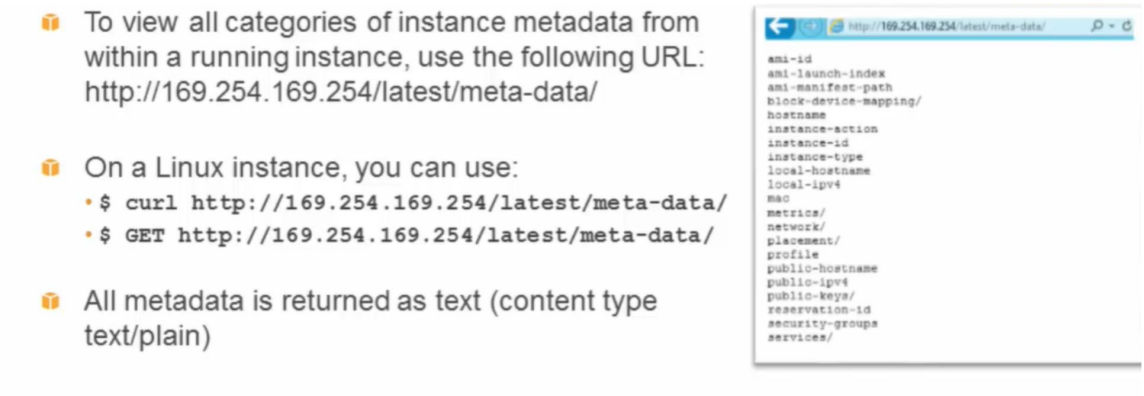
7.6.1 What is Instance Metadata ?

It is the data about you for instance. For example: if you want to know what is a public IP address of my instance or what is the AMI ID of my instance, I can use the instance metadata.

Although you can only access the instance metadata and user data from within the instance itself. The data is not protected by any cryptographic methods. Anyone who can access the instance can view it as metadata therefore you should take suitable precautions to protect sensitive data such as Long lived encryption keys.

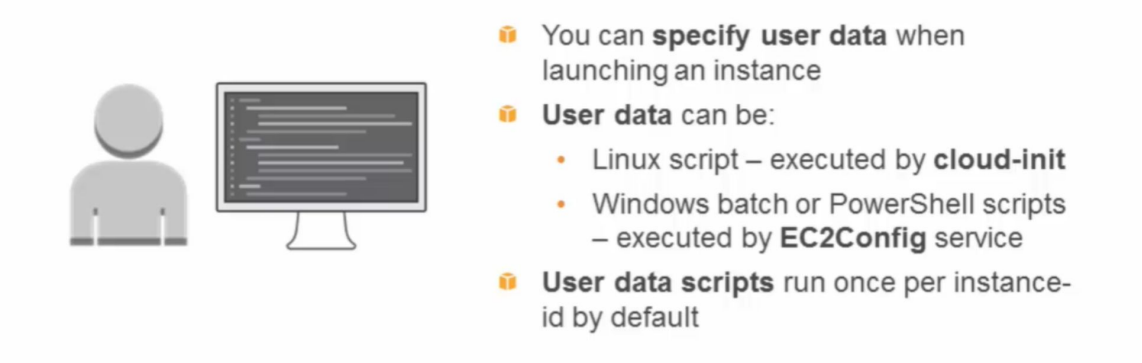
You should also not store any sensitive data such as passwords as user data. To access the Instance Metadata, you connect to a unique URL: http://169.254.169.254/latest/meta-data/

On a Windows machine you can open up the Internet Explorer as you can see in the image or on a Linux machine you can use either the curl or the GET command to connect to that URL.
$ curl http://169.254.169.254/latest/meta-data/
$ GET http://169.254.169.254/latest/meta-data/



Because your instance metadata is available from your running instance, you do not need to use any Amazon APIs or the EC2 console to get the information about the instance. This can be extremely helpful when you're writing scripts to run from your instance.

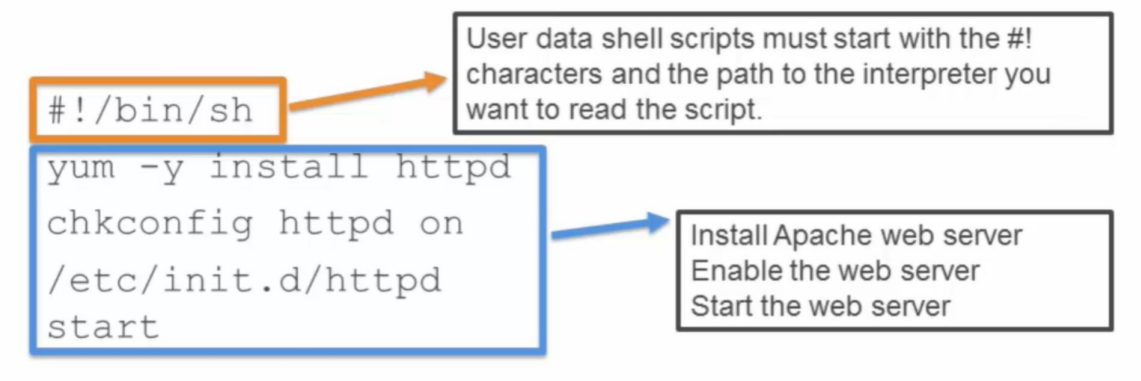
Note that you are never billed for any http request used to retrieve instance metadata or user data. You can specify user data to configure an instance during launch or to run a configuration script. To attach a file, select as the file option or browse for file to attach.



The cloud in a package is an open source application built by the canonical that is used to bootstrap Linux images in a computing environment like Amazon EC2. User data is treated as opaque data. What you give is what you get back.

It is up to the instance to be able to interpret it. User data is limited to 16 KB. This limit applies to data in raw form not in base 64 encoded form. User data must meet base 64 encoding prior to being submitted to the API.

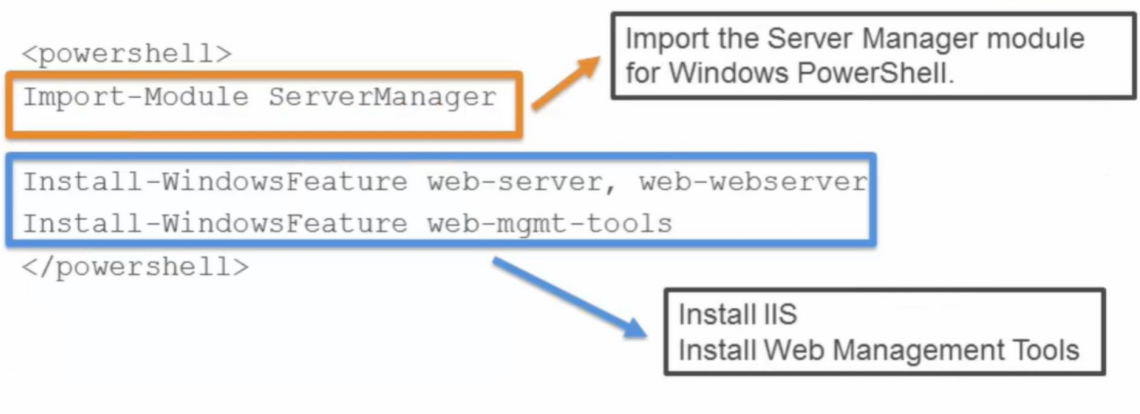
The Amazon EC2 command line tools perform the base 64 encoding for you. The data is decoded before being presented to the instance. User data is executed only at the launch time.



If you stop an instance, then modify the user data and start an instance, the new user data is not executed automatically by default. This example above shows you how we can configure user data in a Linux machine to install a patch of web server, enable the web server and then finally start the web server.

In this example below, we make use of a Windows operating system and we make use of Windows Powershell to configure our operating system ie.. a Windows box

as an IIS server and we also install the web management tools. Notice the difference between the windows user data and the Linux user data.

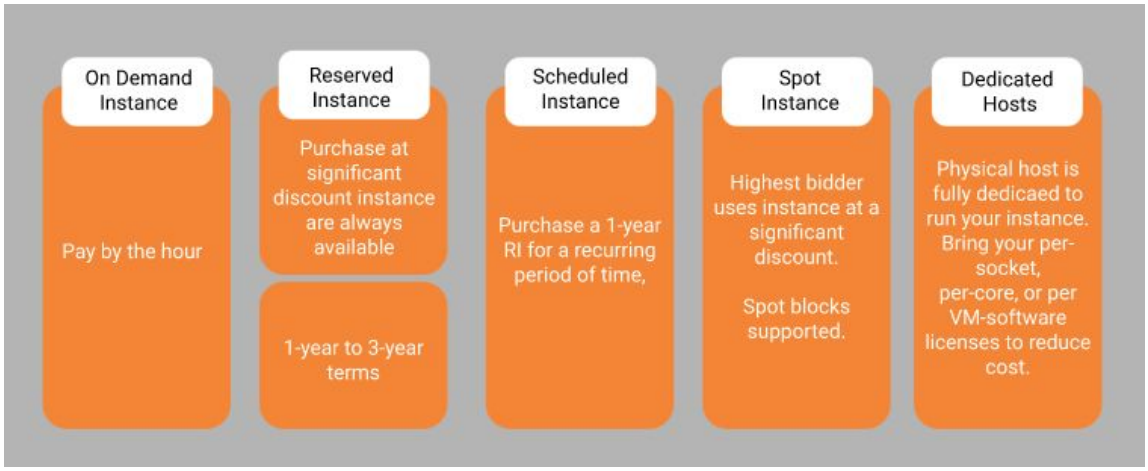


In the case of Windows, we use the Powershell tags. The opening and closing Powershell tags and we embed a partial script between these two tags. To retrieve instance user data, you can connect to the following URL: http://169.254.169.254/latest/user-data

On a Linux instance, you can simply use the curl or GET command.
$ curl http://169.254.169.254/latest/user-data/
$ GET http://169.254.169.254/latest/user-data/

7.7 Amazon EC2 Purchase Options

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:



7.7.1 On-Demand Instances

They are pretty eligible. It has the lowest upfront costs and offers the most flexibility. You pay for an hour at a time with no upfront commitment or long term contracts. This is great for applications with short term, spiky or unpredictable workloads.

For example: If I run an instance for 20 minutes, I get charged for 1 hour. If I run an instance for 1 hour 15 minutes, I get charged for 2 hours. There are no long term contracts and these are great for applications that I want for short term purposes or for unpredictable, spiky workloads. In a nutshell:

Pay, by the second, for the instances that you launch.

7.7.2 Reserved Instances

Amazon's EC2 Reserved Instances pricing, allows you to reserve computing capacity for 1 year to 3 years term at a significantly discounted hourly rate. Reserved instances are building discounts and a capacity reservation that is applied to instances to lower the hourly running cost. This instance is not a physical instance. The discounted usage price is fixed as long as you own the reserve instance, allowing you to predict compute costs over the term of the reservation. If you are expecting consistent and heavy usage, Reserved Instances can provide substantial savings over owning your own hardware and running only on On-Demand Instances.

For example: If I have a domain control that runs 24/7 and 365 days, I would probably go ahead and purchase a reservation of that particular instance time that is running my domain controller.

7.7.3 Scheduled Instances

Scheduled reserve incenses enable you to purchase capacity reservations that are recurring on a daily, weekly or monthly basis with specified duration for one year term.

You reserve the capacity in advance so that you know it is available to you when you actually need them. You pay for the time, instances are scheduled even if you do not use them. Scheduled Instances are a great choice for workloads that do not run continuously but do run on a regular schedule and take finite time to complete.

For example: Let's say you have workloads to run during business hours from 9 to 5. Such workloads can be a great candidate for Schedule Instances.

7.7.4 Spot Instances

Spot instances enable you to build on unused EC2 instances which can lower your Amazon EC2 cost significantly.

The hourly price for a Spot Instance is set by Amazon EC2 and fluctuates depending upon the supply of a demand for Spot Instances. Your Spot Instance runs whenever your bid exceeds the current market price. Spot Instances are a great cost effective solution if you are flexible about when your applications run and if the applications can be interrupted.

Amazon EC2 does not terminate Spot Instances with a specific duration known as Spot blocks when the spot price changes. This makes them ideal for jobs that have a finite amount of time to complete such as batch processing, encoding, rendering, modeling and continuous integration.

7.7.5 Dedicated Hosts

Amazon EC2 dedicated host is a physical server with EC2 instance capacity, completely dedicated for your use. Dedicated hosts allow you to use existing per-socket, per-core of VM software licenses including Microsoft Windows Server, Sequel Server,Soozie, Linux Enterprise Server and so on.

Dedicated host and dedicated Instances can be both used to launch Amazon EC2 Instances on physical servers that are dedicated for your use. There are no performance security or physical differences between dedicated instances and instances on a dedicated host.

However, Dedicated Hosts give you additional visibility and control on how your Instances are placed on a physical server.

8. Amazon Storage Services: Amazon S3

Amazon S3 is designed to make web scale computing easier for developers. It provides a simple web service interface that can be used to store and retrieve any amount of data anytime from anywhere on the web. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network

of websites.

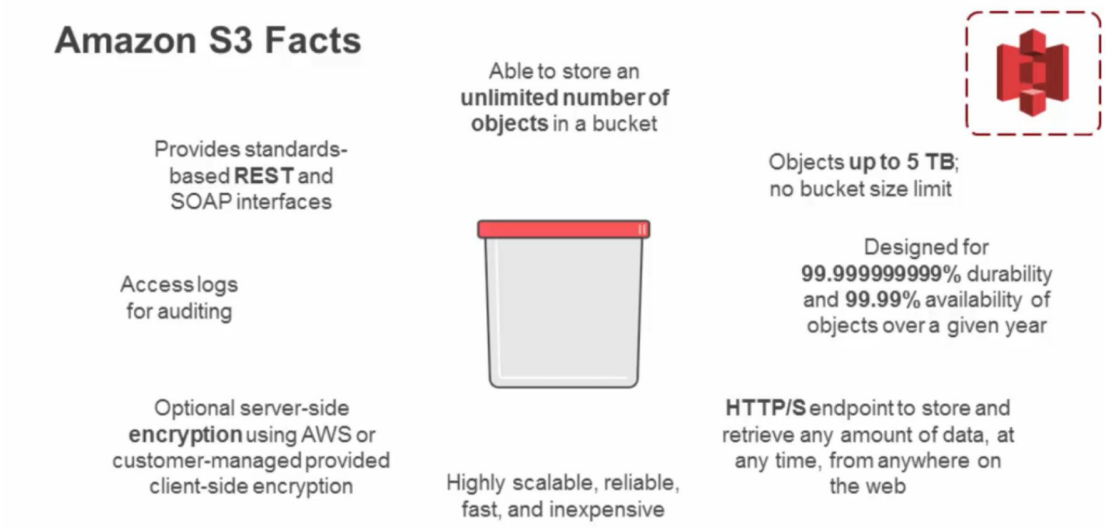
Key Features:

- Storage for the Internet
- Natively online, HTTP access
- Storage that allows you to store and retrieve any amount of data, any time, from anywhere on the web
- Highly scalable, reliable, fast and durable Amazon S3

8.1 Amazon S3 Facts

- S3 can store an unlimited number of objects in a bucket
- S3 objects can be up to 5 TB (terabytes); no bucket size limit
- It is designed for 99.999999999% (11 nines) durability and 99.99% (4 nines) availability of objects over a given year
- It can use HTTP/S endpoints to store and retrieve any amount of data, at any time, from anywhere on the web

- It is highly scalable, reliable, fast, and inexpensive • Can use optional server-side encryption using AWS or customer-managed provided client-side encryption • Auditing is provided by access logs
  S3 provides standards-based REST and SOAP interfacesThere is a 100 bucket limit per account. You can store an unlimited number of objects in a bucket.



8.1.1 Difference between Availability and Durability

We can understand this with an example. Assume my friend travels from Sydney to Okland. It is a three and a half hour journey and after five hours, I give this friend a call on his mobile phone. His phone is switched off. Now the object might be there which means my friend is still there but he may not have turned on his mobile phone so the object is there but I am not able to access that object.

Basically what happens with S3 is, when you upload an object, we replicate this object to multiple facilities within that region which means this object is there and it could replicate it to multiple facilities. It is highly durable.

However, since S3 is accessed over the rest APIs, there might be a case where the endpoint is down ie.. the API endpoint, which is taking a request

that may be down. In such scenarios, we say that we are unable to access that object. So the availability of that object is impacted.

And that is the key difference between durability and availability.

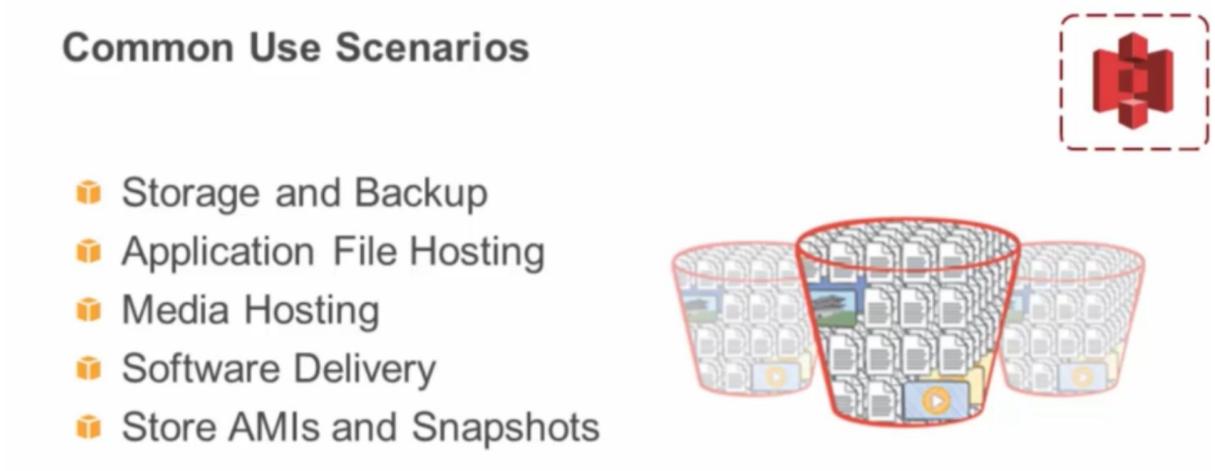
- You can use http and https endpoints to store and retrieve any amount of data, at any time from anywhere on the web.
- Most importantly, Amazon S3 is highly scalable, reliable, fast and inexpensive.
- You can also use server-side encryption using AWS or customer-managed encryption options for Amazon S3.

Now you can go ahead and enable server-side encryption on a S3 which has two options. You can use the S3 encryption or the Amazon Key Management Service (KMS) encryptions.
You can also use your own keys to encrypt data which is stored into S3. These are called Customer-Managed or Customer Provided keys.

The Common Use Case scenarios for Amazon S3 includes:

1. Storage and backups
2. Application File Hosting
3. Media Hosting
4. Software Delivery
5. Store AMIs and Snapshots

Storing your application files, media hosting such as mp3s & mp4s, software deliveries, storing AMIs and snapshots. But let me also talk about some of the advanced use case scenarios.



For example: You can use Amazon Dev pay with Amazon S3. Amazon Dev pay enables you to charge customers for using your Amazon S3 products through Amazon's authentication and billing infrastructure. You can charge any amount for your product including the usage charges such as: storage, transactions and bandwidth, monthly fixed charges and a 1 time charge.

You can direct your clients to torrent accessible objects by giving them a torrent file directly or publishing a link to the bittorrent URL of your object. You can host a static website on Amazon S3 by configuring a bucket for web site hosting and then uploading your website content to the bucket.

Amazon S3 pricing is based on the capacity and bandwidth actually used. Since Amazon S3 is an Internet scaled service that runs natively across the entire region, it can handle significant requests throughput and bandwidth output.

All bandwidth into Amazon S3 is free but AWS charges a rate on bandwidth out. Most importantly, since Amazon S3 can handle any amount of data, it is important for you to know that you only pay for the amount of space you use. Prices are based on a prorated GB per month. There is a price calculator online that can be used as a reference.

To get the most out of Amazon S3, you need to understand a few simple concepts. The core Amazon S3 concepts are:

- Amazon S3 stores data as objects within buckets
- An object is composed of a file and optionally any metadata that describes that file
- You can have up to 100 buckets in each account
- You can control access to the bucket and its objects Amazon S3 Bucket with Objects Bucket Object

Note: To store an object in Amazon S3, you upload the file you want to store into the bucket. When you upload a file, you can set permissions on the object as well as any metadata.

8.2 Amazon S3 Buckets:

Buckets are logical containers for your objects. You can have one or more buckets in your account. For each object you can control access. In other words, you can create delete list objects in the bucket.
You can also view access logs for the bucket and its objects and choose a geographical region where Amazon S3 will store the bucket and its contents.

A bucket is a logical container for objects stored in Amazon S3. Every object is contained in a bucket. Buckets serve several purposes. Their key features are:

- Organize the Amazon S3 namespace at the highest level.
- Identify the account responsible for storage and data transfer charges.
- Play a role in access control.
- Serve as the unit of aggregation for usage reporting.

- Have globally unique bucket names, regardless of the AWS region in which they were created.

You specify the name at the time when you create the bucket. This means that if I have already created a bucket with the name of Amazon, you cannot create that same bucket since they have to be globally unique.

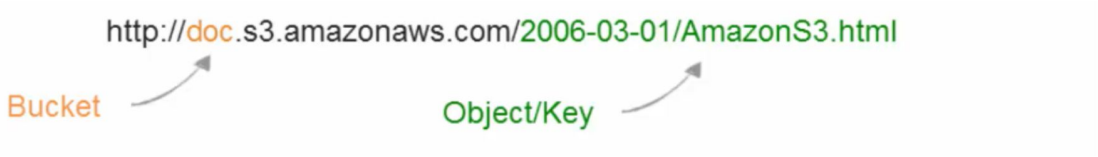
8.2.1 Object Keys

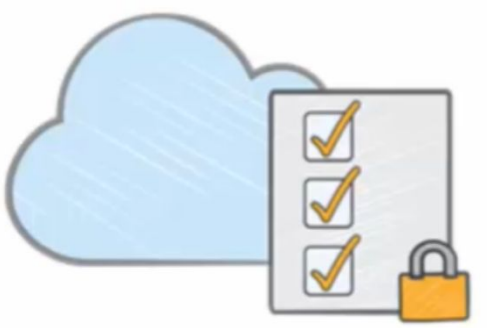
An object key is the unique identifier for an object in a bucket. Because the combination of a bucket key and version ID uniquely identify each object, Amazon S3 can be thought of as a basic data map between a bucket + key + version and the object itself.

eg: **http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.html Bucket Object/Key**

Every object in Amazon S3 can be uniquely addressed to the combination of the web service endpoint, bucket name, the key and a version (optionally if you have enabled the versioning on the bucket).

For example: In this URL, http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.html, “doc” is the name of the Bucket, 2006-03-01/AmazonS3.html is the Object Key.





Amazon S3 Security

You can control access to buckets and objects with:

- Access Control Lists (ACLs)
- Bucket policies
- Identity and Access Management (IAM) policies
- You can upload or download data to Amazon S3 via SSL encrypted endpoints.
- You can encrypt data using AWS SDKs.

With IAM policies, you can only grant users within your own AWS account permission to access your Amazon resources With Access Control Lists (ACLs) you can only grant other AWS accounts, not specific users to access your Amazon S3 resources. Bucket policies in Amazon S3 can be used to add or deny permissions across some or all of the objects within a single bucket.

Policies can be attached to users, groups or Amazon S3 buckets enabling centralized management of permissions. With bucket policies, you can grant users within your AWS account or any other AWS account, access to Amazon S3 resources.

8.2.1 Data Transfer Aspect

For maximum security, you can securely upload or download data to Amazon S3 via the SSL encrypted endpoints. The encrypted endpoints are accessible from both - the Internet and from within the Amazon EC2, so that the data is transferred securely both within AWS and to and from sources outside of AWS.

S3 provides multiple options for pertinent data addressed. Customers who prefer to manage their own encryption keys can use glide encryption libraries like the Amazon S3 encryption client to encrypt data before uploading to Amazon S3.

Alternatively, you can also use the S3 server-side encryption if you prefer to have Amazon S3 to manage encryption keys for you. With Amazon S3 server-side encryption, you can encrypt data on upload simply by enabling. With Amazon S3 server-side encryption, you can encrypt data on upload simply by adding an additional request header when writing the armchair. Decryption happens automatically when the data is retrieved.Amazon S3 server-side encryption uses one of the strongest block ciphers available.

The 256 bit Advanced Encryption Standard ie..the AES 256. With Amazon S3 server-side encryption, every protected object is encrypted with a unique and commission key. This object itself is then encrypted with a regularly rotated Master key. Amazon S3 server-side encryption provides additional security by storing the encrypted data and encryption keys in different hosts.
The server-side encryption also makes it possible for you to enforce encryption requirements.

For example: You can create an apply bucket policy that requires only encrypted data which can be uploaded to your buckets.

Now instead of using the Amazon S3 server-side encryption, you also have the option of encrypting your data before sending it to Amazon S3. You can

build your own library that encrypts Object data on the client's side before uploading to Amazon S3.
Optionally, you can use an Amazon AWS SDK or a software development kit such as Darton SDK or the Java SDK to automatically encrypt your data before uploading it to Amazon S3.

8.3 Amazon S3 Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, restore every version of every object stored in Amazon S3 bucket. With versioning, you can easily recover from both un-attended user reactions and application failures. Here are their key feature:

- Protects from accidental overwrites and deletes with no performance penalty.

- Generates a new version with every upload.
- Allows easy retrieval of deleted objects or roll back to previous versions.

- Three states of an Amazon S3 bucket:
    - Un-versioned (default)
    - Versioning-enabled
    - Versioning-suspended Versioning Enabled Key: photo.gif ID: 121212 Key: photo.gif ID: 111111

For example: In one bucket, you can have two objects with the same key but with different version IDs such as for the photo.gif which is version 111111 and photo.gif which is of version 121212. Once your version enables a bucket, it can never return to an unversioned state. You can however suspend the versioning on that bucket.

8.3 Amazon S3 Storage Classes

Each object in S3 has a storage class associated with it. S3 standard is ideal for performance sensitive use cases and frequently used data. Standard is the default storage class in S3.
S3 infrequent access is optimized for long lived, less frequently accessed data sets as backups and older data that can be accessed less but still require a high performance. S3 Infrequent Access (IA) is optimized for long lived and less frequently accessed data such as backups and all the data that are accessed less but still require high performance.

- Storage Class Durability Availability Other Considerations Amazon S3 Standard 99.999999999% 99.99% Amazon S3 Standard - Infrequent Access (IA) 99.999999999% 99.9%
- Retrieval fee associated with objects
- Most suitable for infrequently accessed data Glacier 99.999999999% 99.99% (once restored)
- Not available for real-time access • Must restore objects before you can access them
- Restoring objects can take 3-5 hours

Storage Class	Durability	Availability	Other Considerations
Amazon S3 Standard	99.999999999%	99.99%	None
Amazon S3 Standard - Infrequent Access (IA)	99.999999999%	99.99%	• Retrieval fee associated with objects • Most suitable for infrequently accessed data
Glacier	99.999999999%	99.99% (after you restore objects)	• Not available for real-time access • Must restore objects before you can access them

Amazon Glacier

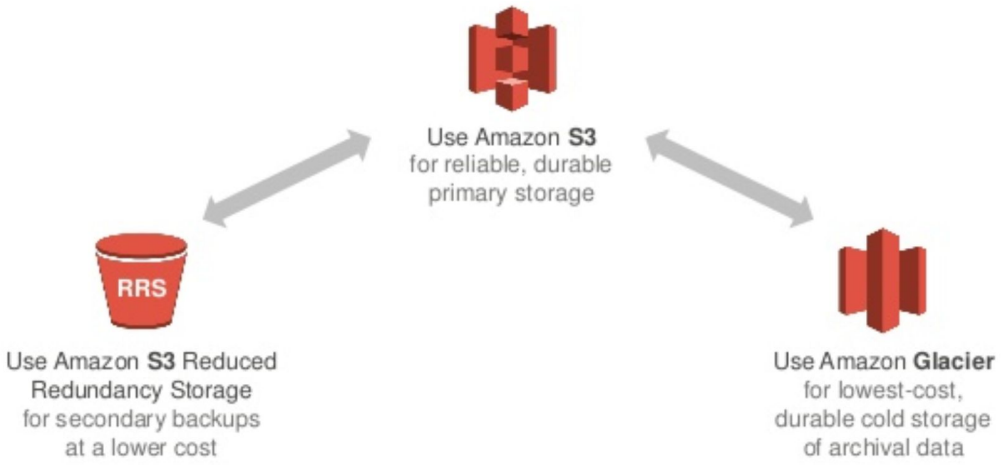
Glacier is suitable for archiving data where access is infrequent and retrieval time of several hours is acceptable. Archived objects are not available for real time access. They must be restored before they can be accessed. The Glacier storage class is very low cost.

The key features to remember are:

- Long term low-cost archiving service
- Optimal for infrequently accessed data
- Designed for 99.999999999% durability
- Three to five hours' retrieval time
- Less than $0.01 per GB/month (depending on region)

8.4 Reduced Redundancy Storage

The S3 Reduced Redundancy Storage (RRS)is designed for non-critical, reproducible data sets that are stored at a much cheaper rate as compared to the other storage classes.



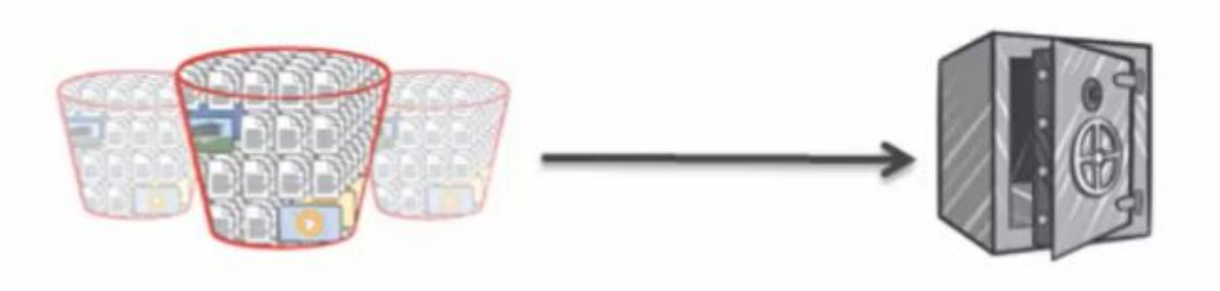
Example: We might consider using the Reduced Redundancy Storage (RRS). Assume, you have a website which allows users to upload files and you can convert those files (images) into grayscale images. You have the source files/original file and your pick and then the application converts this into a grayscale image. Which means even if you lose the output which is a grayscale image, you still have the source file from which you can regenerate the grayscale image.

Since this is a reproducible object, you can store it on the Reduced Redundancy Storage if required to save cost because even if I lose the object I can still go ahead and run the application, get my original file and can put it into a grayscale image.

8.5 Lifecycle Management

Lifecycle management defines how Amazon S3 manages objects during the lifetime. Some objects that you store in an Amazon S3 bucket might have a well defined lifecycle. If you are uploading periodic logs to your bucket,

your application might need these logs for a week or a month after creation. After that you might want to delete them.



Some documents are frequently accessed for a limited period of time. After that, you might not need real time access to these objects but your organization might require you to archive them for a longer period and then optionally need them.
Files such as:

- Log files
- Archive Documents
- Digital Media Archives
- Financial andHealthcare records
- Raw genomics sequence data
- Long-term database backups
- Data that must be retained for regulatory compliance

They are some kind of objects that you might want to upload to S3 primarily for archival purposes. When you configure a lifecycle rule, you specify the storage class you want to transition the object to and a number of days and the object creation to transition it.

You can transition objects to the standard infrequent access storage class, archive them to Amazon Glacier or have them permanently deleted. Standard infrequent access is useful for data such as backups and other older infrequently accessed data where high performance continues to be a requirement. It is more suitable for objects gridding than 128 KB that you

want to keep for at least 30 days. There is a retrieval fee associated with

standard infrequent access.

As discussed earlier Amazon Glacier is designed for long term infrequently accessed data.

For example: If you have objects that need to be stored for regulatory compliance reasons which you cannot delete them Glacier is a great choice.

But keep in mind that when you request an object from Glacier, you need to create a job which takes anywhere between three to five hours for the job to be available, completed and the object to be available for download.

It is very very cheap and depending upon the region, you can be charged as low as 1 cent or 1.2 cents per GB. It offers the same 99.999999999% of durability that S3 Standard storage offers.

8.8 Amazon S3 Pricing

Amazon S3 pricing is based on the capacity and bandwidth actually used. Since Amazon S3 is an Internet scaled service that runs natively across the entire region, it can handle significant requests throughput and bandwidth output. Here are the key S3 pricing features:

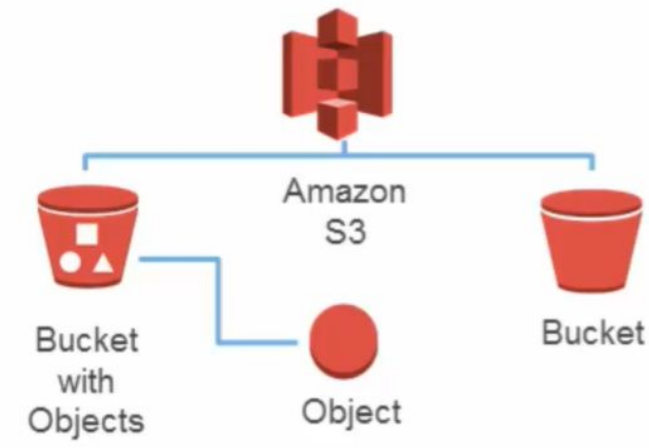
- Pay only for what you use
- No minimum fee
- Prices based on location of your Amazon S3 bucket

- Estimate monthly bill using the AWS Simple Monthly Calculator
- Pricing is available as:
    - Storage Pricing
    - Request Pricing
    - Data Transfer Pricing: data transferred out of Amazon S3

All bandwidth into Amazon S3 is free but AWS charges a rate on bandwidth out. Most importantly, since Amazon S3 can handle any amount of data, it is important for you to know that you only pay for the amount of space you use. Prices are based on a prorated GB per month. There is a pricing calculator online that can be used as a reference.

To get the most out of Amazon S3, you need to understand a few simple concepts.

First, Amazon S3 stores data as objects within the bucket. An object is composed of a file and any metadata that describes that file. To store an object in Amazon S3, you upload the file you want to store into the bucket. When you upload a file, you can set permissions on the object as well as any metadata.



8.7 Case Study of SoundCloud

Here is a simple Case Study of SoundCloud. SoundCloud operates worldwide and it enables users to upload 12 hours of audio material to its platform every minute. Each audio file must be transcoding and stored in multiple formats.

Logs and analyzes billions of events.

They have done this on AWS and they use a mix of S3 and Glacier. The audio files are placed in S3. They are distributed to S3 via the SoundCloud website and then copied into a Glacier which are infrequently accessed.

So any audio file that is not frequently accessed, gets transitioned into Glacier. SoundCloud currently holds (at the time of writing) 2.5 PB (petabytes) of data on Glacier.

9. Amazon Elastic Block Store (EBS)

Amazon Elastic Block Store is also known as Amazon EBS are persistent block level storage volumes for use with Amazon EC2 instances. It offers consistent and low latency performance. Amazon EBS is exceptionally suited for applications that require a database, file system or access to roll

block level storage.

Amazon EBS snapshots are durable and are automatically replicated within their Availability Zone. Snapshots are stored durably in Amazon S3. Amazon EBS provides block level storage volumes for use with Amazon EC2 instances.

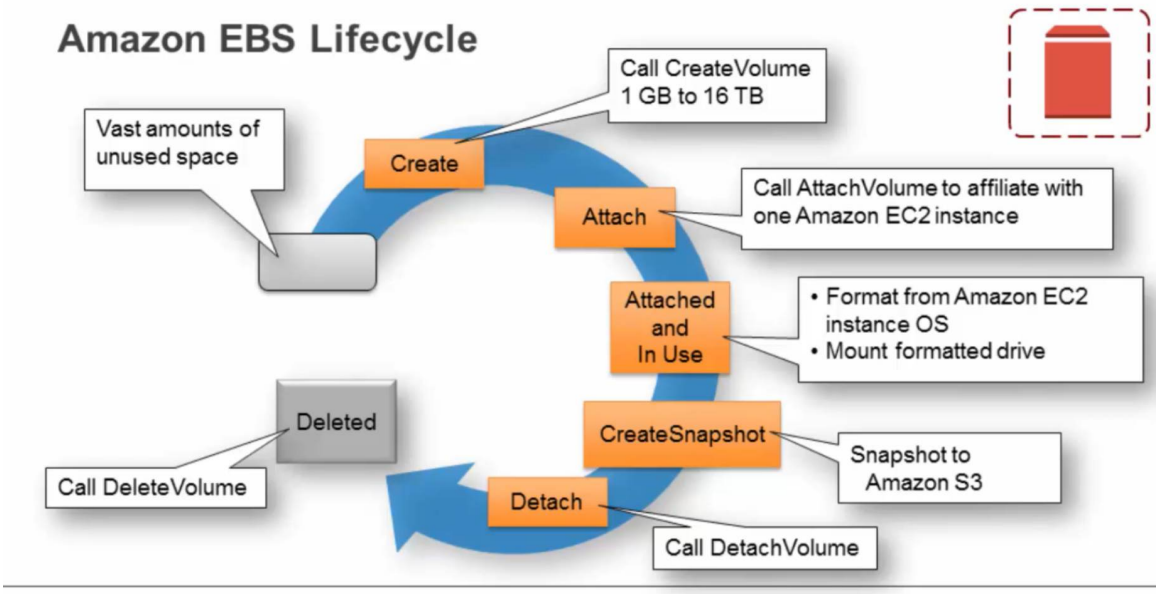
Amazon EBS volumes are highly available and reliable storage volumes that can be attached to any running instance in the same Availability Zone. The Amazon EBS volumes attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance. When the volumes are not attached to an EC2 instance, you only pay for the cost of the storage.

The 3 key points to remember for Amazon EBS:

- Persistent block level storage volumes offer consistent and low-latency performance.
- Stored data is automatically replicated within its Availability Zone.
- Snapshots are stored durably in Amazon S3.

9.1 Amazon EBS Lifecycle

The typical lifecycle of an EBS volume starts with the creation of the volume which you can then attach to an EC2 instance.

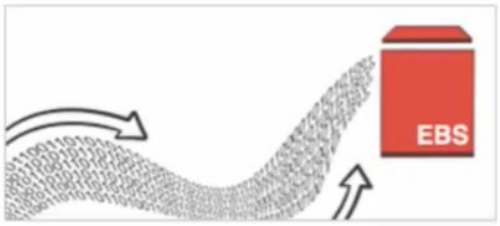


Once it is attached to an EC2 instance, it is then in use byEC2 instance. You can create snapshots which are nothing but a point in time photograph of the EBS volume and then you can detach the EBS volume and maybe attach it to a different EC2 instance in the same Availability Zone. And finally you can always go ahead and delete the EBS volume.

9.1.1 Amazon EBS Facts

Now let's take a look at some of the EBS facts.

- You can create :
    - EBS Magnetic volumes from 1 GB to 1 TB in size.
    - EBS General Purpose (SSD) volumes up to 16 TB in size.

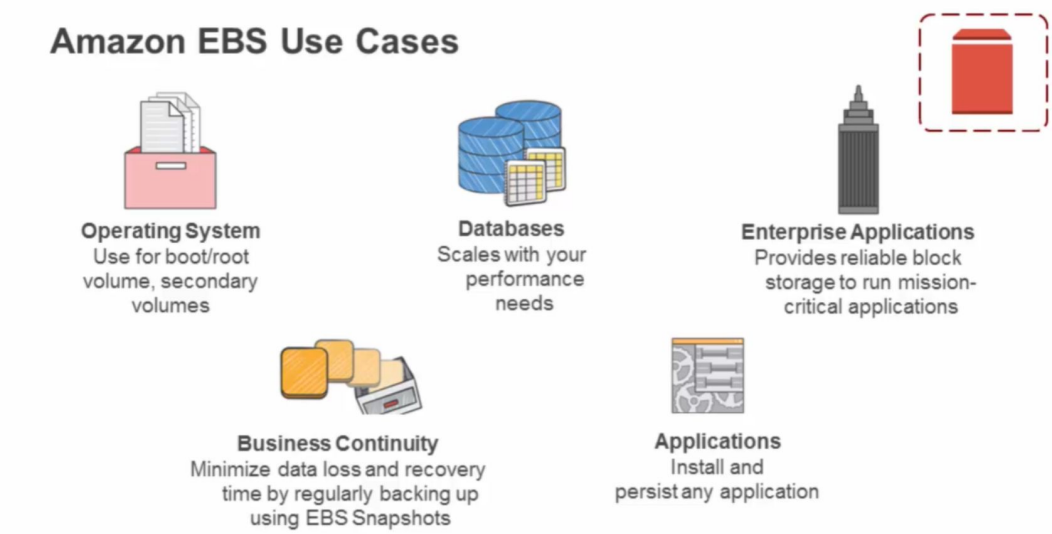


- You can also use encrypted EBS volumes to meet a wide range of data at-rest encryption requirements for regulated/audited data and applications.
- You can create point-in-time snapshots of EBS volumes, which are persisted to Amazon S3.

Amazon EBS is recommended when data changes frequently and requires a long term persistence. EBS volumes are particularly well suited for use as primary storage for file systems, databases or for any applications that require fine granular updates. An access to the raw and formatted block level storage, EBS is particularly helpful for database style applications that frequently encounter many random reads and rise across the datasets.

9.1.2 Amazon EBS Use Cases

The EBS dataset is simply a virtual hard drive. So, a great use case for EBS is when you want the hard drive to persist past the life of the EC2 instance. Before EBS existed as a service, AWS only used physical locally attached drives called ‘ephemeral storage’ or instance store volumes.
The problem with that was that you couldn't stop and start an EC2 instance without losing the data on that ephemeral store or the instance store because of the temporary nature of the local storage. That's why we created the EBS service to decouple the lifecycle of data persistence from the life cycle of an EC2 instance.



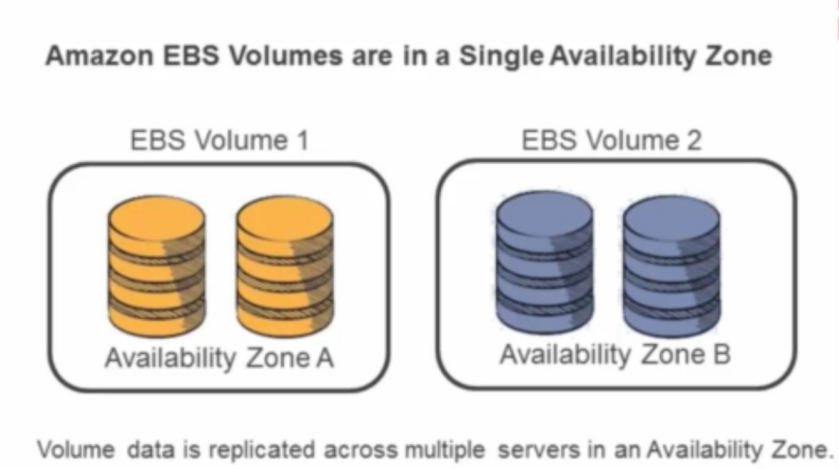
EBS volumes are ideal for root volumes. You need to store and have lock level access to your operating system, database storage, and data sets that are smaller than 1 TB. Given a simple snapshot mechanism, and isn't EBS as is a great use case for simplifying distributed backups as well.

Just keep in mind:

- **OS:** Use for boot/root volume, secondary volumes
- **Databases:** Scales with your performance needs
- **Enterprise applications:** Provides reliable block storage to run mission-critical applications
- **Business continuity:** Minimize data loss and recovery time by regularly backing up using EBS Snapshots
- **Applications:** Install and persist any application

9.2 EBS Pricing and Volumes

EBS pricing is based on allocated storage. Whether you use it or not. This is very unlike Amazon S3 who's pricing is based on the space actually used. Prizes may vary depending upon the region or for IOPS that you provision.



Amazon EBC volumes are designed to be highly available and reliable. EBS volume data is replicated across multiple servers in an Availability Zone to prevent loss of data from failure of any single competent.

The durability of the volume depends upon both:

1. The size of the volume and
2. The percentage of data that has changed since the last snapshot.

EBS volumes are designed for an Annual Failure Rate (AFR) of between 0.1% to 0.2%. Their failure refers to the complete operation loss of the volume depending upon the size and the performance of the volume.

This is compared with commodity hard disks that will typically fail with an Annual Failure Rate (AFR) of our own 4% making EBS volumes 10 times more reliable than the typical commodity hard drives. Since EBS servers are replicated within a single Availability Zone, mirroring data across multiple EBS volumes in the same Availability Zone will not significantly improve the volume durability.

For those interested in even more durability with Amazon EBS, you can create a point in time consistent snapshot of your volumes that are then stored in Amazon S3 and are automatically replicated across multiple Availability Zones. Taking frequent snapshots of your volume is a convenient and a cost effective way to increase the long term durability of the data.

In the unlikely event that you or Amazon EBS volume does feel, all snapshots of that volume will remain intact and will allow you to recreate the volume from the last snapshot point.

9.3 Amazon EBS and S3

The table below demonstrates the significant differences between Amazon S3 and Amazon EBS. Amazon EBS volumes are network attached hard drives that can be written to or read from at a block level.

Amazon EBS and Amazon S3

	Amazon EBS	Amazon S3
Paradigm	Block storage with file system	Object store
Performance	Very fast	Fast
Redundancy	Across multiple servers in an Availability Zone	Across multiple facilities in a Region
Security	EBS Encryption – Data volumes and Snapshots	Encryption
Access from the Internet?	No (1)	Yes (2)
Typical use case	It is a disk drive	Online storage

(1) Accessible from the Internet if mounted to server and set up as FTP, etc.
(2) Only with proper credentials, unless ACLs are world-readable

S3 is an object level storage medium. This means that you must write whole objects at a time. If you change one small part of the file. For example: A character in a Word document, you must still re-write the entire file in order to commit the change to Amazon S3.

This can be very time consuming if you frequently write to the same object. S3 is optimized for writing once, reading many use cases. The other major difference is cost. With Amazon S3, you pay for what you use and with Amazon EBS, you always pay for what you provision.

9.4 Amazon EC2 Instance Store and Reboot

An Instance Store provides temporary block level stories for their instance. This storage is located on disks that are physically attached to the host computer. Instance Store is ideal for temporary storage or information that changes frequently.

Such as buffers, caches, scratch data and other temporary content or data that is replicated across a fleet of instances such as a load balanced pool of observers.

Some key points to remember:

- **Local,** complimentary direct attached block storage resource.
- **Availability**, number of disks, and size is based on EC2 instance type.
- **Storage optimized instance** for up to 365,000 Read IOPS and 315,000 First Write IOPS.
- SSD or magnetic.
- **No** persistence.
- All data is **automatically deleted** when an EC2 instance stops, fails or is terminated.

The table shows the differences between rebooting, stopping and terminating your instance.

Reboot vs. Stop vs. Terminate

Characteristic	Reboot	Stop/Start (EBS-backed instances only)	Terminate
Host computer	The instance stays on the same host computer.	The instance runs on a new host computer.	N/A
Private and public IP addresses	Stay the same.	Instance keeps its private IP address and gets a new public IP address.	N/A
Elastic IP addresses (EIP)	EIP remains associated with the instance.	EIP remains associated with the instance.	The EIP is disassociated from the instance.
Instance store volumes	The data is preserved.	The data is erased.	The data is erased.
EBS volume	The volume is preserved.	The volume is preserved.	The volume is deleted by default.
Billing	Instance billing hour doesn't change.	You stop incurring charges as soon as state is changed to stopping.	You stop incurring charges as soon as state is changed to shutting-down.

One of the key things to understand is when you reboot an instance the underlying host has not changed ie.. you are still on the same host. This means you will not lose your public IP address, you will not lose access to the instance store data.

But when you stop and start, you move from one host to another host and thereby you will lose access to the instance store volumes as well as you lose the Instance's public IP address and you get a new one.

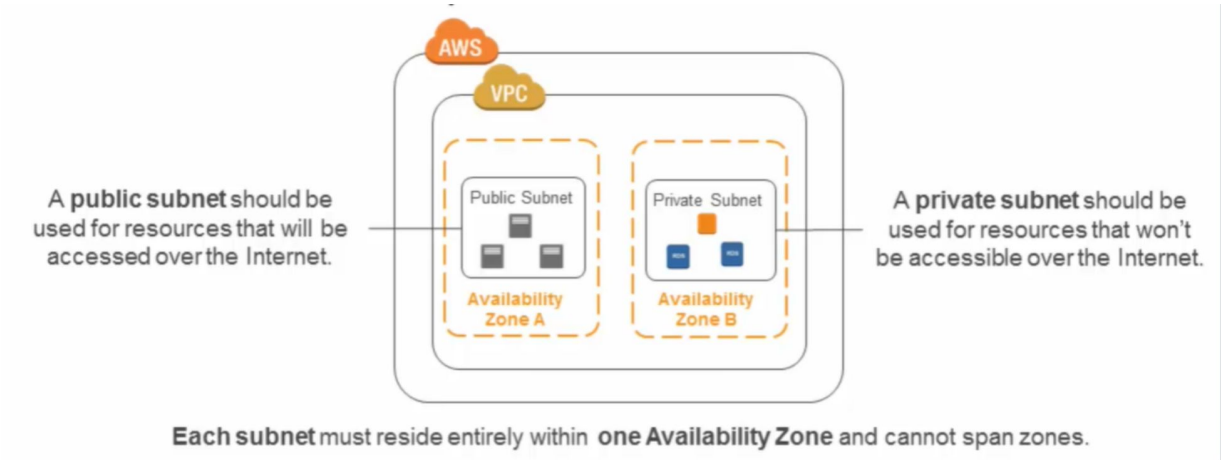
10. Networking

Observe the final section of the infrastructure module: Networking. Here, you are going to talk about Amazon VPC. We can talk about subnets, security, networking and virtual private networks.

With Amazon's Virtual Private Cloud (VPC), you can define it as a virtual network topology that closely resembles a traditional network that you might operate in your data center. You have complete control over your virtual networking environment and you can easily customize the network configuration of your Amazon VPC such as the :

- Selection of IP address ranges
- Creation of subnets
- Configuration of route tables and
- Network gateways

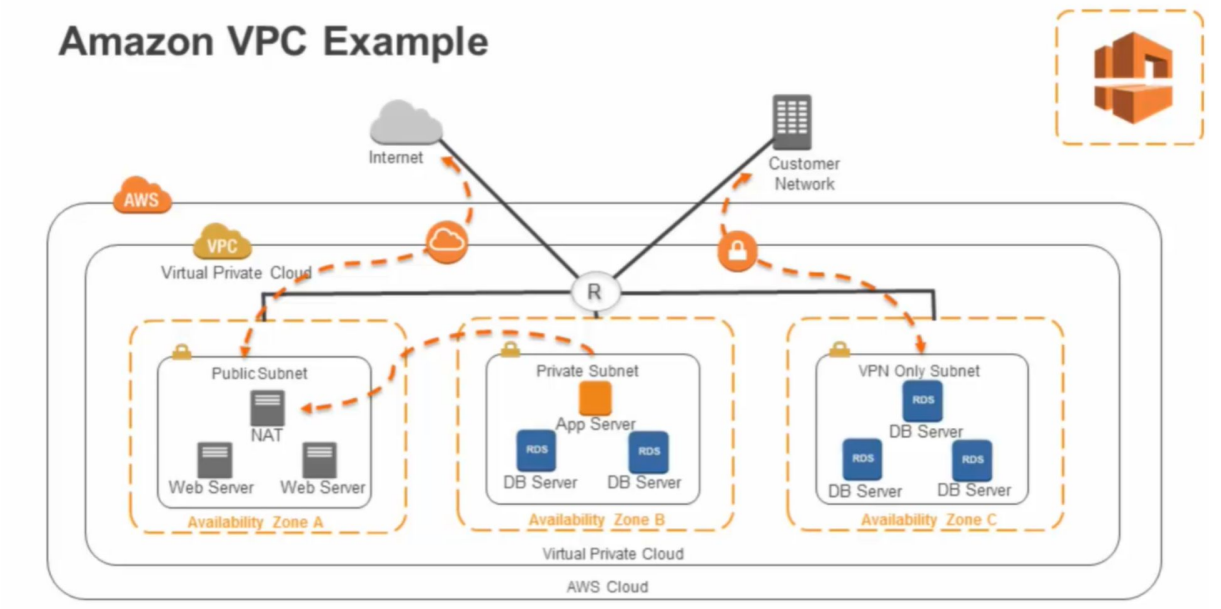
AWS assigns a unique ID to each subnet. A **subnet** can be defined as a range of IP addresses in your VPC. Regardless of the type of the subnet, whether they are public or private. The internal IP address range of the subnet is always private.



10.1 What is a public subnet ?

A public subnet is a subject that has a route to the internet gateway. That is for a web server accessible from the Internet. Whereas a private subnet has no route to the internet gateway.

For example: A banking device like a database server or a batch processing. They are not directly accessible from the Internet. These are those that can only be accessed via the VPC. Also, when you create a subnet you have to choose another Availability Zone a subnet cannot span multiple Availability Zones.



The Amazon Virtual Private Cloud (VPC) allows you to provision a logically isolated section of the tablas cloud where you can launch AWS resources in a virtual network that you define.

You have complete control over your virtual networking environment including:

- Selection of your IP address range
- Creation of subnets
- Configuration of route tables
- Network Access Control List which allows you to define which plot you want to block or allow and
- Network gateways.

You can easily customize the network configuration of your VPC. For example, you can create a public facing subnet for your web servers that require access to the Internet. And place your back at systems such as your database or application servers in a private facing subnet with no internet access.

You can leverage multiple layers of security including security groups which are accessed at the instance level typically at the EU level and the network access control lets you help you control access to the EC2 instances in each summit.

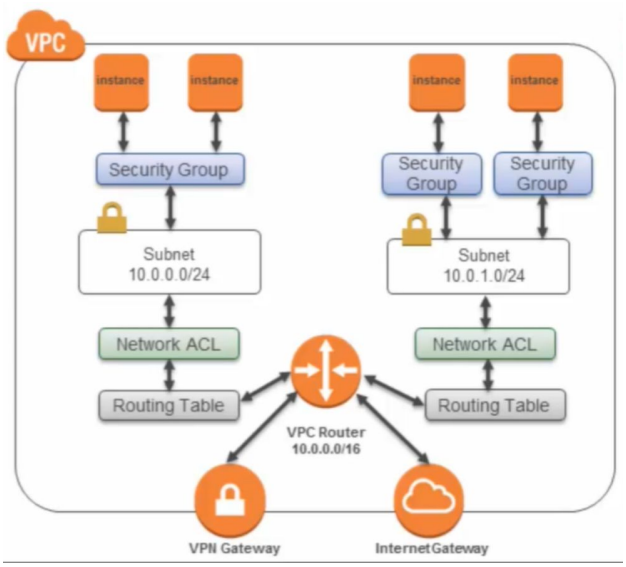
Additionally, you can create a hardware virtual private network connection between your corporate leader's anger and your VPC allowing you to leverage the cloud as an extension of your corporate data center.

10.2 Security

Amazon VPC provides three features that you can use to increase and monitor the security of your Virtual Private Cloud.

- Security groups act as a firewall for the associate and Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level.

- Network access control lists (ACLs) act as a firewall for associated sublets controlling from both inbound and outbound traffic at the subnet level.



You can connect your VPC to a remote network by using a VPN connection.

- You can use different connectivity options like the hardware VPN where you can create an IPsec Hardware VPN connection between your VPC and your remote network.

- You can also use the Amazon Direct Connect or AWS Direct Connect which provides a dedicated private connection from a remote network to your Virtual Private Cloud.

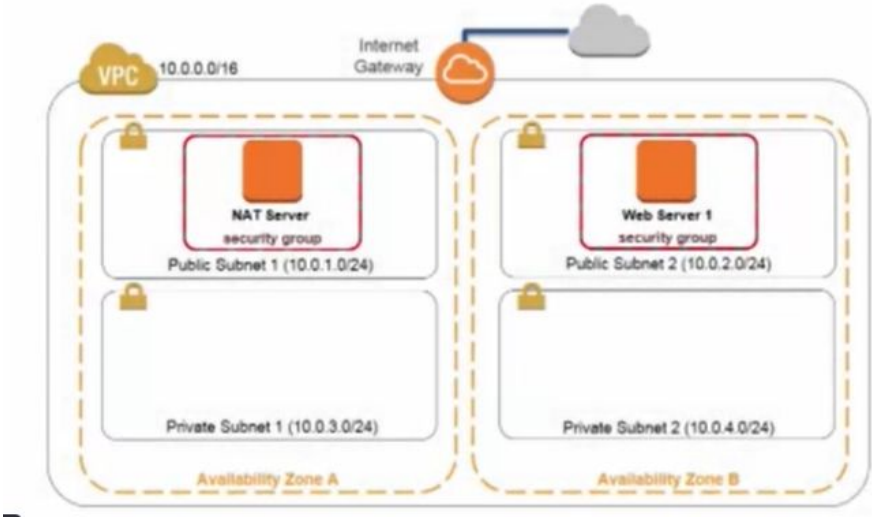
- You can use VPN CloudHub which is creating multiple AWS hardware VPN connections via your VPC to enable communications between remote networks.

- You can use Software VPN connections. When you place an EC2 instance inside of VPC, configure it to be a software VPN server, that is a VPN contemplating in the VPC and then you connect to a VPN concentrator in your remote corporate data center.

VPN Connectivity option	Description
AWS Hardware VPN	You can create an IPsec, hardware VPN connection between your VPC and your remote network.
AWS Direct Connect	AWS Direct Connect provides a dedicated private connection from a remote network to your VPC.
AWS VPN CloudHub	You can create multiple AWS hardware VPN connections via your VPC to enable communications between various remote networks.
Software VPN	You can create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a software VPN appliance.

11. Demo: Build Your VPC and Launch a Web Server

11.1 Lab #1-Build a web server



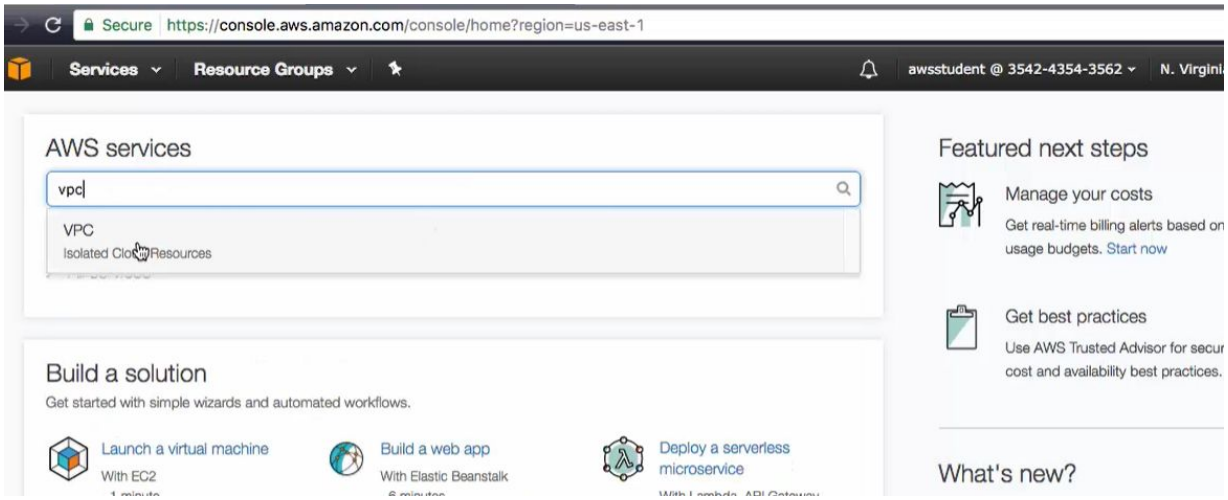
- We are going to use Amazon Virtual Private Cloud (VPC), to create our own VPC and add additional components to it to produce a customized network. By the end of this lab, we will have created a VPC with multiple subnets inside.

- Within the VPC, we will also create security groups to help control what is allowed to talk to our underlying EC2 instances.

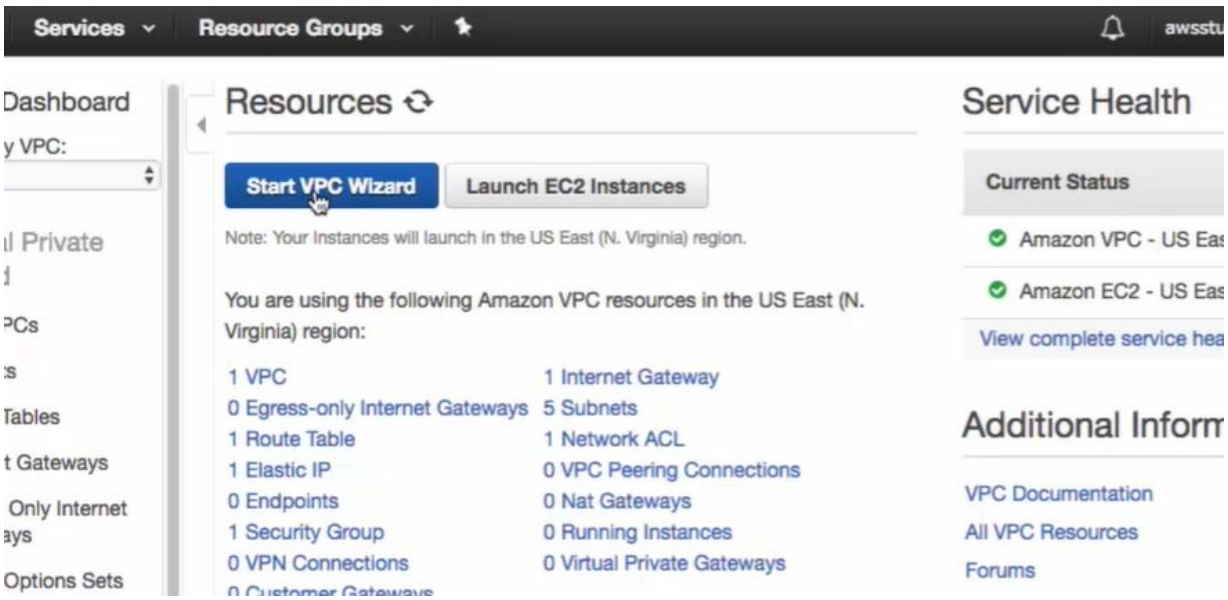
- We will be attaching an internet gateway.

Note: Internet gateway is a component of our VPC that allows communication between instances in the VPC and the Internet.

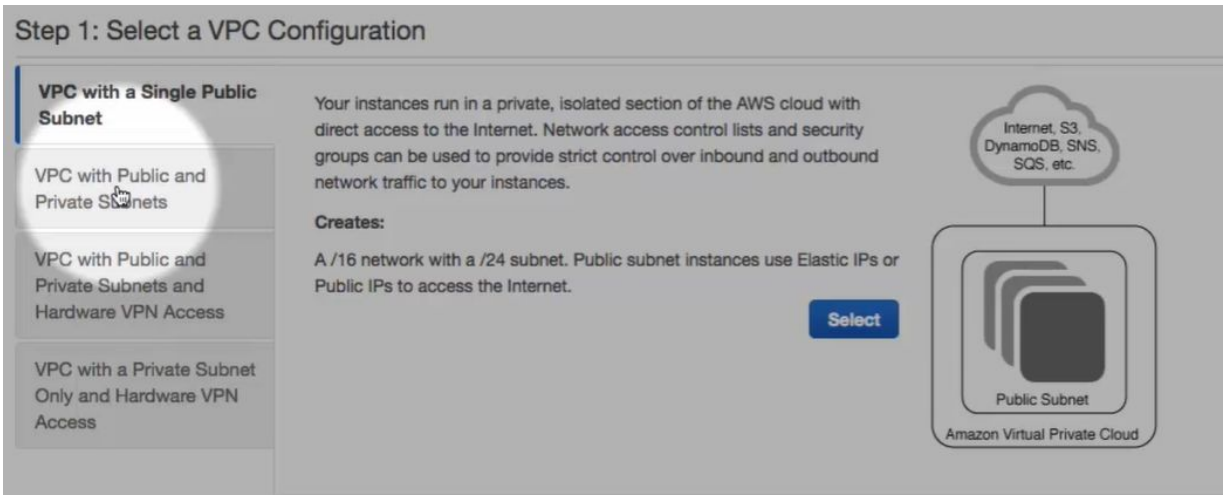
Inside the Console, we are going to want to move to the VPC service. The easiest way to find any service is by going to the search bar right here at the top of the page and typing in the name of the service we are looking for. We can then click and be redirected.



To begin, we are going to click on the Start VPC wizard.



Select the option “VPC with Public and Private Subnets” as shown in the figure.



For the site or block of our VPC, we are actually going to leave the IPv4 CIDR block at the default (10.0.0.0/16). We are going to give the VPC a name however. Call this “My Lab VPC”. This will allow us to find this VPC and pick it out when we move on later in the lab.

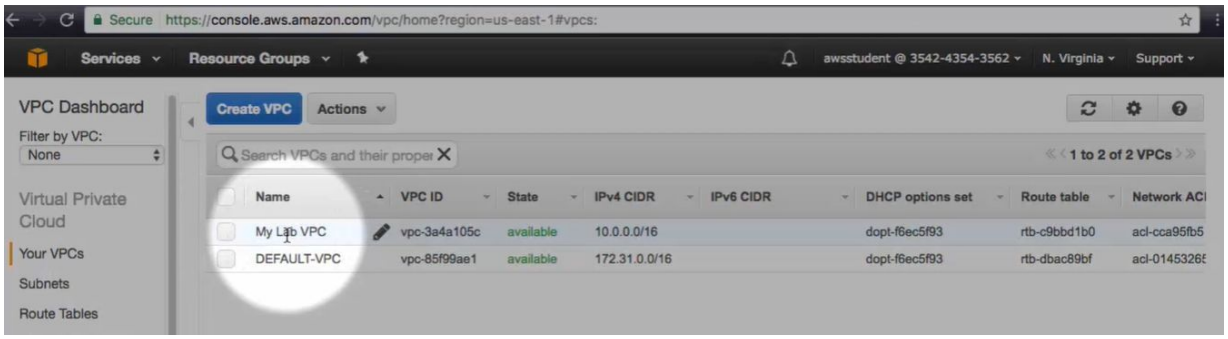
We are going to change the public subnet (IPv4 address) to 10.0.1.0/24 and we are going to change the IP address of our private subnet: 10.0.3.0/24

Now, we are not going to leave No Preference for Availability Zone which would essentially randomize which AZ these instances will be launched inside of. We are going to pick us-east-1a for both Public and Private

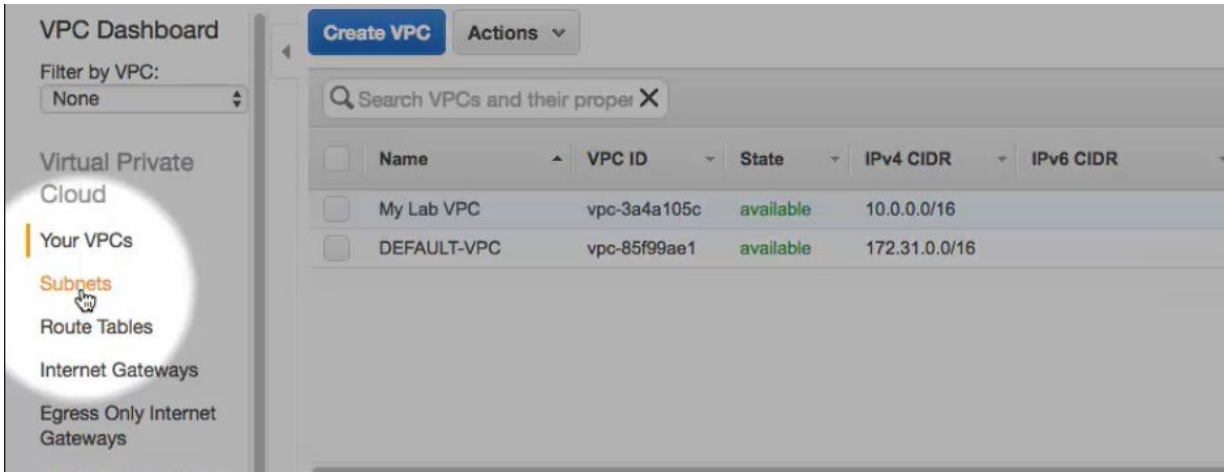
subnets. This will make sure any resources that are launched inside of these subnets will be launched in the same Availability Zone.

When we get down to the NAT (Network Address Translation), we are actually going to use a NAT Instance instead of the NAT Gateway. We can leave it at the default instance type and select the key pair that has already been created for us. We do not need to change anything else on the wizard. So we can hit Create VPC.

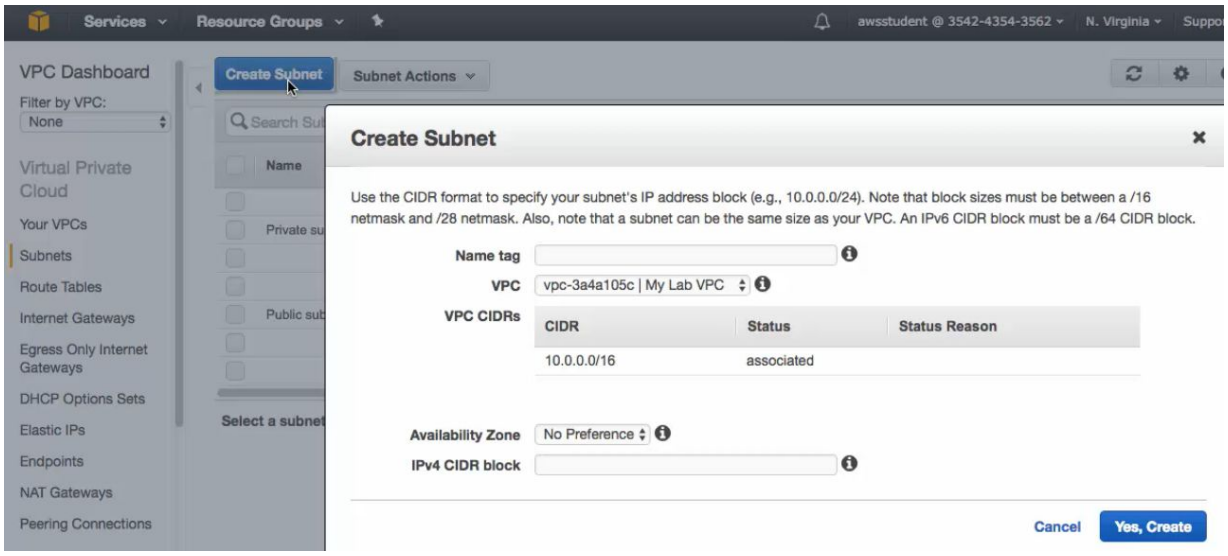
From here, the VPC wizard is going to go ahead and create the VPC, the subnets, the NAT instance and stand all of this up and configure it for us. Now that the VPC wizard has finished its job and created our VPC, you can now click on OK and see that “My Lab VPC” is listed in our listing of the VPCs.



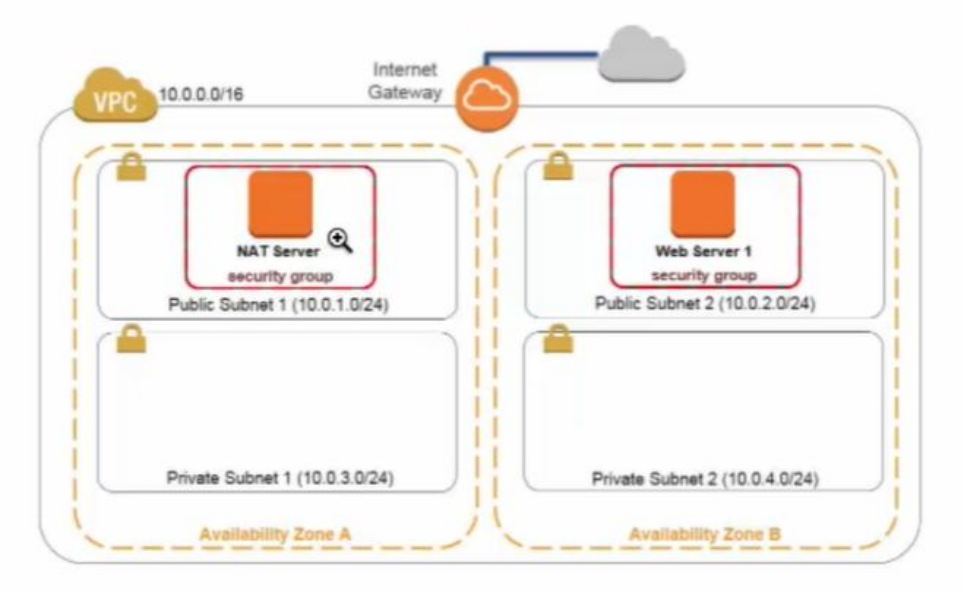
For the next task in our Lab, we are going to add some subnets to another Availability Zone. So select Subnets on the left hand side navigation panel.



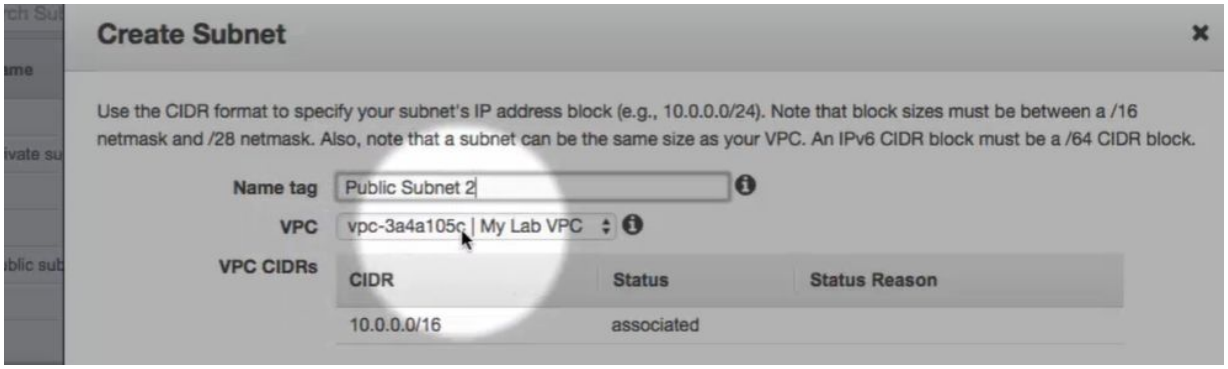
And then click on Create Subnet.



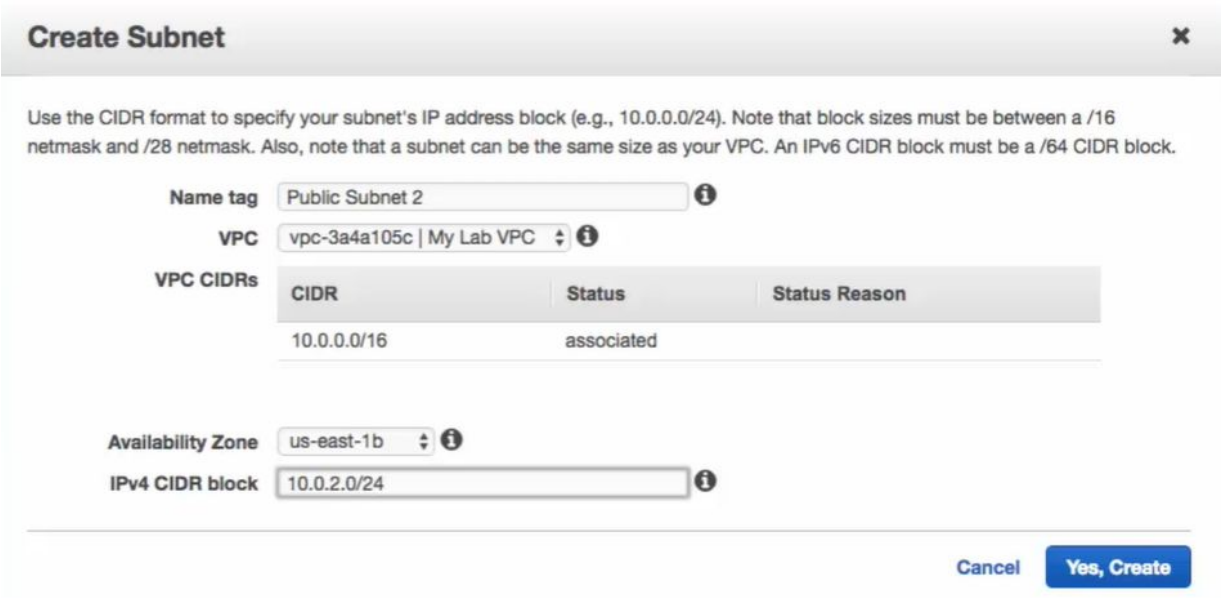
If we take a look back at our diagram, we have essentially created the first set of Subnets in Availability Zone A. We are now going to set up the subnets in Availability Zone B.



Coming back on the Create subnet Window, we will have the Name Tag: Public Subnet 2 as we are creating a Public Subnet. And we do want to make sure that My Lab VPC is selected. And the reason we gave it that name earlier is because otherwise we would get a randomly generated VPC ID which isn't going to be very easy for us as humans to figure out what we are looking for.

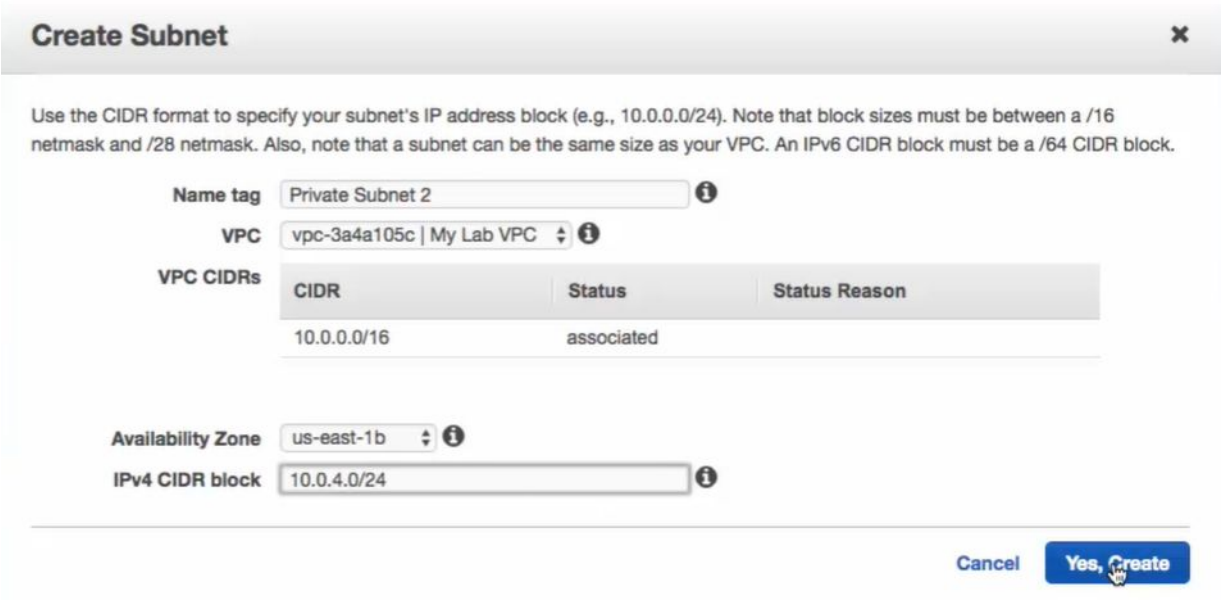


For our Availability Zone, we are going to select us-east-1b and then give it our desired CIDR block 10.0.2.0/24 and click Yes, Create as shown in the next figure.

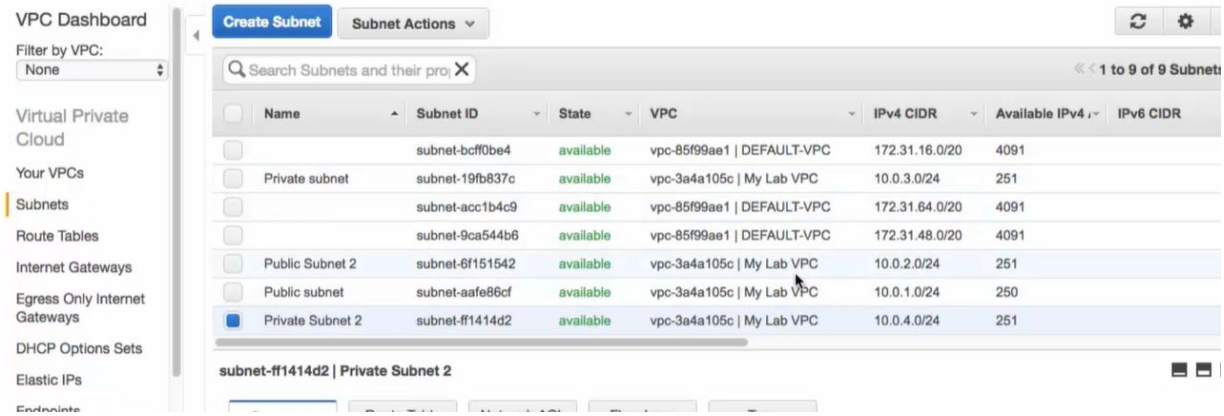


Now that subnets have been created, we can see it listed in our subnets and we are going to go ahead and repeat that process to create an additional Private Subnet. Following the previous steps, we can add a name in the Name Tag as Private Subnet 2 as it is a Private Subnet.

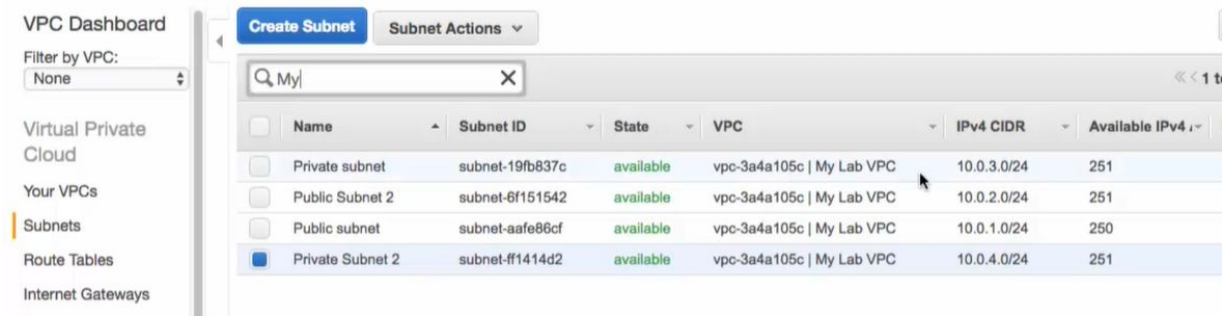
Again, we make sure that we have My Lab VPC selected. Availability Zone as us-east-1b and CIDR block as 10.0.4.0/24 and we will click Create.



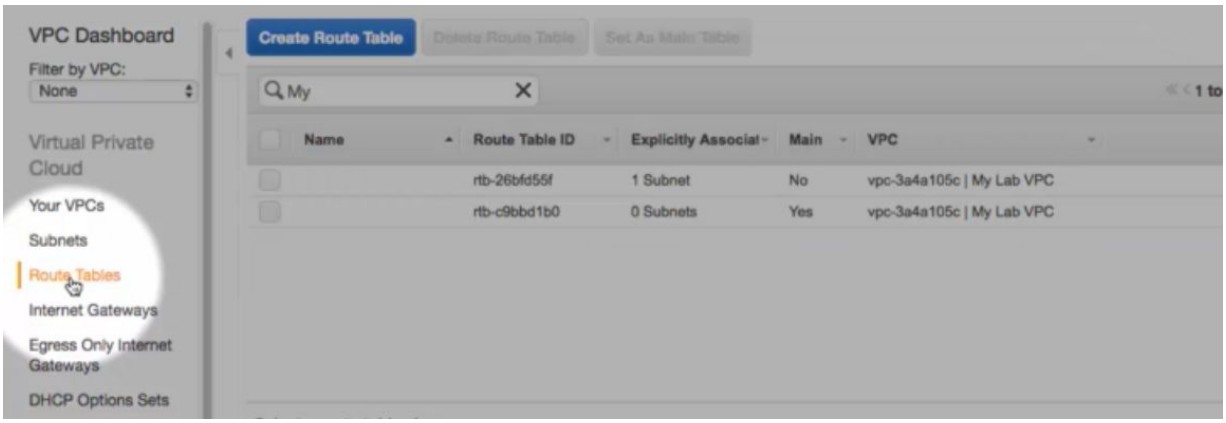
The Private Subnet has been created and is reflecting on our list. And as you can see, it can start to get a little bit cluttered when we are looking at the resources.



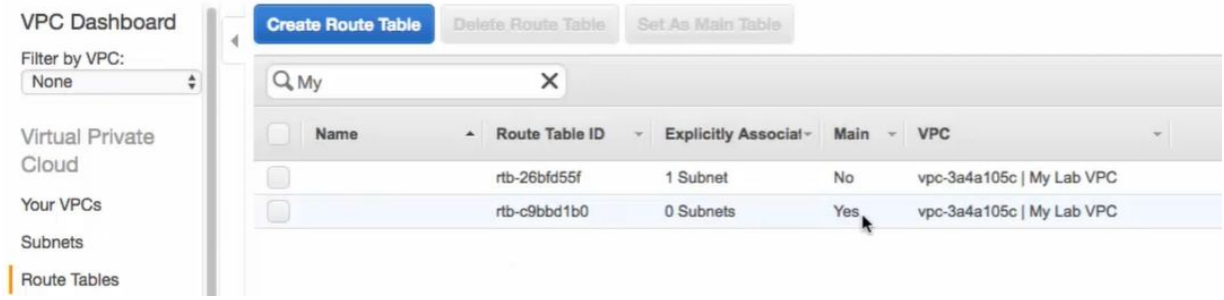
One thing that is handy to do is to use the search bar at the top of the VPC Dashboard. If we begin typing “My”, we can get filtered out for only the resources that exist in My Lab VPC. This is another handy feature to have by naming the VPC when we created it.



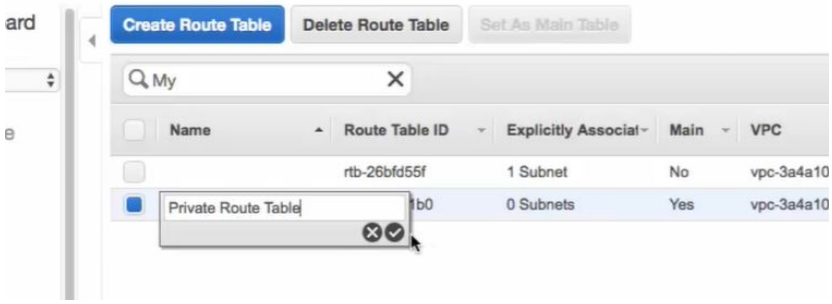
The next thing we need to configure for our Lab is the Route Table. So we are going to select router tables from our navigation panel.



What we are looking for is the route table with “Yes” listed under Main,which is under My Lab VPC.

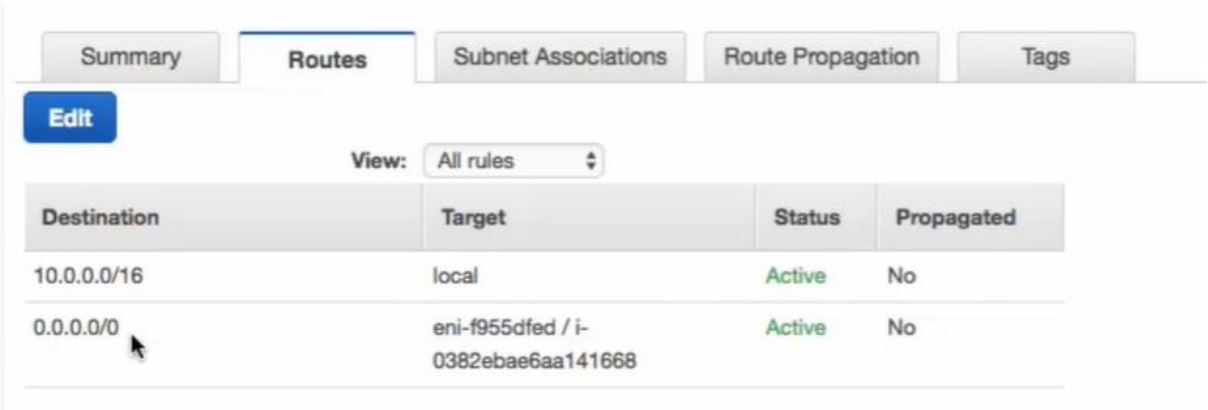


We are actually going to tag this one and give it a name. If we click and double click in the name column, we are able to give it a label so this will be our Private Route Table.



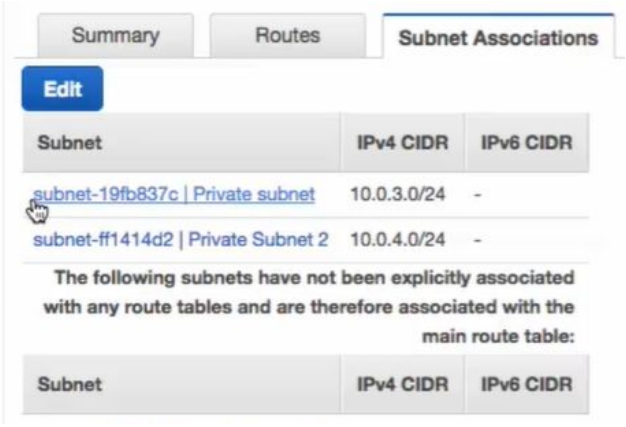
Now that we have given our route table a name, we could take a look at the routes that are associated with it. We can see that this route table is 0.0.0.0/0 entry, which is a wildcard that directs all traffic to the ENI (Elastic Network Interface) which is represented as “eni-f955dfed” attached to the EC2 instance “i-0382ebae6aa141668” as shown in the below figure.

We haven't taken a look at this yet but this is the EC2 instance that the VPC wizard created on our behalf when we ran through the wizard.

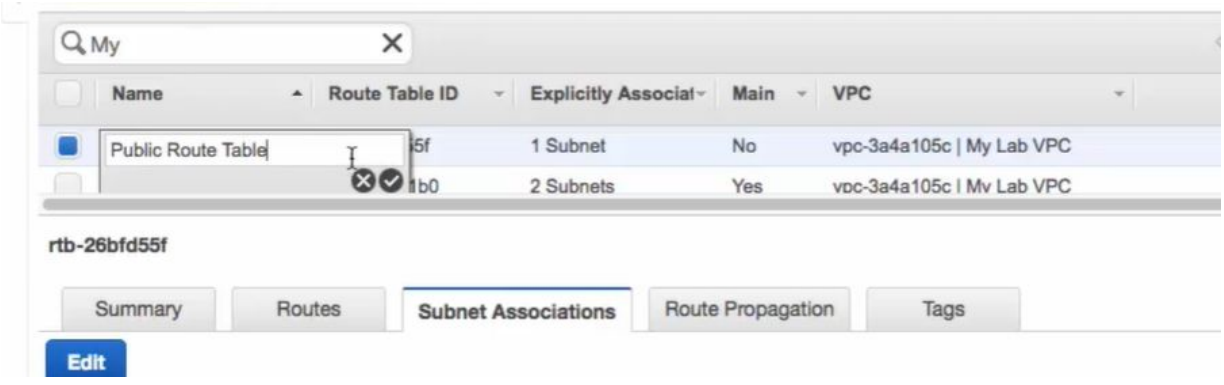


So our next step will be to assign specific Subsets to this Route Table. We want to select Private Subnet 1 and Private Subnet 2 and then click Save.

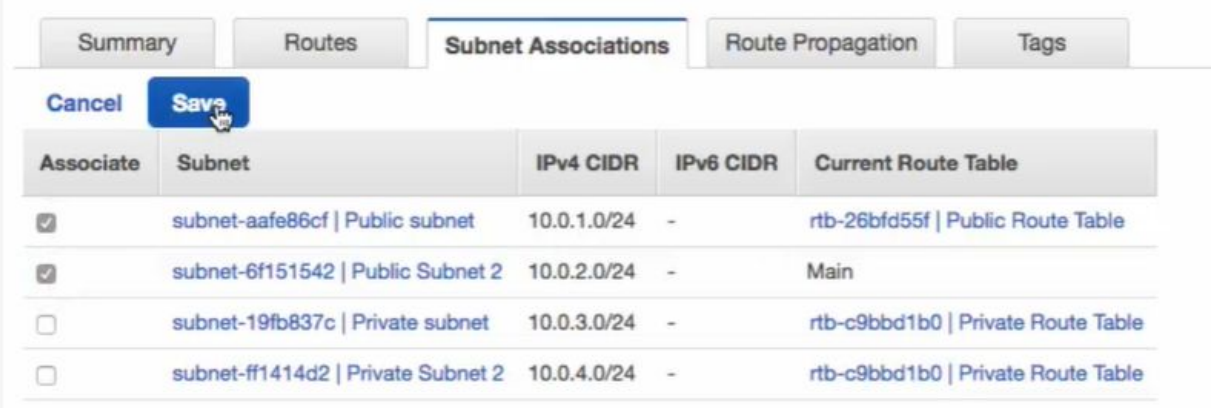
Now that the Settings have been saved, we can see that under our assigned subnets, we have both Private Subnets getting reflected as shown in the below figure.



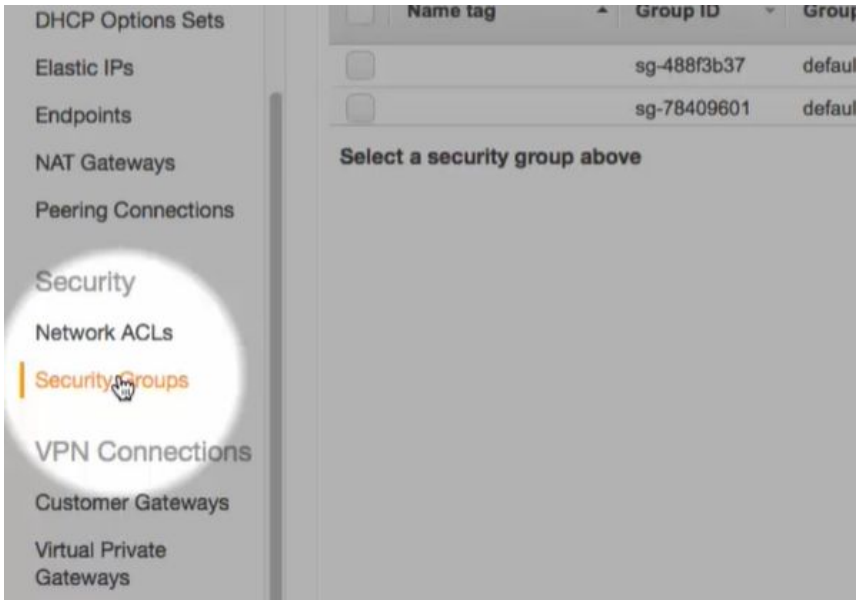
We are now going to repeat this with our other Route Table. Giving it a name Public Route Table.



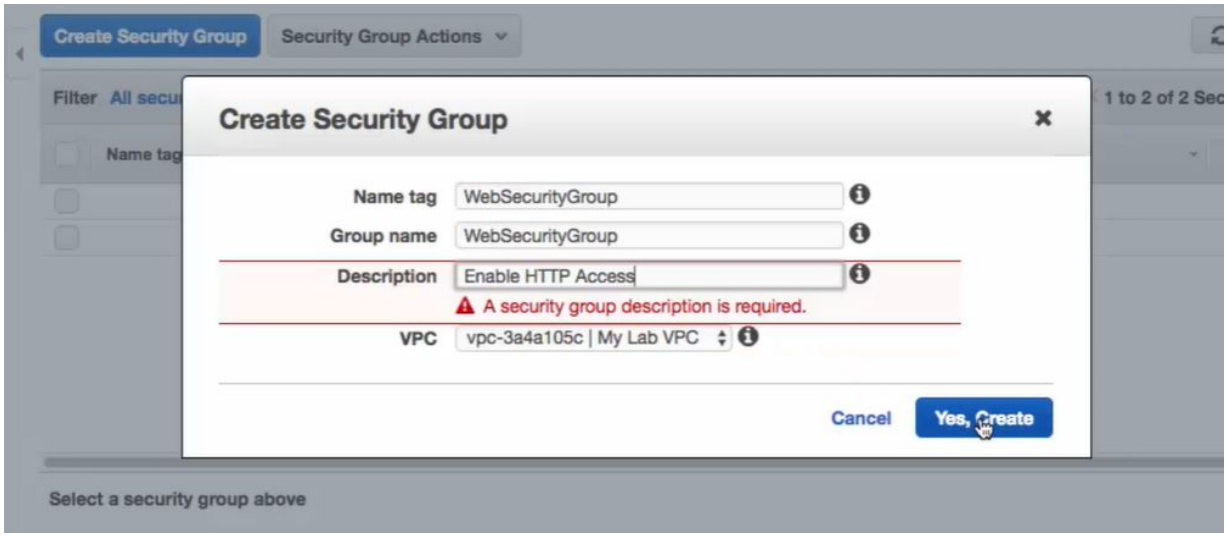
And under Subnet Associations, we want to make sure we have both Public Subnets selected and click Save as shown below. So now we can see that our Public Route Table has both of our Public Subnets associated with it.



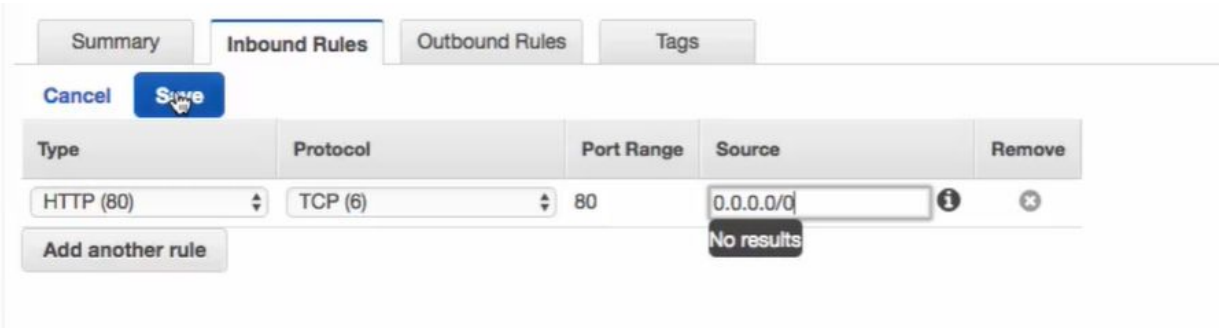
The next task that we are going to take care of is configuring and creating a brand new Security Group. So the Security Group is under Security as shown here.



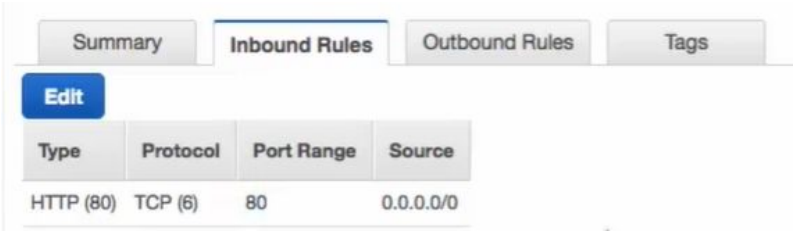
To create our Security Group, we will select Create Security Group and give it a name. In this case, we use WebSecurityGroup. The description will be : Enable HTTP Access and click Yes, Create.



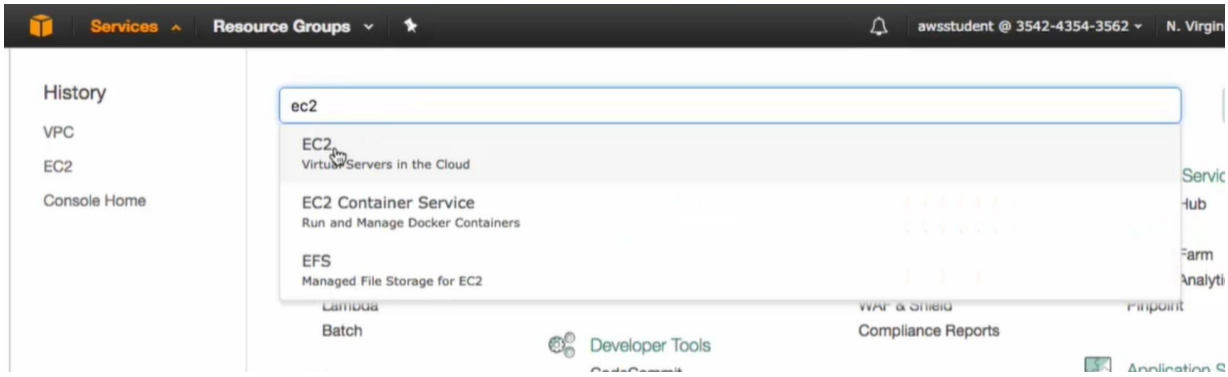
By default when we create Security Groups in AWS, their default position is to have no inbound rules meaning that they are not going to allow any inbound traffic. So we actually want to enable HTTP traffic. In the below figure, we see that we have predefined rules for HTTP which will open up Port 80 and for our source we want this to be able to come from anywhere ie.. 0.0.0.0/0 and click Save.



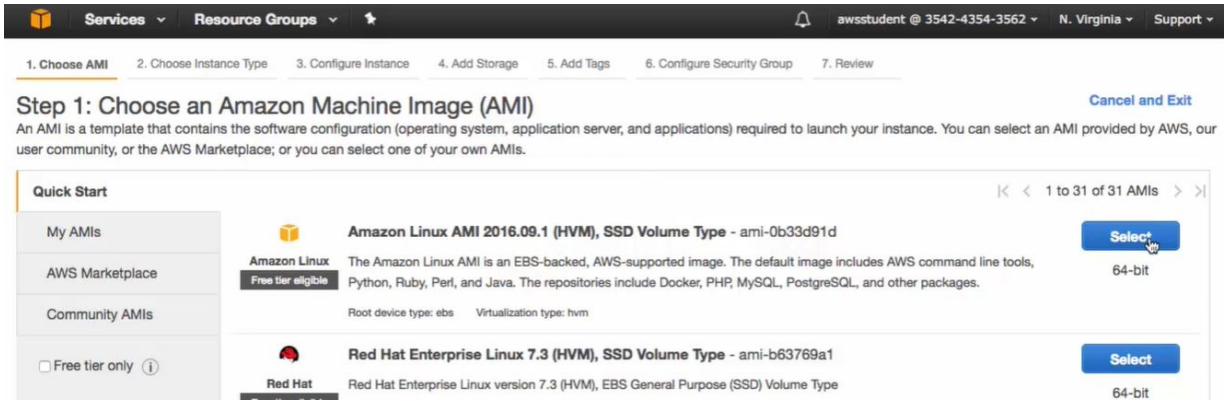
This rule will now allow traffic to come from any source and go through the Security Group on Port 80.



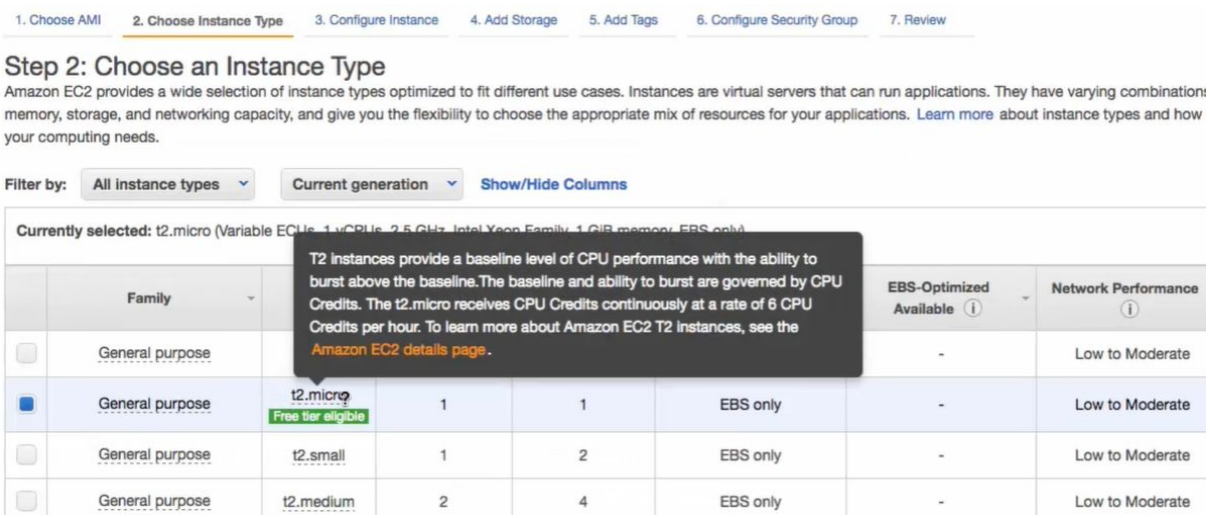
Now that we have successfully configured our VPC, it is now time for us to launch our EC2 Instance. We go to services at the top, type EC2 in the search bar and select the EC2 Service.



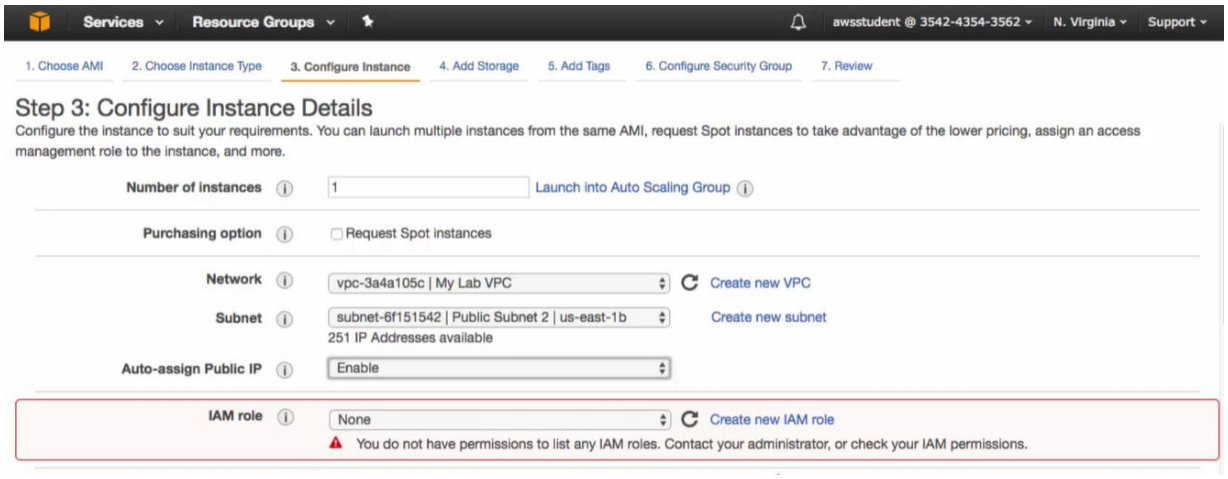
We want to launch a brand new Instance and for this instance, we get to select Amazon Machine Image (AMI). This is essentially an image of an operating system to be applied to our EC2 Instance. We are going to select the Amazon Linux AMI.



Leave it at the default of a t2.micro.

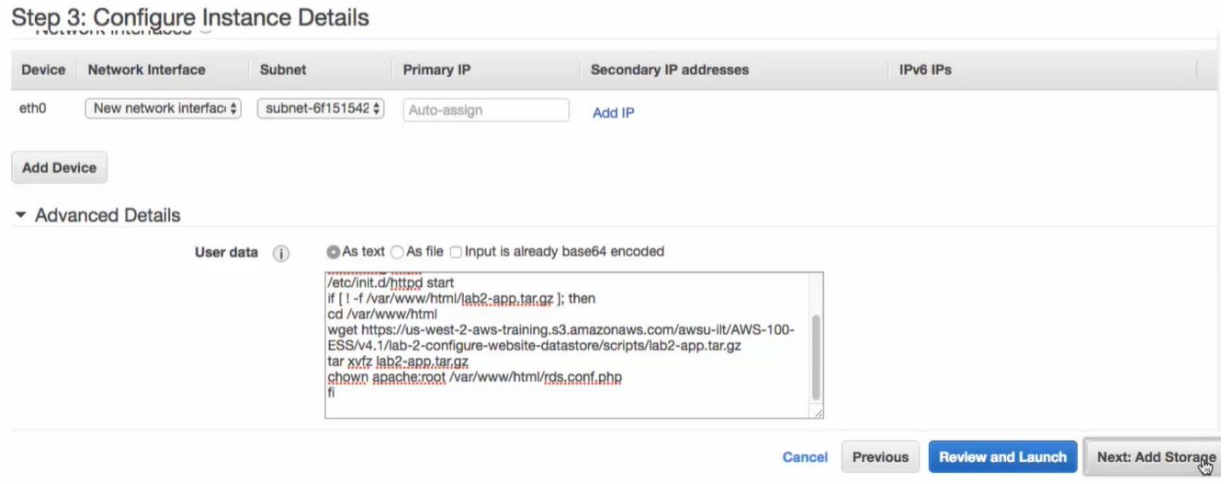


Next, we will configure the Instance details. Here, we are requesting one Instance and we do want to make sure that we select My Lab VPC and in this case, we are going to select Public Subnet 2 for instance.To assign Public IP, we will select Enable. We do not have to have an IAM role, so we do not need to worry about this warning.

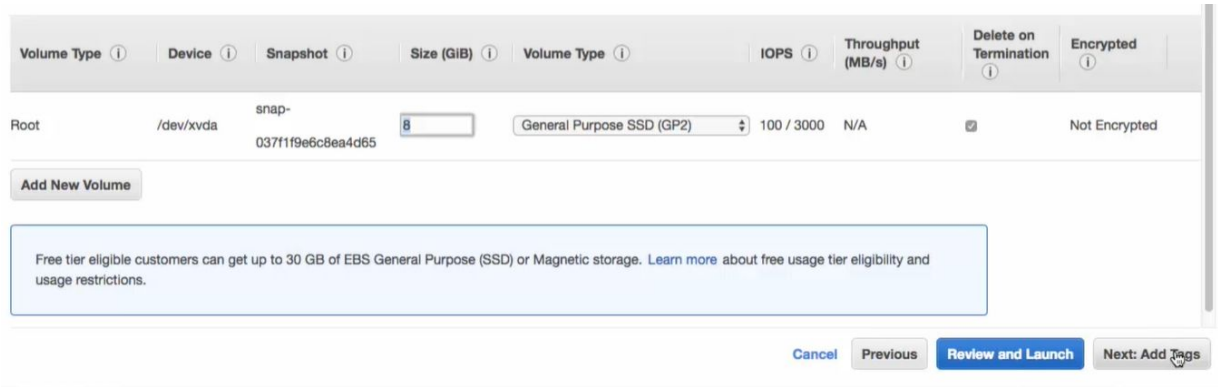


All the way at the bottom, we are going to expand advanced details. What this expansion shows is the user data block. User data allows us to do what is called bootstrapping with our EC2 Instance. When the EC2 instance comes up for the very first time, User Data can be applied to install things like the Apache Web Server.

So in this text box I am actually going to paste a block of code. From this point, we are going to look at Add Storage.

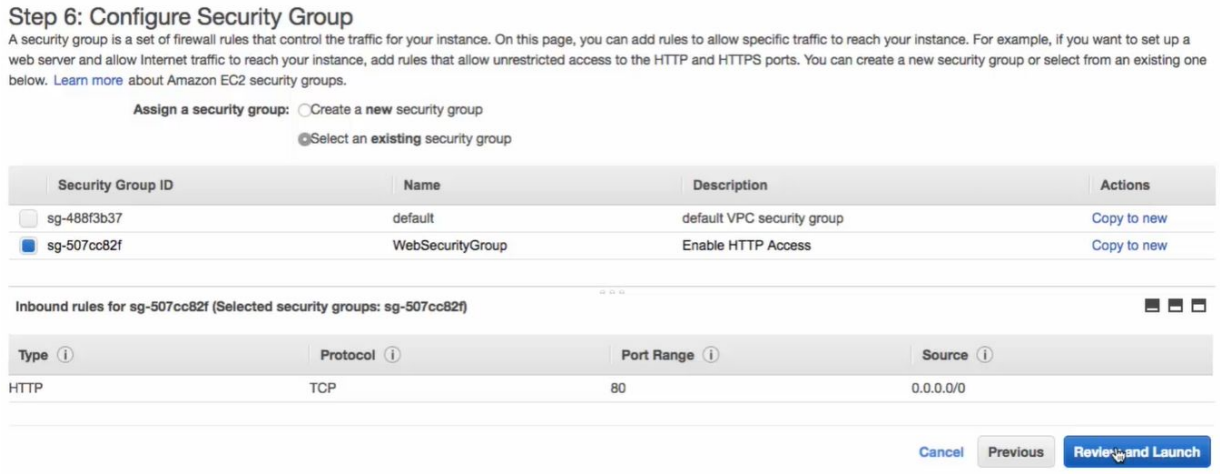


By default you see Amazon Linux AMI will be given an 8 GB main Drive. Now click on Next: Add Tags.

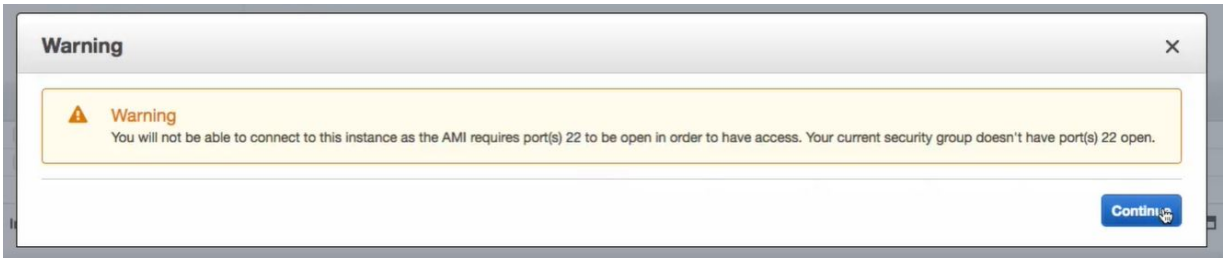


Under Tags, you see that we can give it an additional name. This name, similar to naming our VPCs will allow us to have a human readable friendly name rather than a Unique Resource ID. So we are going to call this Web Server.

Next, we will configure the Security Group. Here, select an existing Security Group of that Web Server Security Group. It allows us to have HTTP access from any source. At this point, we are done configuring our EC2 Instance. Click Review and Launch.

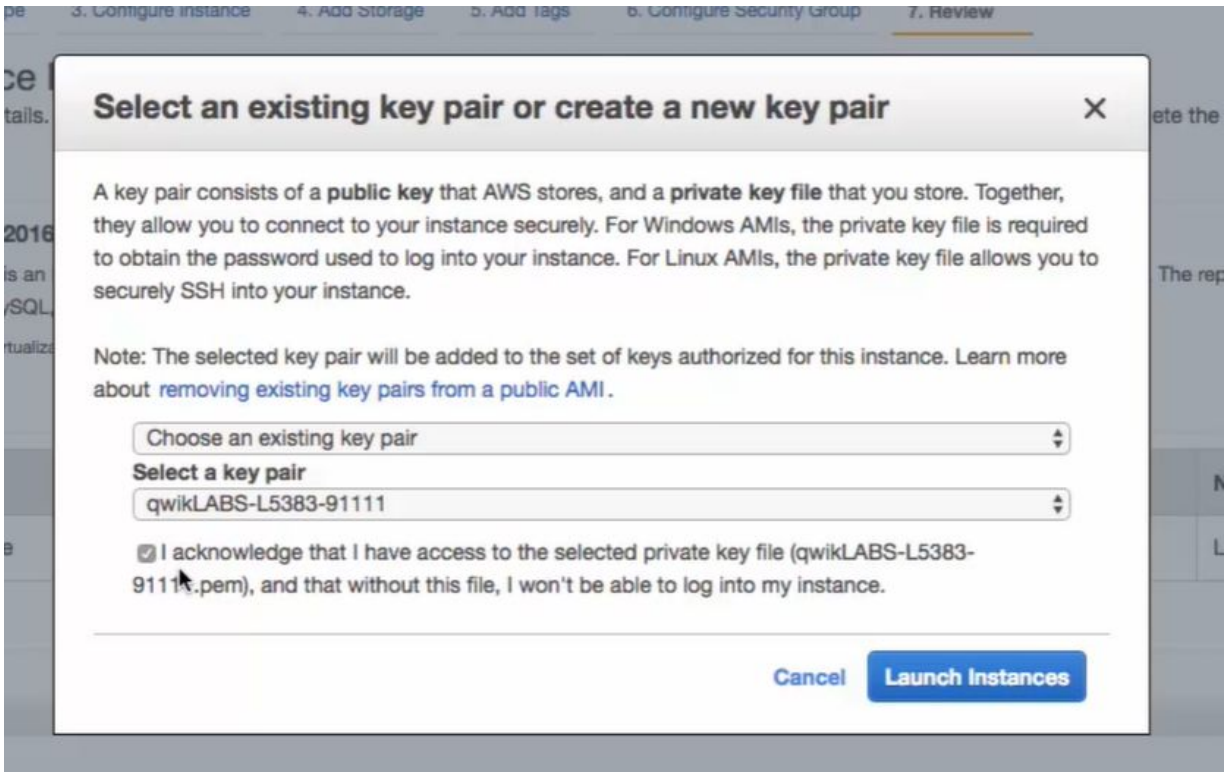


A warning window will appear. We can safely ignore the warning that is being presented. What this warning is letting us know is that if we want to have any remote administration to this server, we currently do not have access to port 22 which would allow us to have SSH access to the Linux instance. We can click continue.

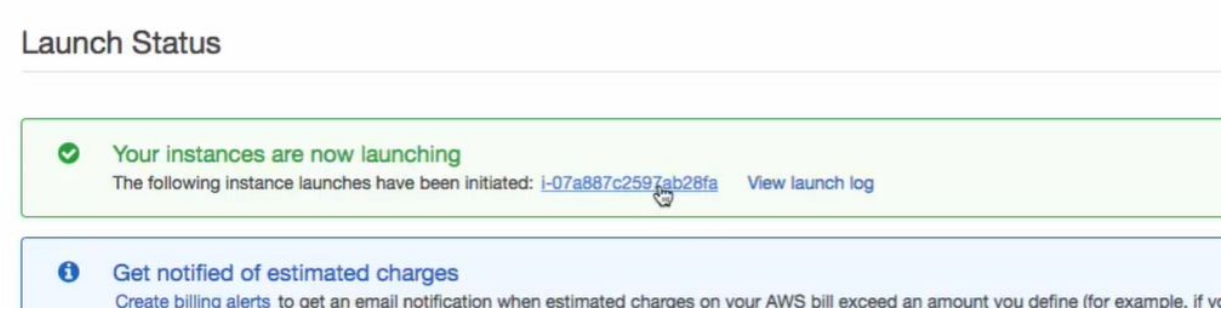


Without making any changes to the new window, click Launch. At this point, we are selecting a key pair. The reason that you have to check a box that says I acknowledge that I have access to the private key is because AWS does not hold a copy of the Private Key. This Private Key is what you use when you are ready to do that remote administration when you're ready to connect over SSH to that instance.

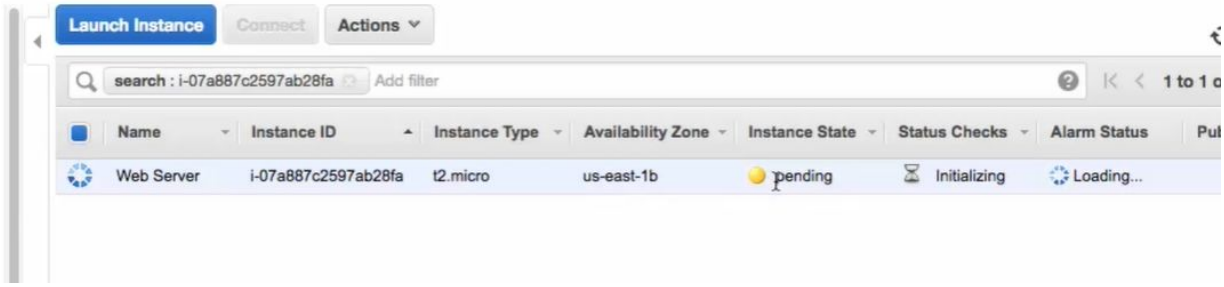
So we will check our box and click on Launch Instances.



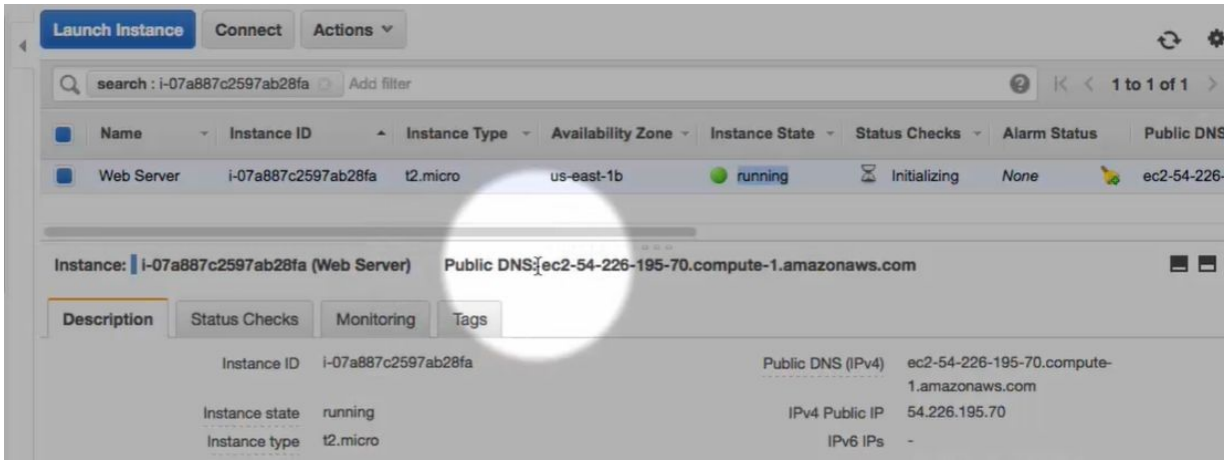
And our instances are now launching. If we click the link on the page that is given to us, it will take us to our EC2 Instances and automatically filter it for the one we just created.



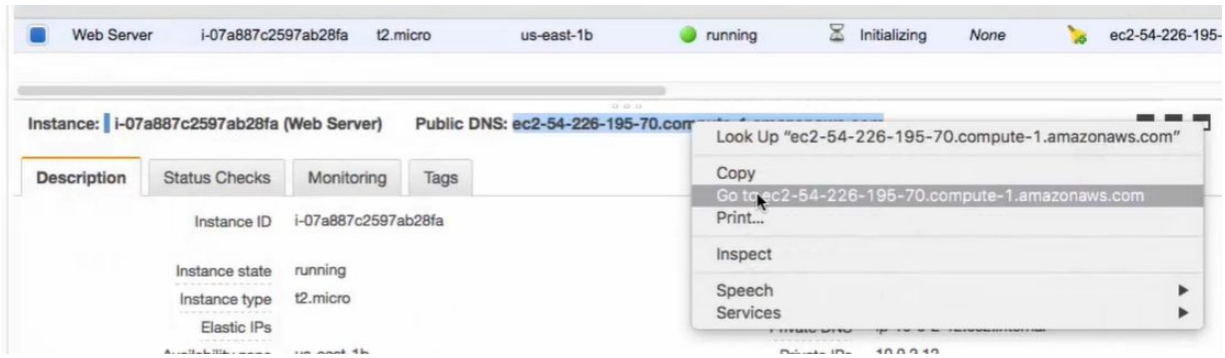
We can see that this instance for the web server is now in a pending state meaning that it is being brought online.



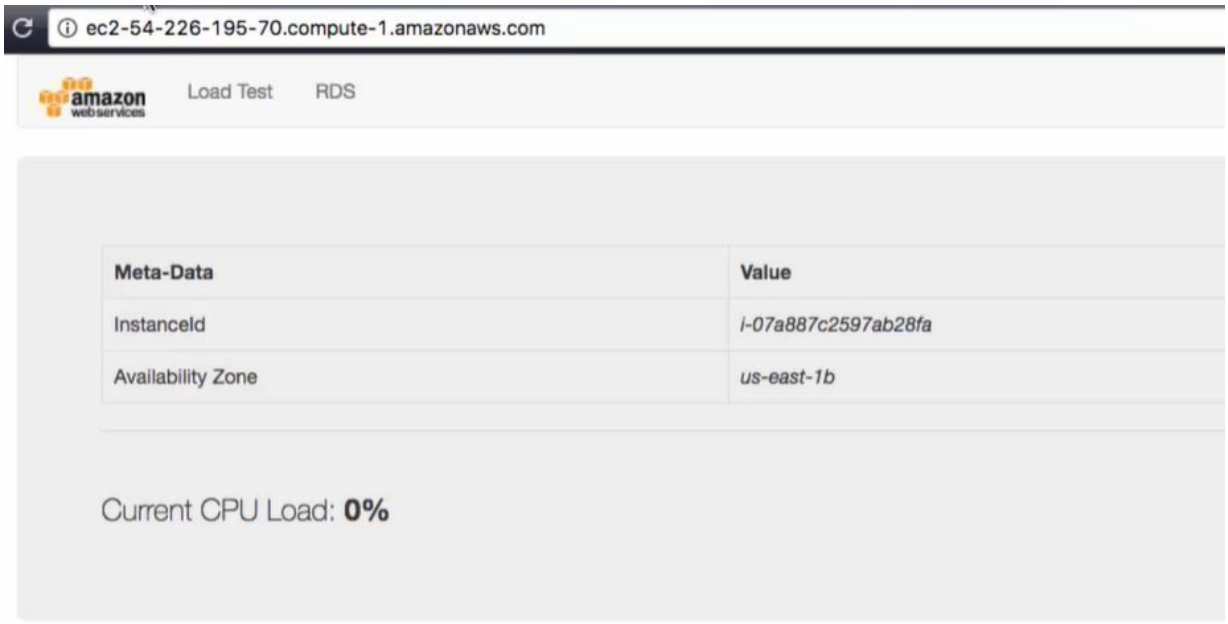
So now that our web server has moved to the running state, it means that AWS has successfully launched the EC2 instance. If we want to make sure that all of the settings that we put into place are working properly, we can take a look at the Public DNS name.



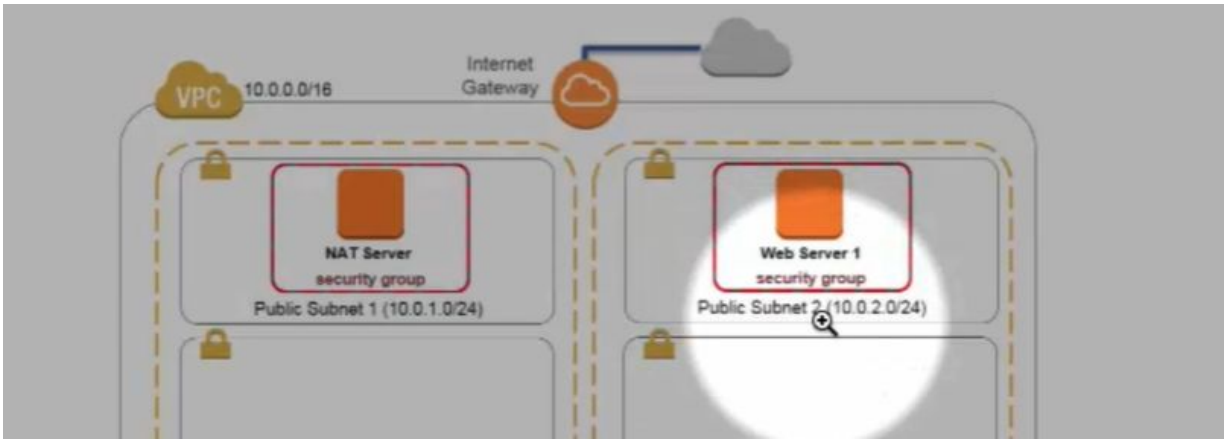
We can go ahead and highlight it. Right click and then click go to... which is going to open up a new tab as shown in the next figure.



In this tab, we see that we have a web page running. This Web page has been configured to pull some metadata from the server including its Instance ID as well as which Availability Zone it is running in.



If we take a look back at the diagram from our Lab guide, we can see that we have now launched this initial web server in Public Subnet 2 and it is able to talk out to the Internet.



And that brings us to the conclusion of the Lab.

Section 4: Module Three: Security and Identity & Access Management

11.2 Introduction to Security and Identity & Access Management

Observe at AWS through the lens of security, privacy and compliance. AWS works closely with their customers in order to help them to achieve their security objectives quickly and easily. Cloud security at AWS is the highest priority. As an AWS customer, you will benefit from a data center and network architecture built to meet the requirements of the most security sensitive organizations.

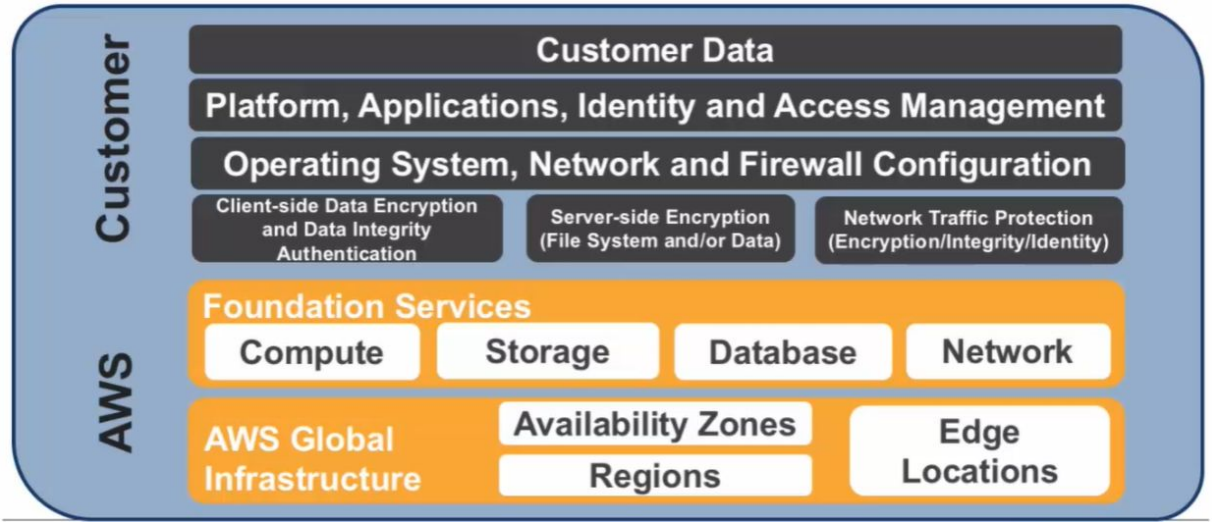
AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users.

Using IAM, one can :

- Create and manage AWS users in groups
- Use permissions to selectively and precisely grant or deny access to resources.

12. Shared Security Model

AWS shared Responsibility ModelAll three security identity and access managemeet's start with a really important diagram which is the AWS shared responsibility model.



This model is generally referred to as the “For” and “In” model.

12.1 For

AWS is responsible for the cloud. It is responsible for the foundation services, compute storage, database network for the global infrastructure and the physical environment, where we host those services.

12.2 In

That is the responsibility of the customer. The customer is responsible for everything they put in the cloud. Critically starting with customer data, probably when we move to cloud computing the thing that has true value.

Try and understand the split of responsibility.

- AWS : For the cloud

- Customer: What you put in the cloud.

This shared responsibility model changes, depending upon the type of service that you are using. For example, maybe you want to spin up an AWS EC2 Elastic Compute Instance.

AWS are responsible for all the foundation's services that make up that hypervisor, the environment where that physical EC2 is going to run.

- You are responsible for the configuration of that DC to the operating system that you put on it
- You are responsible for the network whether the operating system or the EC2 is running the security group/firewall that EC2 instance is protected by.
- You are responsible for patching, maintaining the operating system by updating it from time to time.

AWS will be responsible for the underlying hardware. You are responsible for a lot of work when it comes to EC2. But you can contrast that with another service such as Simple Storage Service S3.

AWS does a lot more than foundation’s services. AWS is also responsible for how that service is maintained. You do not ever have to patch or update the software that runs S3; we do all of the management of the application. You just purely focus on the thing that delivers value to you as a customer which is using it as a storage solution.

- Putting data in
- Taking data out.

One other thing that is very important for you as a customer is that at AWS, they go through obtaining a very wide array of industry certifications through independent third party audit and attestations. You inherit some of those. We will talk a little bit more about some of those regulatory inheritance controls that you get from AWS.

We have one more example of how this is very very important and how it could also go wrong. We mentioned earlier that we have security groups. If you create a new EC2 instance, you put it in a security group or firewall.

AWS gives you the construct of a security group and you write the configuration of that security group.AWS or you could make a rule that says anyone from anywhere can access on any port.
It is a rule (no matter how good or bad) but if you give AWS that rule, they will robustly enforce it. That is very much a good example of a shared responsibility.

We supply the construct of a security group. You write the rule and we enforce the rule. So please write good rules.

12.3 Physical Security

As part of our responsibility for the cloud, one of the things that AWS does is we take physical security of our data centers extremely seriously. They have many years of experience of designing, constructing and operating these large scale data centers.

It is very important that AWS take that experience and provide very good physical security around those environments. The data centers which build our Availability Zones and our Availability Zones that go toward building

our Regions, are typically in very nondescript environments.

AWS never discloses those locations. If you want to get access to those sites, then it is always a two-factor authentication for ingress. There is lots of authorization required for data center access only for approved people. You have to have a specific need.
And when you are accessing our physical environments, there is continual monitoring, logging and auditing of what you were doing on that side.

We strongly recommend that you go and pay a visit to aws.amazon.com/securityfor some really good information about how AWS conducts itself from a security standpoint.

In a nutshell:

- 24/7 trained security staff
- AWS data centers in nondescript and undisclosed facilities
- Two-factor authentication for authorized staff
- Authorization for data center access

13. Hardware, Software, Auditing and Compliance

All changes to AWS hardware and software are managed through a centralized and automated change control process. All access to hardware or software must be authorized and that authorization must be periodically renewed.
Also, privileged access to software and systems require authenticated log on only three Bastion servers. Those Bastion services record all access attempts.

AWS networked devices such as firewalls and other boundary devices that monitor and control communications at the external boundary and also key internal boundaries of the network.

Finally and very importantly, AWS monitoring tools are designed to detect unusual or unauthorized activities and conditions of ingress and egress communication points. These tools monitor server network utilization. They monitor for Port scanning activities, application usage and unauthorized intrusion or take attempts.

We have a world class security team that supports these functions within AWS. And just to highlight one point that we raised there is that AWS are looking for in detecting port scanning activities. Be very careful about doing those sorts of activities in your AWS accounts.

If you want to do a penetration test then there is no problem. They perfectly accept that.
Look at aws.amazon.com/security page as that will give you some guidance on prerequisites before you do pen testing.

All of these tests, change control and monitoring that they provide, gives us the ability to achieve a very high amount of certification and accreditation on the platform and you will see some of those described there.

So for those of you in the financial world, you are taking credit card payments, we have PCI. If you are in health care, we have things like Hiper, NIST federal ram ITAR lots of different accreditation and we believe that AWS has nearly 20 different sets of occasions and accreditation is on the platform.

Non AWS users had to be PCI compliant and no-one enjoys the process of going through that compliance standard. So for any one to now work for

AWS and to realize that they have to do that for PCI and for all of these other standards. This demonstrates a significant amount of trust in the platform.

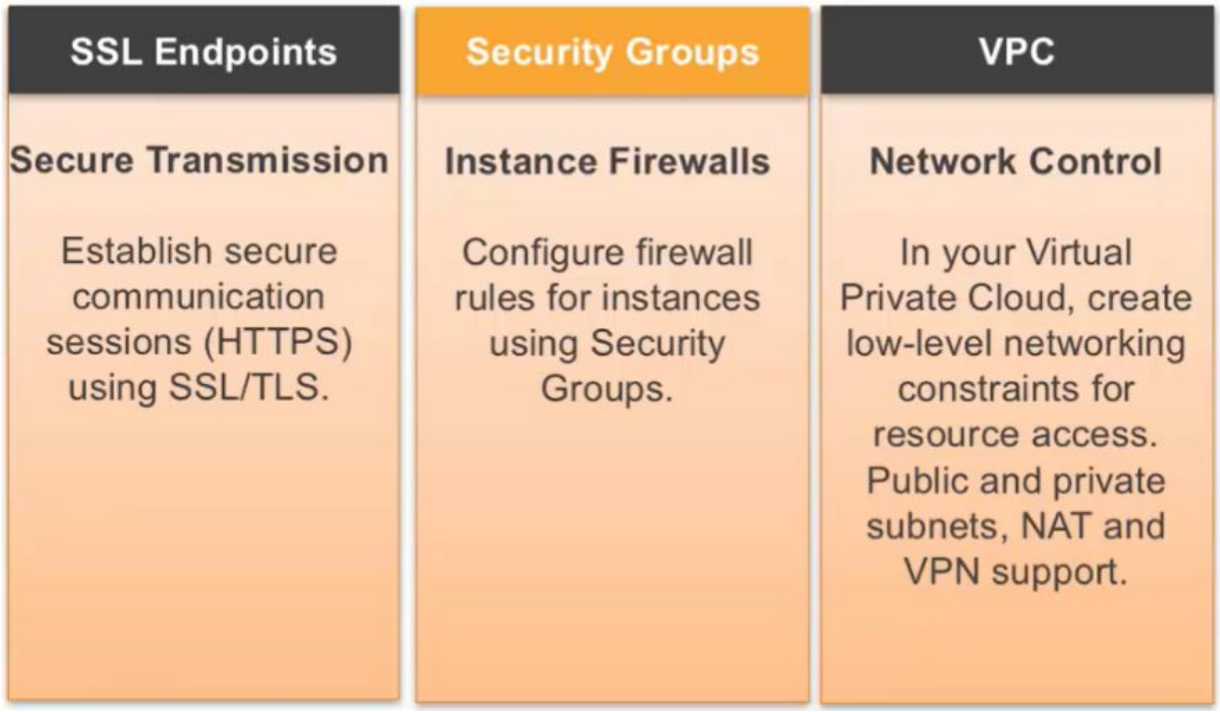
In a nutshell:

- Automated change-control process
- Bastion servers that record all access attempts
- Firewall and other boundary devices
- AWS monitoring tools

13.1 Security Groups

AWS provides you with customer access points also called APIN points that will allow HTTPS access so that you can establish secure communication sessions with your AWS services.
SSL/TLS encrypts the transmission, protecting each request and all responses from being viewed in transit.

AWS also has Security Groups. They provide these Security Groups and they act like built in firewalls for virtual servers. You can control how accessible your instances are by configuring the Security Group rules.



From the public to completely private, or somewhere in between (somewhere in between is normally the right choice). When you insist on residing within a Virtual Private Cloud (VPC), you can control egress as well as ingress traffic.

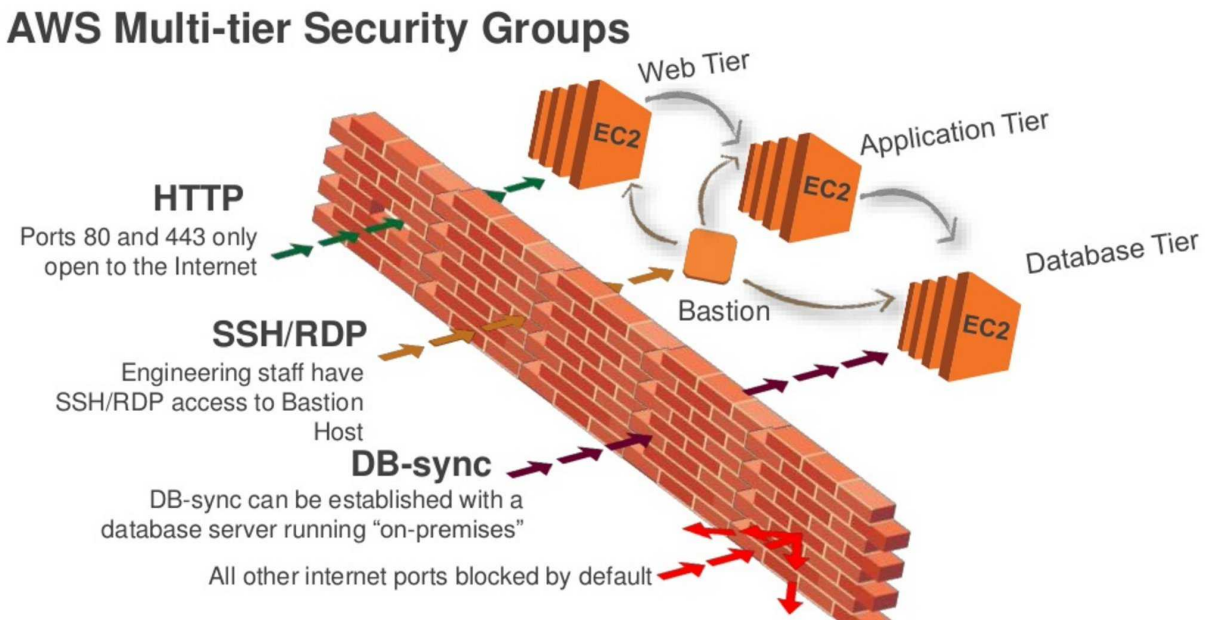
Security Groups can be used by a wide variety of services such as Amazon RDX (Relational Database Service), Amazon Redshift Data Warehousing Service, Amazon Elastic Map Reduce and Amazon Elasticache.

There is a wide list of services that you can configure security group or firewall rule sets around those instances.

13.2 Multi-tier Security Groups

It is very important when we look at Security Groups that way as we are thinking about least privilege. We will talk more about least privilege later within this book.

So as per the figure, we have got a three tier application or a multi-tier application.



At the top, that is our Web Tier. It is the customer presentation layer. On the left hand side, we are running an HTTP with enabled Port 80 and 443 so we could put a slash (/) on the end of that as well.

There is a secure connection and this is allowing Internet traffic to hit the web server. But do we want any other type of Internet traffic to reach the other tiers of my network? And the answer is probably no.

We can write security group rules that say: “do not accept connections to my EC2 instances for the App tier.” Or “Do not accept connections for my database tier.” We can have different security groups around each of these three tiers as well, changing, contrasting and tightening the rules set depending upon the type of access that's required.

Also internally, we can control whether the web tier talks directly to my database here. Always remember, write the correct rule sets that block out and minimize communication. Always very important when engineering security is that you think about that terminology minimizing blast radius.

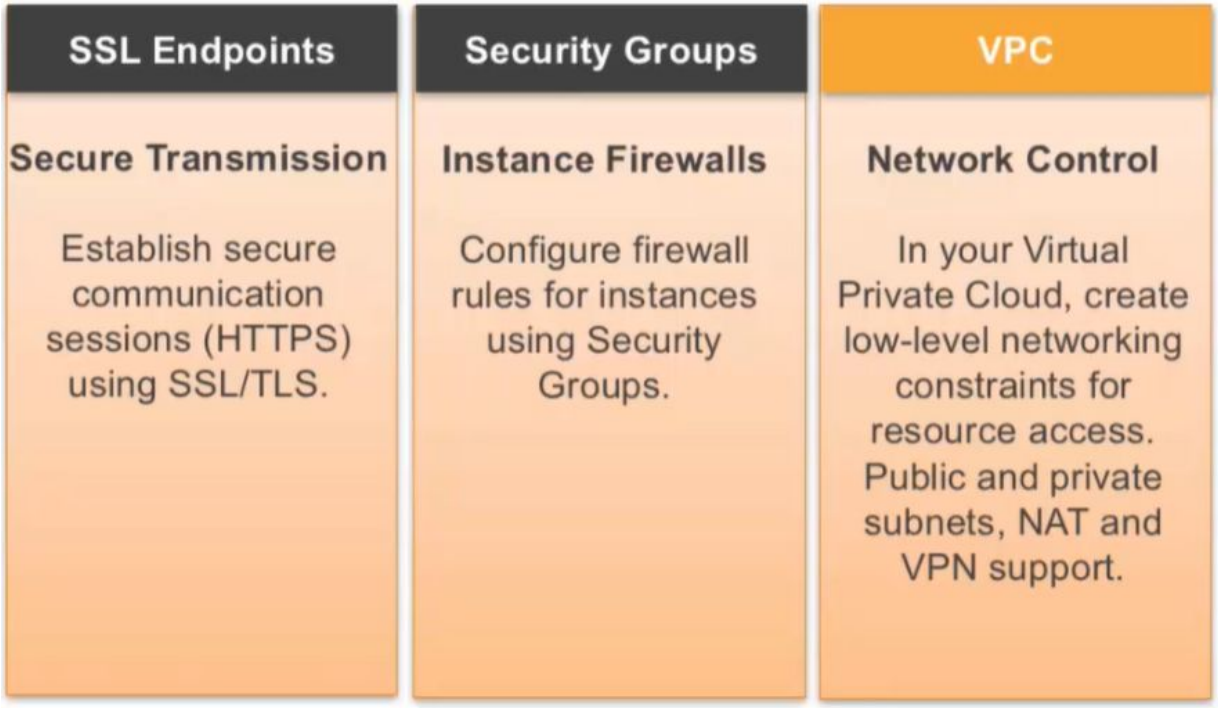
By using Security Groups is one of the ways that we can start to minimize blast radius by reducing the domain of conversation that it can have with other elements or other resources in that AWS environment or in the applications stack.

13.3 Amazon Virtual Private Cloud (VPC)

We have Amazon Virtual Private Cloud (VPC) service which allows us to add another layer of security to our instances by applying network security groups.

In this case network access control lists.

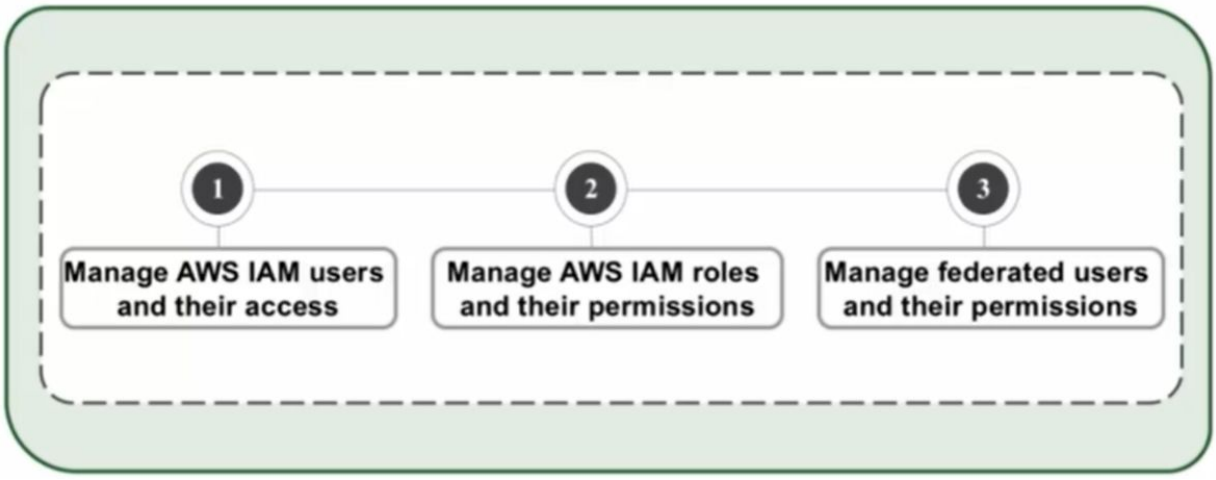
You can also define your own network topology including the IP address range, the number of subnets that you declare and the IP address range for each of those subnets. You can declare how that VPC is connected with Internet gateways, routing tables, virtual private gateway, VPN connections and the PC pairing. There are lots of different ways that we can control how traffic flows within our network.


SSL Endpoints
Secure Transmission
Establish secure communication sessions (HTTPS) using SSL/TLS.
Security Groups
Instance Firewalls
Configure firewall rules for instances using Security Groups.
VPC
Network Control
In your Virtual Private Cloud, create low-level networking constraints for resource access. Public and private subnets, NAT and VPN support.

14. AWS Identity and Access Management (IAM) Authorization

Using IAM, you can create and manage users in groups and use permissions to allow and deny their access to AWS resources.

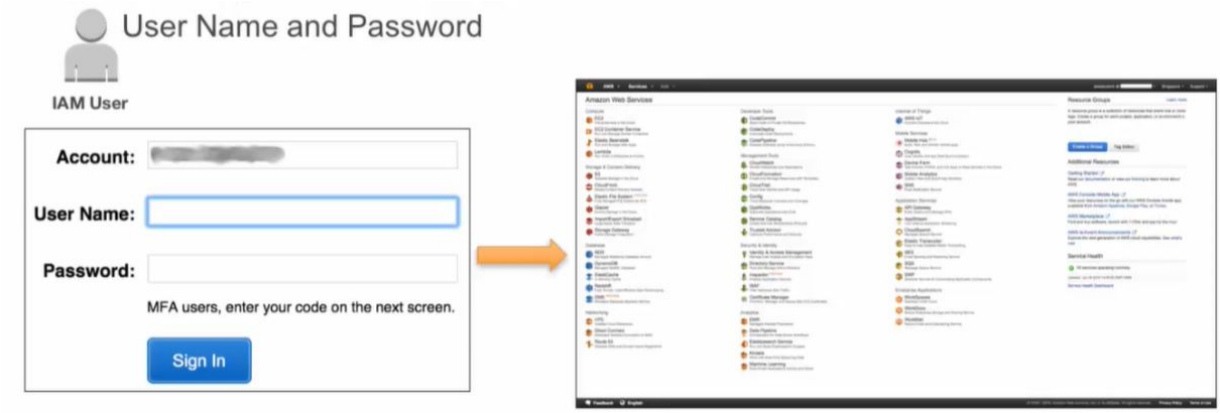
You can also use existing corporate identities, something like Active Directory to grant secure access to AWS resources such as Amazon S3 buckets without creating new AWS identities. It is referred to as Identity Federation.



There is a lot of information on our website link: https://docs.aws.amazon.com/ to talk about how IAM integrates with AWS services.

AWS services and resources can be accessed using the AWS management console, the AWS command line interface or through SDKs and API from a wide range of supported platforms users.

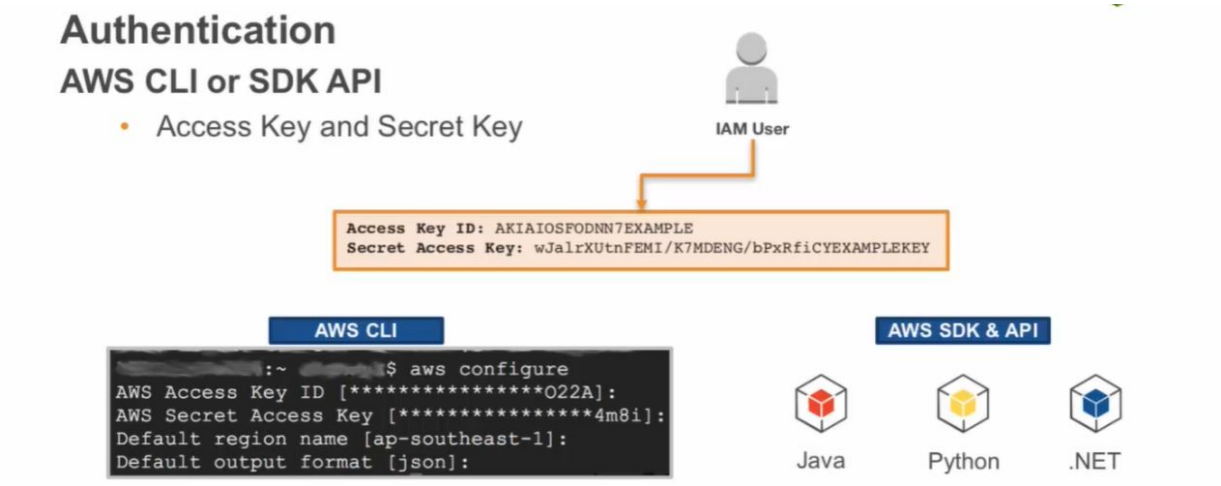
The systems have to be authenticated before they can access AWS services and resources.



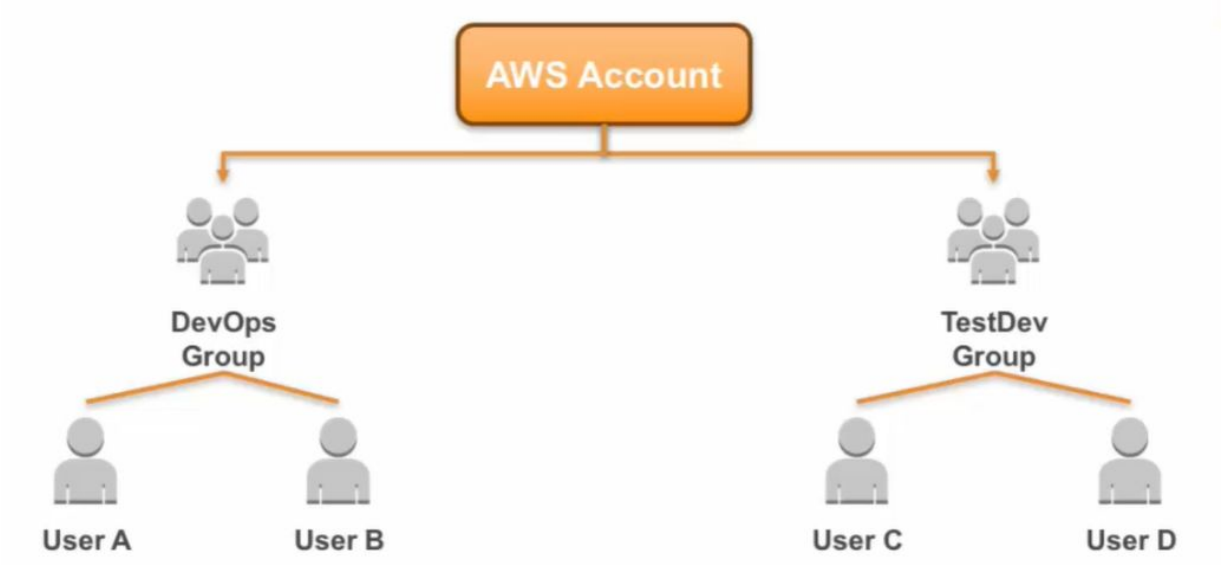
The management console provides a web based way to administer AWS services. If you are the account owner, you can sign into the console directly using the root account. However, do not do that. The reason behind not doing this is that the root account carries permissions to do everything. They are:

- Create
- Manage
- Maintain and
- Delete.

It is very easy to make mistakes when you have all of those permissions. Also you cannot take away permissions from the root account. So we strongly advise best practice to use Identity Access Management (IAM). The good news is IAM is a complementary service. There is no charge for creating and managing and maintaining users through the IAM interface.



When you create an IAM user, you can optionally also create an Access Key ID and a Secret Access Key as well for that user. And these Access Keys and Secret Access Keys can be used by our AWS command line interface. Also by our SDKs and API interface to the AWS platform. This opens up a wide variety of automation opportunities for you within AWS.



Along with users, you also have the ability to create groups. It is highly recommended that when assigning permissions, we should assign permissions to groups and we put users in those groups. This is a much more scalable way of managing user permissions. It is also possible for one user to belong to multiple groups.

14.1 Policies

After the user system has been authenticated, they then have to be authorized to access an AWS service. To assign permission to a user, a group, a role or a resource, you create a policy which is a document that explicitly lists permissions.

An IAM role is similar to a user such that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with a person a role is intended to be assumable by anyone who needs it.

Also our role does not have to have any credentials ie.. a password or access key associated with it. Instead, if a user is assigned to a role, access keys are created dynamically and provided to the user. We will cover policies and roles in more detail in the coming topics.

14.2 IAM Policies

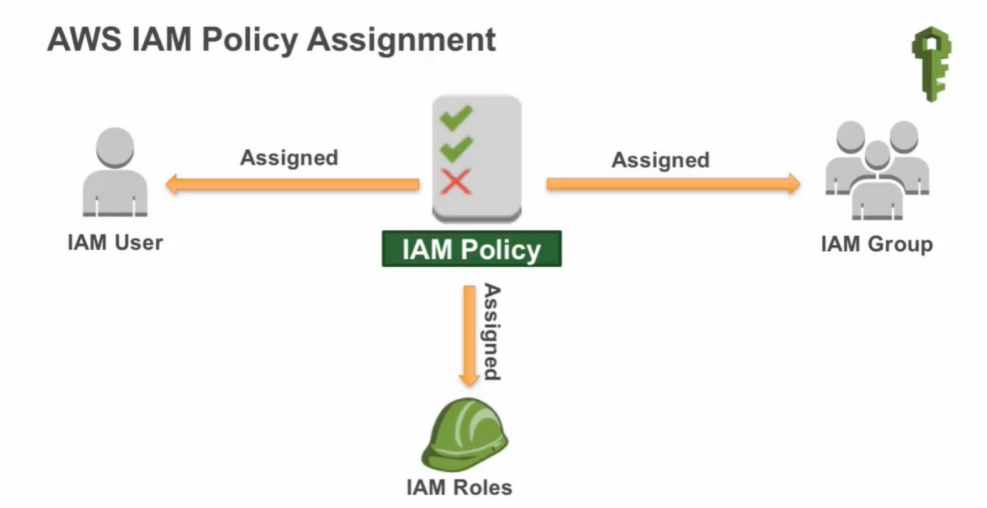
A policy is a document that we write in javascript object notation and it consists of a number of different statements.

Importantly, within those statements we have an effect. It is either allow or deny. It is the principal who is being given permission, the resource what type of AWS services are trying to be acted upon and the condition: Where is the connection coming from? and What type of authentication is being used?

The good news is that we have a policy generator that can help you build these policies. We have something called the policy simulator which will help you test the policies that you write. And also we have a number of policies that are preexisting in your AWS account. So these are referred to as AWS managed policies.

You can create your own customer management policies or even use another form of policy called in line policy. The language and the method of creating them is the same regardless of whether they are AWS managed, customer managed or in line policies. The thing to remember that is significantly different is, with an in line policy it is tied to a resource, user or group. If you delete that resource user or group, the policy document also gets deleted as well.

So once we have created an IAM policy, we can assign that to either a user, a group or a role. Remember that our role is similar to the user as it is an AWS identity with permission policies that determine what the identity can do or cannot do in AWS.

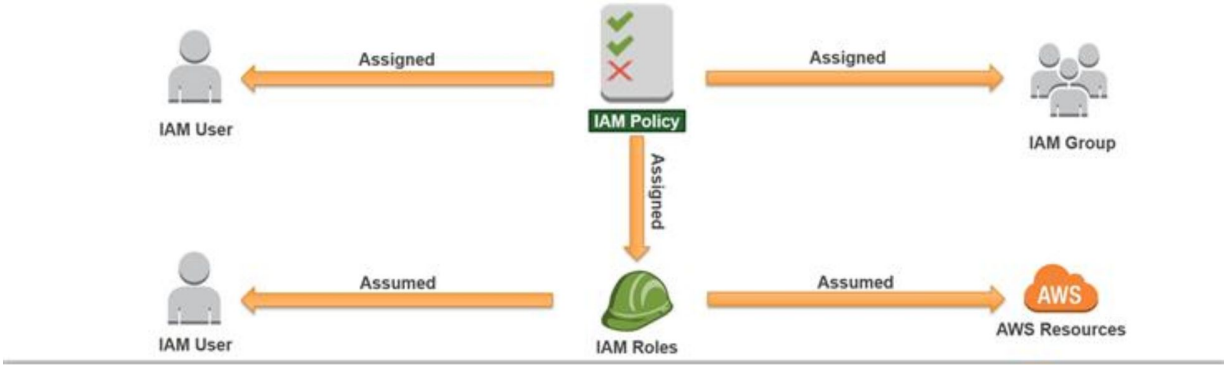


However instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also our role does not have credentials, password or access keys associated with it. Instead if a user is assigned the role, access keys are dynamically created.

But the important element here is that a single IAM policy could be assigned in multiple different places. All through our best practice here, when we are looking at users and groups is to assign policies to groups and put the user in the relevant group.

14.3 Assume Role

We can say this diagram expands a little bit more about IAM Policy Assignment and you can say that we have got a single policy that is assigned to a user that is assigned to a group and it is assigned to a role.



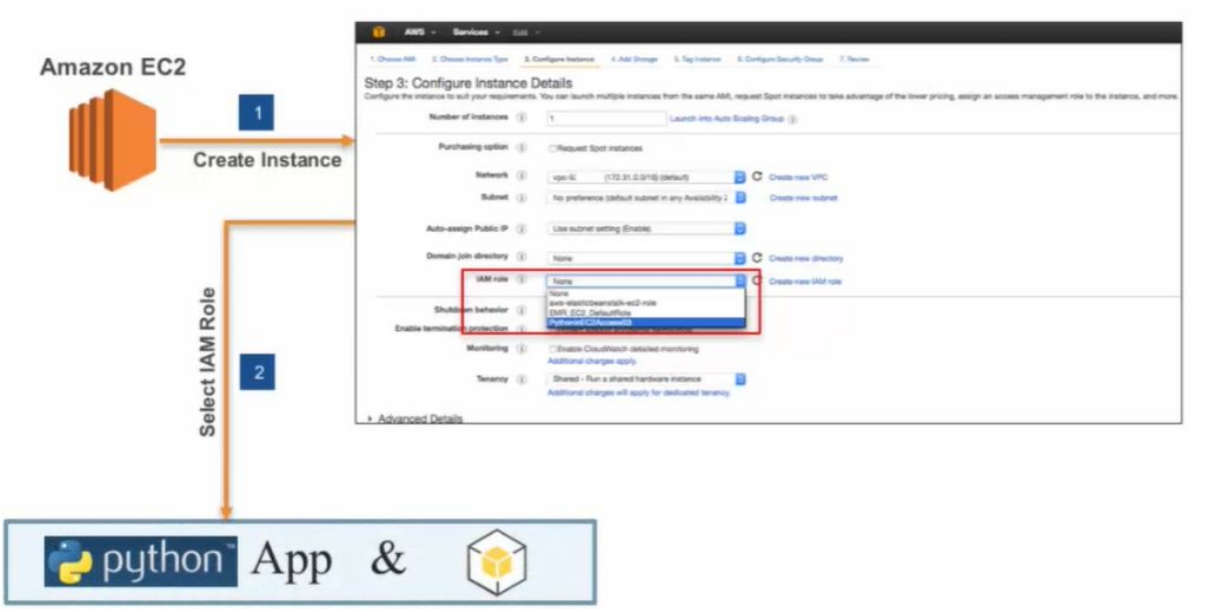
What we have the ability to do now with that IAM role is to have another IAM user or even an AWS resource. Assume that role.This means that they assume for a short period of time, temporary credentials that allow them to action the IAM policy embedded or assigned to the role.

Here is a really significant benefit of the role and the ability for AWS resources to assume that role. In this particular example we can talk about the Python application that we are running on an Amazon EC2 instance and that Python application needs to interact with Amazon S3.

So traditionally, we could just write our Python application and store that access ID and secret key in our Python script. But obviously that has some potential security issues. Maybe we accidentally publish our Python script to get other people. They will lead other people to view that and they would be able to get our access ID in a secret key.

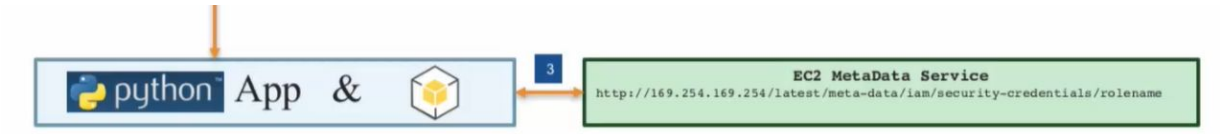
One thing that we can do with our role is we can stop that type of activity and instead we can assign a role to the EC2 instance that has the necessary permissions to interact with S3. Now we no longer have to embed credentials in code. Instead, the code inherits or assumes the set of policy permissions that have been assigned to the role that the EC2 instances is running under.

Here is an IAM, assuming a roll process with a sample workflow with the example that we have just used. We are going to create an Amazon EC2 instance. And during that process of creating the instance, we can assign the instance role called Python EC2 access S3.



This is very important because you can only assign an IAM role at the time that you create the instance. You cannot go back and add a roll to an existing EC2 instance. Once we have created the instance with the correct role assigned to it, we can launch our Python application.

When we launch the python app on the EC2 instance, it will interact with the EC2 metadata service which will give it the ability to obtain temporary security credentials. We will talk a little bit more about metadata and Temporary Security Credentials a little bit later on.

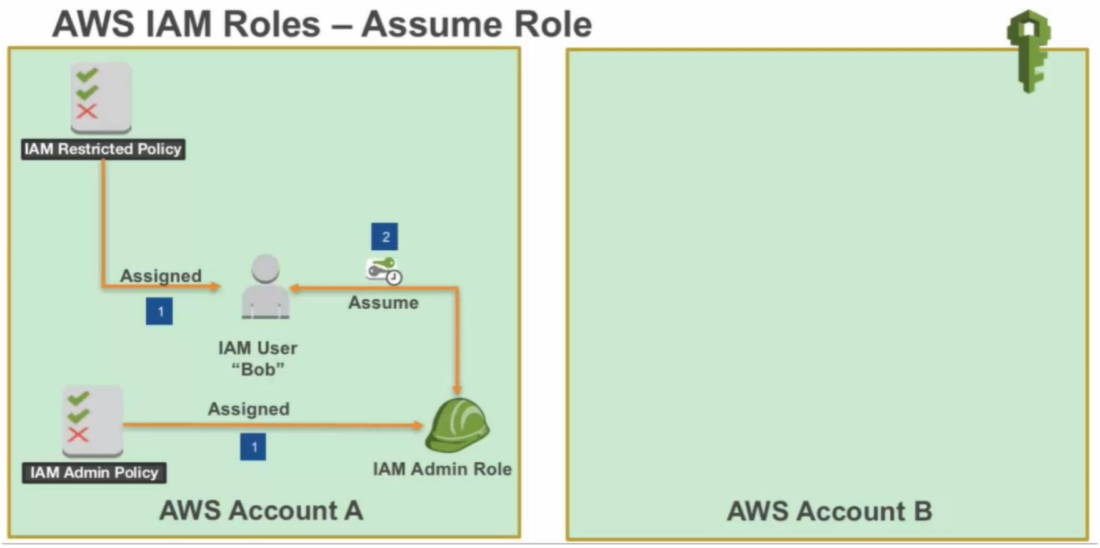


Once the python application is inherited, the permissions from the Python EC2 access S3 role, then it has the ability to interact with and carry out whatever that gets and puts all that it needs to work correctly with our Amerson S3 bucket.

14.4 Assume User

In the previous topic, we looked at how an EC2 instance could assume a role. In this topic, we are going to look at how an IAM user can assume a role.

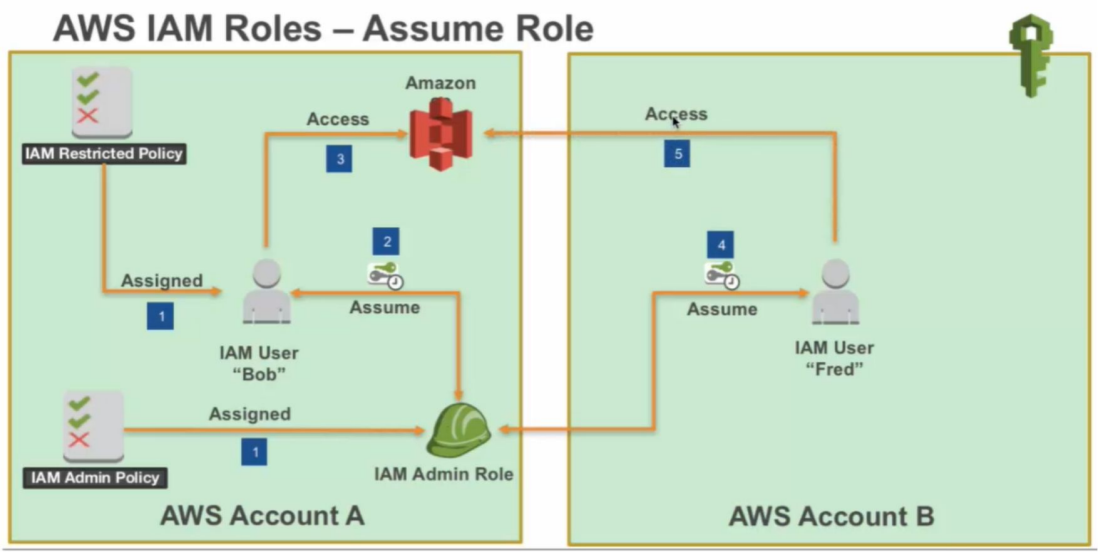
Here we have got a user called Bob and he has a very limited set of permissions only allowing him to access a restricted number of AWS resources. But occasionally, Bob needs to carry out some administrative duties.



What we have done here is we have created a role called the IAM admin role. We have also given Bob the permission and we have done that explicitly. We have given Bob the permission to assume that role. So when he has to carry out those admin functions, he can assume the IAM admin role which gives him a temporary set of credentials. He then drops his normally restricted set of credentials and uses the new assumed credentials to access the rules resources that he needs.

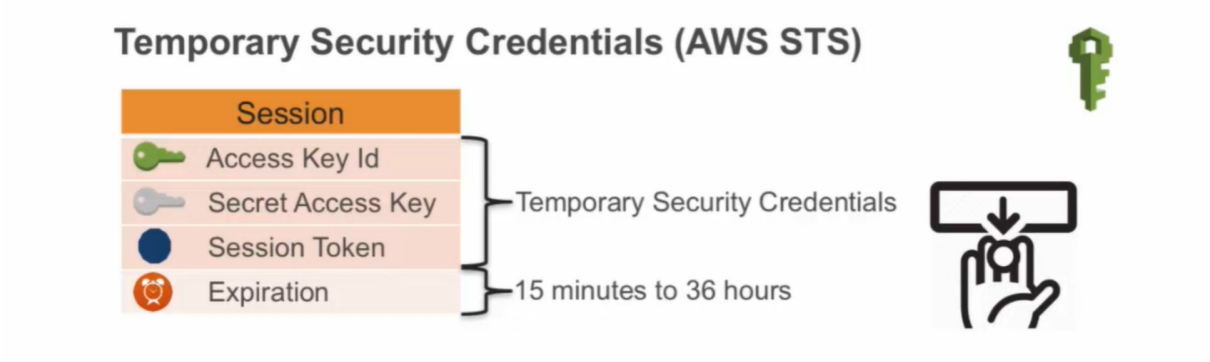
One other benefit of the IAM assume role capability is that you can enable cross account trust relationships. So I could have a user called Fred in another AWS account. And I can explicitly declare that Fred has the

permission to assume this role and then when he goes through that assume roll process he then also has the ability to access the resources in the first AWS account.



14.5 Security Credentials and IAM Authorization

So as we just saw in the previous two examples temporary security credentials are required. And these are generated by the AWS Security Token Service or STS service. These credentials are short term but work identically to long term access key credentials. The credentials are generated dynamically and provided to the user when requested.

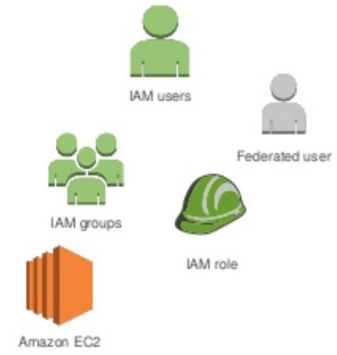


So a session established with AWS STS consists of an Access Key ID, Secret Access Key, a Session Token and has an Expiration time. The Expiration time could last between 15 minutes all the way up to 36 hours.

The keys are used to sign API requests and pass in the token as an additional parameter which AWS uses to verify that the Temporary Access Keys are valid. There is lots of use cases for Temporary Security Credentials (STS) such as :

- Cross account access
- Mobile Users
- Key rotation for Amazon EC2 based applications but also very importantly for
- Account Federation.

So if you are an existing active directory customer, you already have a very large AD built with all of your users and groups. You can use roles and STS to do Federated access for your AD users to use and inherit permissions inside of your AWS account.



IAM is not suitable to be used for your operating system or for your application authentication. It is to be used for your AWS Management Console which contains the Username and password. Or by your AWS

Command Line Interface (CLI) or STK API using Access Key and Secret Key or authentication.

The Authorization occurs with the permissions that are given in the Policies.

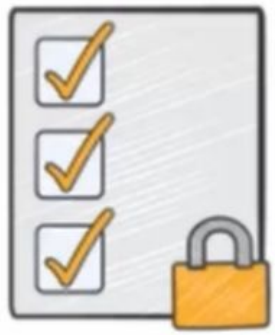
15. Security Best Practices

These are the following IAM best practices.

- ❑ **Delete** AWS account (root) access keys
- ❑ **Create** individual IAM users
- ❑ Use groups to **assign permissions** to IAM users
- ❑ Grant **least** privilege
- ❑ **Configure** a strong password policy
- ❑ **Enable** MFA for privileged users

It is very important to delete AWS account root access keys. We really do not want anyone to interact with our AWS account programmatically with the root account. And in actual fact, we shouldn't be using the root account on a daily basis to interact with our environment because it goes against the principle of granting least privilege.

Instead of granting least privilege, we can create individual IAM users and we are going to put those users in groups and assign permissions to those groups. We are going to configure a strong password policy and enable MFA access. For privileged users, you could consider using MFA for all your users.



- ❑ **Use roles** for applications that run on Amazon EC2 instances
- ❑ **Delegate** by using roles instead of by sharing credentials
- ❑ **Rotate** credentials regularly
- ❑ **Remove** unnecessary users and credentials
- ❑ Use **policy conditions** for extra security
- ❑ **Monitor** activity in your AWS account

Remember the ability to use roles for applications that run on top of Amazon EC2 instances. This means that you no longer have to put credentials inside your code and run the risk of getting those exposed.

Delegate by using roles instead of by sharing credentials. You should never have users sharing multiple or single accounts multiple users sharing single accounts. It is very difficult to go back through an audit and identify who was the actual person that made a change.

Retain your credentials, regularly, remove all unnecessary uses and credentials.Use Policy conditions for extra security such as a particular action is only allowed if the user is authenticated from an IP address.

And very importantly, monitor the activity in your AWS account. We have a service called AWS Cloud Trail that will give you who has made what changes ? Was it successful ? Where were they logged in from ? Look at that information on a regular basis and monitoring is done with AWS Cloud Trial.

15.1 AWS Resource-Based Policies

We had focused on how we assign policies to our IAM users, groups and roles. And also we have looked at the temporary credentials that can be inherited with the STS service as well.

But one thing to remember is that there is an alternative to IAM, some services give you the ability to assign a policy directly to the service. This can be very useful if you are trying to grant cross-current access to their resources or if you are trying to create a more granular set of permissions for an application.

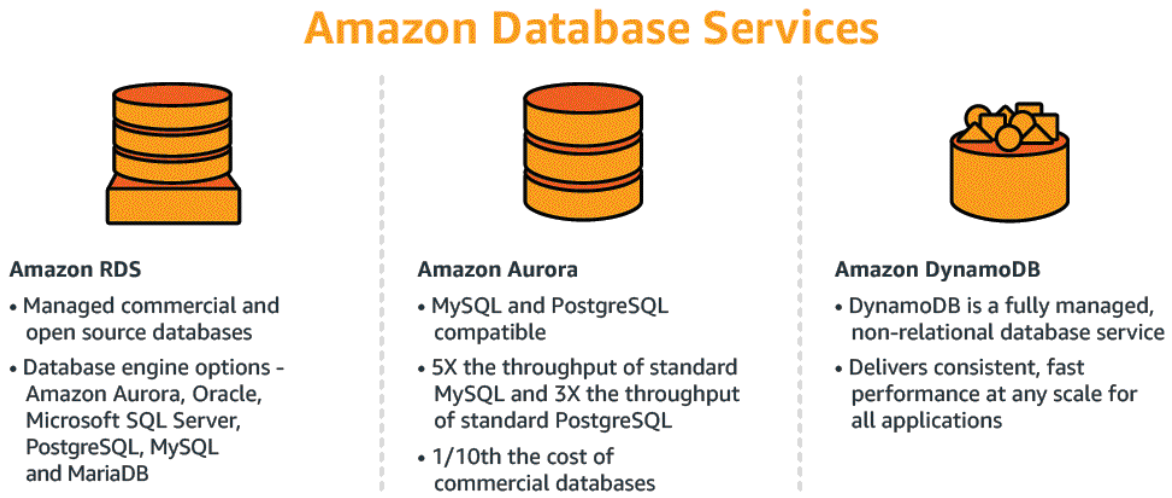
There is a wide array of services that support policies including things like:

- → Amazon S3
- → Amazon SNS
- → Amazon SQS
- → Amazon Glacier
- → AWS Opswork
- → AWS Lambda

There is a big list of services that will enable you to do that. One thing to remember whether we are writing an Iranian policy for a user or for a service we always want to have our focus on least privilege.

16. Introduction to AWS Databases

AWS offers a wide range of database services to fit your application requirements. These database services are fully managed and can be launched in minutes with just a few clicks. AWS provides fully managed relational and no SQL database services as well as in-memory caching as a service and a petabyte scale data warehouse solution.



The 3 main Database services are:

- Amazon RDS
- Amazon Aurora
- Amazon DynamoDB

Amazon Relational Database Service (RDS) makes it easy to set up, operate and scale a relational database in the clouds. RDS provides cost-efficient and resizable capacity while managing time-consuming database administration tasks.

Amazon RDS provides you 6 familiar database engines to choose from. Amazon dynamoDB is a fast and flexible non-sequel database service for all applications that need consistent single digit no second latency at any scale.

It is a fully managed cloud database and supports both document and key values store models. The flexible data model and reliable performance make it a great fit for mobile, web, gaming and tech iOS and many other applications.

16.1 AWS Databases

By the end of this topic, you will understand the following:

- Differences between the NOSQL and SQL database models.
- Understand Relational Database Service (RDS) as well as some of its core concepts like Database Instances, Security Groups and DB parameter groups. And
- Understand DynamoDB and some of its core concepts like the Data Model, Supported Operations and how we provision throughput.

16.2 Data Storage Considerations

When it comes to picking our database, there really is no one size fits all solution. We are going to need to take a look at all the data we plan on story and analyze a couple different factors. Things like :

- The format of the data to be stored.
- How much data do we need to hold in our database.
- What kind of frequency are we going to be querying the database
- How quickly do we need those results to be returned back and
- How long do we need to hold on to the data that's being stored in the database.

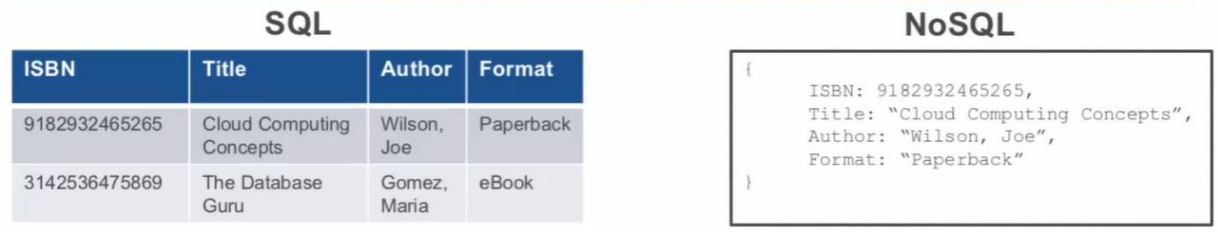
16.3 SQL and NoSQL Databases

Below is a detailed table that demonstrate the primary differences between a SQL and NoSQL database.

	SQL	NoSQL
Data Storage	Rows and Columns	Key-Value
Schemas	Fixed	Dynamic
Querying	Using SQL	Focused on collection of documents
Scalability	Vertical	Horizontal

A SQL database stores all of its data in rows and columns. The rows contain all of the information about an entry and the columns are its attributes. A SQL database schema is fixed. All of those columns must be locked in before we can begin entering data.

We can modify the schema after we have put the database online but it will require taking it temporarily offline for change. The data in a SQL database is queried using SQL. This is for structured query language which can allow for complex queries. SQL databases tend to scale vertically by increasing the amount of hardware power

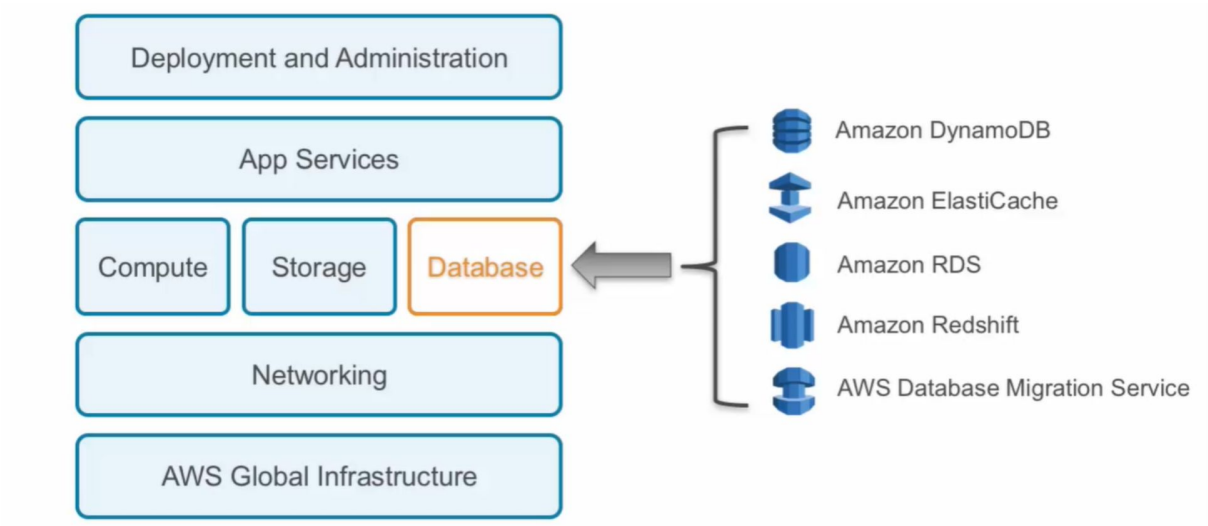


NoSQL databases store their data using one of many storage models including key value pairs, documents and graphs. NoSQL schemas are dynamic or sometimes referred to as schema lists. And information can be added on the fly.

Each row doesn't necessarily have to contain data for each column. Data in a NoSQL database is queried by focusing on collections of documents.

AWS has several managed database services. These services include:

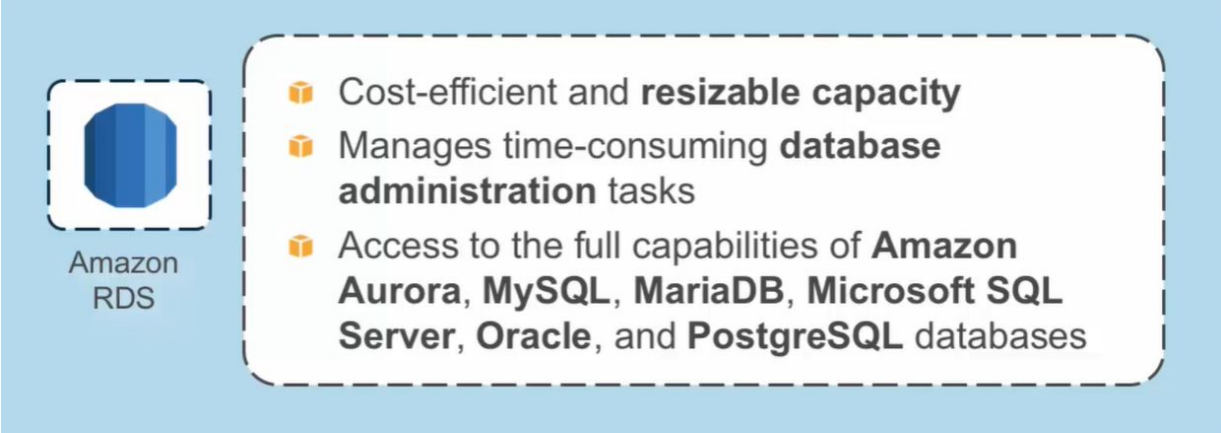
- → Amazon DynamoDB
- → Amazon ElastiCache
- → Amazon RDS
- → Amazon Redshift and
- → AWS Database Migration Service.



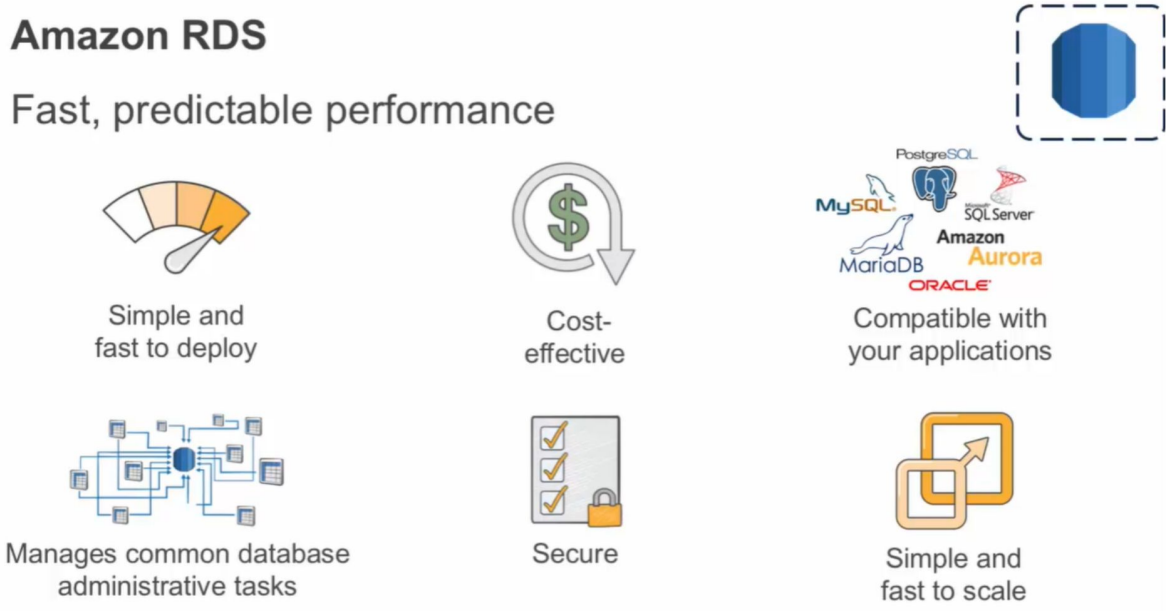
We are going to be diving into Amazon DynamoDB and Amazon RDS.

16.4 Amazon RDS Concepts

Start off with the Amazon Relational Database Service(RDS). RDS makes it really easy to set up, operate and scale relational databases in AWS. RDS provides cost -efficient and resizable capacity, while also managing some of the more time consuming database administration tasks. This is going to free you up to focus on your applications and your business rather than having to run the database server itself.



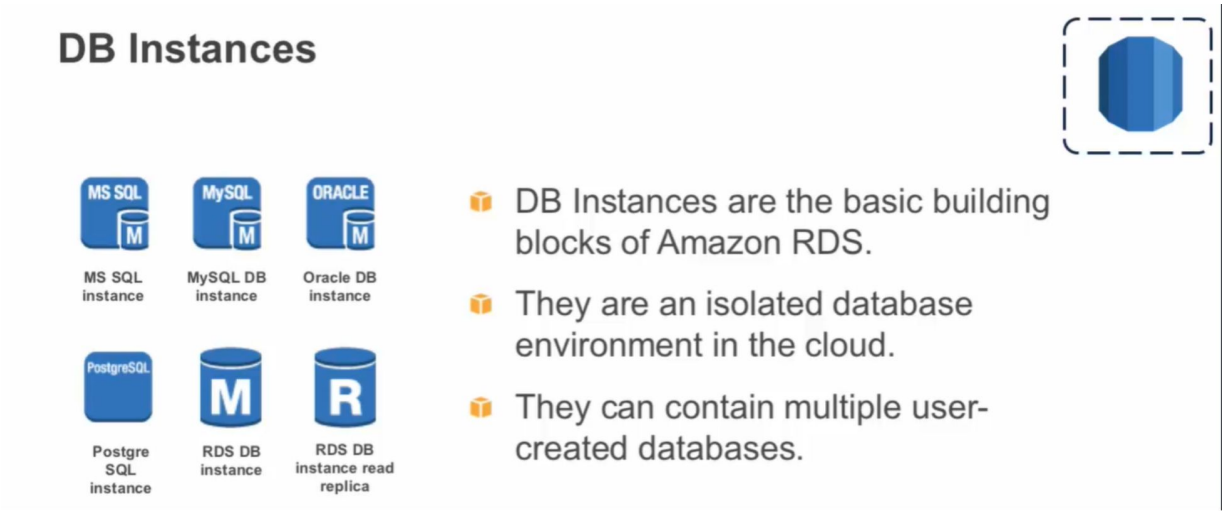
RDS currently supports six different relational database engines. Depending on the engine you select, RDS has some additional replication features available. These features include multiple AZ deployments, read replicas and cross region replication.



Amazon RDS gives us access to the full capabilities of its supported engines. It means that the code, applications and tools that you are already using today with the existing databases can be used with RDS.

The service also automatically patches the database software and backs up our database. Those backups can be used for point in time recovery of our

database. The basic building block for Amazon RDS is the database instance.

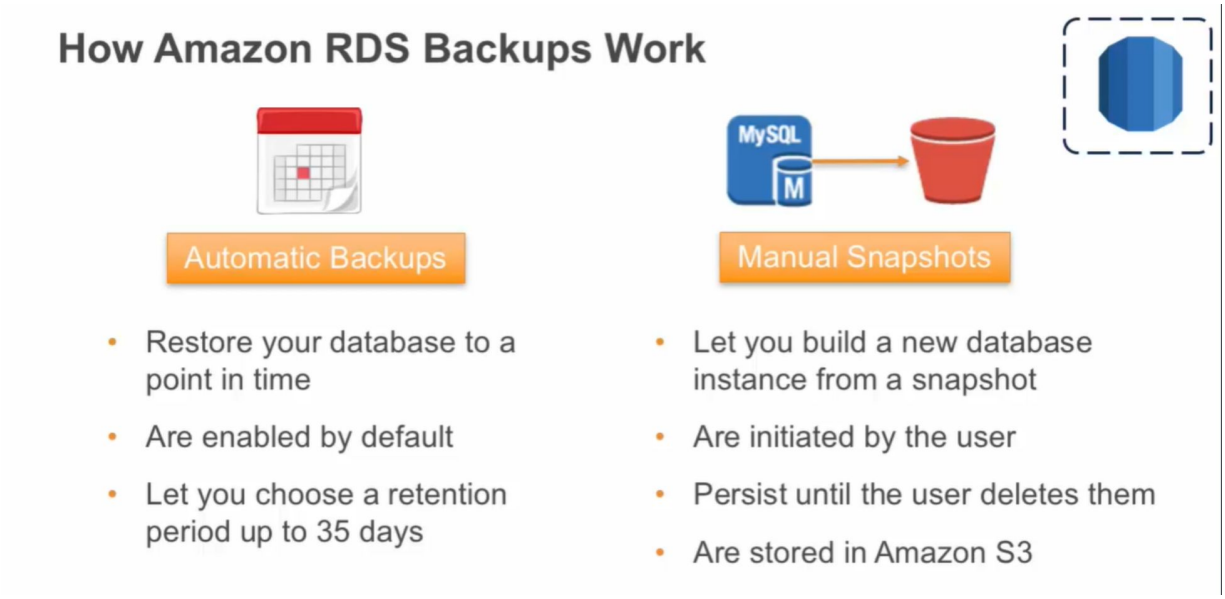


The Database instance is an isolated database environment in the cloud. A database instance can contain multiple user created databases and you can access it by using the same tools and applications that you would use with a standalone database. You can create and modify a database instance using the AWS Management Console, AWS Command Line interface or the Amazon RDS APIs.

16.4.1 Automated Backups

When automated backups are turned on for your database instance, Amazon RDS automatically performs a full daily snapshot of your data. This snapshot is taken during a backup window that you configure. The backup will also contain the transaction logs. When you initiate a point in time recovery, those transaction logs are applied to the most appropriate daily backup in order to restore the database to the specific time you requested.

Automatic backups are only retained for a limited period of time. By default, they are set to only be held for one day but you can configure this retention period for anything up to 35 days.



16.4.2 Manual Snapshots

Amazon RDS already provides the option for manual snapshots. Manual database snapshots are user initiated and enable you to back up your database as frequently as you want. You are then able to restore to that specific snapshot at any time. Unlike automatic backups which have a limited retention period, manual database snapshots will be held until you explicitly delete them.

The snapshots that you take will be stored in S3 so it will be extremely durable.

16.4.3 Cross Region Snapshots

Cross region snapshots are available for all supported Amazon RDS engines. These copies can be moved between any of the public AWS regions and you can copy the same snapshot to multiple regions simultaneously.

Using one of these copies, we can restore our database in a different region.

16.5 Database Security

When it comes to our data and databases, security is very important. RDS provides us many options for managing access to not only the service but the databases as well. The method you use to manage that access is going to depend on that task that needs to be performed in RDS.

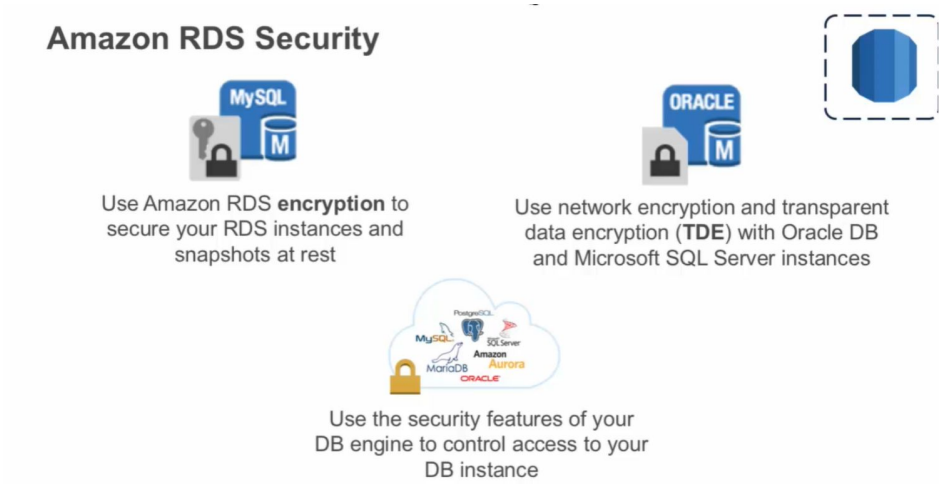
For managing network access, running our database instance in a VPC will provide the greatest possible control. We can use AWS Identity and Access Management policies to assign permissions that determine who is allowed to manage RDS resources.

For example, you can use IAM to determine who is allowed to create, describe, modify or delete database instances. These policies can also control who is allowed to tag our resources as well as modified database security groups.

Security groups are going to control which IP addresses or EC2 Instances can connect to your database. When you first create a database instance, its firewall prevents any database access except through rules specified in the security group.

To protect your communications with your databases, you can use SSL connections to encrypt all of the traffic.

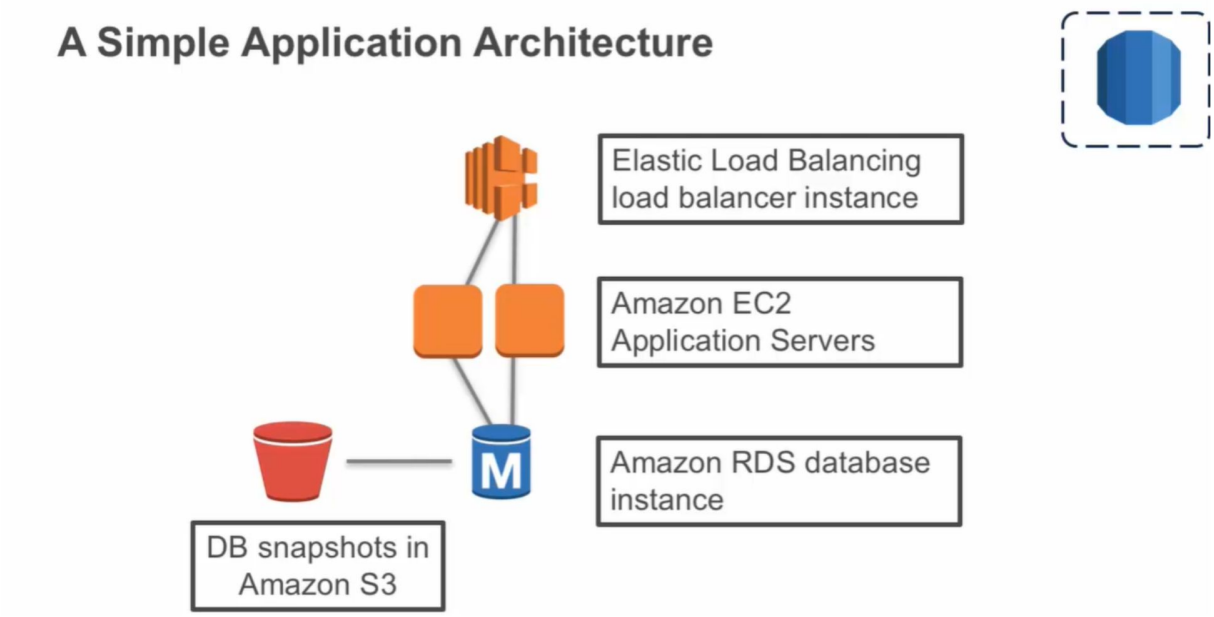
Amazon RDS provides us the option to encrypt our databases. When you enable encryption, all data at rest for not only the database, but any snapshots taken from that database will be encrypted.



And finally, we can use the security features of the database engine to control who can log into the database on the database instance. Just like you would if the database was running on your local network.

16.6 Amazon RDS Architecture

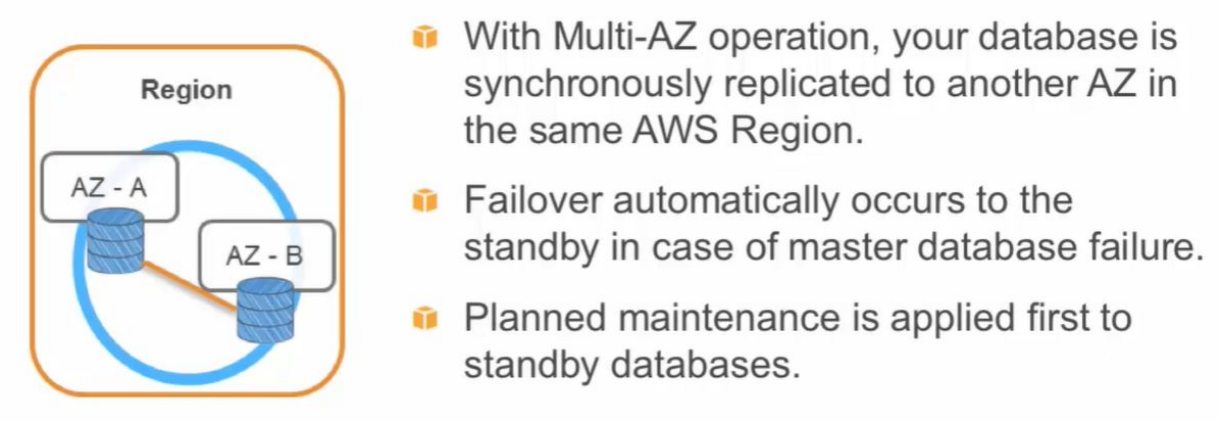
The sample architecture on this slide shows a simple application stack with an application running on an Amazon EC2 instance supported by a master database running on RDS. By presenting the application behind Elastic Load Balancer, we are allowing for the future use of other scaling features like Auto Scaling.



Amazon RDS multi AZ deployments provide enhanced availability and durability for your database instances, making it a natural fit for any production database workloads. When you provision a multi-AZ database instance, RDS automatically creates a primary database instance and then synchronously replicates the data to a standby instance in a different Availability Zone.

As we had discussed in the infrastructure topic, Each AZ runs on its own physically distinct and independent infrastructure and is engineered to be highly reliable. In the case of an infrastructure failure, things like the underlying instance hardware fails or storage failure or even network disruption.

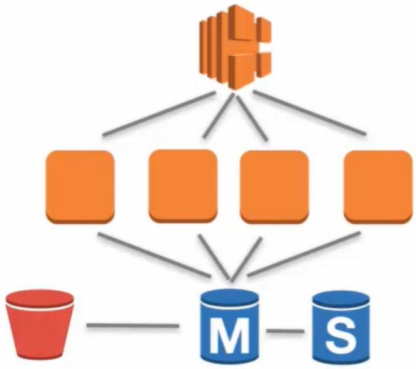
Amazon RTX performs an automatic failover to the standby so that you can resume database operations as soon as the failover is complete. Since the end point for your database instance remains the same after failover, your application can resume database operations without the need for any manual intervention.



When it comes time for patching or applying upgrades, any planned maintenance activities are performed on our standby instance first.

Building on our original architecture example, we will add in some additional E.C to instances and bring in a new standby database. In the event we have an outage with our master database, RDS will take a standby instance and move it into the master role. It will then launch a new standby instance to replace the old one.

A Resilient, Durable Application Architecture



Elastic Load Balancing load balancer instance

Application, in Amazon EC2 instances

Amazon RDS database instances:
Master and Multi-AZ standby

DB snapshots in Amazon S3

16.7 Database Parameters

We do not have direct access to the engine running under RDS, we do have the capability of customizing the configuration that it is using. The first of these options is parameter groups.

16.7.1 Parameter Groups:

The Parameter Groups will contain many of the options that we might find if we were standing up the database on our own. Each supported engine and its available versions will contain its own set of available parameters.
When we create a parameter group, that group can be reused with any other RDS instances that are leveraging the same engine and version of that engine.

- → Contains engine configuration values that can be applied to one or more DB instances of the same instance type.
- → Amazon RDS applies a default DB parameter group when you create a DB instance, which contains defaults for the specific database engine and instance class of the DB instance.

16.7.2 Option Groups:

Some engines support a second method of customizing the configuration leveraging what are called option groups. These options can contain things like enabling and configuring mem-cache D for our MySQL instance or

enabling transparent data encryption for our Oracle instances. Each supported engine will have its own set of available options.

→ Tools that simplify database management.
→ Currently available for Oracle, Microsoft SQL Server and MySQL 5.6 DB instances.

16.8 Amazon RDS Best Practices

Here are some of the best practices when working with RDS.

- We always want to be monitoring our memory, CPU and storage usage of the service to make sure that our databases are operating efficiently.

- By using multi AZ deployments, we are adding in that high availability and automatic failover in the event that any problems occur.

- We want to make sure that automatic backups have been enabled.

- The backup window for those automatic backups is set to occur during a low period in the right volume to our database.

- When it comes time to scale the I/O capacity of our database, there are a couple of options that we can go through.
  → The first is to migrate our database instance to a class with higher I/O capacity.

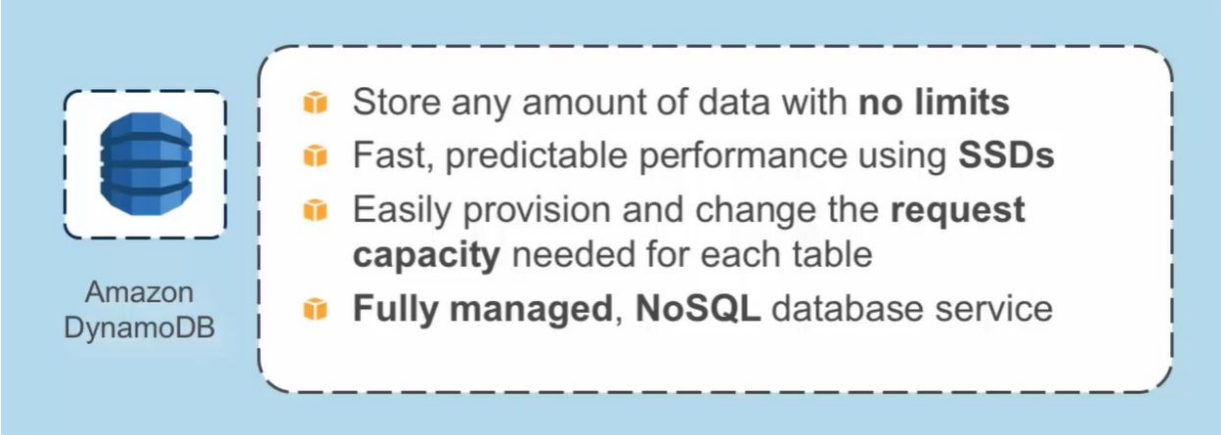
- → We can convert from standard storage to provisioned IOPS volume and
- → If we are already using provisioned IOPS volumes, we can provision it for even greater throughput capacity

- If our clients are doing any DNS caching or holding onto the IP addresses for DNS entries, we need to make sure that we have a TTL of less than 30 seconds set. This is going to come into play when we have a failover event from a master to a standby as these changes are made at a DNS level.

- Finally, it is very important to test the failover of your database instance. without testing, you are running the risk that it fails over if the event might not happen the way you have planned it to.

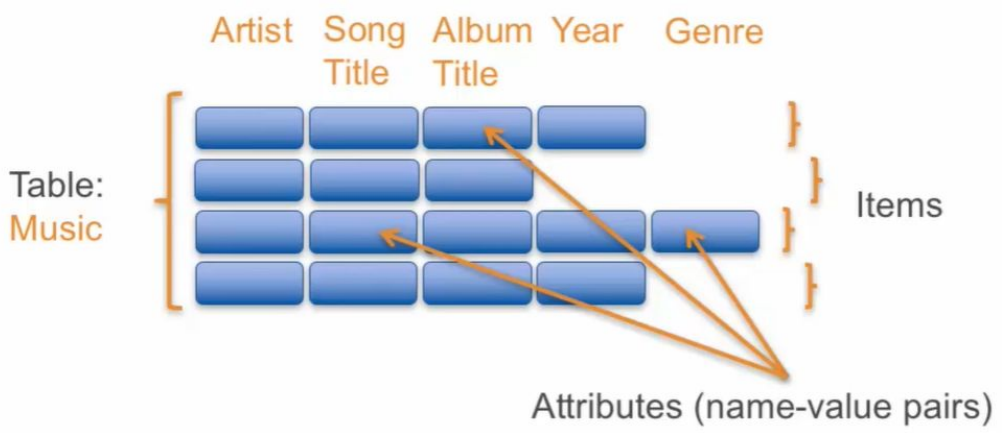
17. Amazon DynamoDB

DynamoDB is a highly performant, scalable, fully managed, NoSQL database service. The databases that are created in Dynamo, do not have any limit on their size. The back end storage is all being performed by solid state drives, so they are going to be highly performant.



With Dynamo, we are not actually picking a specific instance size like we do with RDS, instead we provision capacity units for read and write depending on what we need. In all of the capacity, scaling and performance, is being handled for us by AWS in a fully managed service.

Observe the DynamoDB data model. In Amazon DynamoDB, a table is a collection of items and each item is a collection of attributes. Each attribute is a name value pair and can contain a single value, a Jaison document or a set of values.



When you create your table, in addition to the table name you must specify a primary key to use on the table. As with other databases, a primary key in DynamoDB uniquely identifies each item in the table. So no two items can have the same key. When you add, update or delete an item in the table, you have to specify the primary key value for that item.

DynamoDB supports two different kinds of primary keys:

- Partition Key
- Composite Key

17.1 Partition Key

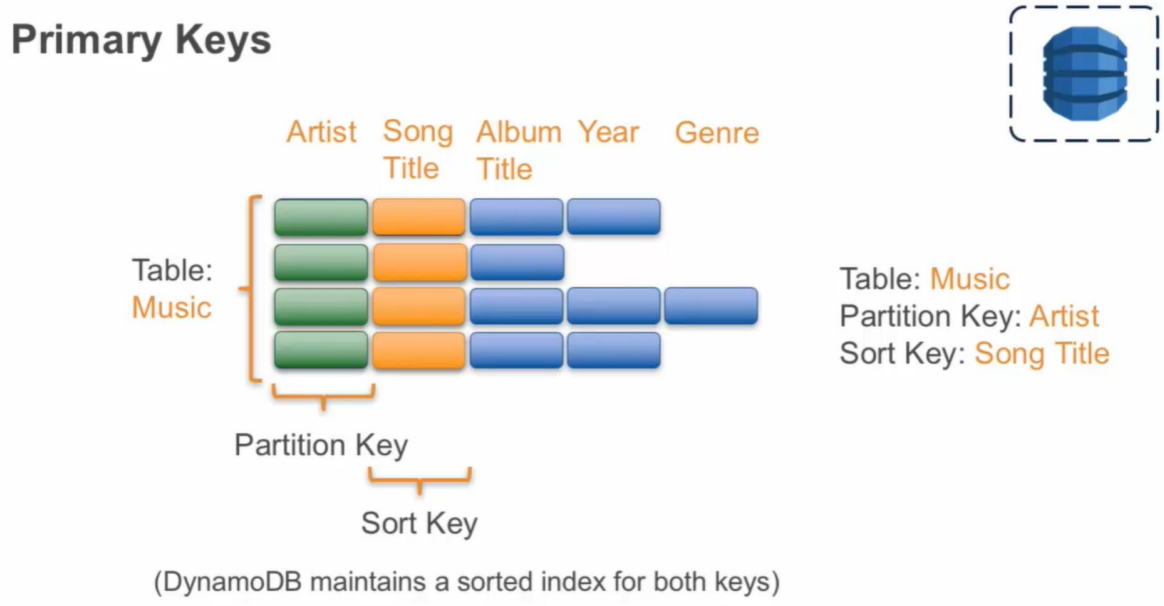
The first is a partition key. Partition key is a simple primary key composed of just one attribute known as the partition key. DynamoDB uses the partition key's value as input to an internal hash function.

The output from the hash function determines the partition where the item is stored. No two items in a table can have the same partition key value.

17.2 Composite Key

The second type of primary key is a composite key using both a partition key and a sort key. The first attribute is the partition key and the second attribute is a sort key. DynamoDB uses the partition key value as input to an internal hash function.

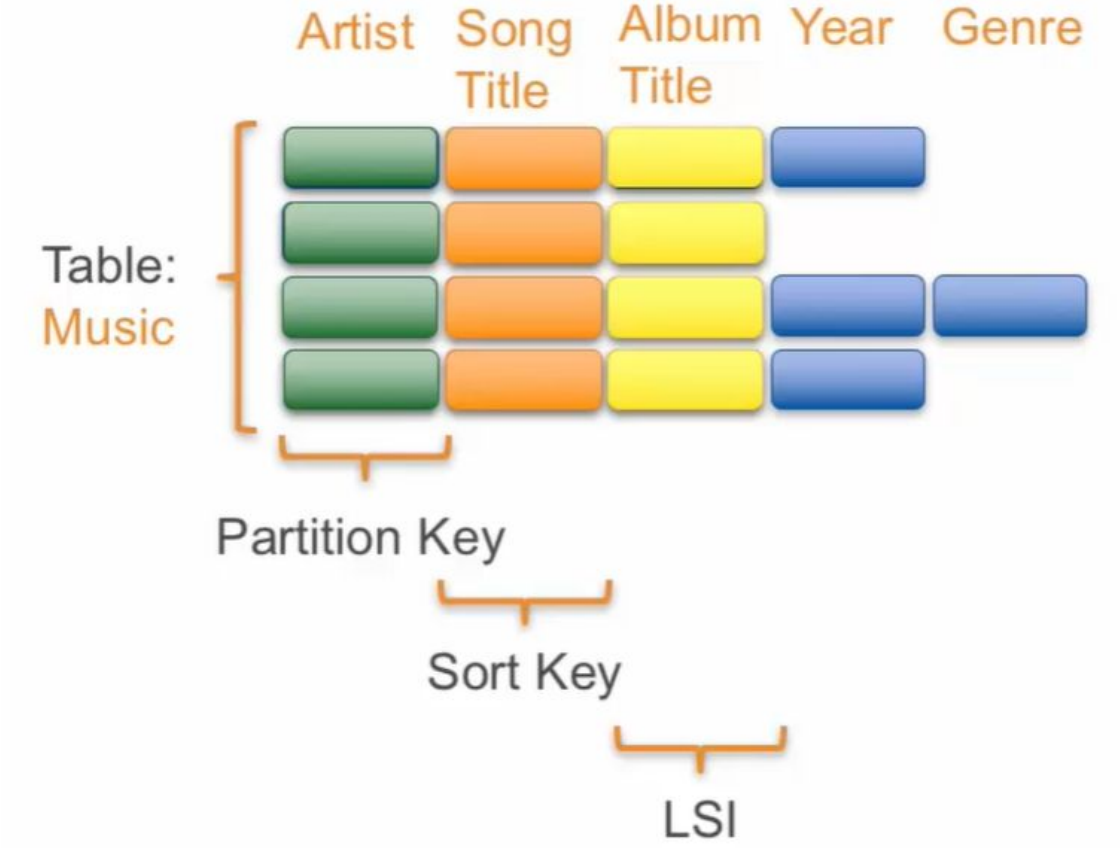
The output from the hash function determines the partition where the item is stored. All items with the same partition key are stored together in a sorted order by the sort key value. It is possible for two items to have the same partition key value but those two items must have different sort key values.



So for our sample table, we have a table name of music, a partition key of the artist and our sort key is the song title.

17.2.1 Local Secondary Index (LSI)

In DynamoDB, each table can have up to five Local Secondary Indexes (LSI). These local secondary indexes come into play when we want to query a non key value from our database. And what we mean by non-key value is not on a partition key, not our sort key. So if we wanted to query my table based on the album title we could create a local secondary index with the album title as a key.



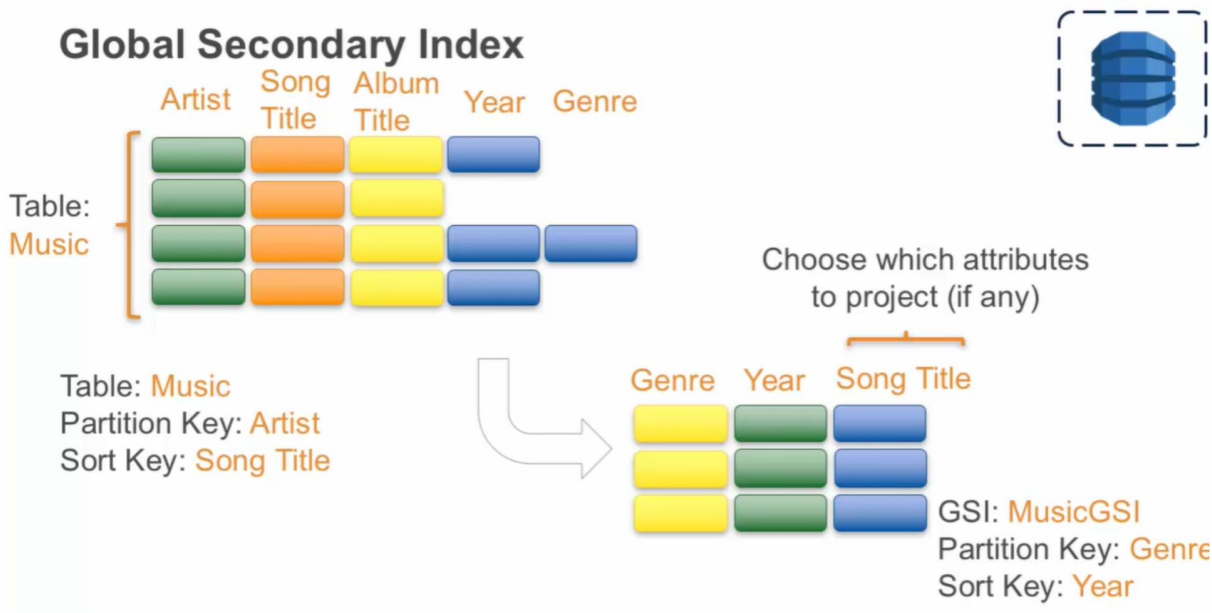
The local secondary index key axes our new sort key. So we are going to have the same partition key as the original table.

17.2.2 Global Secondary Index (GSI)

A Global Secondary Index is an index with a partition key and sort key. They can be different from those on the primary table. They can be thought

of as pivot charts for our tables.

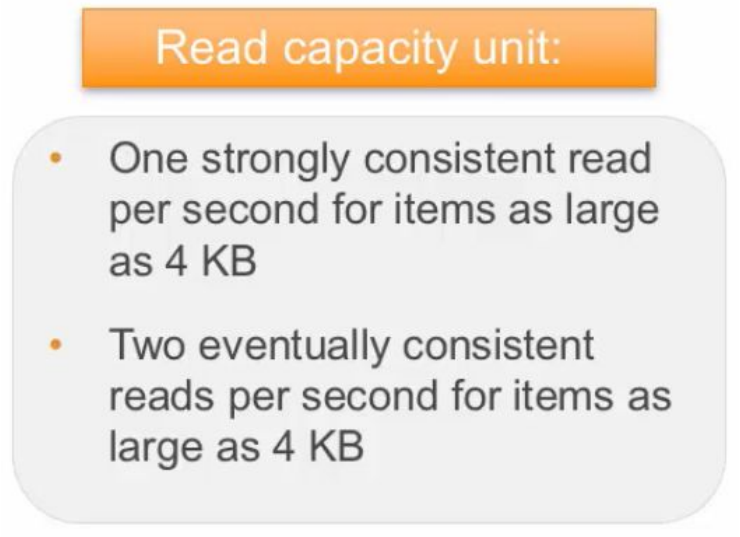
So in our primary table called music as shown in the figure, we have a partition key of the artist in a sort key of the song title. With our global secondary index, we can move over to a partition key of genre in a sort of year.



17.3 Supported Operations

When you create or update a table, you specify how much provision throughput capacity you need for reads and writes. Amazon DynamoDB will automatically allocate the necessary machine resources to meet your

throughput needs while also ensuring consistent low latency performance.



This throughput is measured in capacity units. A unit of read capacity represents one strongly consistent read per second or two eventually consistent reads per second for any item as large as 4 KB.

A unit of write capacity, represents one write per second for items as large as 1 KB.

DynamoDB supports two operations.

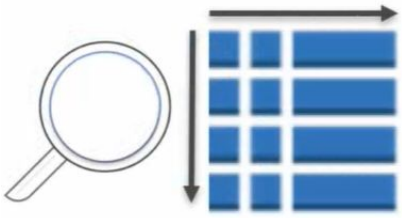
- Query
- Scan

Query

A query operation enables you to query a table using the partition key and an optional sort key filter. If the table has a secondary index, you can also query the index using its key. A query is going to be the most efficient way to retrieve items from a table or a secondary index.

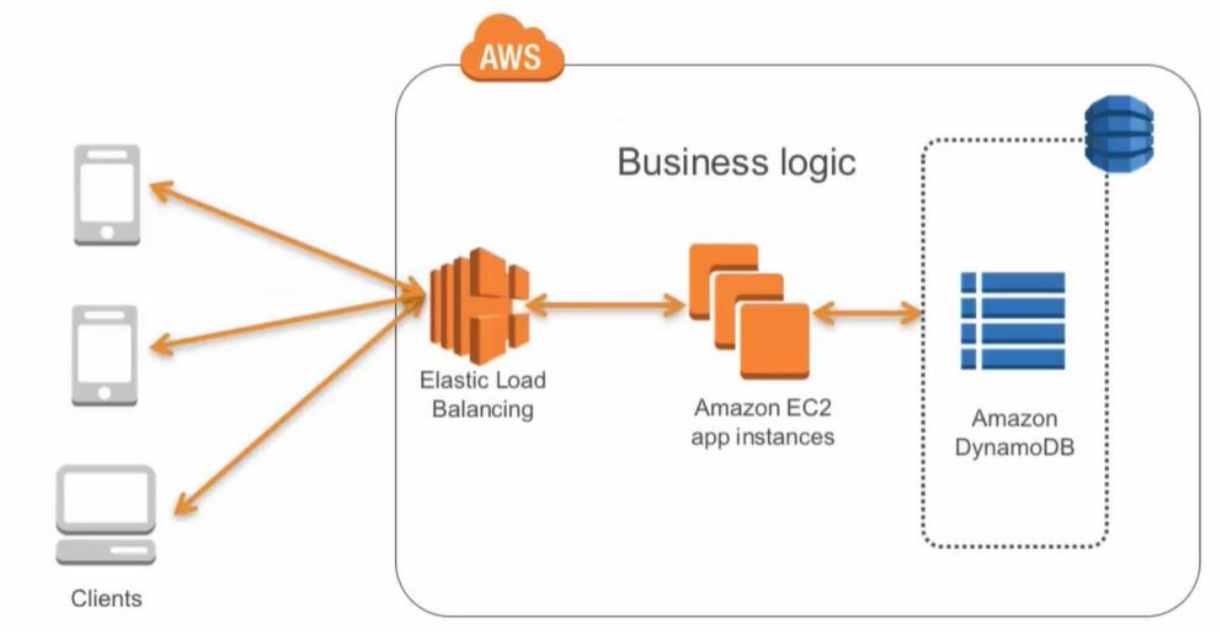
Scan

Scan can be used with both tables and secondary indexes. The scan operation is going to read every item in the table or secondary index. So for tables and secondary indexes that are very large, a scan can consume a large amount of resources. For this reason, we recommend that you design your applications so that you can use the query operation mostly and only use scan where appropriate.



You can also use conditional expressions in both the query and scan operations to control which items are returned.

Similar to our reference: RDS infrastructure, this figure shows a simple application architecture using DynamoDB to store data processed by applications on Amazon EC2 instances.



17.4 Amazon RDS and Amazon DynamoDB

As you already know, one type does not fit all solutions. The choice depends on several factors and you can use both relational and no sequel databases in one application depending on the requirements. This table is a great side by side comparison of Relational or Non-Relational databases and can be a good source of reference for you.

Factors	Relational (Amazon RDS)	NoSQL (Amazon DynamoDB)
Application Type	• Existing database apps • Business process–centric apps	• New web-scale applications • Large number of small writes and reads
Application Characteristics	• **Relational** data models, transactions • **Complex** queries, joins, and updates	• Simple data models, transactions • Range queries, simple updates
Scaling	Application or **DBA–architected** (clustering, partitions, sharding)	**Seamless, on-demand scaling** based on application requirements
QoS	• Performance–depends on data model, indexing, query, and storage optimization • Reliability and availability • Durability	• Performance–**Automatically optimized** by the system • Reliability and availability • Durability

AWS provides a number of database alternatives for developers. You can run fully managed relational and NoSQL services or you can operate your own database in the cloud on Amazon EC2 and Amazon EBS.

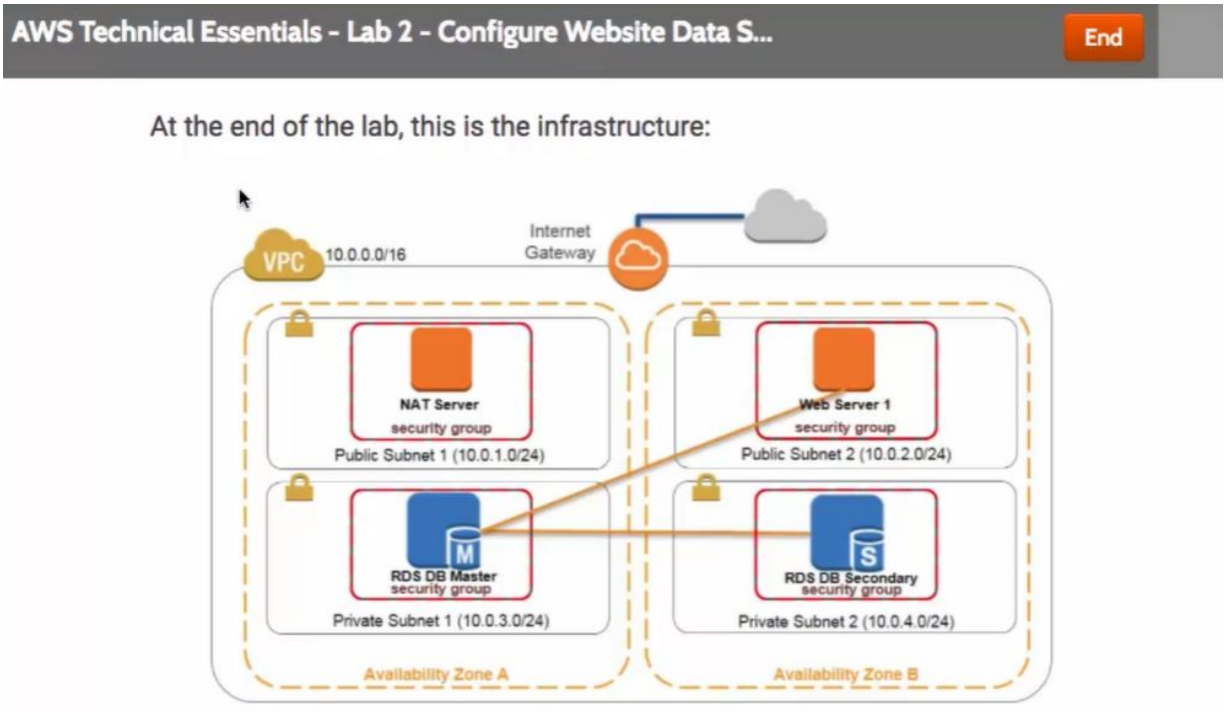
If you need a Relational database service with minimal administration, consider using Amazon RDS. If you need a fast, highly scalable NoSQL database service, consider using Amazon DynamoDB. If you need a Relational database, you can manage on your own. Consider it using your choice of Relational AMIs.

If You Need	Consider Using
A relational database service with minimal administration	**Amazon RDS** • Choice of Amazon Aurora, MySQL, MariaDB, Microsoft SQL Server, Oracle, or PostgreSQL database engines • Scale compute and storage • Multi-AZ availability
A fast, highly scalable NoSQL database service	**Amazon DynamoDB** • Extremely fast performance • Seamless scalability and reliability • Low cost
A database you can manage on your own	Your choice of **AMIs** on Amazon EC2 and Amazon EBS that provide scale compute and storage, complete control over instances, and more.

17.5 Demo: Build Your Database Server

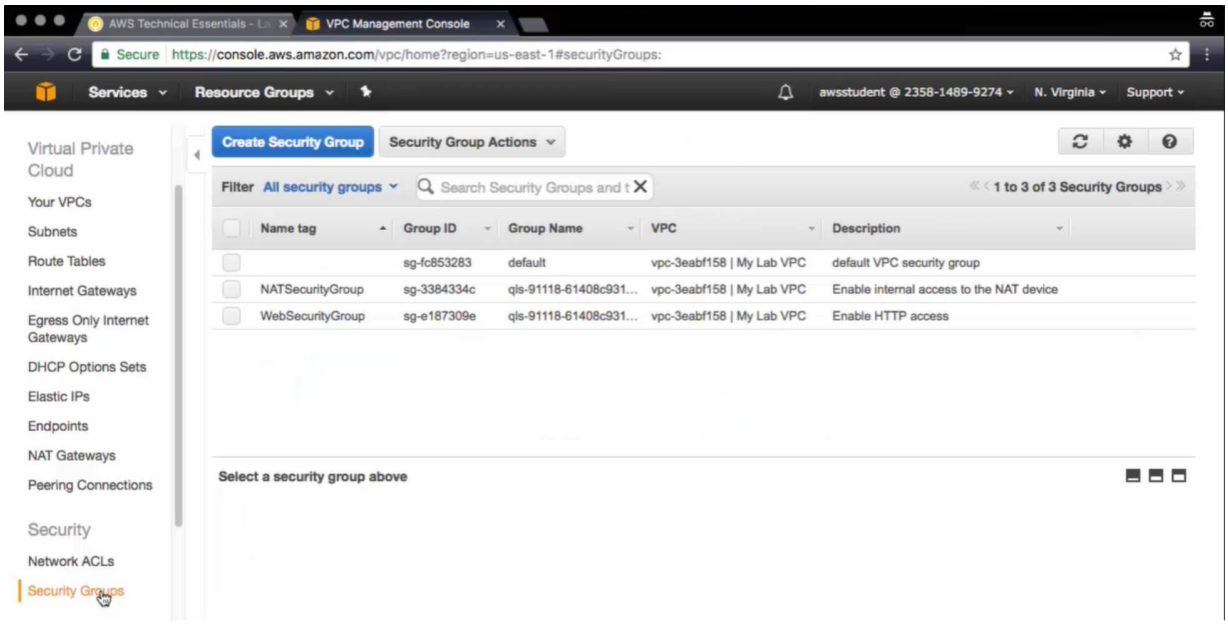
17.5.1 Lab #2: Configure Website Data Server

In Lab #2, we are going to be configuring a website datastore and this lab is going to be building on our previous Lab.

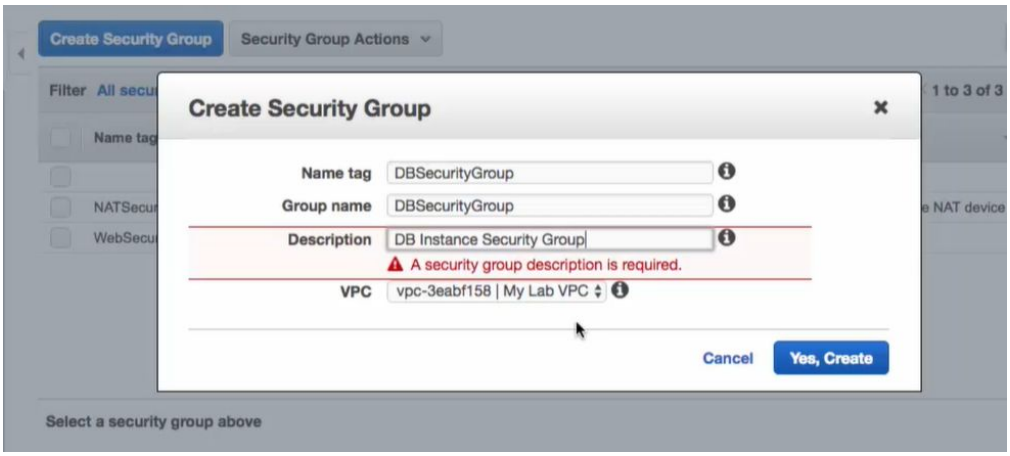


We can see that we still have our web server as it was configured in the previous Lab but we are also going to be adding a new RDS database in a multi AZ configuration. So back in the Management Console, we are picking up where we left off. And one of the first things we need to do is configure a new security group.

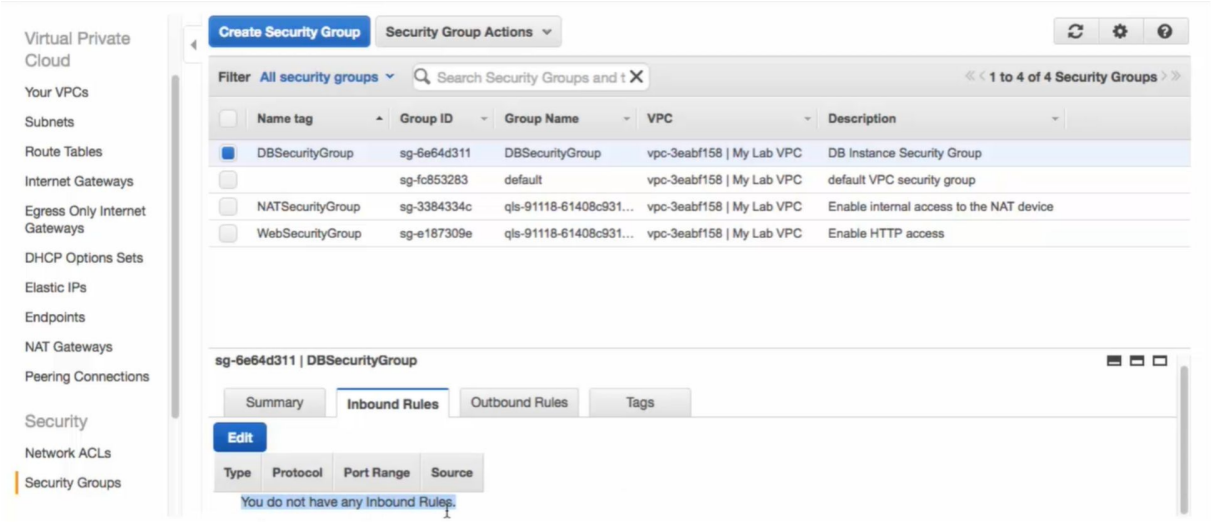
If you remember, the only security group that we created was to allow HTTP access to our web server. Now that we are going to be launching in our RDS instance, we are going to need to be able to open up ports so that something can talk to our database.



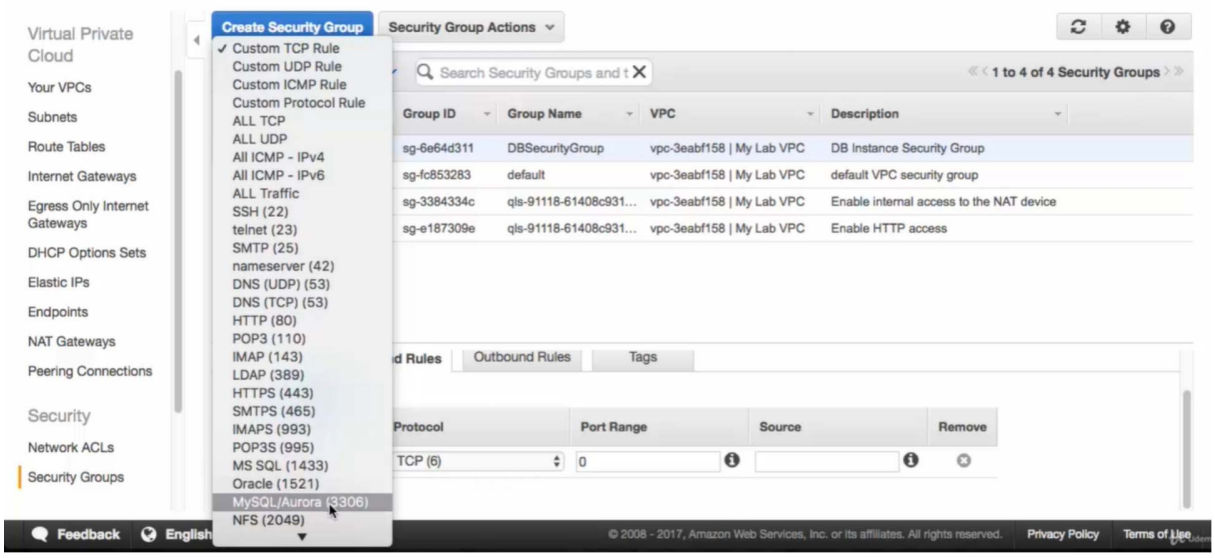
To create a security group, We will go ahead and click on create security group at the top and for our name tag. This is going to be our DBSecurityGroup, with a description of DB Instance Security Group.



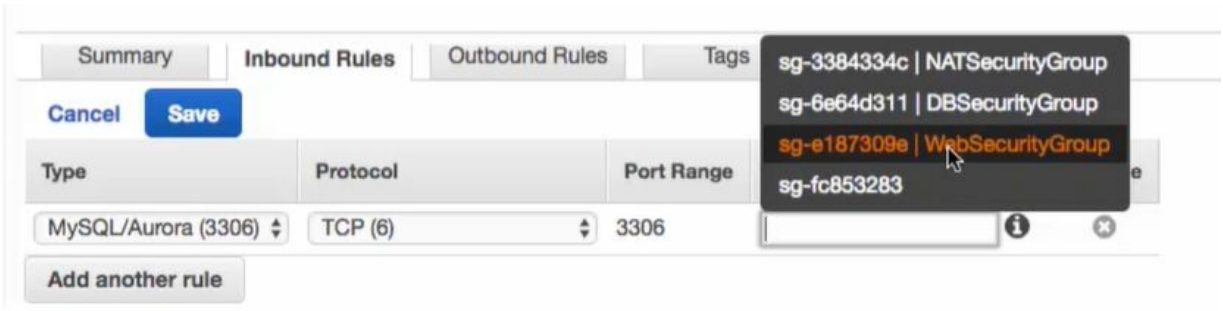
Just like the last lab, we are going to make sure that we have selected my lab at the PC and then click Create. Our security group is going to have no default inbound rules which means that there are no inbound packets.They are going to be allowed to talk to anything with the security group.



So we need to go ahead and open up some ports and for our ports if we scroll down, we are looking for MySQL/ Aurora 3306.

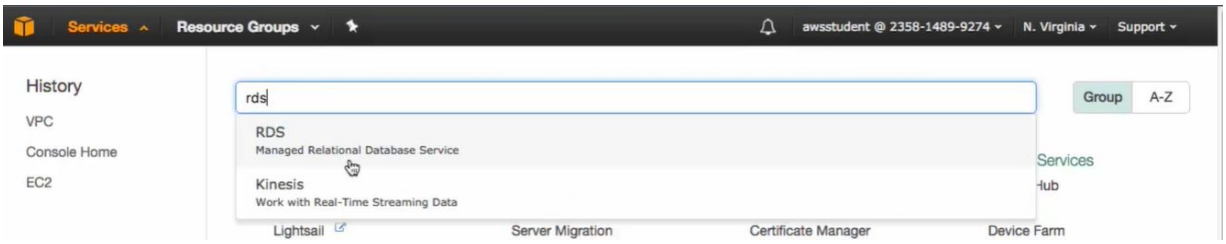


Our source for this is actually going to be our web security group. One of the features of security groups is that we can reference other security groups in the VPC.

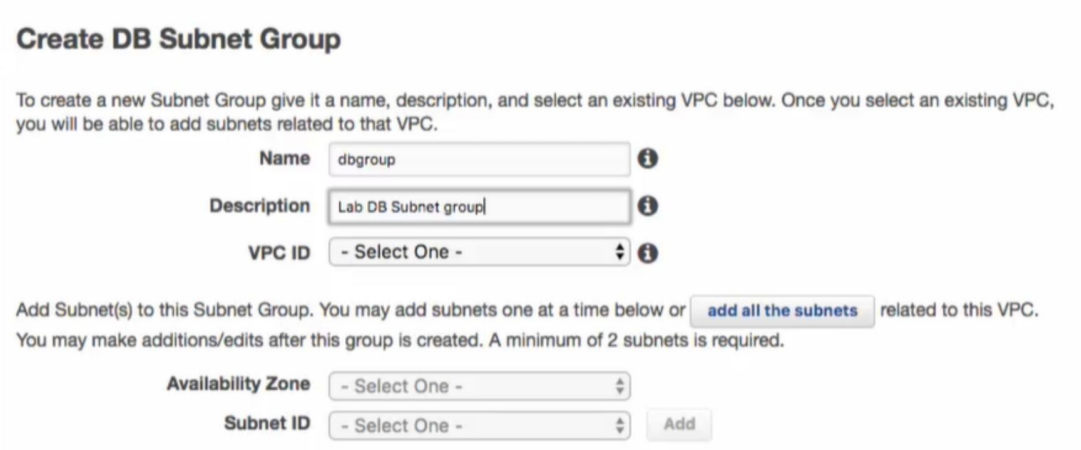


This allows us to have a lack of knowledge of what the underlying IP addresses are and just be able to target specific security groups from other instances. We will now go ahead and click Save.

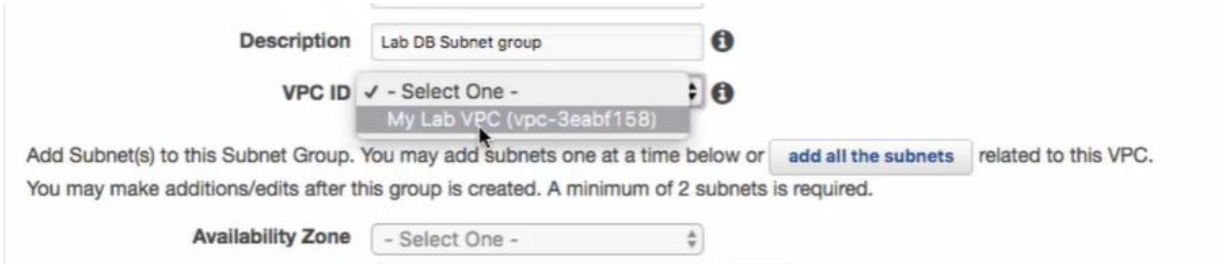
Now that we have our security group in place, the next thing we need to do is start configuring RDS. So we are going to go back up to services at the top of the screen. Type in RDS and go ahead and click.



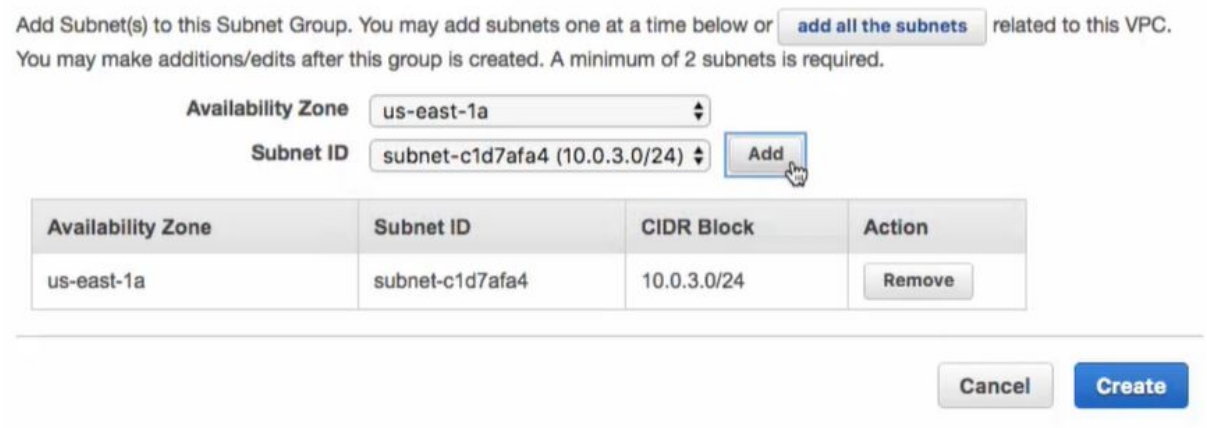
The first thing we need to configure in RDS is our subnet group. The subnet group essentially tells the RDS service which subnets it is allowed to launch its instances inside of. So we are going to create a DB subnet group here and we are just going to call this DB group with a description of the Lab DB Subnet group.



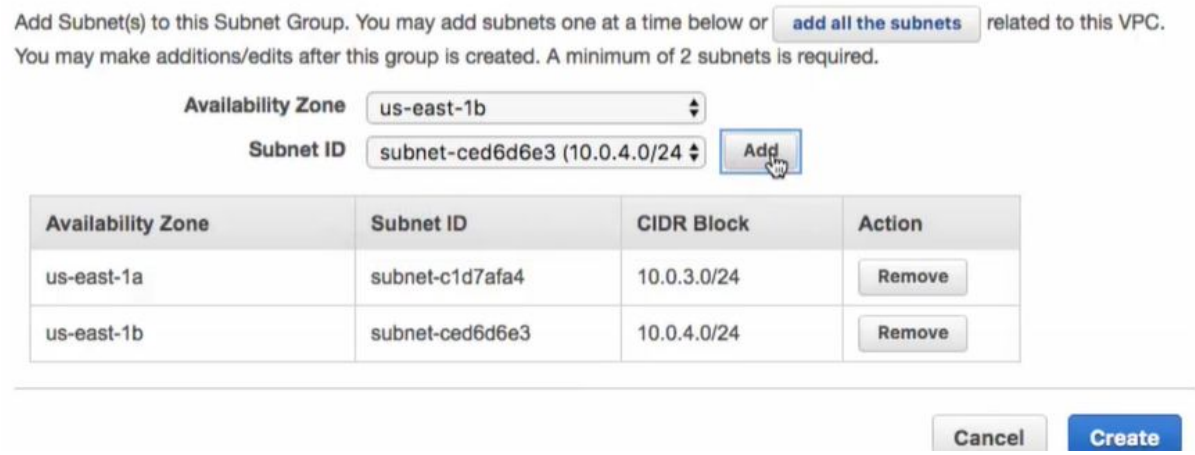
For the VPC, we do need to make sure we select My Lab VPC and then we need to add some subnets.



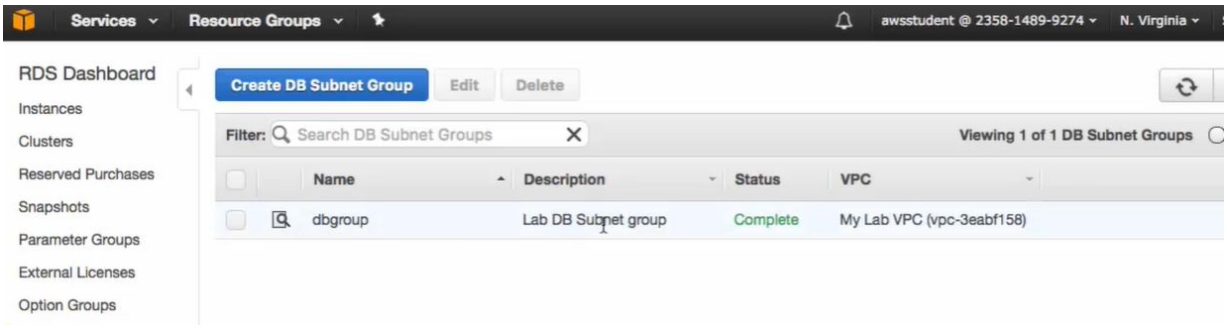
So first, we are going to pick us-east-1a and the second subnet in the list which is 10.0.3.0/24 which is private subnet number one and will add that to our list.



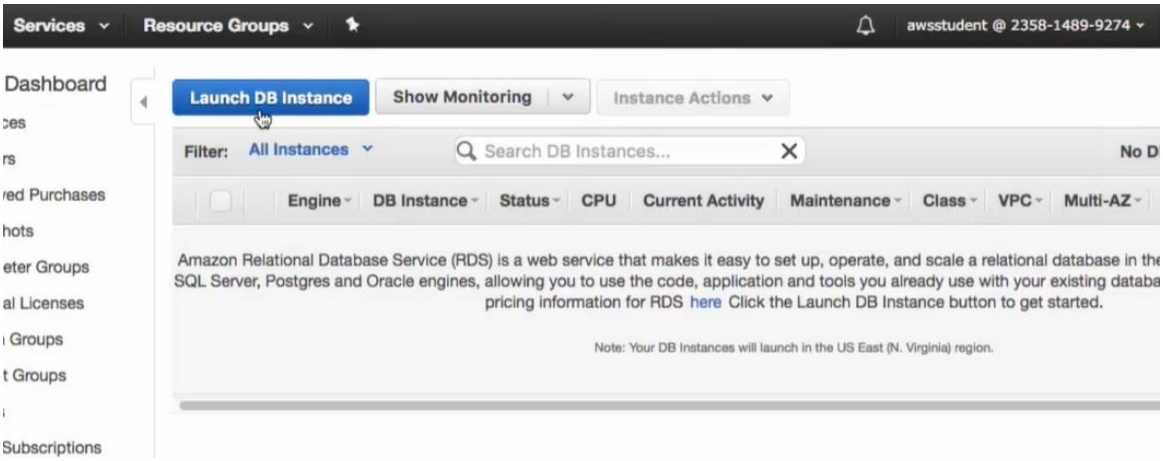
And then we are also going to select from us-east-1b and Subnet ID: 10.0.4.0/24 which is private subnet number two. We will Add that in and click Create.



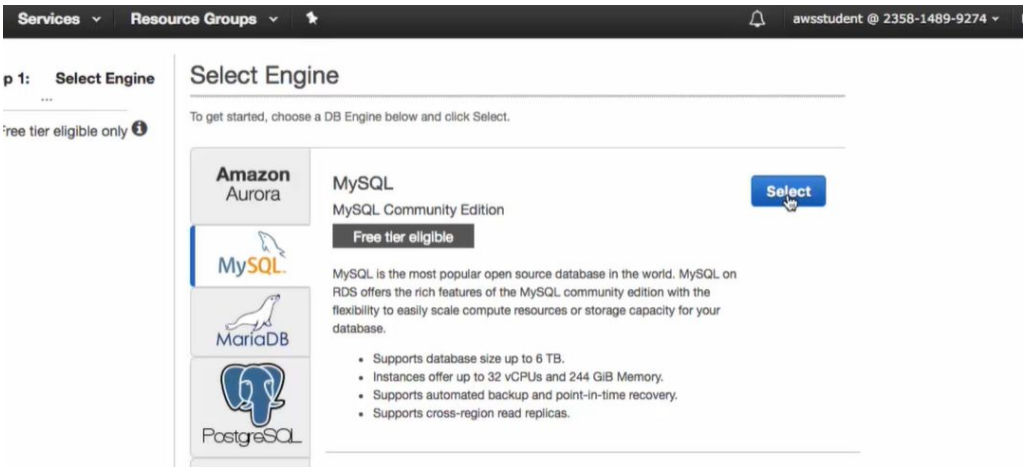
If we refresh our console, we can see we have a Lab DB Subnet group.



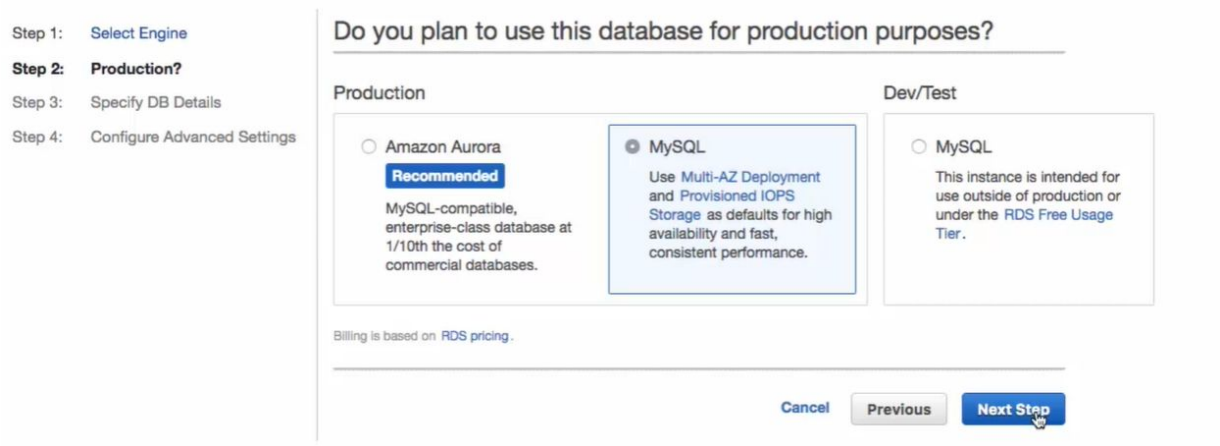
Now that we have our subnet group created, we are ready to launch our instance. So we will go over to instances here on the left hand side and click on Launch DB instance (below figure).



The type of instance we are going to launch today is a MySQL server. So once we have selected that we can click on Select as shown in the below figure.

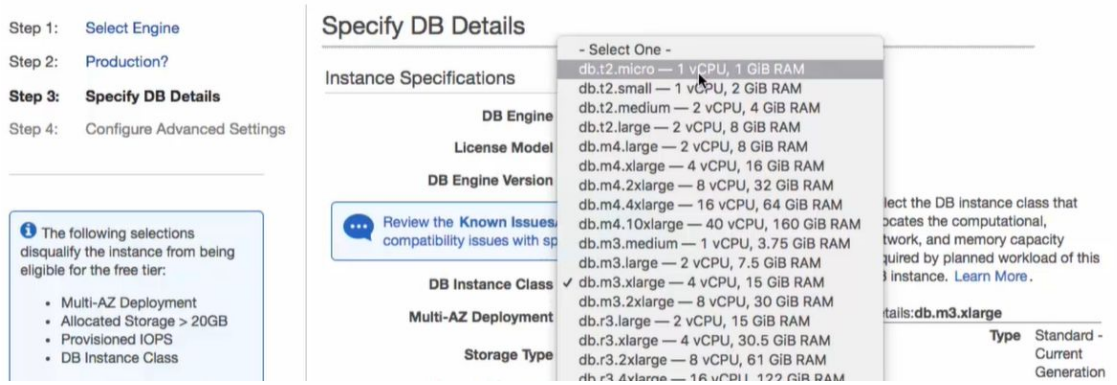


Under Production, we are going to select MySQL for the Engine and click on Next Step.



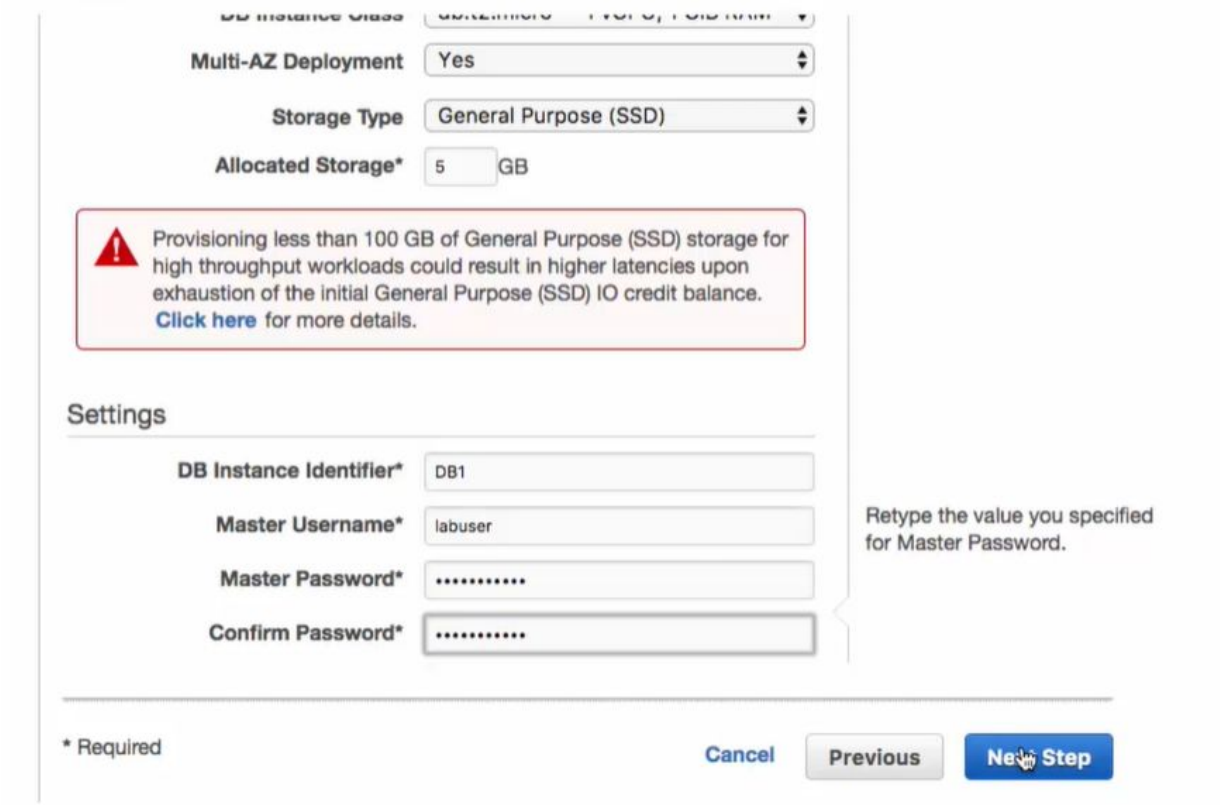
On our Specified DB Details page, we are actually going to leave the License Model and DB Engine Version.

We are going to change the DB Instance Class to the first entry in our list which is a db.t2 micro.

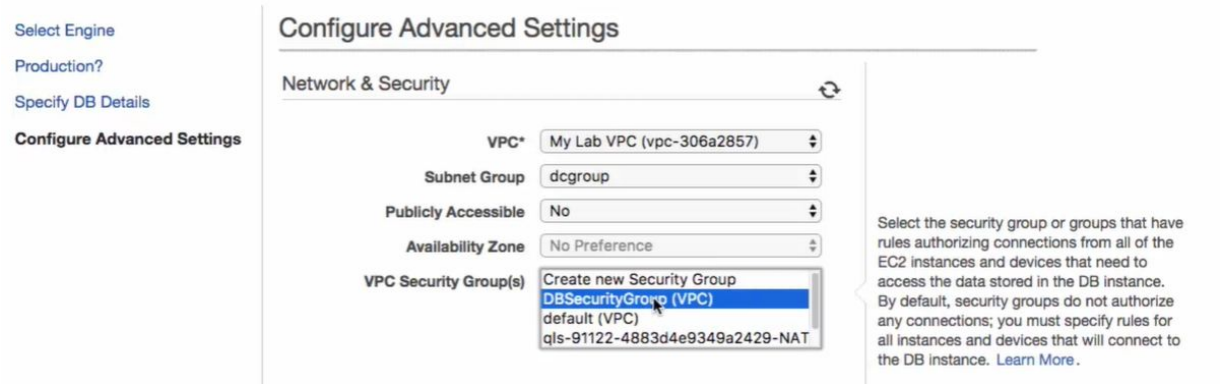


This is going to be a multi AZ deployment and we do not need to change this storage type or allocate storage.

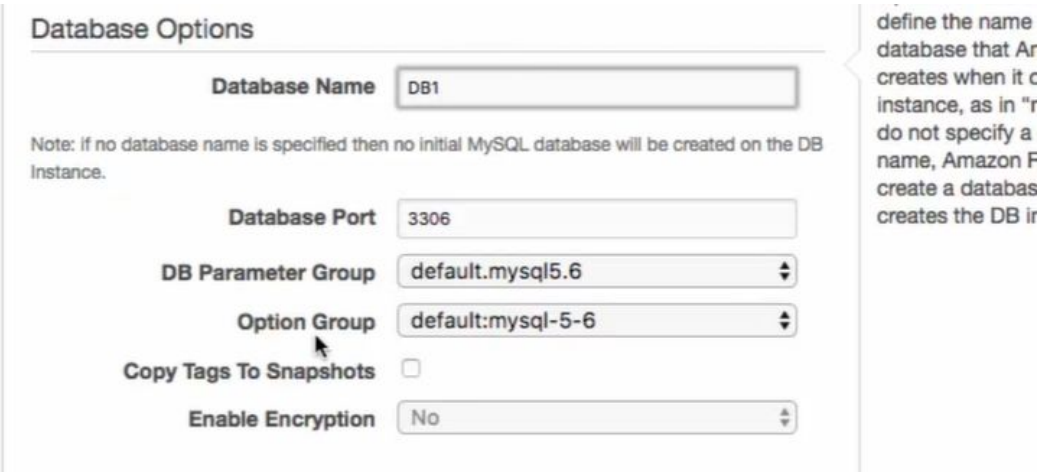
For our DB Instance Identifier, we are just going to enter DB1. This just needs to be a unique name across all of our AWS account owned resources for this database instance. For our Master Username, we are going to use labuser. Set your master password of the lab password and after confirming the lab password, we can click Next Step.



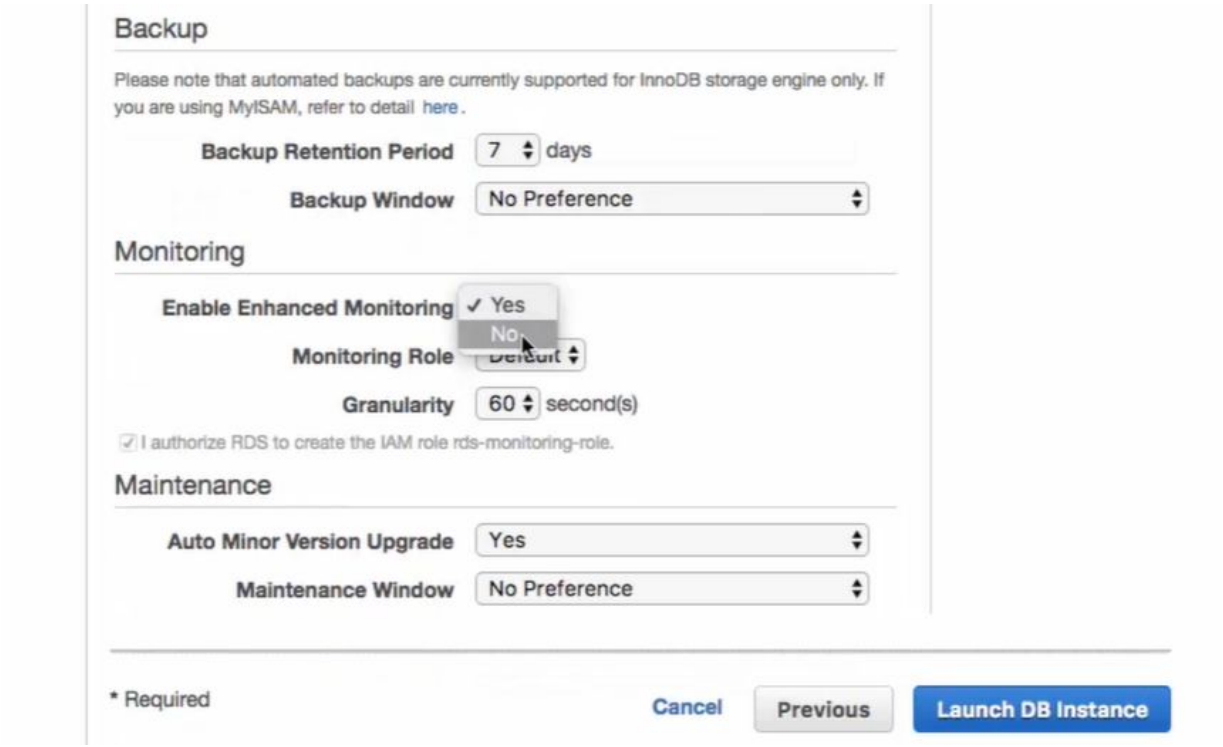
Under our Advanced Settings, you go to the VPC and make sure we select My Lab VPC and we should see the Subnet Group that we created previously. This is not going to be a publicly accessible database instance. And for our security groups, we are going to make sure to select the DBSecurityGroup which we created earlier.



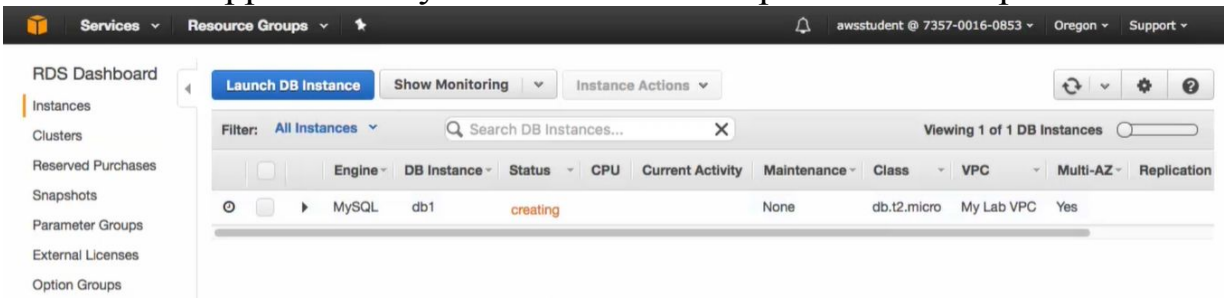
For a database name, we are just simply going to call this DB1 and we can leave most of the rest of the settings to their default state.



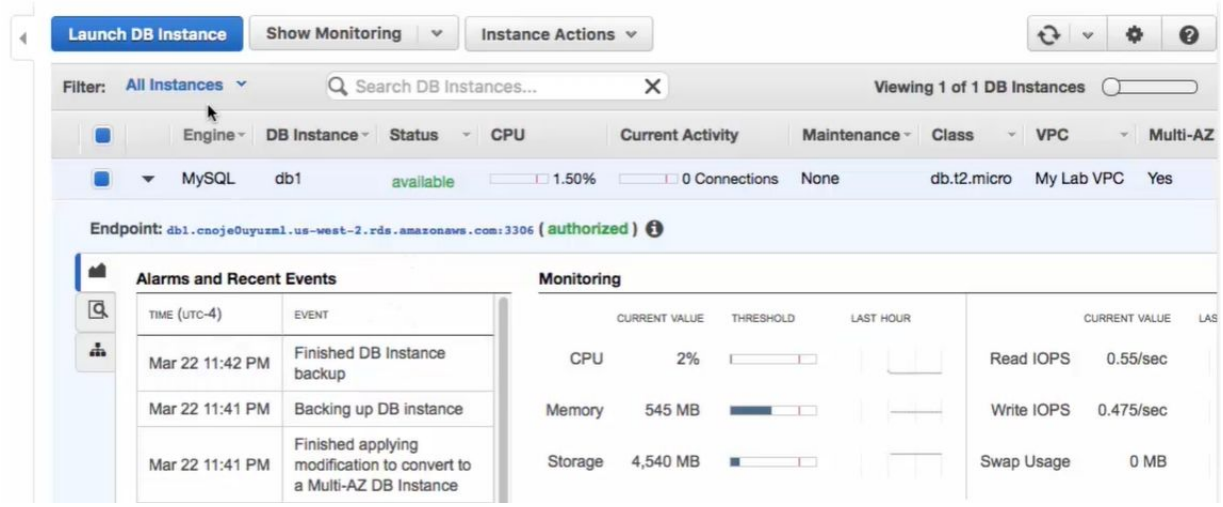
For Enhanced Monitoring for this lab, we are going to set it to No and then we click on Launch DB Instance.



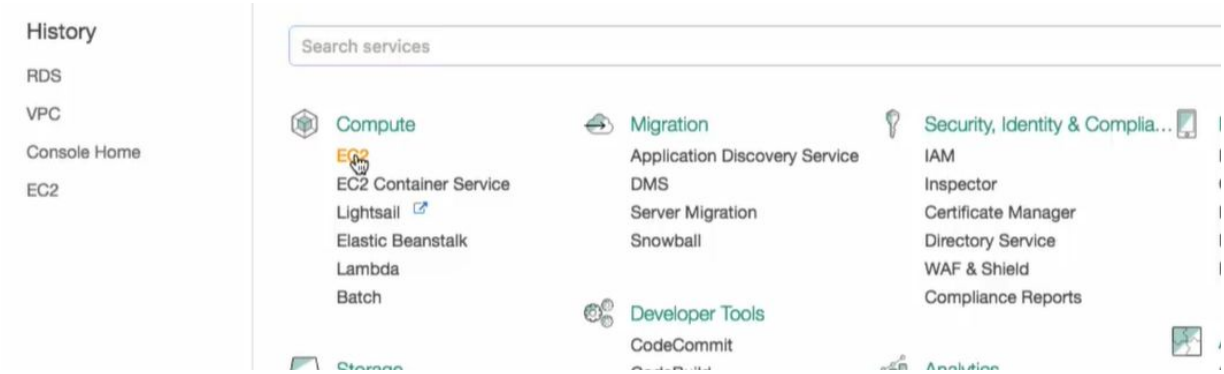
At this point our database instance is being created. And if we go back over to view our Database Instance we can see that it is in creating state. We will have to wait approximately 10 minutes for this process to complete.



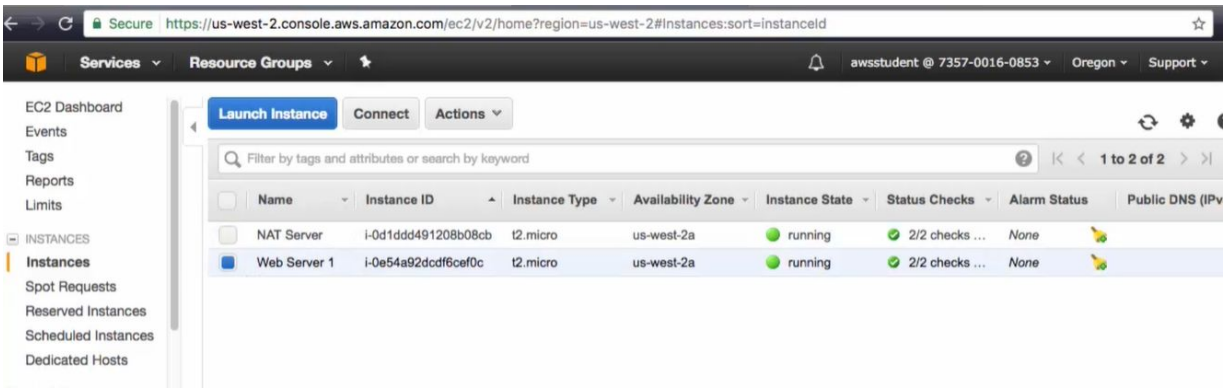
Now that our MySQL Instance is listed as available. you can take a look at some of the details revolving around it.



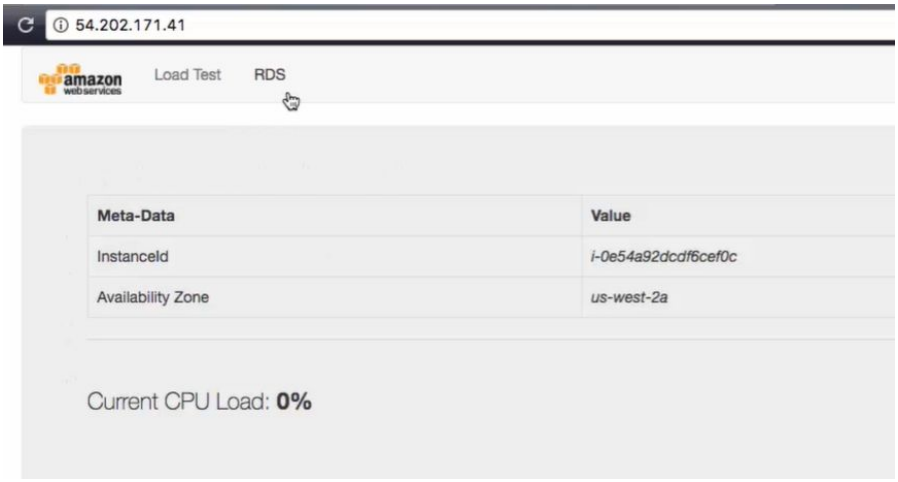
We are going to go ahead and attach to this Database Instance, leveraging our web server.



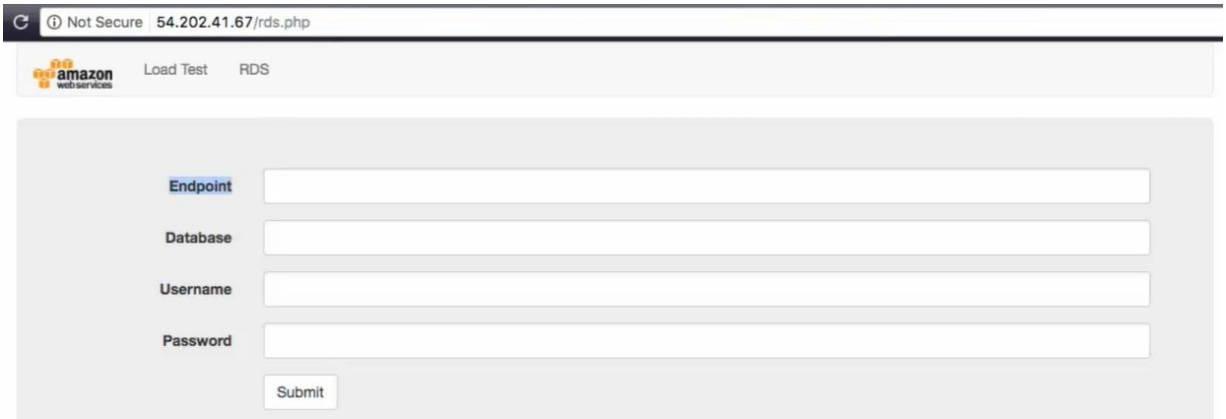
This web server has stood up for us, for this lab and we can find it under EC2 which I'm going to open a new tab. Go over to our running instances and we can find our web server.



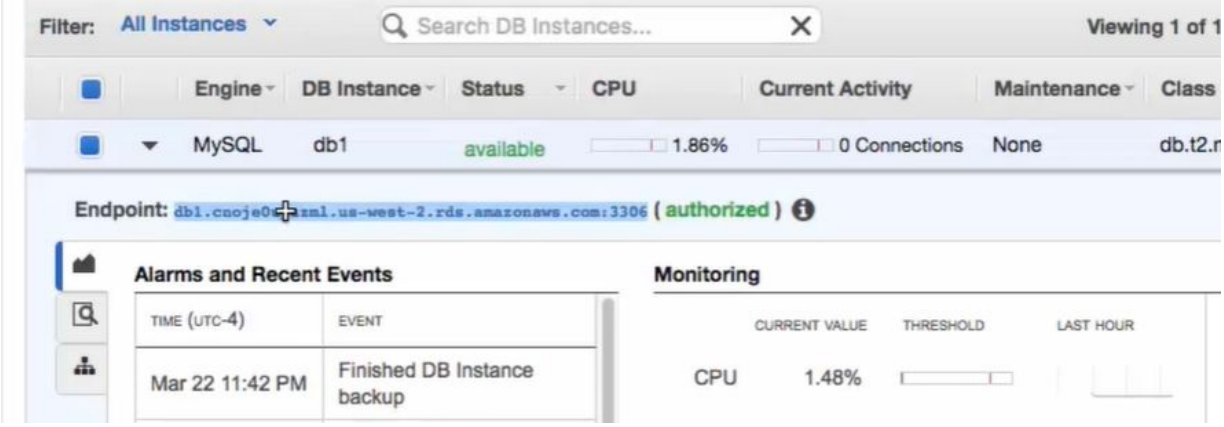
The web server is automatically going to have a public IP address available which we can go to and we can see the same page that we saw loaded on Lab No 1.



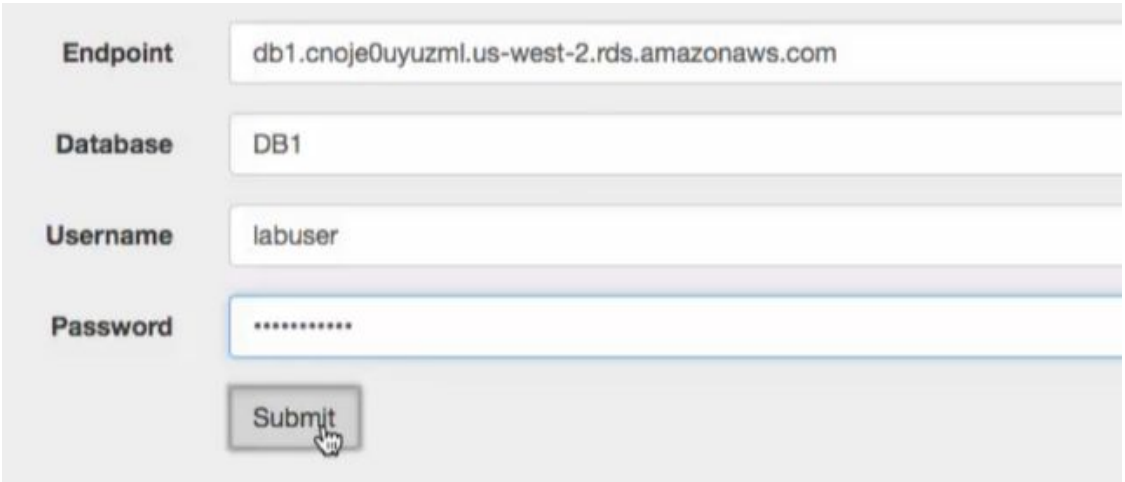
Now at the top we do have an option for RDS and under the RDS tab, we are asked for a couple of bits of information. The first of which is the Endpoint to our RDS Instance.



So we are going to go ahead and copy that and point address and paste it in.



Now for this lab, we do not need the port number on the end. The database we created is DB1 with a Username : labuser with a relevant password. We can now click on Submit.



This is writing out a file to the local web server and it is going to import an address book into
that MySQL Instance.

In the figure below, we can see our current Address Book.

Now in this Address Book, we can edit or remove a contact. We can add a new contact in. So in this case below, we are going to have the details: Name: Doe John, Phone: 555-555-555 and Email Id: johndoe@amazon.com and lets Submit this Contact.

We can see that it has now been added into our database.

This takes us to the end of our Lab.

In this lab we built on Lab No.1 by adding in a multi AZ RDS deployment and then connected that to one of our web servers.

18. Introduction to Elasticity and Management Tools

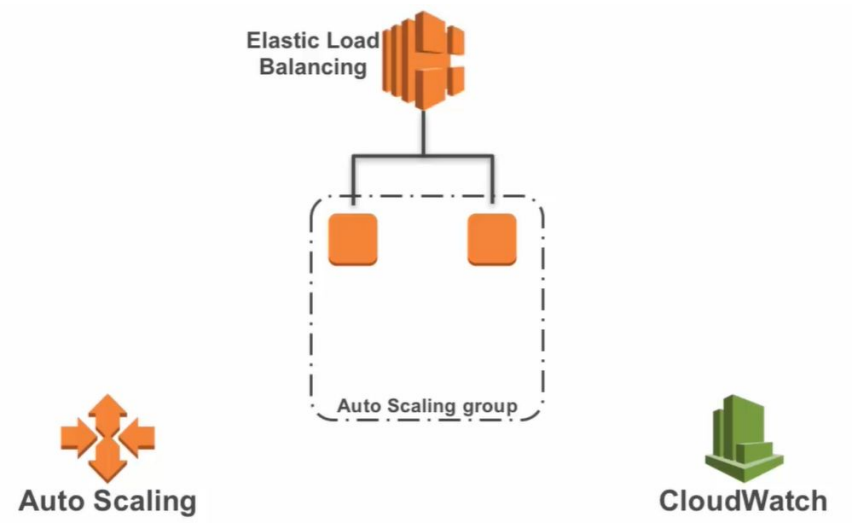
AWS provides you with services to help with the deployment and management of your applications.
This tool includes:

- → Scaling your Compute capacity automatically and dynamically
- → Monitoring your applications and
- → Running status checks to optimize the performance security costs and fault tolerance of your resources.

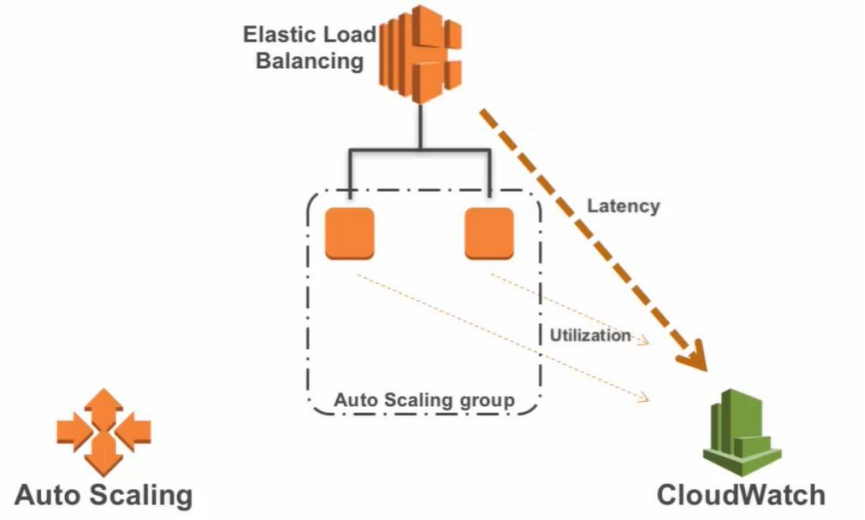
18.1 Elasticity and Management Tools

In this Chapter, we are going to be introduced to try out services including: Auto Scaling, Elastic Load Balancing and Amazon CloudWatch. We will also take a look at Trusted Advisor as one of the Management Tools in AWS.

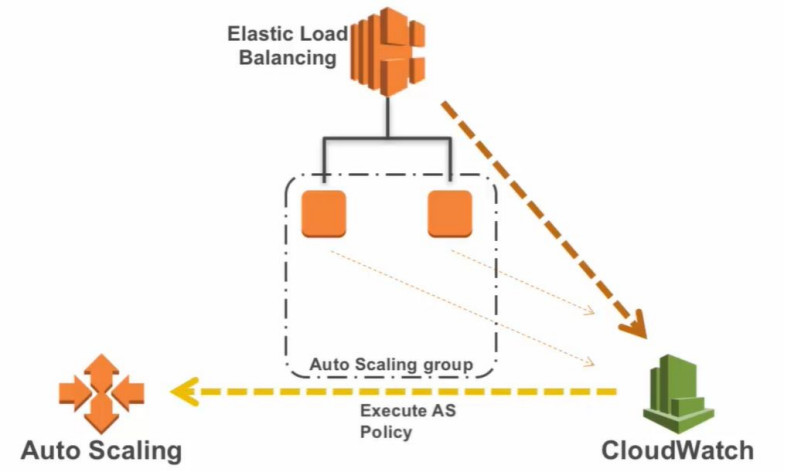
By the end of this Topic, you will have a better understanding of some of the core concepts for things like Auto Scaling Groups, Elastic Load Balancer types and How to access Amazon CloudWatch. Start off taking a look at that Triad of Services and how they work in sync to help us manage elasticity in our environment.



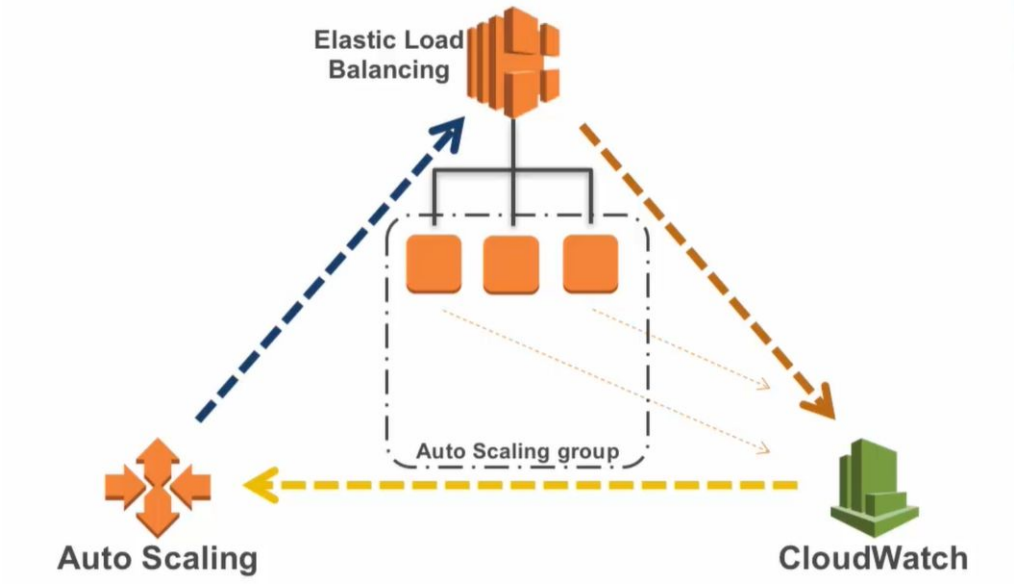
In this example, we have 2 back end instances registered with our Load Balancer. The Load Balancer and EC2 Instances will be sending data in the form of metrics down to Amazon CloudWatch as shown here. These metrics could be things like latency from our load balancer and CPU utilization from our instances.



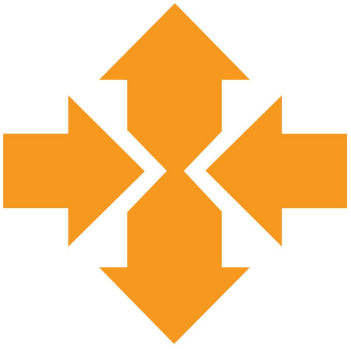
In CloudWatch, we can set up alarms to be triggered. If any of these metrics exceed asset value, our alarm is going to trigger an Auto Scaling Policy.



This Auto Scaling Policy will tell auto scaling that we need an additional instance to be created. Once those instances come online, Auto Scaling will add it to our Auto Scaling Group.



Once the instance has been added to our group, Auto Scaling will register with the Load Balancer, so it can join our fleet.



18.2 Auto Scaling

The first service that we are going to take a look at from the Triad is Auto Scaling. This service helps us to automatically scale our EC2 capacity and make sure that we always have the right number of instances running to meet our current workload demands.

Auto Scaling works really well when we have workloads that tend to experience weekly, daily or even hourly fluctuations in how much capacity they need. One of the best parts is that the Auto Scaling service doesn't have any additional charges. We are just paying for the EC2 Instances that the service launches for us.

When we bring in Auto Scaling, we are going to see a bunch of benefits. Adding in Auto Scaling to our architecture is one of the ways that we can maximize the benefits of being in the AWS Cloud. The Benefits are:

- → Better Fault Tolerance
- → Better Availability
- → Better Cost Management

18.2.1 Better Fault Tolerance

When we use Auto Scaling, our applications can gain some benefits like better Fault Tolerance. Auto Scaling can detect if one of our instances will become unhealthy. It can then terminate it and launch a new Instance to replace it.

We can even configure Auto Scaling to use multiple Availability Zones. In the event when one of those Availability Zones goes down, Auto Scaling will automatically launch instances in the other to compensate.

18.2.2 Better Availability

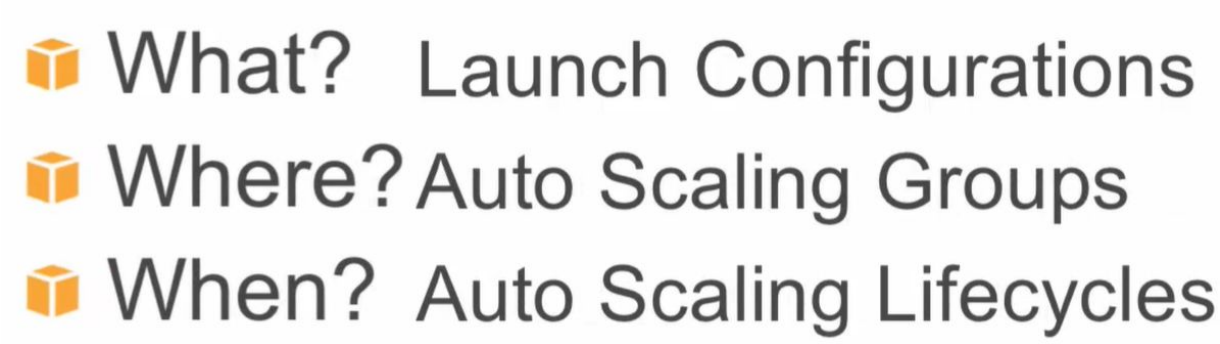
We are also going to gain better availability. Auto Scaling is going to help us ensure that our application has the right number of instances running to meet whatever our current workload demands are.

18.2.3 Better Cost Management

We will get better cost management because Auto Scaling can dynamically increase and decrease the number of instances we have running. We can save money by only launching instances when we need them and terminating them when we do not.

When we are dealing with Auto Scaling there are three questions we have to answer. We need to know: WHAT? WHERE? And WHEN?

- What is going to be a Launch Configuration.
- Where will be in Auto Scaling Group and
- When is going to be part of the Auto Scaling Lifecycle.



Whenever we create an Auto Scaling group, we have to specify a Launch Configuration. The Launch Configuration is just a template that the Auto Scaling service is going to use whenever it launches a new EC2 Instance for us.

The launch config templates are going to have the values we would typically set when launching a new EC2 Instance. Things like :

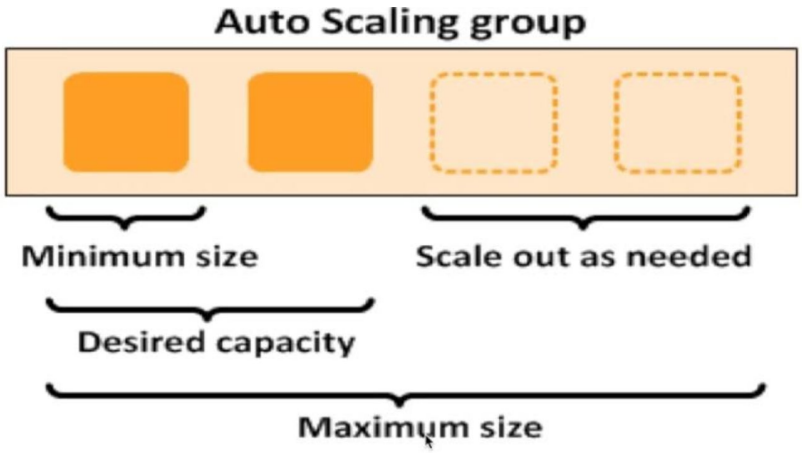
- → AMI ID
- → Instance type
- → Key Pairs
- → Security Groups we want associated with,
- → Any Block device mapping for EBS volumes and
- → User data (if we want to do bootstrapping)

There are some important things that we need to keep in mind when we are dealing with Launch Configurations. Once we have created a launch config, we are not going to be able to modify it.

If we want to change the Launch Configuration used with our Auto Scaling Group, we will need to create a new one and then associate it with the group. If we update our Auto Scaling Group with the new Launch Configuration, it is not going to modify any of our existing EC2 Instances

but what will happen is any new instance Auto Scaling launches, we will use the new configuration.

Auto Scaling Groups are going to be made up of a collection of EC2 Instances. All of the instances in our group are going to be treated as one logical unit when it comes to Scaling and Management.



There are a few values that we can set when dealing with our Auto Scaling groups. They are:

- → Minimum Size
- → Maximum Size
- → Desired Capacity
- → Scale out as needed

The first value is the Minimum Size. When you set a minimum size, Auto Scaling will ensure that we never have fewer instances in the group than that number.

The second number is the Maximum Size. The maximum size is the ceiling or the most instances that we can ever have inside of the group despite any Scaling operations taking place.

The third one is a little bit trickier which is the Desired capacity. The Auto Scaling engine is always working towards that desired capacity. When the desired capacity is above the current instance count, Auto Scaling will look

up our launch configuration and launch a new instance. If the desired capacity moves below our current count, Auto Scaling will terminate one of those instances.

The difference between the Minimum size and the Maximum size is our current Scaling capabilities. One of the ways that we like to picture the desired capacity number is thinking of it like the thermostat at a house. When anyone sets a desired temperature on their thermostat, the system is constantly working to attain that temperature.

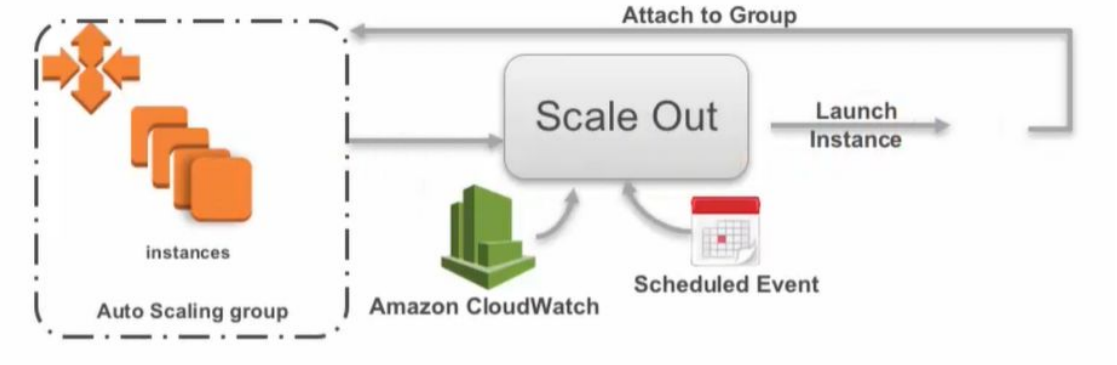
So we will be continually monitoring and should any environmental changes take effect. If a cold breeze comes through the house, the system will kick back on and bring the temperature back up to a desired amount.

Auto Scaling works the same way with desired capacity. If 1 of us becomes unhealthy and has to be terminated. Auto Scaling will see that we no longer have met our desired capacity and will add a new instance.

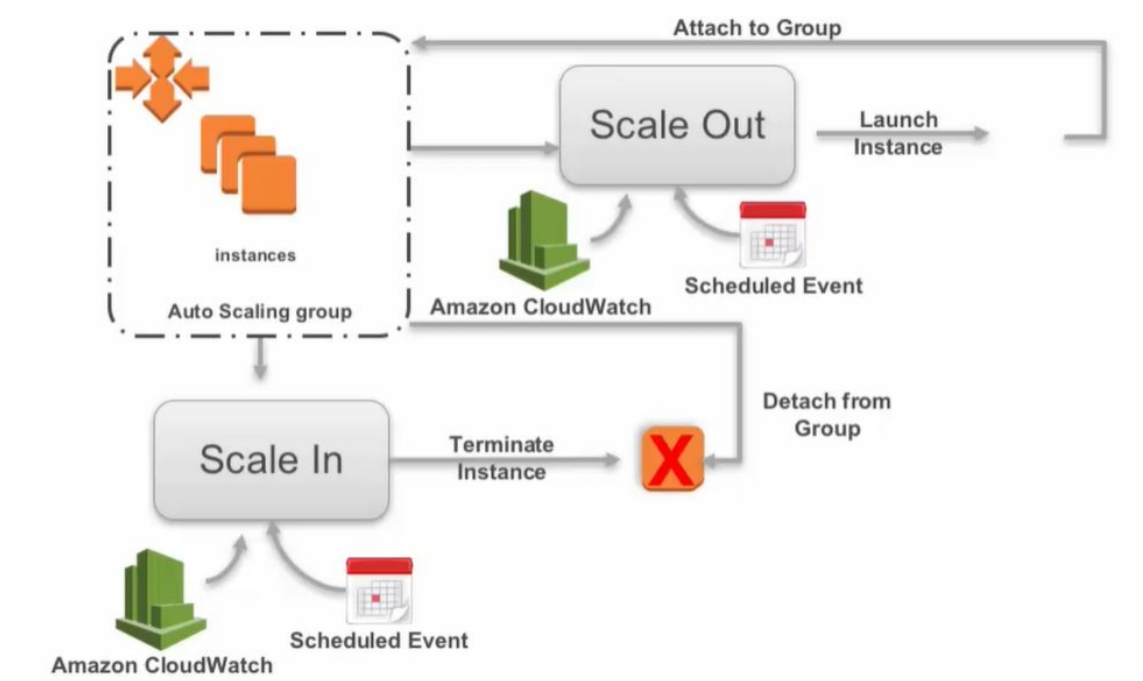
And the final one is the Scale out as Needed. This is the backup instance which is used only when the requirement arises. They are not usually required but are still considered if the right time comes. These are still under our Scaling capabilities as it is within the Maximum size.

Observe some of the life cycles when dealing with Auto Scaling. When we have a scale out event, either from a scheduled event or because an Amazon CloudWatch alarm has been triggered, Auto Scaling will launch an instance.

Once those instances come online, it will get attached to our Auto Scaling group.



On the other hand, if we have a Scale in event either from a scheduled event or another alarm being triggered, that instance is immediately detached from the group and then terminated.



Using what are called Lifecycle hooks, we can tie into any one of these steps in the process.So maybe when we initially launched an instance on a scale out event, we want to trigger some additional configuration actions or

maybe we want to notify an administrator that that scaling event is taking place.

When we go to pull an instance out of the group and decommission it, maybe we want to take all of the logs off of that instance and ship them to my S3 bucket. So maybe we can take a look later on.

18.3 Elastic Load Balancing

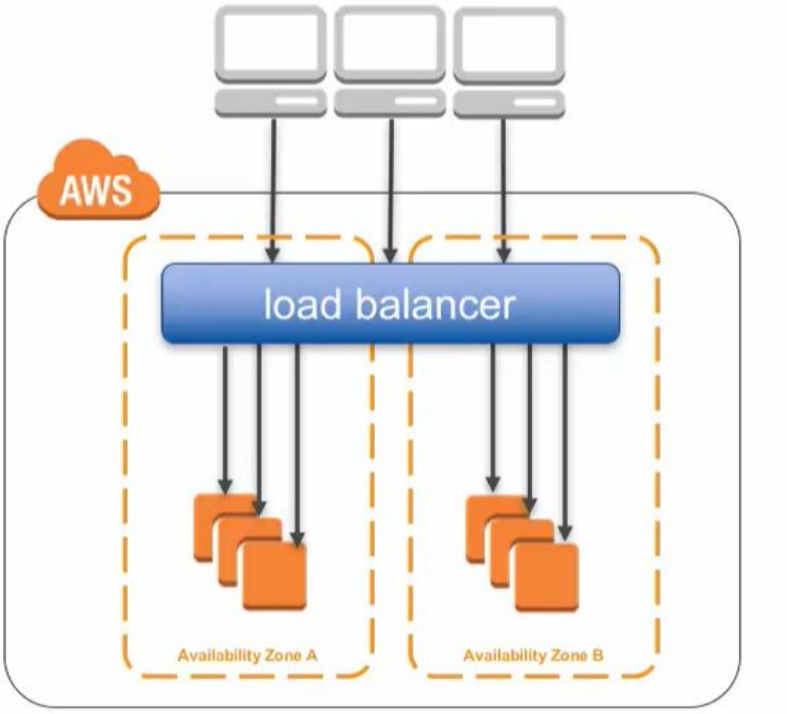
The next service that we are going to cover from the triad is Elastic Load Balancing. By using Elastic Load Balancing in conjunction with Auto Scaling which we just covered, we are able to evenly distribute the incoming requests among all of the members of the Auto Scaling group.

Because Elastic Load Balancing supports health checks of other instances, we are able to achieve a higher level of Fault Tolerance for our applications. Just like Auto Scaling, Elastic Load balancing is able to operate in a single Availability Zone or span multiple Availability Zones per even higher availability.

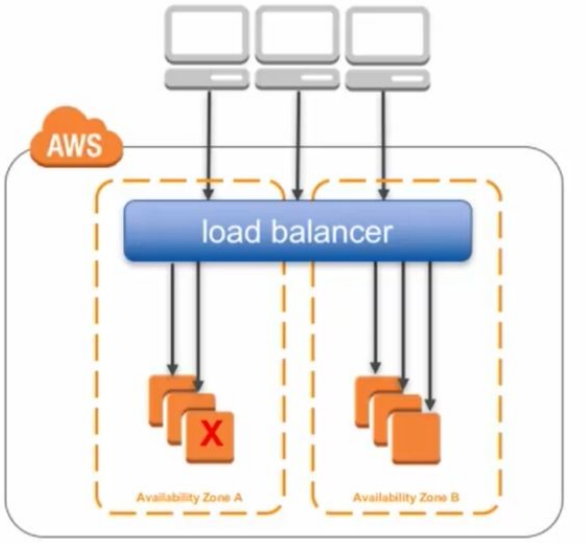
When it comes to handling traffic, our Elastic Load Balancers are able to handle incoming requests in the form of HTTP, HTTPS and TCP traffic toAmazon EC2 instances.

We will take a moment to cover how the Load Balancing service itself works. The Load Balancers are going to accept incoming requests from our clients and evenly distribute those requests among the back end instances. As we just mentioned, the Load Balancers are able to perform health checks on all of those backend instances. If for some reason the backend instance

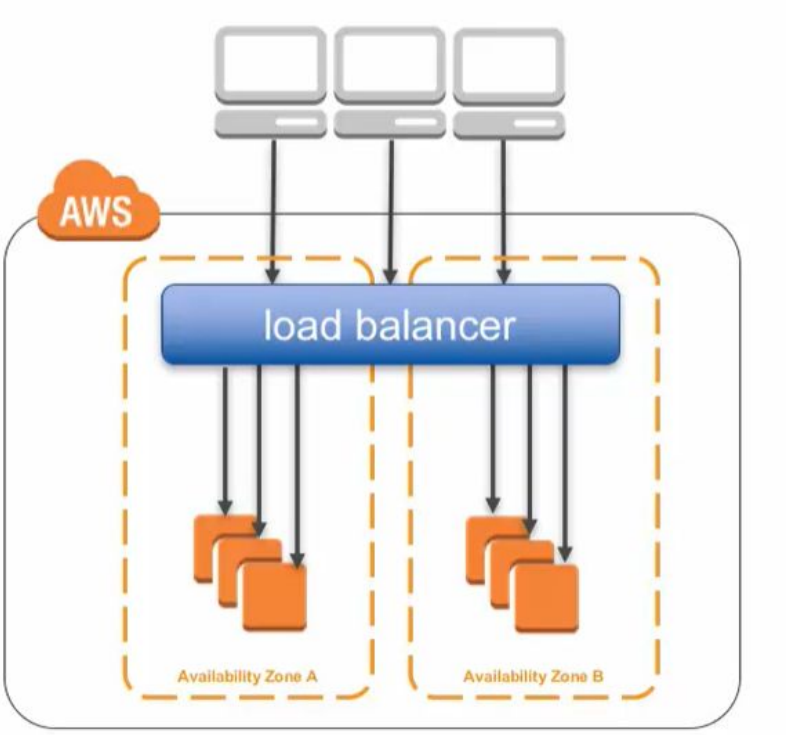
begins to fail its health checks, it will be marked as unhealthy by the Load Balancer.



Once an instance is marked as unhealthy by the Load Balancer, it will no longer receive new incoming requests. While the instance is unhealthy, the Balancer is continuing to perform health checks among all of the instances.



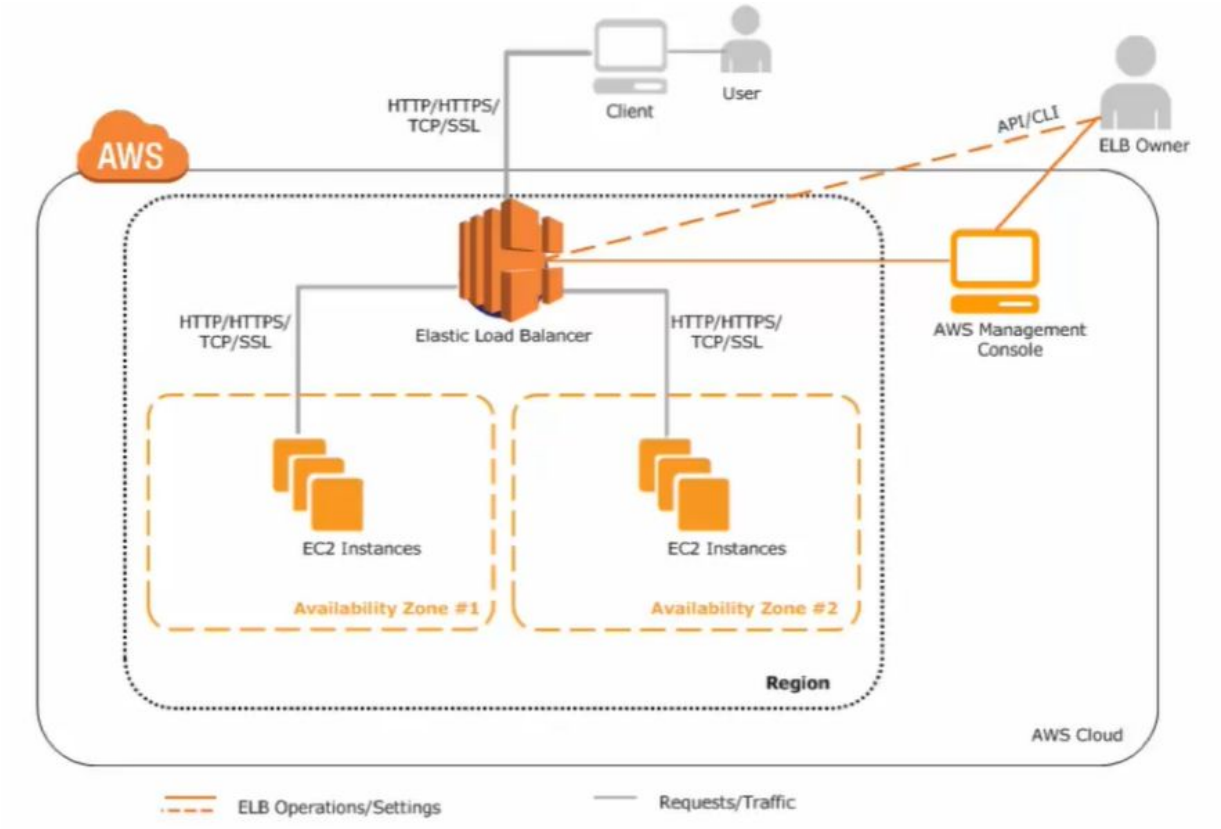
When the unhealthy instance has been determined as running in a healthy state again, the Load Balancer will reattach and begin routing traffic to it again. These health checks are an ongoing process. The Load Balancer will be continuously checking the health status of each back and instance and reporting that state.



If we choose, we can use this health status from the Load Balancer regarding the backend instances in conjunction with Auto Scaling. If an instance remains in an unhealthy state beyond a set grace period, Auto Scaling can automatically terminate that instance and replace it with a new one.

By taking advantage of the combination of services in this way, we can get closer to that goal of a self-healing architecture where our services are monitoring themselves and automatically replacing instances that are found to be acting up.

When it comes time to manage the Load Balancer itself, we have a couple of options. We can go directly within the AWS Management Console and interact with the service there or we can interact with it via the Command Line Interface (CLI) or directly via APIs as shown in the below figure.

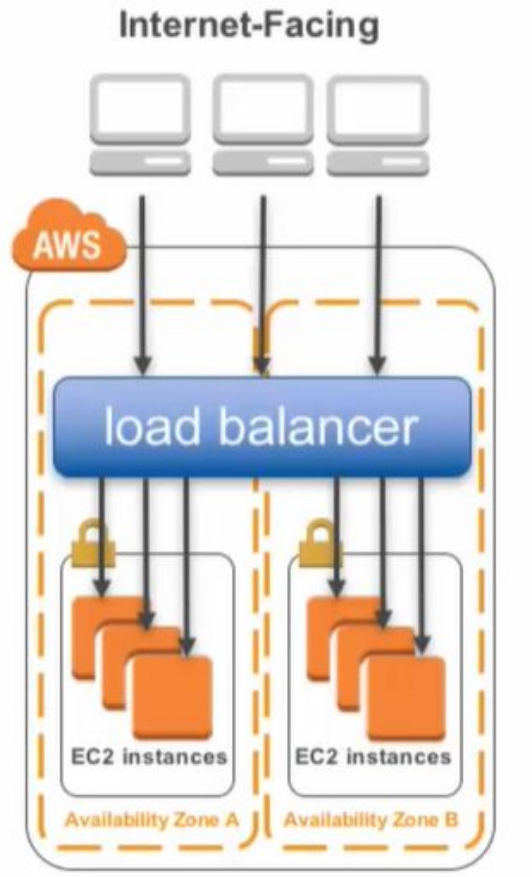


When it comes time to select the Load Balancer type, we have a few options for how to configure and locate that Load Balancer in our environment. There are 3 types:

- → Internet-Facing
- → Internal
- → HTTPS

18.3.1 Internet-Facing

The first scenario is referred to as Internet Facing. In this scenario, our Load Balancer accepts incoming requests directly from clients over the Internet and then routes those requests to our Backend Instances.



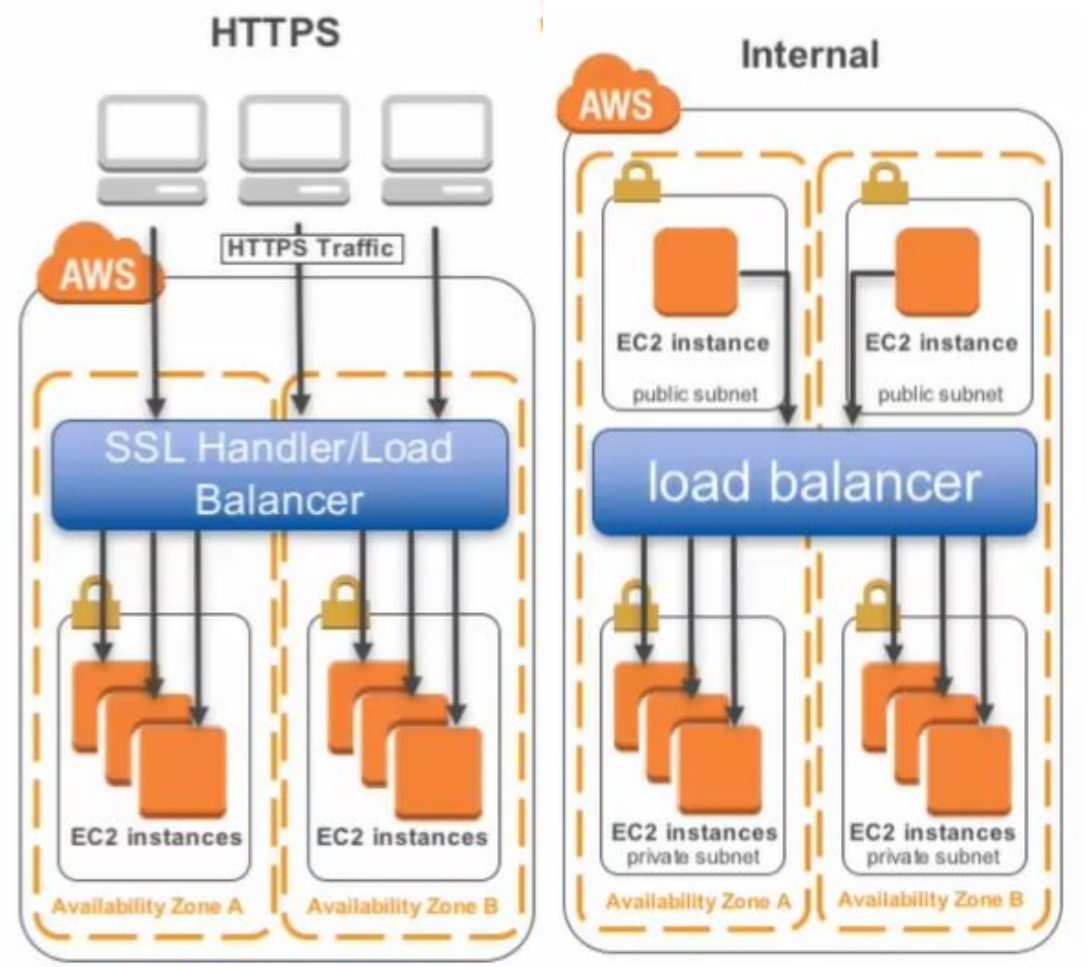
18.3.2 Internal

Our second option is to configure the Load Balancer to be Internal. With the Internal scenario, all incoming requests for our load balancer are being generated from within our VPC. Not from the Internet. In fact, there is no way for an internet facing client to interact with this Load Balancer.

18.3.3 HTTPS

No matter if we choose the Internet-Facing scenario or the Internal scenario, we have the option of enabling HTTPS Load Balancing. When we configure our Load Balancer for HTTPS, we are creating a Load Balancer that is using the SSL TLS protocol for encrypting all incoming connections.

When used in this scenario, our Load Balancer has the option of routing that traffic to our backend instances over HTTP or re-encrypting and sending it over HTTPS.


HTTPS
HTTPS Traffic
AWS
SSL Handler/Load Balancer
EC2 instances
EC2 instances
Availability Zone A
Availability Zone B
Internal
AWS
EC2 instance
public subnet
EC2 instance
public subnet
load balancer
EC2 instances
private subnet
EC2 instances
private subnet
Availability Zone A
Availability Zone B

19. Amazon CloudWatch

The final service for my triad is Amazon CloudWatch. CloudWatch is our monitoring service. It is a centralized location where we can collect and track metrics, collect and monitor log files, set alarms and take actions based upon changes that we see in our AWS resources.

Amazon CloudWatch is going to give us visibility into a large set of default metrics. These default metrics include things like:

- CPU utilization for EC2 instances
- The number of read and write operations occurring on our EBS volumes and
- The number of simultaneous connections to our RTX databases

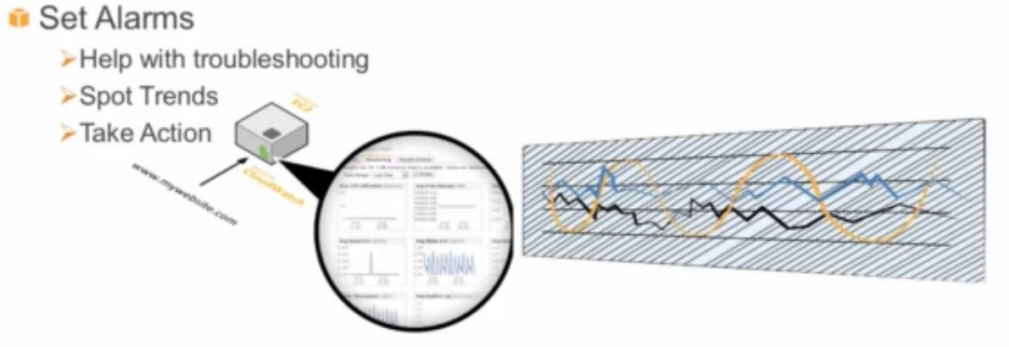
By default, CloudWatch is not capable of collecting metrics from our operating systems or applications. If we want we can push metrics from those sources in the form of customer metrics to CloudWatch to be monitored. We can do this by using the API or the CLI.

All of this metric data being collected by CloudWatch is accessible to us through the AWS Management Console directly on the web, through the APIs, SDKs or CLIs.

19.1 CloudWatch Facts

Here are some of the Facts related to CloudWatch:

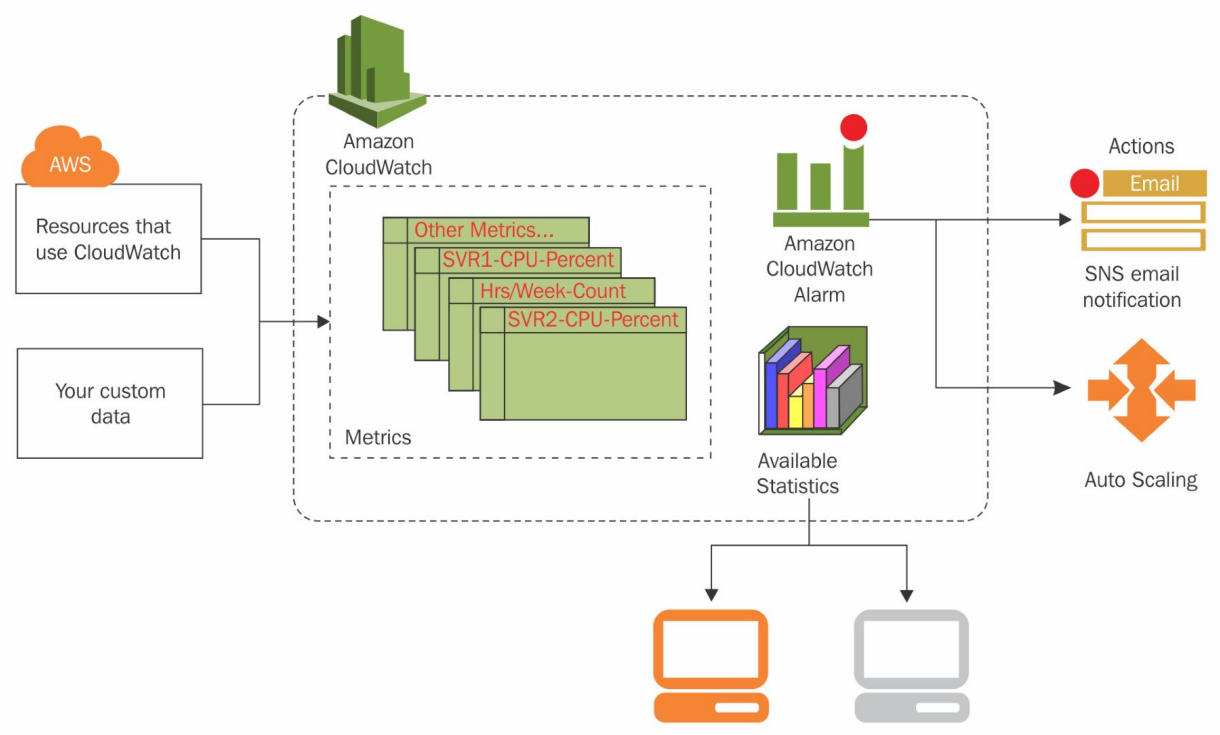
- → Amazon CloudWatch enables us to monitor our AWS resources in near real time.
- → With basic monitoring, CloudWatch collection reports metric data once every 5 minutes.
- → If we need even greater frequency, we can enable CloudWatch, a detailed monitoring which provides the same metrics but at a 1 minute interval.
- → In CloudWatch, we can set alarms based on these metrics to do things like :
  - ◆ Send an email to an administrator or
  - ◆ Send a text message to another user or
  - ◆ Trigger an Auto Scaling Policy.



19.2 Amazon CloudWatch Architecture

Amazon CloudWatch is a metric repository. Supported AWS resources put metric data into CloudWatch for analysis and we can even do custom metrics with our applications. All of these metric data can be analyzed and viewed straight through the AWS Management Console or using the APIs, pulled out into third party utilities.

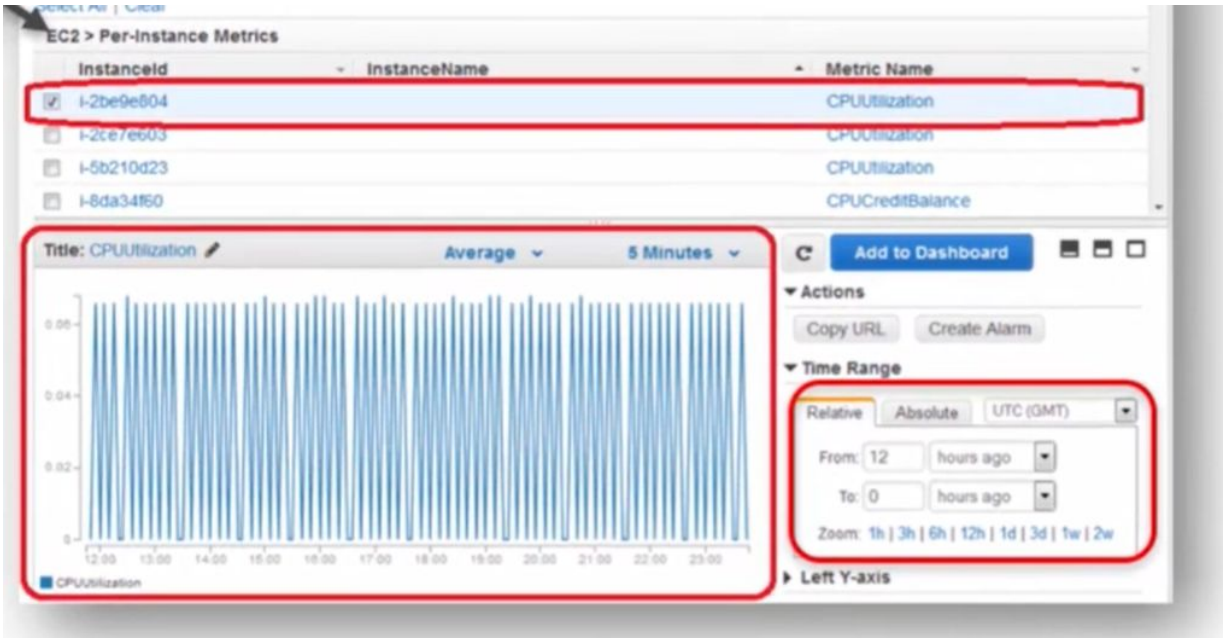
Amazon CloudWatch alarms enable us to send notifications, trigger Auto Scaling events based on thresholds that we set for these metrics. This diagram is showing the Amazon CloudWatch Console.



19.3 CloudWatch Metrics

In the following image, we have selected an EC2 per instance metric of CPU utilization. In the lower right hand corner, you can see that we can specify the time range of the metric values that we would like to see now.

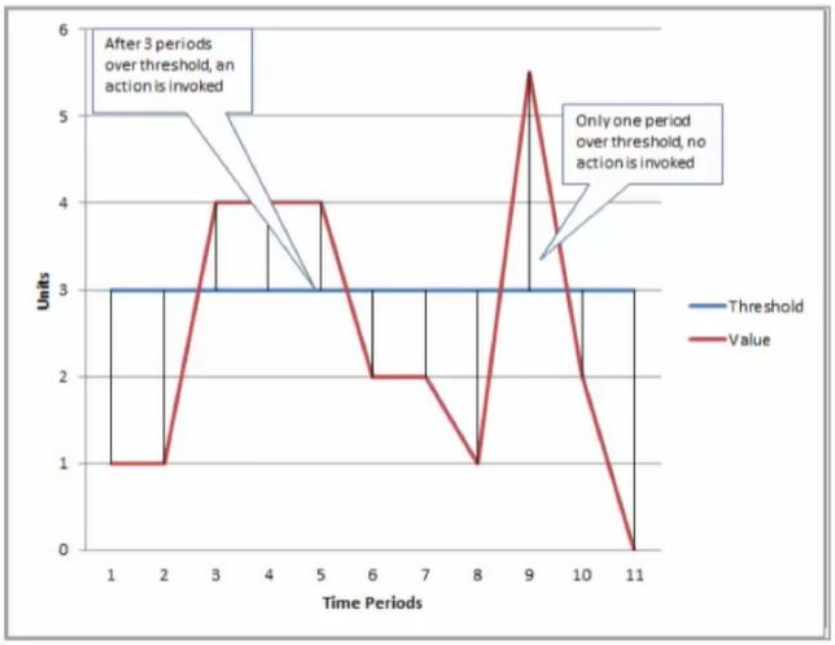
An important note when you are looking at the graphs as with looking at any graph is to see the total scale. While it might look as though the CPQ for this instance is moving up and down very rapidly. If you look at the scale you see that it is only between 0 and point 0 6% of the CPU.



So the scales are staying pretty stable.

19.4 CloudWatch Alarms

CloudWatch Alarms watch as a single metric over a time period that you specify and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods.

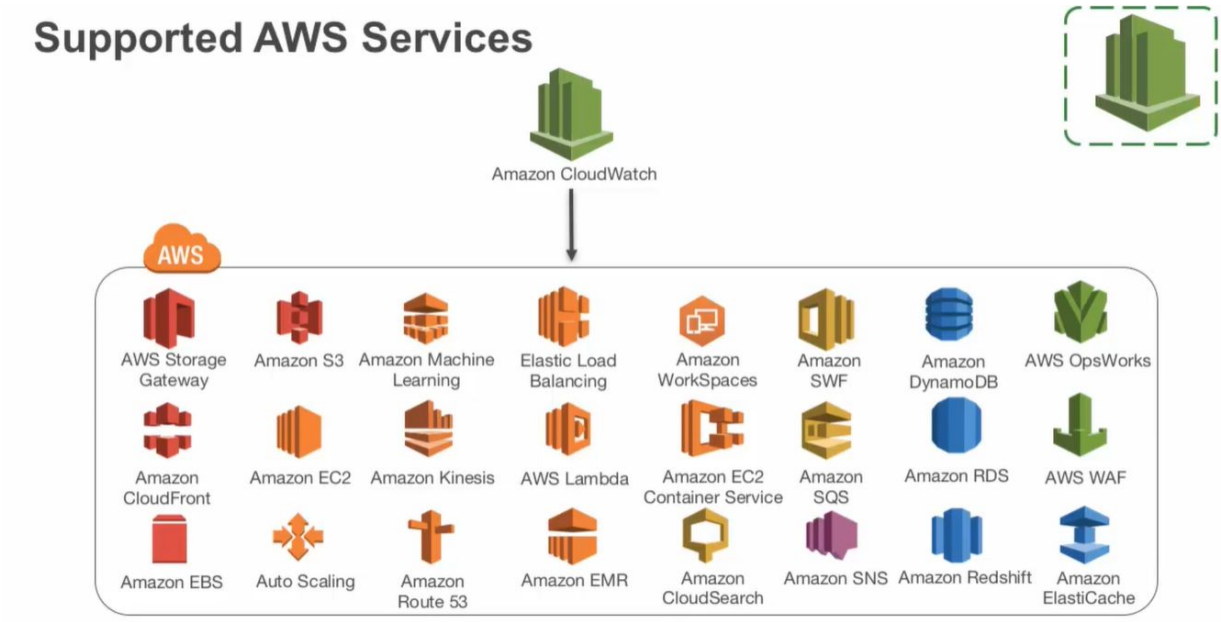


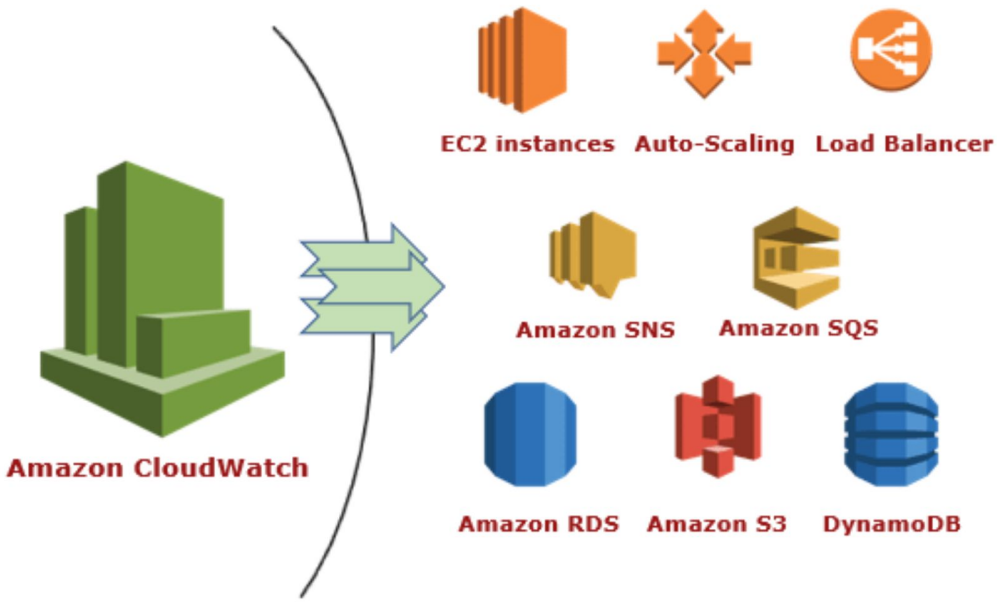
The action is a notification sent to an Amazon SNS topic or an Auto Scaling Policy. The alarm's invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state. The state must have changed and be maintained for a specified number of periods.

In the figure, the alarm threshold is set to 3, and the minimum breach is 3 periods. The alarm invokes its action only when the threshold is breached for 3 consecutive periods. In the figure, this happens with the 3rd through 5th time periods and the alarm is triggered. At periods 6, the value dips below the threshold and the state is set to OK. Later during the night time period, the threshold is breached again but not for the necessary 3 consecutive periods. Consequently, the alarm state remains OK.

19.5 Supported Services

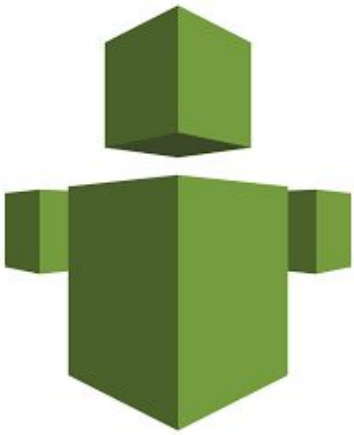
This figure shows the supported services that send metric data to Amazon CloudWatch as of the writing of this book. It is important with all services and AWS to check the online documentation page to see what the current supported resources are.




EC2 instances
Auto-Scaling
Load Balancer
Amazon SNS
Amazon SQS
Amazon CloudWatch
Amazon RDS
Amazon S3
DynamoDB

20. AWS Trusted Advisor

The last service that we are going to take a look at is a really useful one. AWS Trusted Advisor. To simply put, AWS Trusted Advisor is a recommendation engine for best practices. The service is going to perform



checks in four categories. Which are:

→ Cost Optimization
→ Security
→ Fault tolerance and
→ Performance improvements

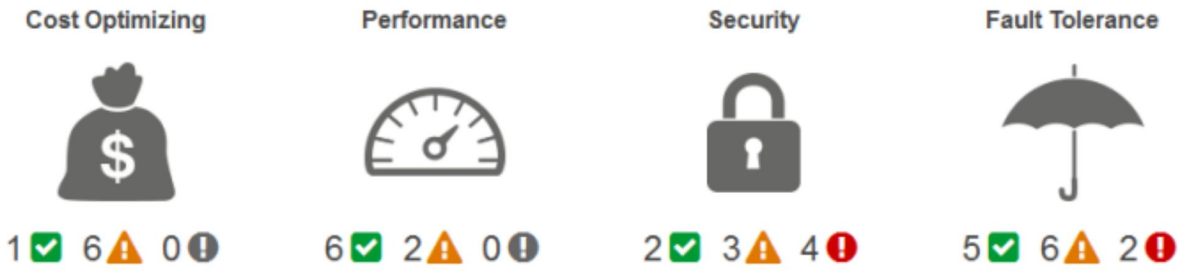
The checks that are performed by Trusted Advisor are going to be reported on a dashboard in an easy to understand, 3 color scheme.

- The first color Red means that some type of action is recommended.
- Yellow means that you might want to investigate that finding and
- Green means no problem was found

For each of the checks, we can go and review a detailed description of the recommended best practice, a set of the alert criteria meaning: what was Trusted Advisor looking for for that check. The guidelines for taking action and a list of useful resources that you can use to find additional information on that topic. They are:

→ Cost Optimization

→ Security
→ Fault Tolerance
→ Performance Improvements



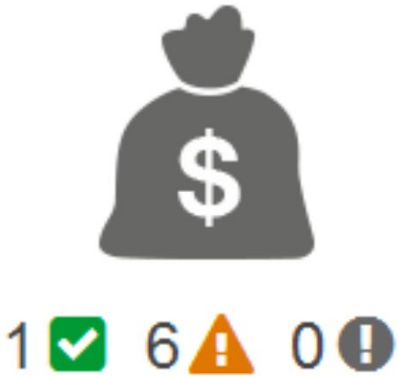
20.1 Cost Optimization

Trusted Advisor can help us save money on AWS by checking for unused and idle resources. As an example, the following cost optimization checks are available and Trusted Advisor.

20.1.1 Amazon EC2 reserved Instance optimization

This check reviews your previous month's Hour by hour usage aggregated across all of your consolidated billing accounts and calculates an optimal number of partial upfront reserved instances

20.1.2 Low Utilization Amazon EC2 Instances



This check is going to look through your account for any instances that in the last 14 days have spent at least 4 of them under 10% of CPU utilization.

20.1.3 Idle Load Balancers

This is going to check for any Load Balancers in your account that are not actively being used

20.1.4 Underutilized Amazon EBS Volumes

This check is going to look through the Elastic Block Store volume configurations and warn you when it sees volumes that appear to be underutilized.

20.1.5 Unassociated Elastic IP addresses

Elastic IPs are an interesting item in AWS. It is one of the services that we are going to charge you for if you are not using it. So this check is going to go through and look for any elastic IP addresses that are not associated with an instance that is currently online.

20.1.6 Amazon RDS Idle Database Instances

This check is going to look for any RDS instance that has not had a connection in an extended period of time. It will then recommend that you turn off that instance to save some money.

20.2 Security

Trusted Advisor can help us improve the security of our applications by closing gaps, enabling various AWS security features and examining the sets of permissions that you have in place.

The following security checks are going to be made available to us in Trusted Advisor.

20.2.1 Security Groups

This check is going to look for any specific ports that are totally unrestricted. It is going to check security groups for rules that allow unrestricted access. Unrestricted access increases the opportunity for malicious attacks because it is going to be wide open to the Internet.

Trusted Advisor is going to analyze what the port is used for and make a recommendation based upon that. The ports with the highest risk are going to get flagged Red. Those with a little bit less risk are going to be flagged Yellow. Ports that get flagged Green, are typically used by applications that require unrestricted access like HTTP and SMTP.

20.2.2 AWS IAM Use

This check is simply making sure that you are taking advantage of AWS Identity and Access Management.

20.2.3 Amazon S3 Bucket Permissions

This check is looking for any buckets in the Amazon simple storage service that have open access permissions allowing access from anywhere in the world.

20.2.4 MFA on Root Account

Now this really is a best practice check. We always recommend that after creating your root account for AWS, you immediately enable MFA on that account to prevent any security credentials from becoming compromised easily.

20.2.5 AWS IAM Password Policy

This check is simply making sure that you have a password policy enabled for your IAM users and that password requirements have been set.

20.2.6 Amazon RDS Security Group Access Risk

This check is going to take a look at security group configurations for RDX and warn you when a security group rule might be granting overly

permissive access to your database.

20.3 Fault Tolerance

For Fault Tolerance Trusted Advisors helping us to increase the availability and redundancy of our applications by taking advantage of things like Auto Scaling, Health Checks, Multi-AZ deployments and backup capabilities

Trusted Advisors going to be performing the following checks:

- → Checking to make sure we have recent Amazon EBS Snapshots of our volumes
- → Making sure that our load balancers are set to balance across Availability Zones
- → Ensuring that our Auto Scaling group resources are also set to move across Availability Zones
- → Checking to make sure our audience is set to multi AZ
- → Amazon Route 53 name server delegations (This check is just validating that we are pointing to the correct DNS servers for Route 53)
- → ELB Connection Draining (This check is looking for any load balancers that do not have connection draining enabled)

20.4 Performance Improvements

AWS Trusted Advisor is going to help us improve the performance of our services by checking for things like service limits, ensuring that we take advantage of provisioned throughput where possible and monitoring for overutilized instances.
The checks being performed byTrusted Advisor are things like:

20.4.1 High Utilization of Amazon EC2 Instances

This is looking for any instance that in the last 14 days has maintained higher than 90% CPU utilization for more than four days.

20.4.2 Service Limits

Trusted Advisor is going to take a look at the service limits currently on our account and warn us when we exceed 80% of that current limit.

20.4.3 Large Number of Rules in EC2 Security Group

If an EC2 Security Group has an excessive number of rules, your network performance can be degraded.

20.4.4 Over Utilized Amazon EBS Magnetic Volumes

This check is looking for any Magnetic EBS volumes that might benefit by being moved to either a general purpose or provisioned IO volume.

20.4.5 Amazon EC2 to EBS Throughput Optimization

This check is looking for any EBS volume that may be impacting the throughput capabilities of an EC2 instance

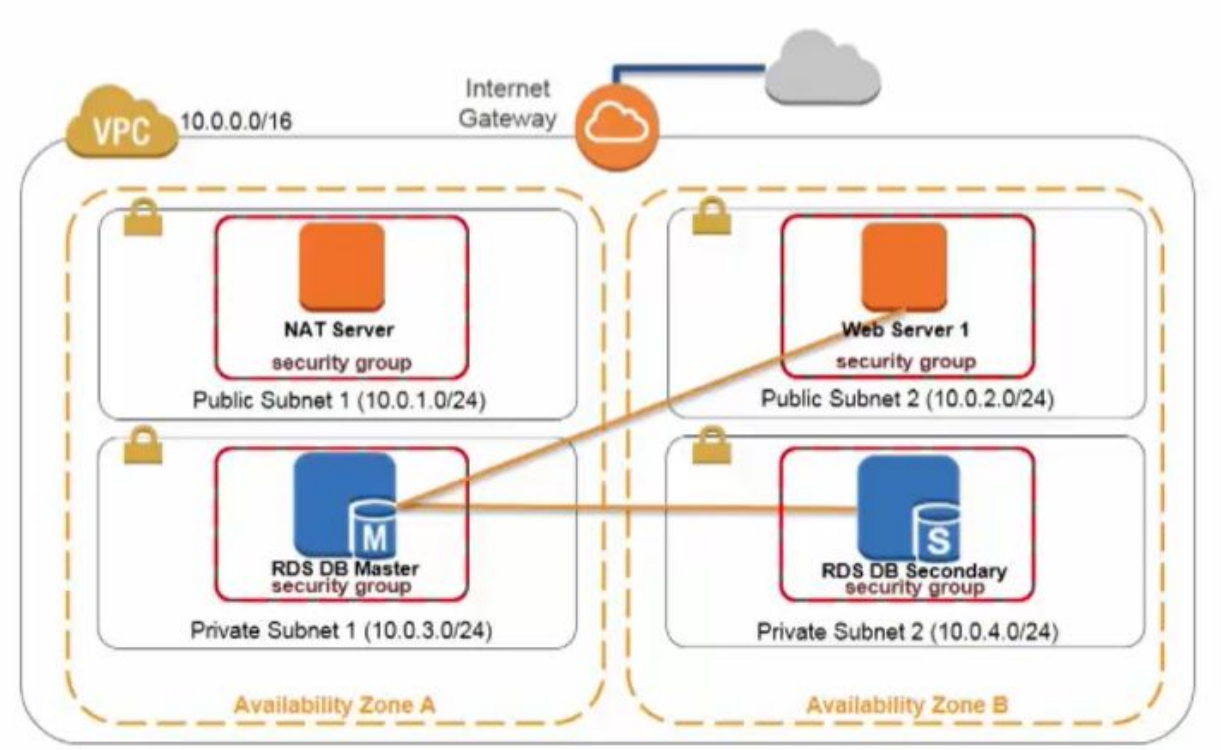
20.4.6 Amazon CloudFront Alternate Domain Names

This check is looking for any CloudFront distributions for alternate domain names with incorrectly configured DNS settings.

21. Demo: Scale and Load-Balance Your Web Application

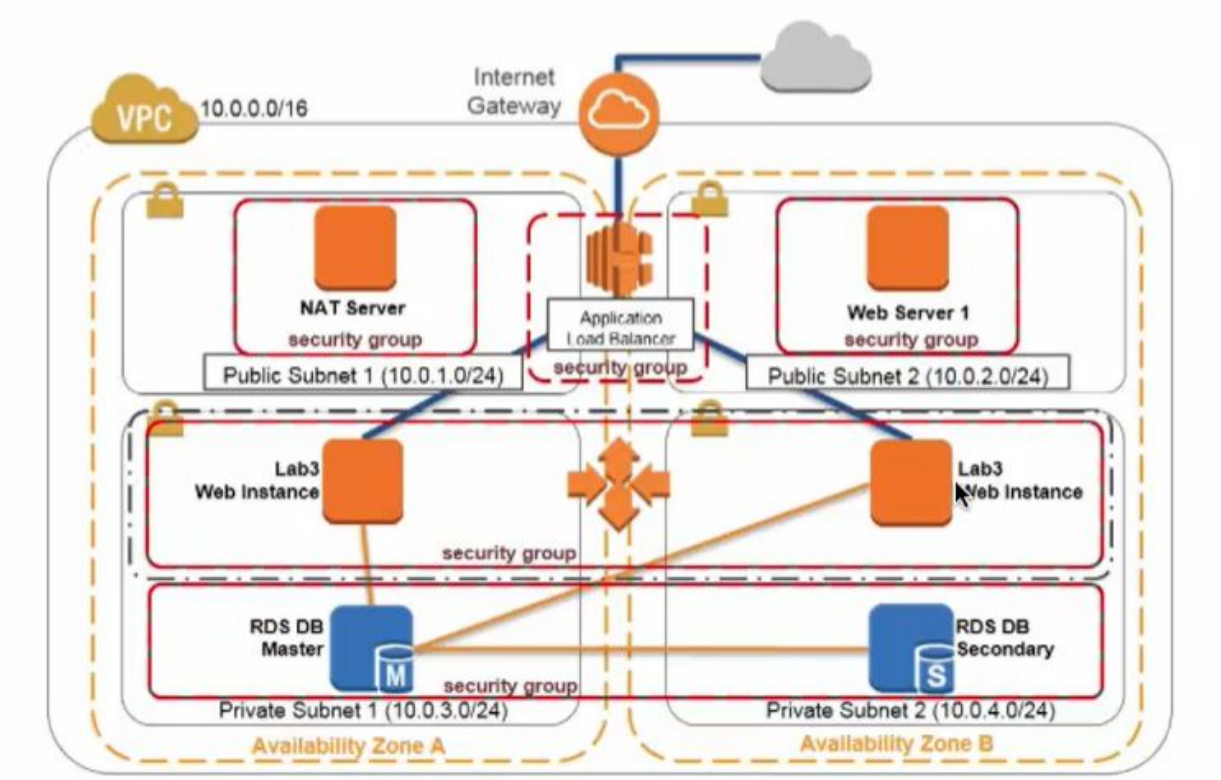
21.1 Lab #3: Managing Your Infrastructure

Here, we are going to be picking up where we left off from Lab #2. So we can see at the end of Lab#2, we had our VPC with subnets in two different Availability Zones. A web server sitting in the public subnet with a multi AZ RDS deployment.

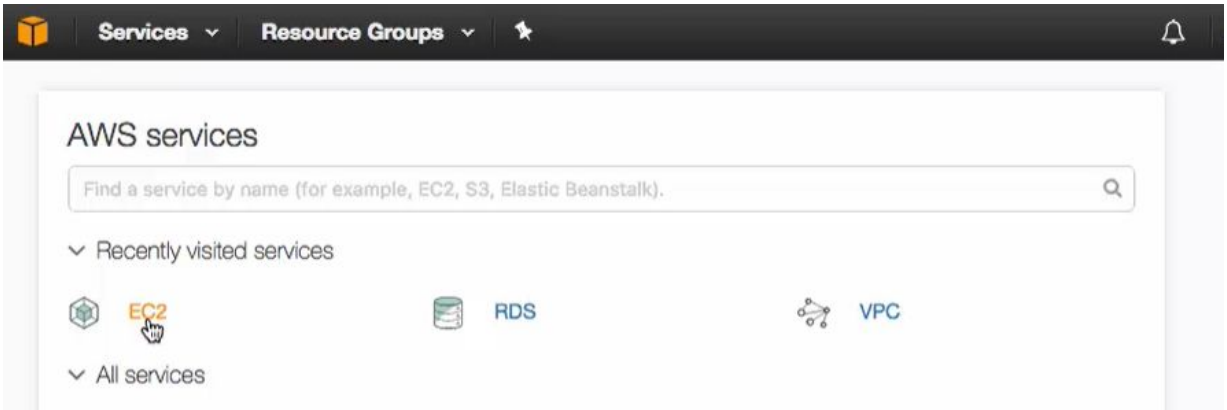


In this lab will be adding an application Load Balancer to balance incoming traffic to backend instances. We will be creating an AMI from Web Server 1 and then using that AMI to launch 2 additional web server instances back in our private subnets.

These Web instances will then be connected to our database on the backend. We will also be taking a look at CloudWatch and how we can use it to monitor the performance of our infrastructure.

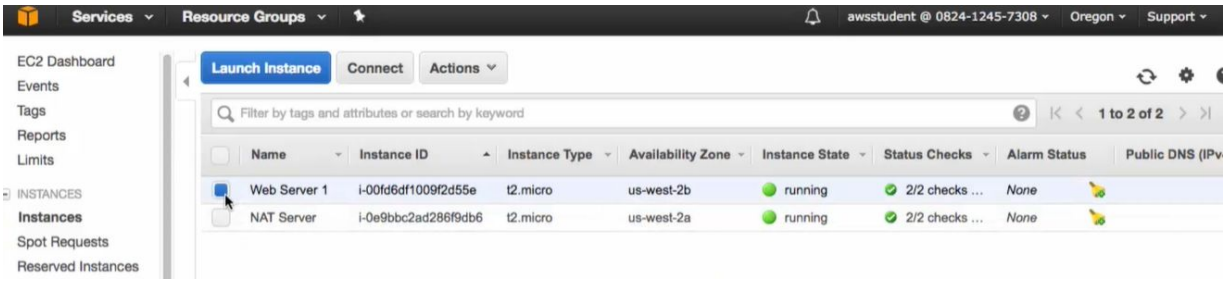


The first task that we are going to complete in this lab is creating our custom AME. We can create our AMIs under the EC2 service.

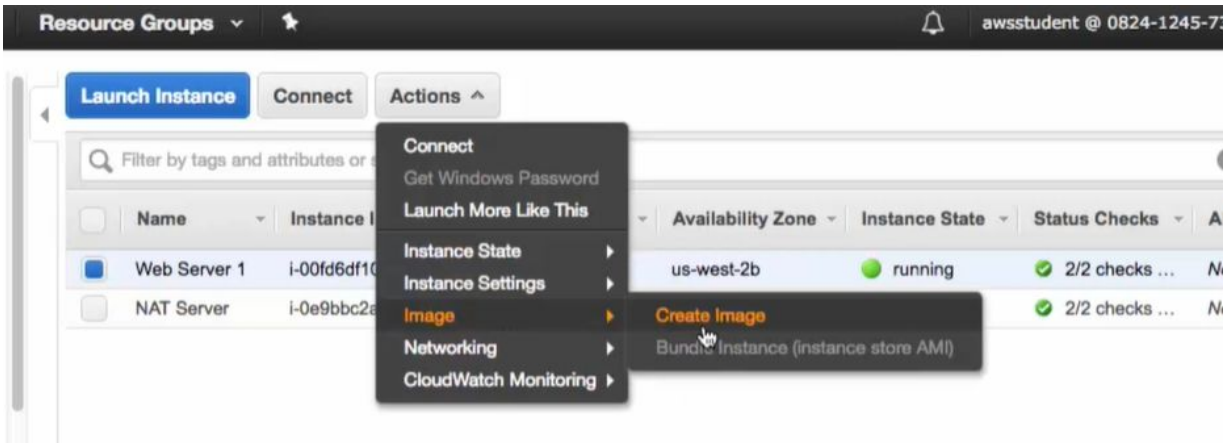


Once we are under EC2, we are going to take a look at our running instances and identify our Web Server Instance.

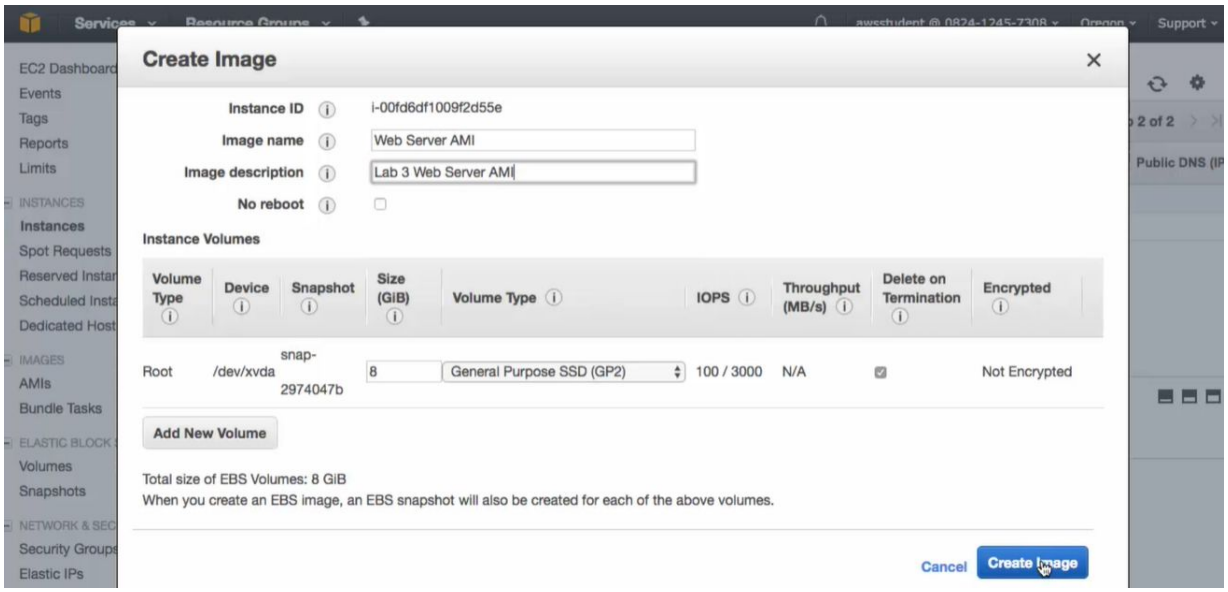
That is our first one of the list here.



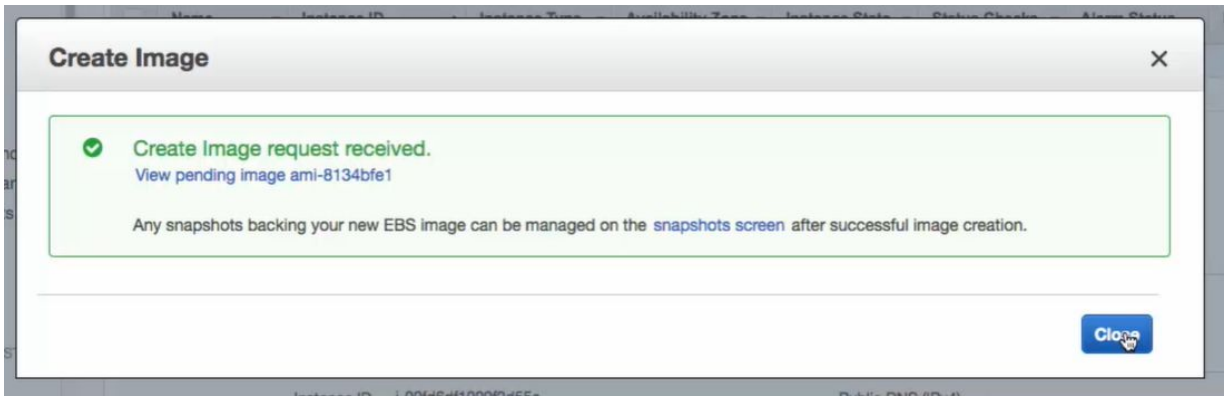
Once we have selected our Web Server Instance, we can go up to Actions ➜ Image ➜ Create Image.



Under the Create Image Properties, we will give an Image name: Web Server AMI and for the description, we will just call this Lab 3 Web Server AMI. Now we can just click Create Image.

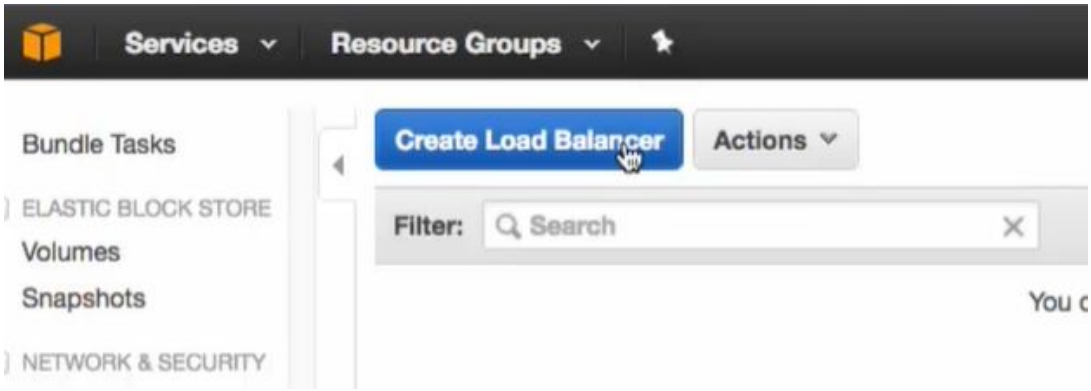


The request has been received and this will take a couple of minutes for it to be created while we are waiting for this.

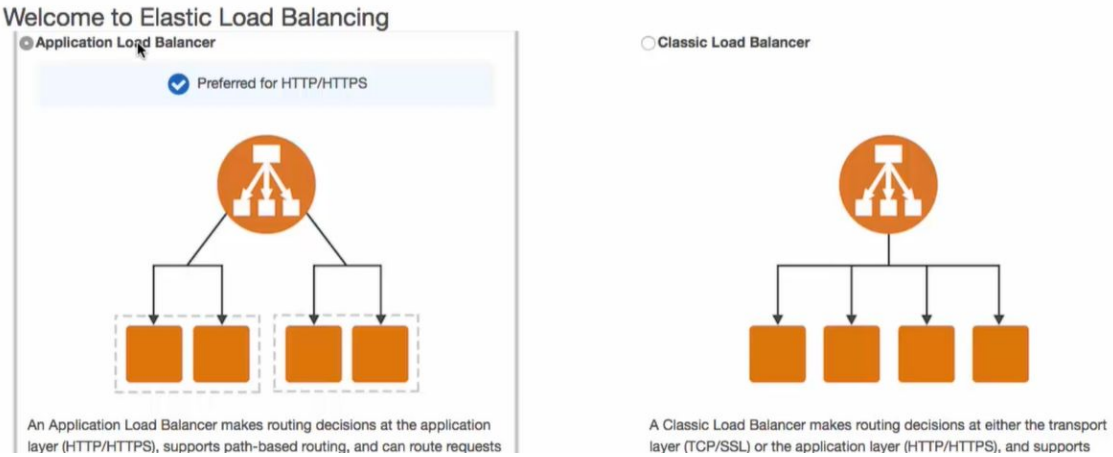


The next task we are going to complete is creating a Load Balancer. To get to our Load Balancers, it is under the EC2 service but we can go over to the navigation pane on the left hand side and we will find Load Balancers.

Click Create Load Balancer

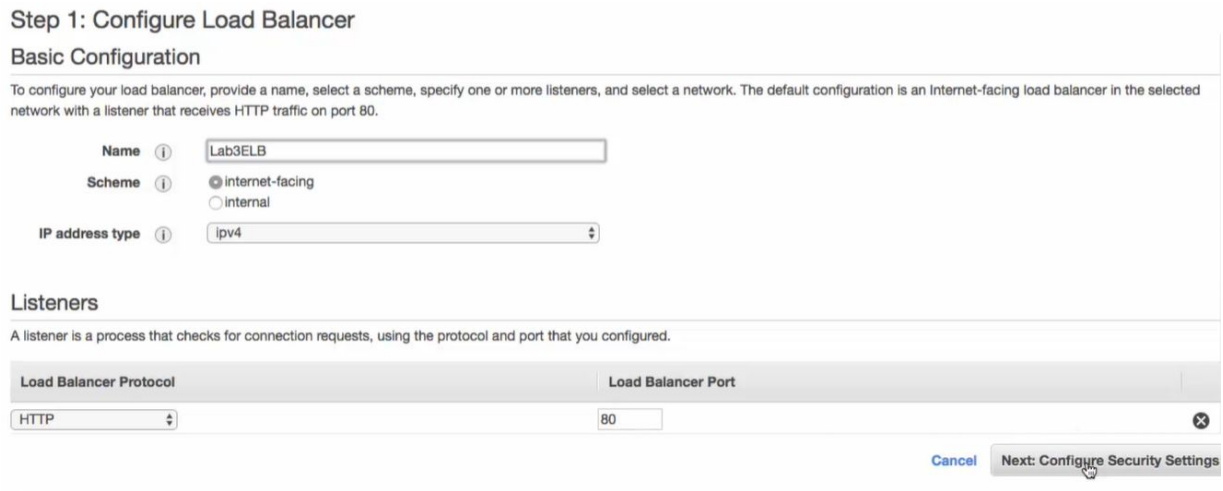


For this lab, we are going to be using the Application Load Balancer and click Continue.

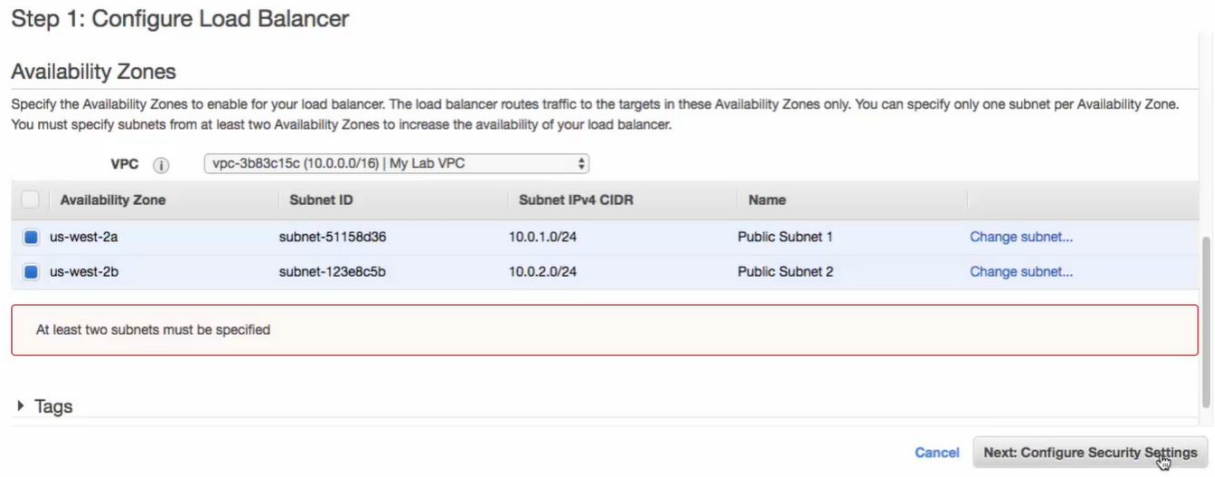


For our name, we will keep Lab3ELB.
Scheme: Internet-facing
IP address type: Ipv4
Here is another section for Listeners. A listener is a process that checks for connection requests, using the protocol and port that is configured.
Load Balancer Protocol: HTTP
Load Balancer Port: 80
He will be an elastic load balancer.

All of the above settings are the default settings. Next we will click on Configure Security Settings.



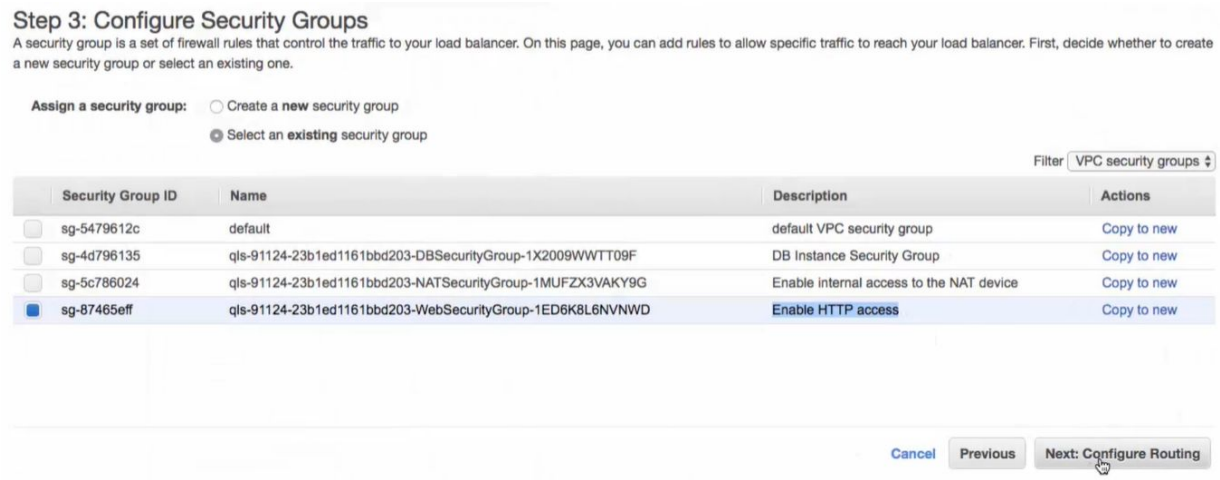
Under Security Settings, the first thing we need to select is My Lab VPC,just like the other Labs. Under Subnet us-west-2a, we want to make sure that we are listening on Public Subnet 1. Likewise under us-west-2b, listen to Public Subnet 2. We can now go ahead and click Next: Configure Security Settings as shown in the below figure.



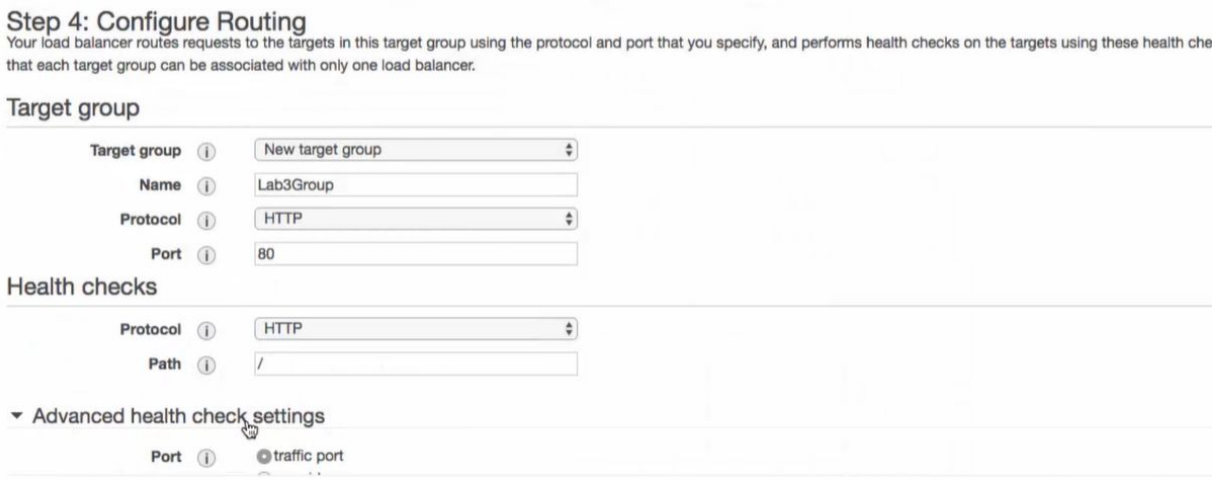
The warning that we are going to get is just notifying us that we are not currently leveraging the HTTPS protocol. This means that we are not leveraging secure connections.

This is absolutely something you would want to do in a production environment to make sure that all of the traffic being sent through the load balancer is being delivered in an encrypted state. We are going to continue on and click on Configure Security Groups.

In the Step3, we can select one that has previously been created for us which is the Enable HTTP access. You can find the name under the description (as shown in the figure). Click Configure Routing.



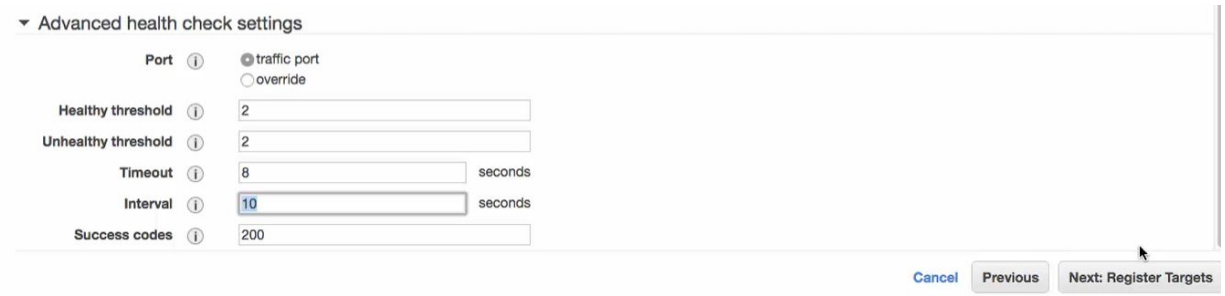
For our Target group in Step 4, we are creating a New Target group. Our name will be Lab3 Group. We are using HTTP Protocol on port 80. Our health check will also be running in HTTP protocol.



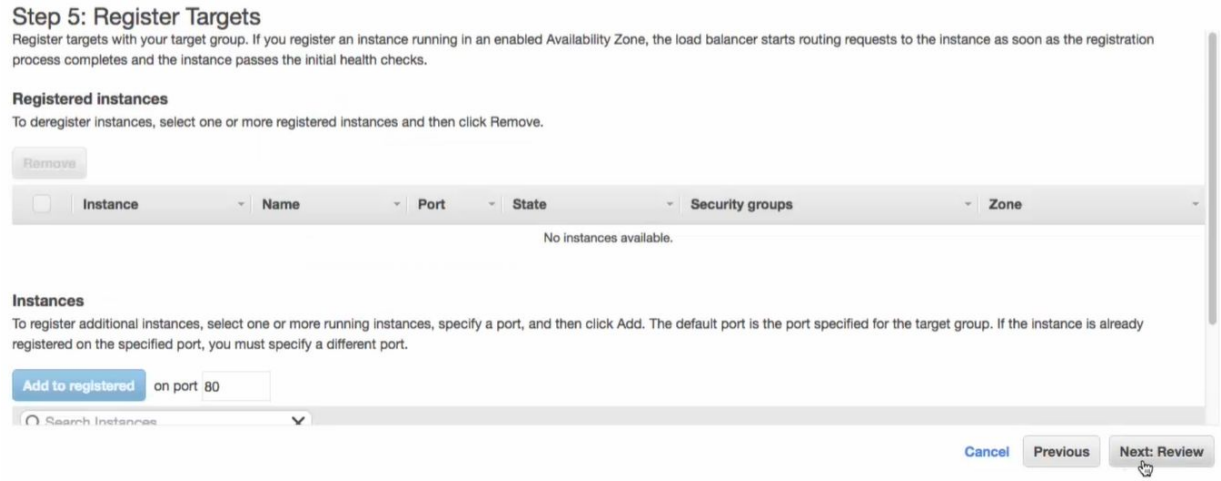
We do want to configure a couple of Additional settings. So we are going to expand the Advanced health check settings and scroll down. We are going to set our Healthy threshold to 2.

This indicates that an Instance only needs to pass a health check twice in order to be marked healthy. For the time out, we are going to set this to 8 seconds per request. We are going to run these requests every 10 seconds. So for an instance to be marked healthy, it will need to pass 2 times once every 10 seconds.

So it will ping at 10 second intervals and it will need to pass 2 in a row to be marked as healthy. Now go ahead and click Next: Register Targets.



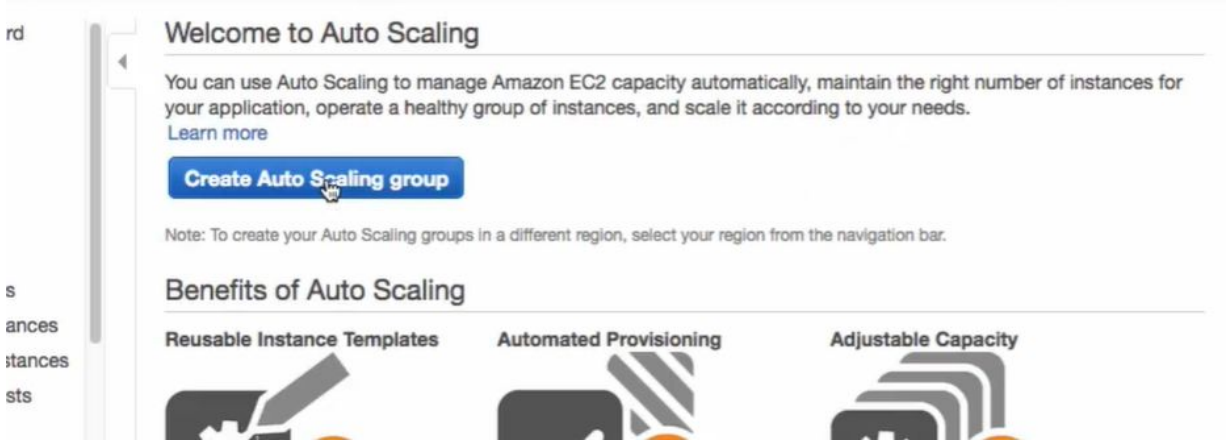
And then Next to review. Review is the 6th and the last step.



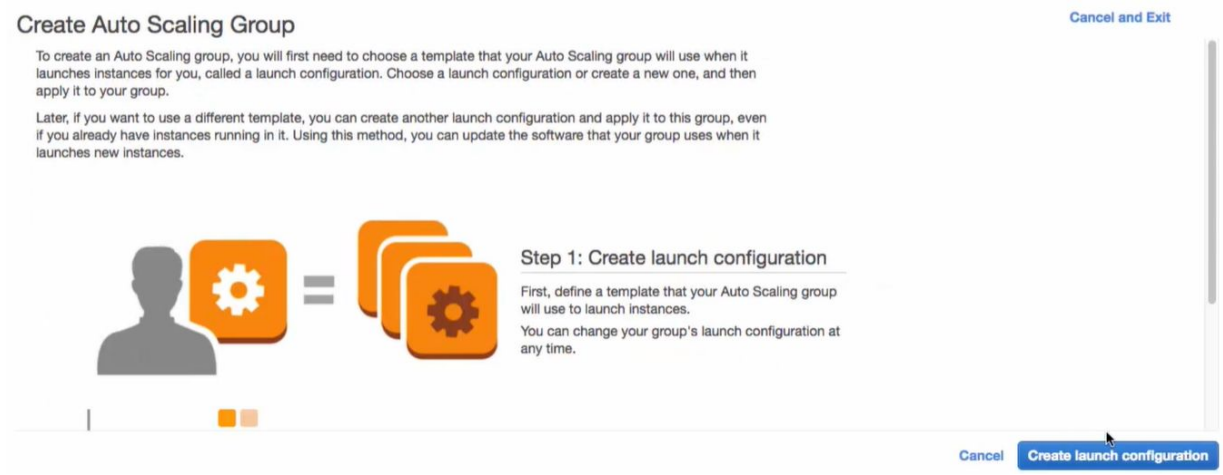
So, we take a look to review and make sure that all of the settings are what they are supposed to be. Once the complete reviewing of the details are done, we can go ahead and click Create. This is how Load Balancer is being created and we are ready to continue on with the rest of the Lab.

The next test for our Lab is going to configure Auto Scaling. The first step in configuring Auto Scaling is to create a Launch Configuration.

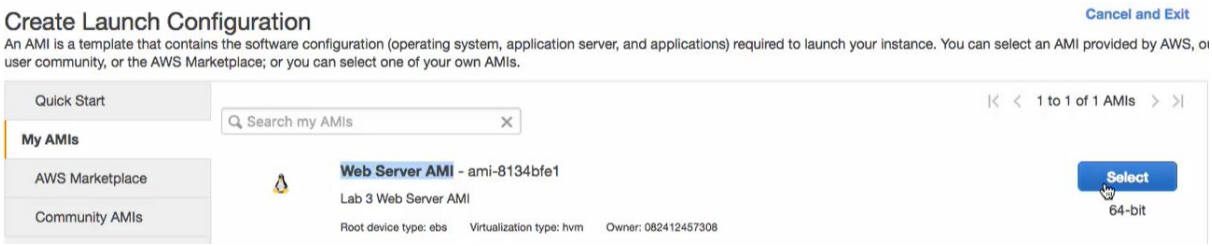
Under our Launch Configurations, the first thing we are going to click is Create Auto Scaling group.



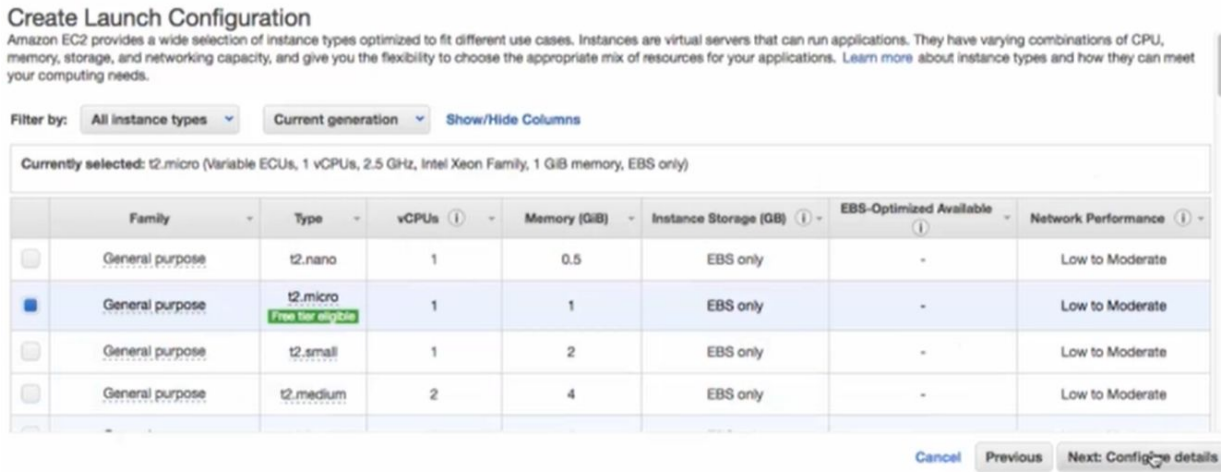
And under Create Auto Scaling group, we will do nothing else but to click Create Launch Configuration.



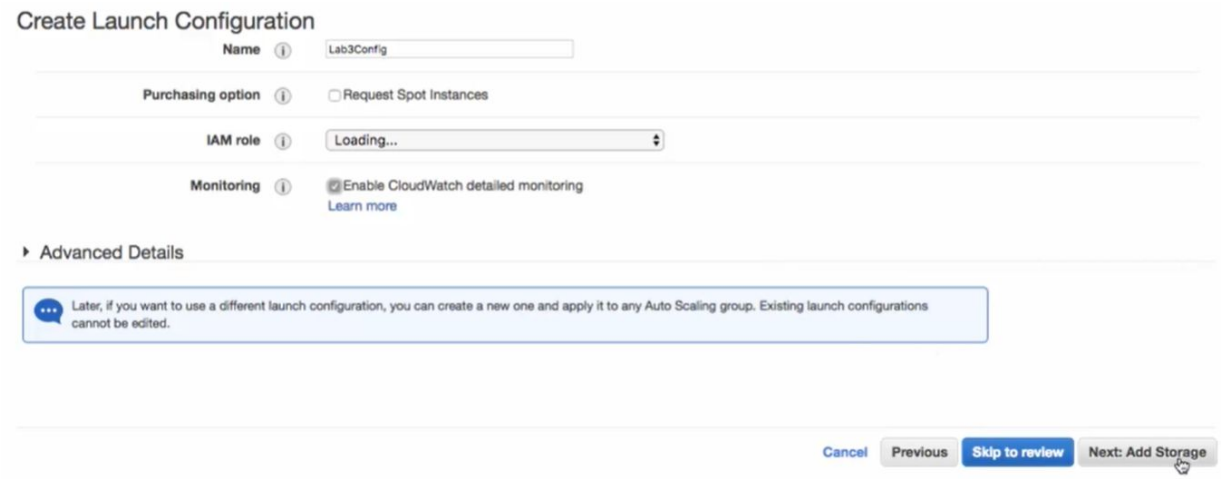
Under Create Launch Configuration, I am actually going to select My AMIs. This is going to allow us to select that Web Server AMI that we created at the beginning of the web. Click Select after that.



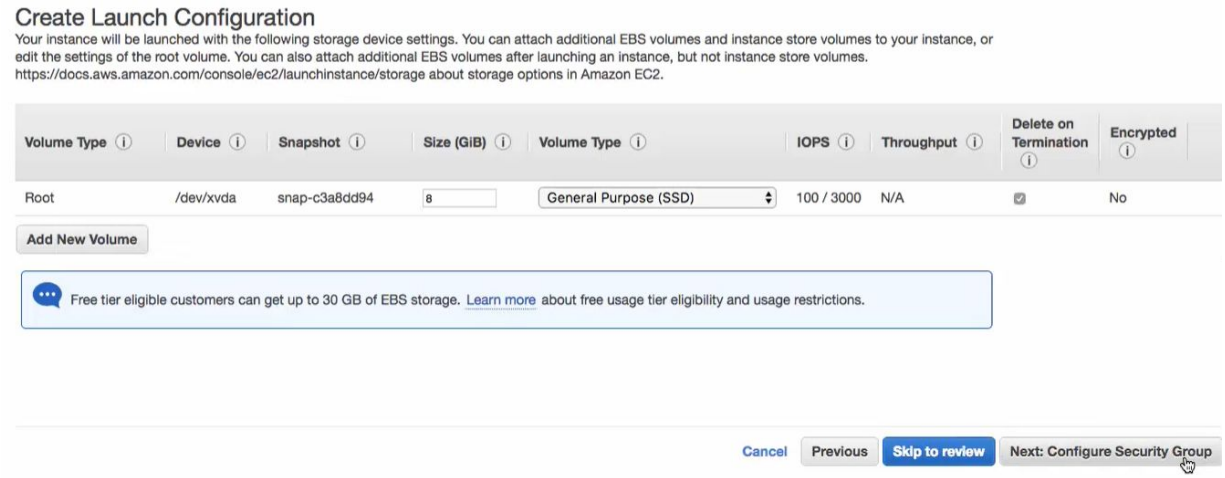
Once we select this, we are going to stick with the default of t2.micro and then click Next: Configure details as shown below.



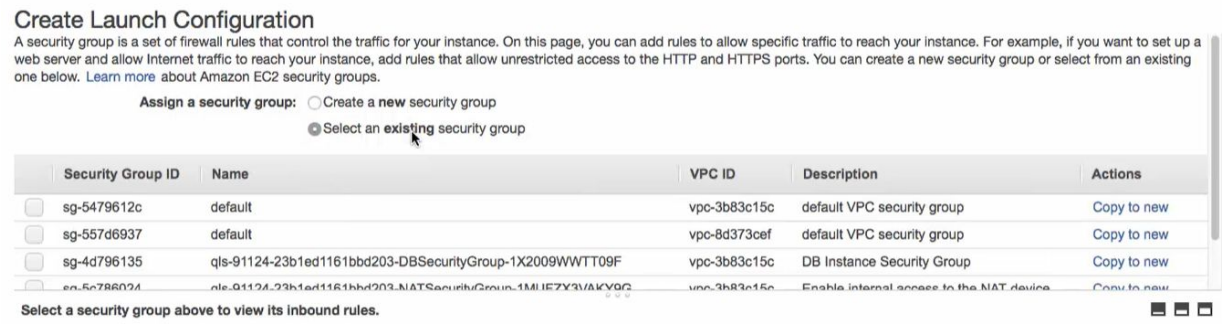
For the name, we are going to call this Lab3Config. Under Monitoring, we are going to check the box for Enable CloudWatch detailed monitoring. We do not need to worry about anything under Advanced Details so we will click on Next: Add Storage as shown in the figure.



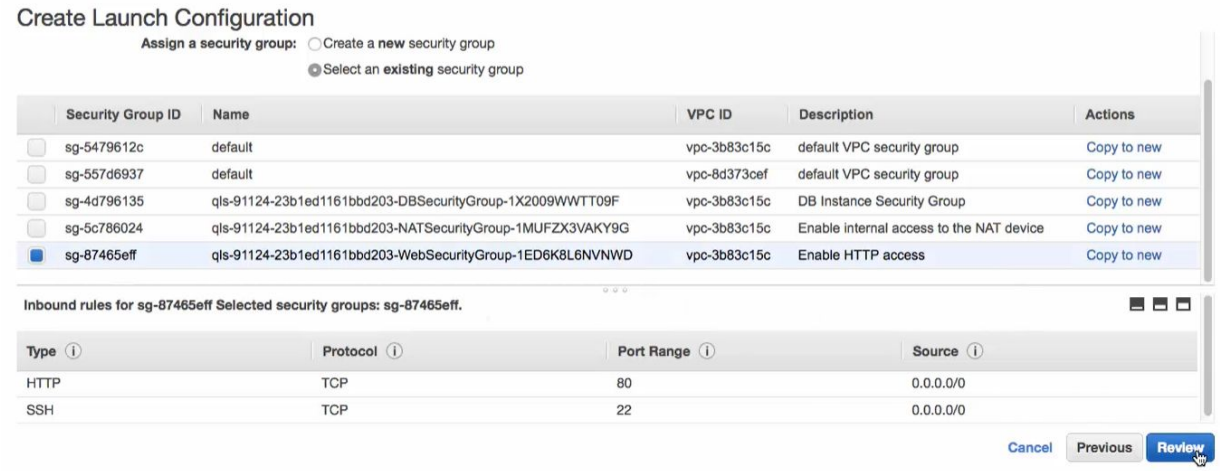
And then skip to Configure Security Group without making any changes to the existing window.



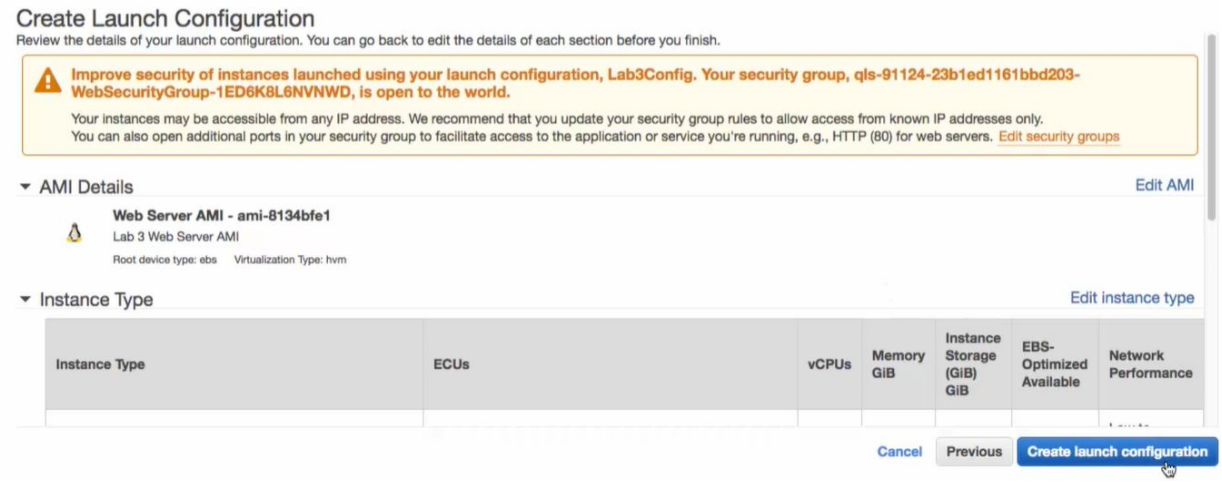
We are going to leverage an existing Security Group that has been created for us. We will be checking the box as shown in the screenshot.



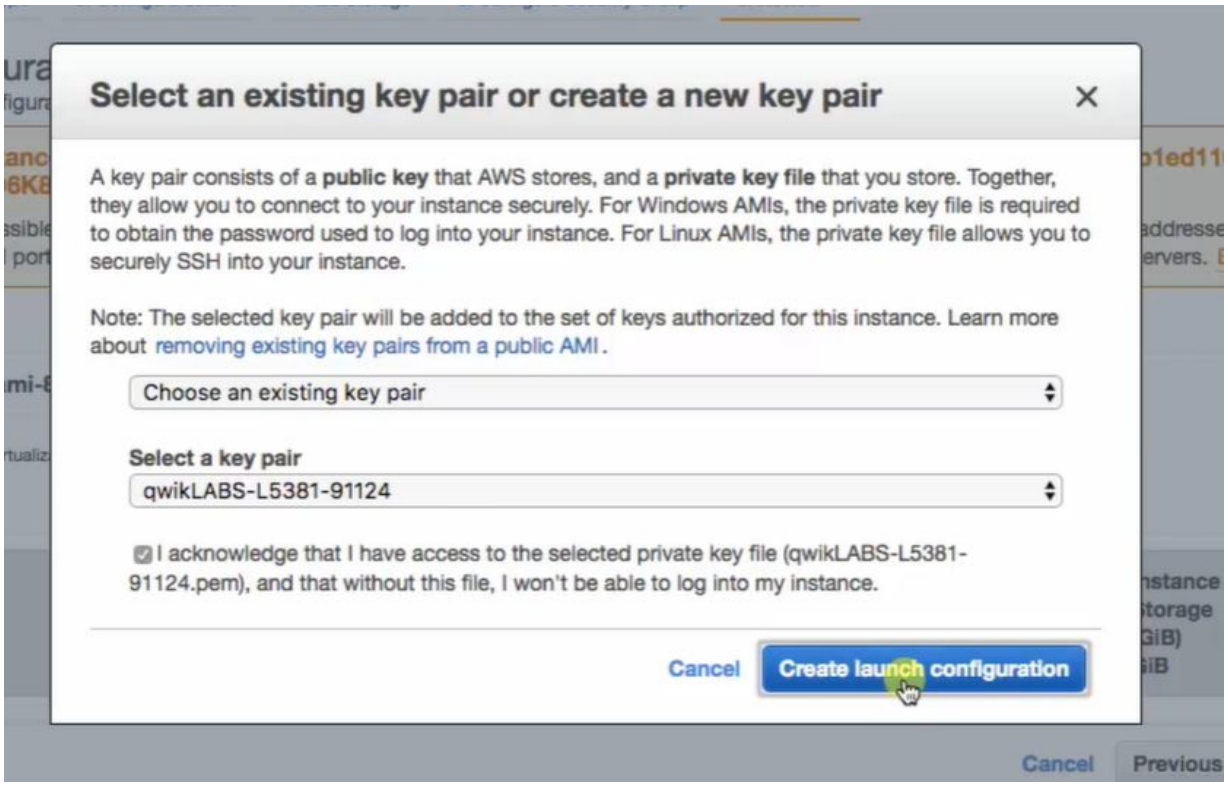
Here, we are looking for the item that says Enable HTTP access. So go ahead and select that and click on Review.



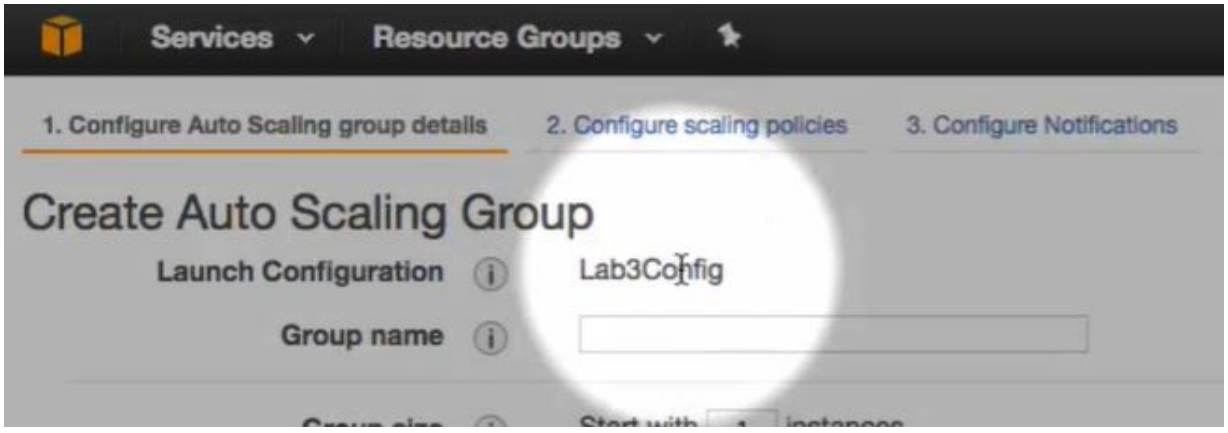
Now you will notice in the below screenshot that we have a warning, stating that this security group is open to the world. This is simply letting us know that the Security Group that we selected has a rule set in it that has 0.0.0.0/0 as the source of any packet. This means that it is listening to the open Internet. For the purposes of this lab, we are going to go ahead and continue with Create launch configuration.



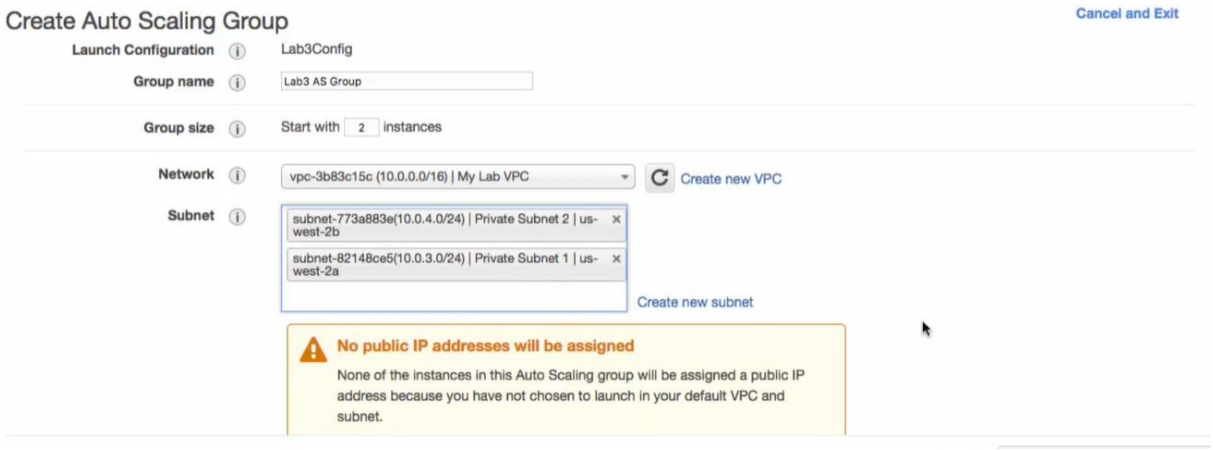
After doing so, you will get another popup window, here we will acknowledge that we have access to the key pair and click Create launch configuration.



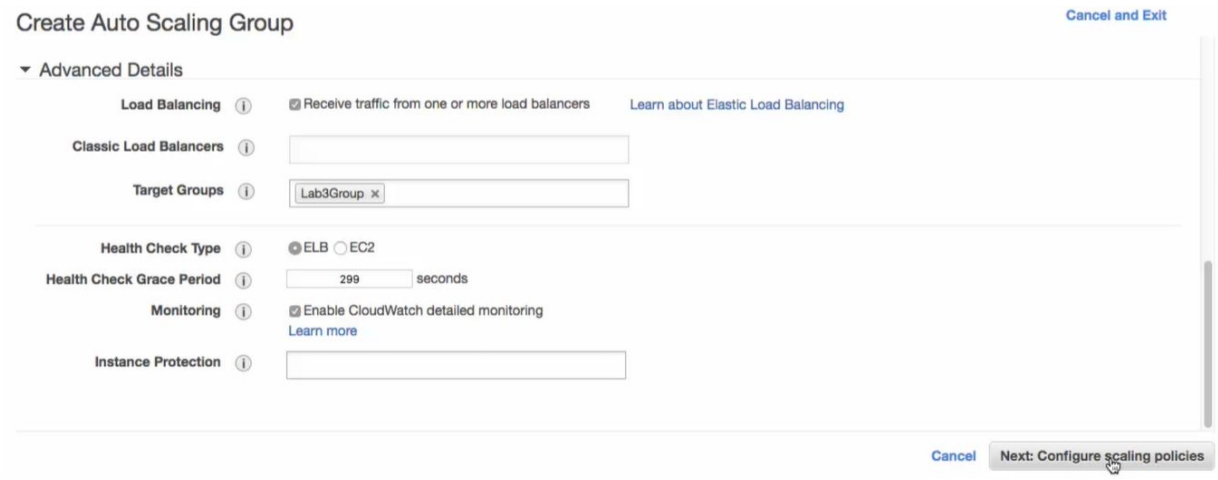
Now we are ready to create the Auto Scaling Group that this Launch Configuration will be a part of. We can see our Launch Configuration listed here is the first property as shown here.



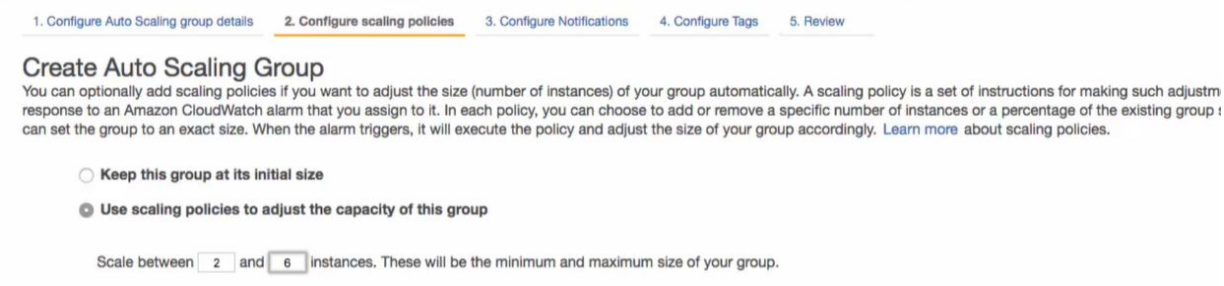
Our Group name is going to be Lab3 AS Grouplab 3 (AS stands for Auto Scaling). We want to start off with 2 instances and for the network, we need to make sure we select My Lab VPC. For Subnet,we need to put both of our private sublets. Select Private Subnet 2 and 1.



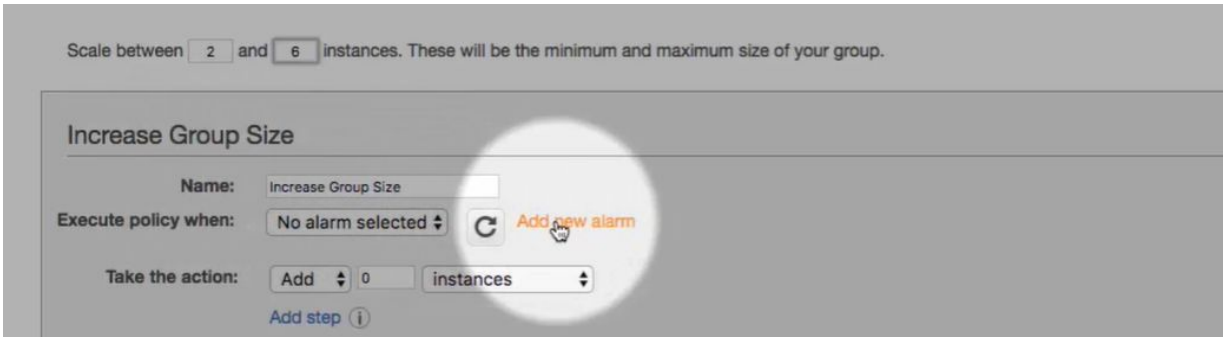
On the same page if we scroll down, we would expand the Advanced Details. In Advanced Details, we are going to click the check box: Receive traffic from one or more load balancers as shown in the screenshot. Our Target Group will be Lab3 Group. Health Check Type will be ELB and not EC2.
What this will allow us to do is use the ELB health checks that we configured previously for the
2 consecutive with a ping every 10 seconds versus just the standard EC2 instance and host checks.
For Monitoring, we are going to select the check box: Enable CloudWatch detailed monitoring. This will bring us down to a 1 minute polling interval versus a 5 minute polling interval. And then we will click Configure scaling policies.



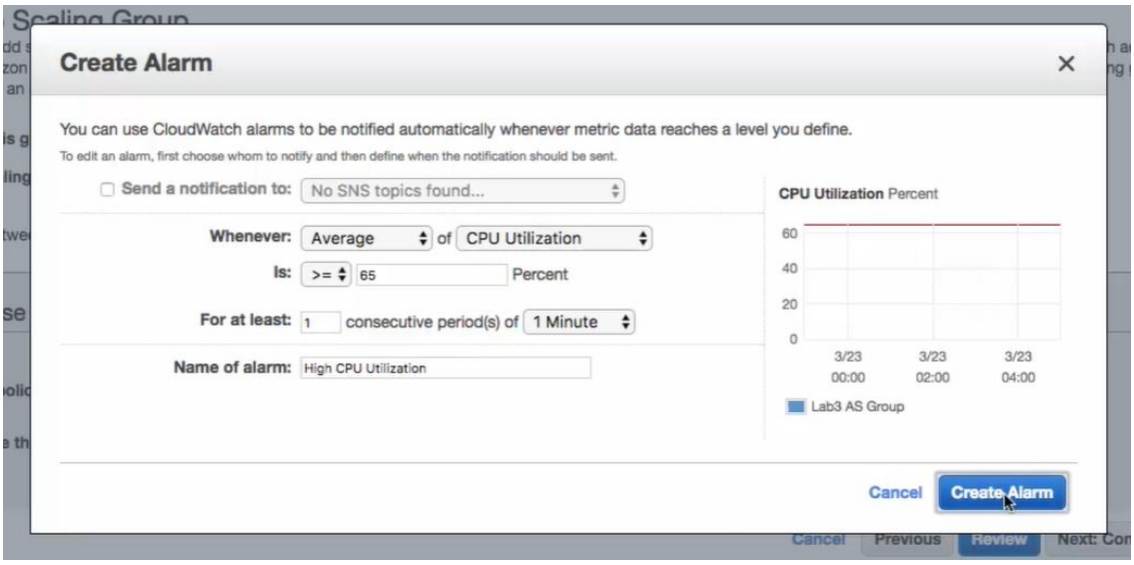
Under Configure scaling policies, we do want to leverage scaling policies to adjust the capacity of the group. We are going to scale anywhere between 2 and 6 instances.



In our Increase Group Size, we are going to select Add a new alarm.

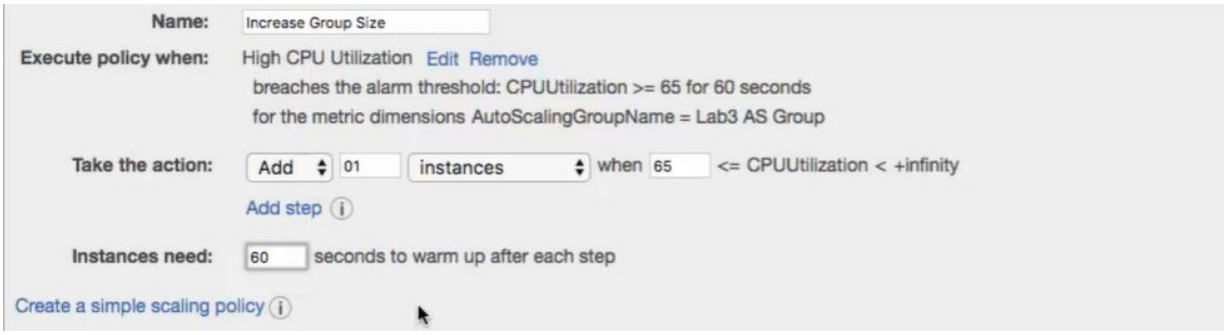


This will open a new window called Create Alarm. We are going to uncheck Send a notification to:.Next, what we are looking for is when the Average CPU Utilization is going to be greater than or equal to 65%. We are looking at this for at least 1 consecutive period of 1 minute (change details if any other time is mentioned). For the name of the alarm, it would be High CPU Utilization. When everything is set, go ahead and click Create Alarm.



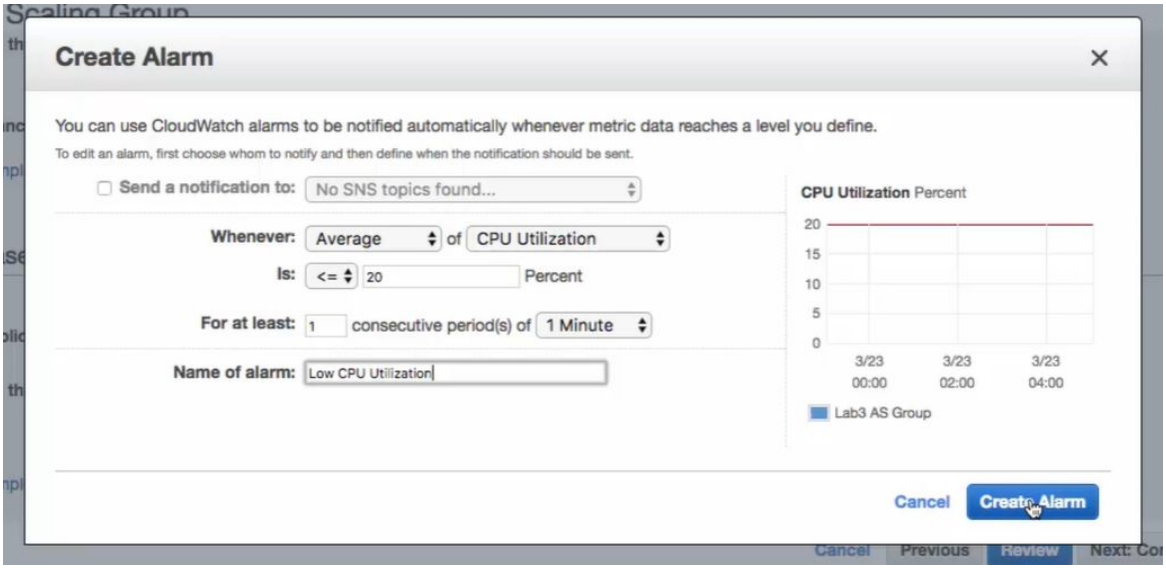
Now that we have created our Alarm, we need to make sure we know what is going to happen when that alarm is triggered. So when that alarm does get triggered, we are going to Take the action of Adding 1 instance whenever 65 <= (less than or equal to) CPU Utilization which is under Infinity. In this case, we are going to configure the action to add 1 instance whenever 65 is below our CPU Utilization or in other words, once CPU Utilization goes above 65.

We also need to tell Auto Scaling how long the instances need to come online. In this case, we are going to give them a full 1 minute (60 seconds).

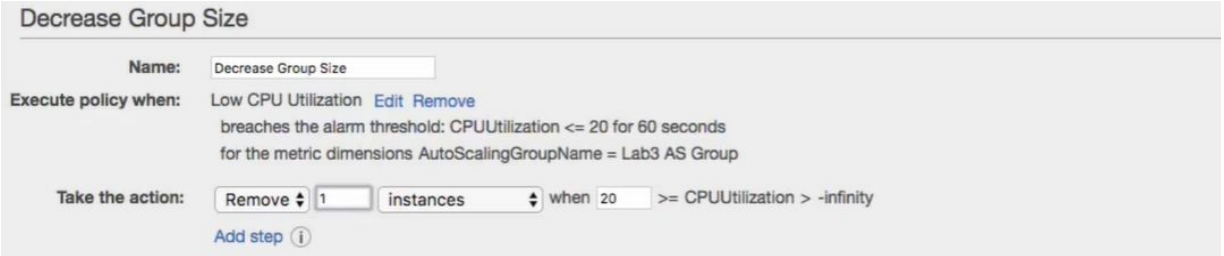


Now we have configured our Increased Group Size, we are going to go ahead and do the same for Decrease Group Size. So first we are going to click on Add a new alarm. This will open a popup window of Create Alarm just as what we previously did for the Increased Group Size.

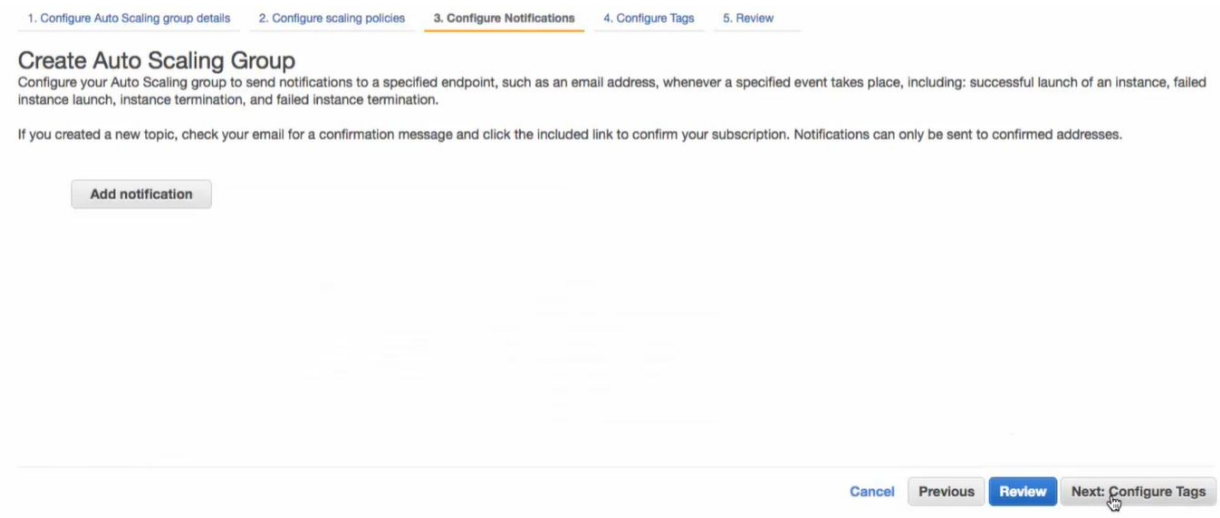
We are going to uncheck Send a notification to:.Next, what we are looking for is when the Average CPU Utilization is going to be less than or equal to 20%. We are looking at this for at least 1 consecutive period of 1 minute (change details if any other time is mentioned). For the name of the alarm, it would be Low CPU Utilization. When everything is set, go ahead and click Create Alarm.



Now in Take the actions, we are going to remove 1 instance when 20 is greater than or equal to CPU Utilization or in other words one CPU Utilization drops below 20.

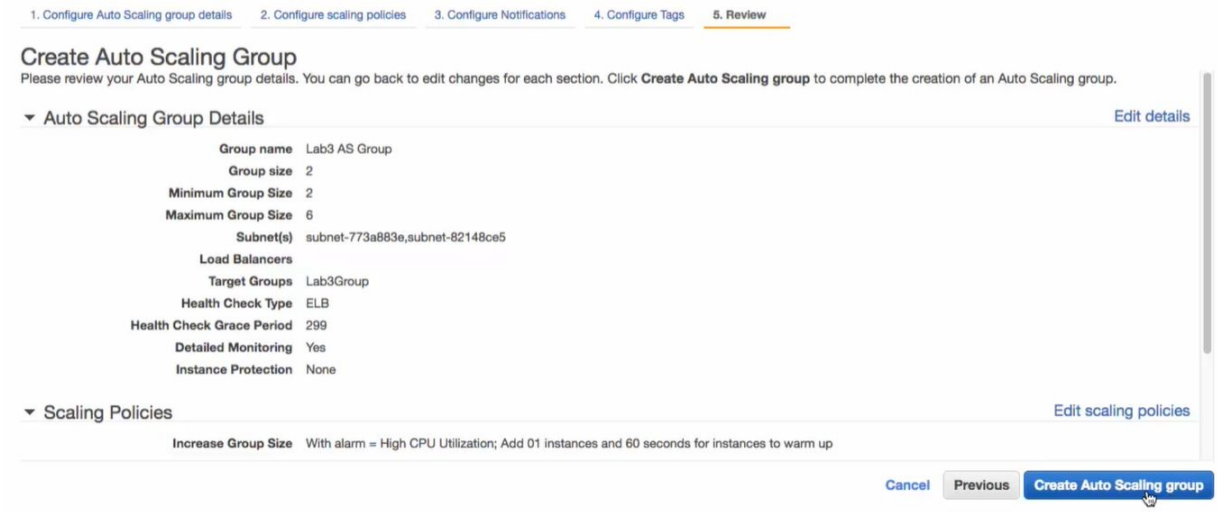


With the scaling policies done, we can go ahead and click Configure notifications and Configure Tags.

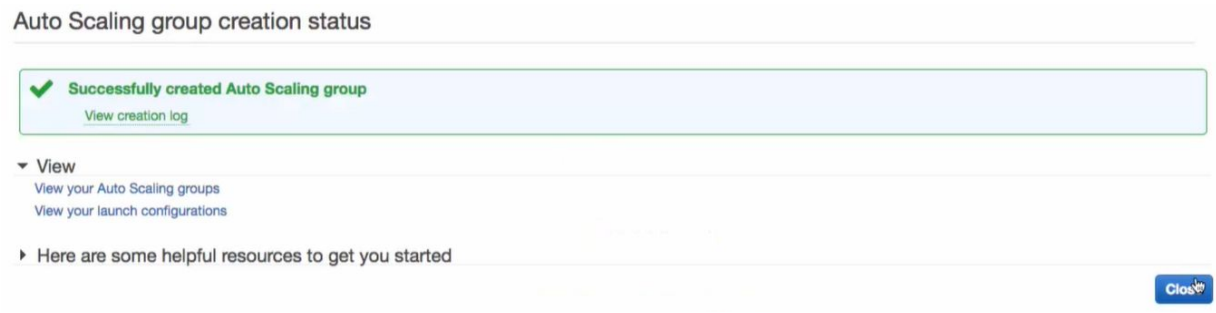


Under Tags, these are tags that will be applied to each EC2 Instance that gets launched. So we would like to be able to append a Name of Lab 3 Web Instance to each one of those instances as they come online. Go ahead and click Review.

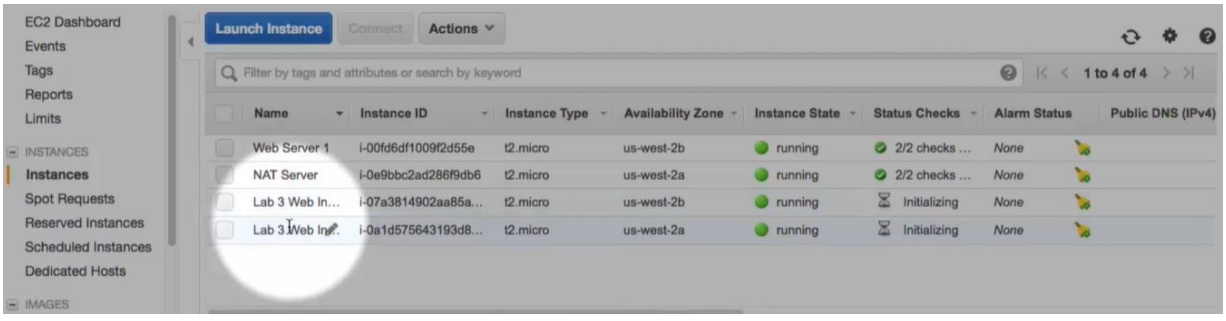
And then click on Create Auto Scaling group



Once the Auto Scaling group has been created, we will go ahead and click Close and everything is ready to go.



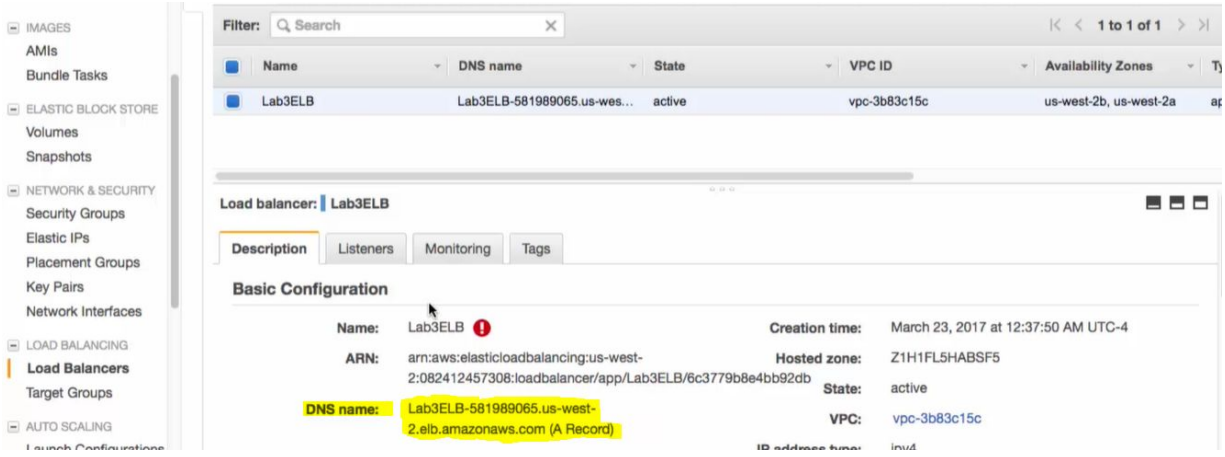
Now we told our Auto Scaling group that we would like 2 instances running at any time. It is what is under our desired number. We can validate that it has gone ahead and launched those by going under instances.



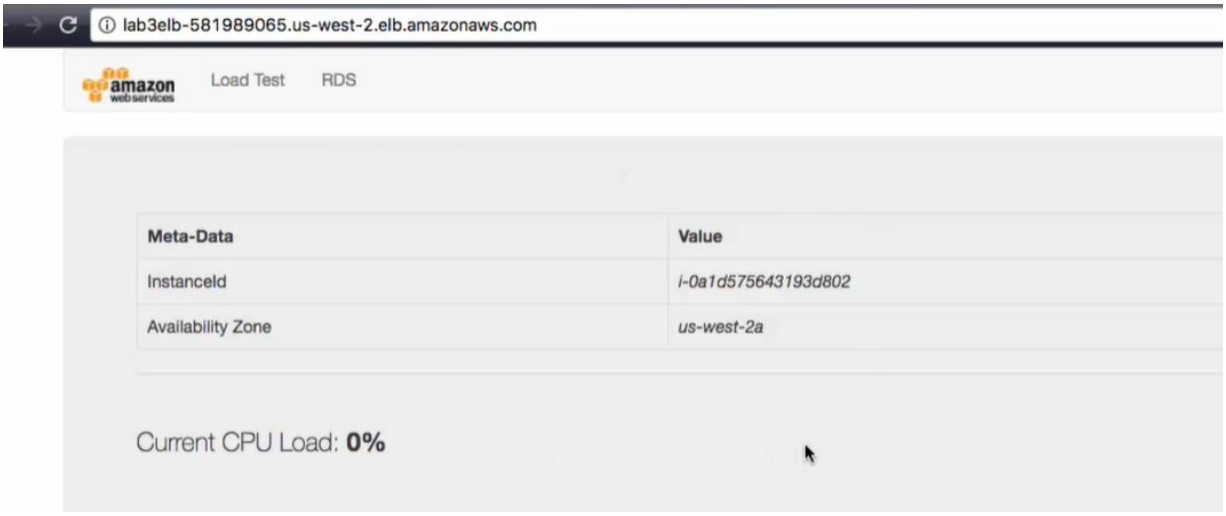
Now that we see that we have both our Lab 3 Web instances running. We can also verify that they are connected to our Load Balancer. For that, we go over to the navigation panel on the left hand side and scroll down to Target Groups. Click it.

Once we select our target group, we can see that we have the Lab 3 Group and other targets as shown below. If we scroll down we can see that registered instances are both the Lab 3 web instances that have started returning a healthy status.

Next, we are going to want to validate that the Load Balancer is properly sending traffic and returning requests from our Load Balancer instances. So if we go under Load Balancers, we can spot the Lab3ELB. Under the description, we will be able to find the DNS name of this Load Balancer as highlighted in the screenshot.



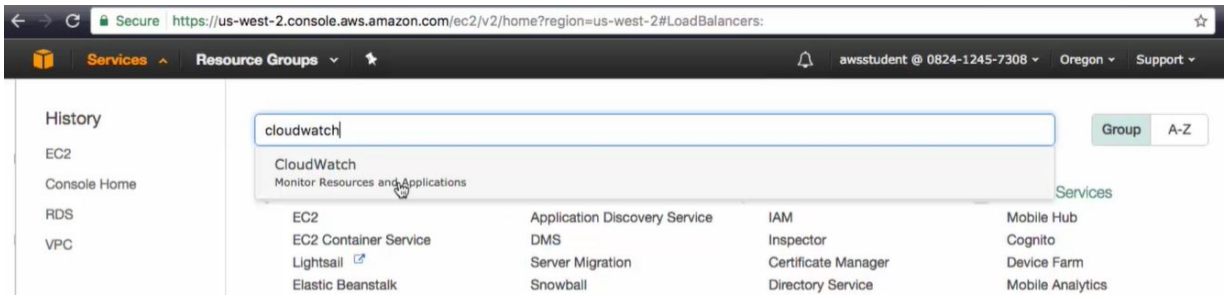
So we will copy that and open it in a new tab. We have got the same web page from Lab #1 and Lab #2. We can see the instance ID.



And if we refreshed the page we can see that that instance ID changes to the other instance ID in our Auto Scaling Group.

Now in order to test our Auto Scaling, we know that we configured our alarms to trigger based on CPU load. Currently, we can see that the CPU load is 0% on this instance.

If we go back over to our Management Console and then to our Services, we are going to head over to CloudWatch to take a look at our alarms and what their current states are.

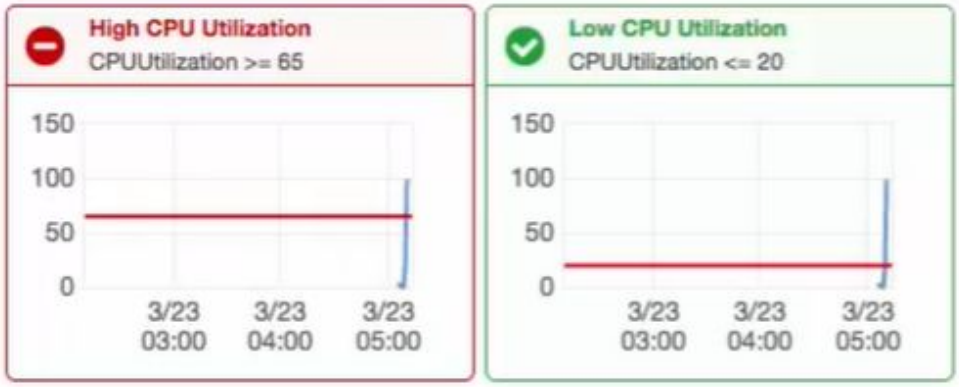


The Low CPU Utilization Alarm is currently in an alarm state (Red) meaning it has been triggered. We can see where the line has been drawn for this metric and we can see what our current data point is in the lower right hand corner.

We can see our high CPU Utilization is currently in Green meaning an okay state. We see the line that has been drawn and what our current data point is.

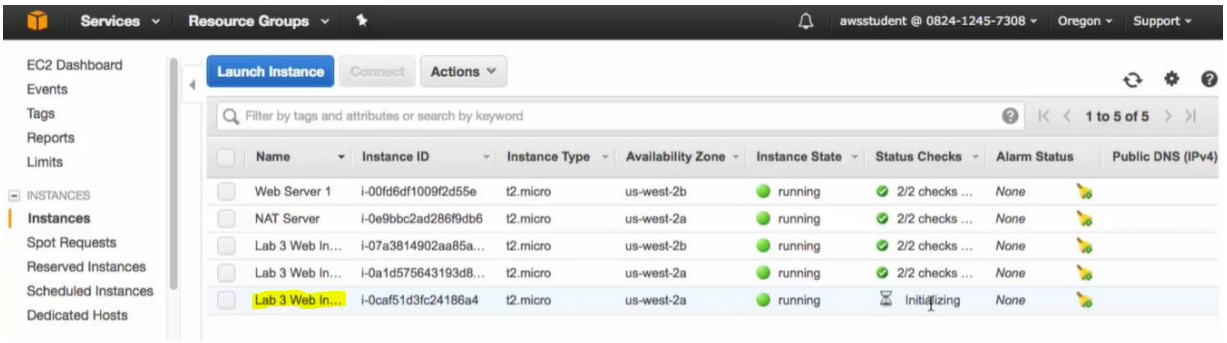
Now we want to go ahead and trigger a high CPU Utilization. To do this, this lab has a Load Test that has been installed. The Load Test is automatically going to generate a CPU load on the system.

So if we go back over to ClubWatch, we can track the CPU Utilization because of the Load we just generated. Go ahead and refresh to see the changes. It has been about 2 minutes since the last time we went to the load test and we can see that the alarms have already switched states.



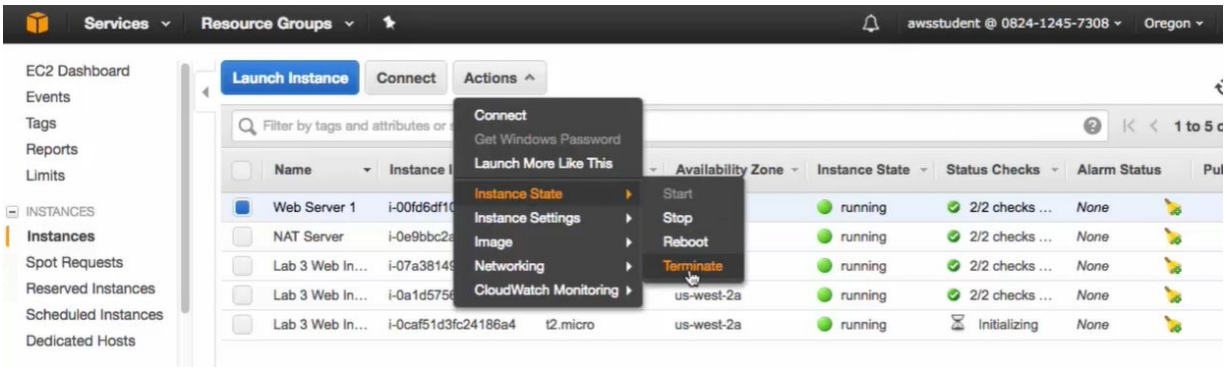
So high CPU utilization is currently in the alarm state while low CPU utilization is in the OK State. We can see our data points have both moved beyond the lines which has actually triggered our alarm. Now because we triggered this, we should be able to see that instances have been launched.

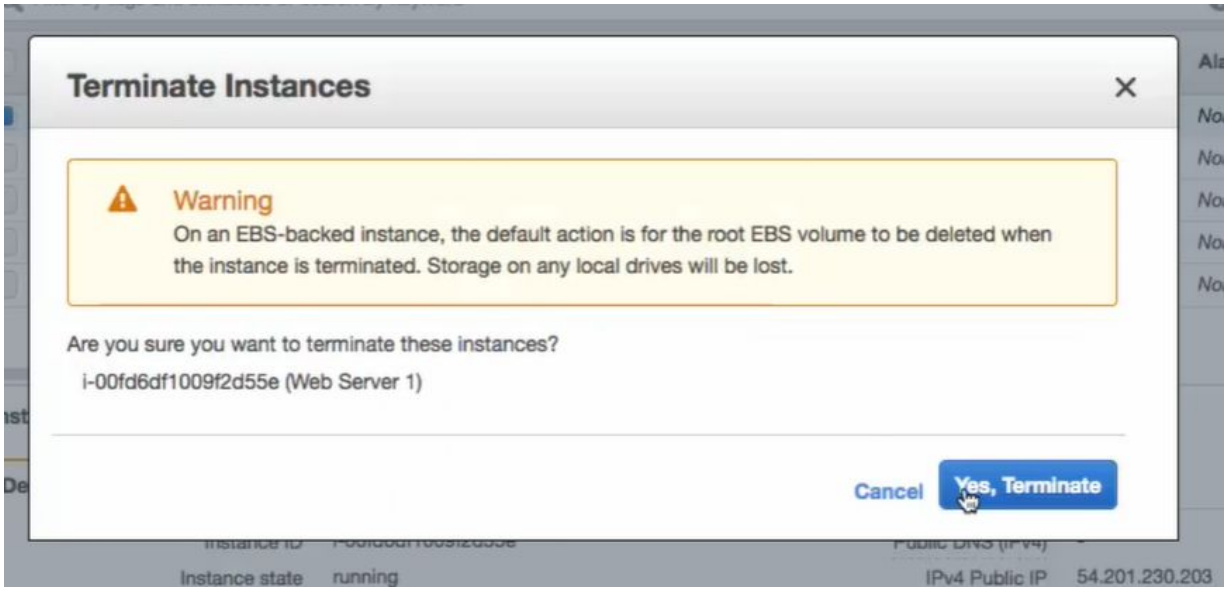
Back over to EC2 and under Running instances, we can see that another lab has launched three web instances. We see that it is currently initializing. So Auto Scaling has just kicked that off.



Now that we have validated that our Load Balancer is working in conjunction with our Auto Scaling Groups to serve traffic to those back end Private Subnet instances. We actually no longer need Web Server 1 sitting in the Public Subnet.

So we can select Web Server 1 and terminate that server as shown in the screenshots.





It is a good job for Lab #1 and 2. But now that we have reached the end of Lab #3, it is no longer necessary.

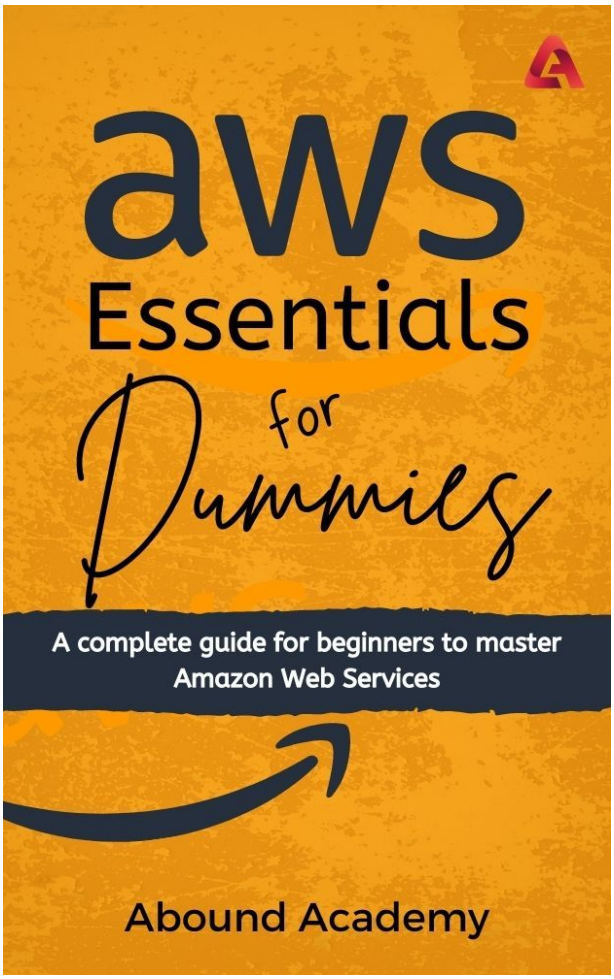
We have reached a desired configuration state of having our load balancer directly delivering traffic back to instances in a Private Subnet. These instances are connected back to the databases and ready to continue serving traffic.

And that brings us to the end of Lab #3.

Conclusion

You should now have a foundational understanding of AWS Core Services, Compute, Storage, Networking and Database. See how they can serve as building blocks for your Cloud based solutions.

You should also be familiar with the services to help you manage your infrastructure along with the methods and tools to secure your environment. To learn more please visit the AWS website.
https://aws.amazon.com/


aws
Essentials
for
Dummies
A complete guide for beginners to master
Amazon Web Services
Abound Academy