AWS Config is used as a dectective and preventive guardrail to check the configuration compliance around your environment

Scp security control policies is used to deny certain activities over yr environment

Sample SCPs

1. VPC Flow Logs

[PreventUsersfromDeletingAmazonVPCFlowLogs.json](https://onedrive.live.com/embed?resid=8751035A609F4B69%21965&filename=PreventUsersfromDeletingAmazonVPCFlowLogs.json&authkey=!AMbrAu_7VrAFBl4)

1. EC2 Instance Type

[Require AmazonEC2InstancestoUseaSpecificType.json](https://onedrive.live.com/embed?resid=8751035A609F4B69%21966&filename=Require%20AmazonEC2InstancestoUseaSpecificType.json&authkey=!AKu7z35Ae1bJZTw)

1. Require MFA to Stop an Amazon EC2 Instance

[RequireMFAtoStopanAmazonEC2Instance.json](https://onedrive.live.com/embed?resid=8751035A609F4B69%21967&filename=RequireMFAtoStopanAmazonEC2Instance.json&authkey=!AI2VydvIObBRnoA)

1. Require a Tag Upon Resource Creation

[RequireaTagUponResourceCreation.json](https://onedrive.live.com/embed?resid=8751035A609F4B69%21968&filename=RequireaTagUponResourceCreation.json&authkey=!AMHekx6fX3CQ0Eg)

**Conclusion**

In this session, we have seen how to patch AWS Systems Manager Managed Instances across multiple AWS accounts using Automation. We started by showing you how to create the required IAM service roles used by Automation in the management and target accounts. After you created your IAM roles, we saw how to create a custom Automation Document for executing patch baseline operations. Following this, we saw how to execute Automation Documents that targeted your managed instances via resource groups in target accounts.

You can customize your workflow further by creating your own Automation Document based on the document AWS-PatchInstanceWithRollback.

After you have created your Document, you can write additional steps and add them to the workflow.

Example steps include:

* Updating an ALB Target Group to remove the target during patching operations and add the target back to the group following successful patching (aws:executeAwsApi action type)
* Validating your OS-level application states using Run Command with the AWS-RunPowerShellScript or AWS-RunShellScript Documents (aws:runCommand action type)
* Creating an AM

**Feb 15, 2020: First Installment Due: $1,000**

Week1-Day 1:

* Cloud Practitioner Essentials
  + Define what the AWS Cloud is and the basic global infrastructure
  + Describe the key services on the AWS platform and their common use cases
  + Cloud Terminologies
  + Describe basic AWS Cloud architectural principles
  + Setup AWS Accounts

Week1-Day 2:

* Cloud Practitioner Essentials
  + Describe basic security and compliance aspects of the AWS platform and the shared security model
  + Describe the billing, account management and pricing models
  + Identify sources of documentation or technical assistance (e.g; whitepapers, support tickets)
  + Setup AWS Accounts

Week2-Day 1:

* Cloud Practitioner Essentials
  + Describe the AWS Cloud value proposition
  + Describe basic/core characteristics of deploying and operating in the AWS Cloud
  + Setup AWS Accounts

Week2-Day 2:

* Identify and Access Management (IAM)
  + IAM 101
  + IAM Lab

Week3-Day 1:

* AWS Object Storage and CDN-S3, Glacier and CloudFront
  + Introduction to S3
  + Create an S3 bucket lab
  + Version Control Lab (Skip for CI/CD or DevOps section)

Week3-Day 2:

* AWS Object Storage and CDN-S3, Glacier and CloudFront
  + Cross Region replication
  + Lifecycle Management and Glacier
  + CloudFront -CDN overview
  + S3 security and encryption

Week4-Day 1:

* AWS Object Storage and CDN-S3, Glacier and CloudFront
  + Storage gateway
  + Snowball
  + Transfer Acceleration

* Project: Create a static website using S3

Week4-Day 2:

* Exam 1

**Marc 15, 2020: 2nd Installment Due: $500**

Week5-Day 1:

* VPC
  + Introduction
  + Build your own Custom VPC

Week5-Day 2:

* VPC
  + Network Access Control Lists and Security Groups
  + VPC Peering

Week6-Day 1:

* VPC
  + VPC Peering
  + Transit Gateway
  + NAT Gateway
  + VPC Flow logs

Week6-Day 2:

* Compute
  + Introduction on EC2 instance
  + Launch an EC2 instance
  + Security Group -Basics
  + Creating an AMI -lab

Week7-Day 1:

* Compute
  + EBS Root volume VS instance store
  + The AWS command line and EC2
  + Using IAM role with EC2
  + EC2 Backup Strategies

Week7-Day 2:

* Compute Cont:
  + EC2 instance metadata
  + An EC2 Placement Groups -an exam must know
  + EFS Concepts and labs
  + Lambda Concepts

Week8-Day 1:

* Route 53
  + Simple routing policy-lab
  + Weighted routing policy-lab
  + Latency routing policy-lab
  + Failover routing policy-lab
  + Geolocation routing policy-lab
  + DNS Exam Tips

Week8-Day 2:

* RDS
  + RDS Introduction
  + Creating MySql Database
  + Creating and Restoring from Snapshot
  + Resizing Database

Exam

**April 15, 2020: 3rd Installment Due: $500**

Week9-Day 1:

* Security
  + Delegation-Cross Account Trust
  + Understanding Federation
  + CloudTrail
  + The AWS Organization
  + Trusted Advisor
  + Understanding and Mitigating DOS and DDOS attacks
  + Data Encryption

Week9-Day 2:

* AWS Secrete Manager
  + Introduction
  + Secrete Manager Demo
  + Demo for RDS and Secret Rotation
  + Auditing Secrets Access
  + Restricting Access to Secrets

Week10-Day 1:

* Operations
  + CloudWatch Events
  + KMS Essentials
  + IAM Review
  + Quiz

Week10-Day 2:

* Deployment with Autoscaling
  + Deployment Concepts with Autoscaling
  + AutoScaling Termination Policies
  + Suspending AutoScaling Policies

Week11-Day 1:

* Deployment with Autoscaling
  + AutoScaling Lifecycle Hooks
  + AutoScaling CLI

Week11-Day 2:

* High Availability and Business Continuity
  + Scalability with RDS Read Replicas
  + Automatic Failover with RDS Multi-AZ Deployments
  + Disaster Recovery Models
  + Understanding self-healing-based architectures

Week12-Day 1:

* RDS
  + Creating Read Replica
  + Configuring Multi Availability Zone
  + Monitoring

Week12-Day 2:

* **Applications**
  + **Introduction**
  + **Simple Queue Service SQS**
  + **SQS and EC2 AutoScaling**
  + **CloudFront**

**May 15, 2020: 4th Installment Due: $500**

Week13-Day 1:

* Applications
  + Kinesis
  + AWS Billing
  + Service Limits

Week13-Day 2:

* Applications
  + S3 Server Side Encryption
  + Logging
  + S3 to SNS

Week14-Day 1:

* Elastic Search
  + What is Amazon Elasticsearch Service?
  + Elastisearch Use Cases
  + SQS and EC2 AutoScaling
  + Master Node and Domain in ES
  + Demo of Elasticsearch and Kibana

**SA Professional/DevOps**

Week14-Day 2:

* DevOps
  + What is DevOps
  + What is CI/CD
  + AWS CodeCommit
  + AWS CodeBuild
  + AWS Deploy

Week15-Day 1:

* DevOps
  + AWS CodePipeline
  + AWS CodeBuild
  + AWS Deploy
  + Artifacts
  + Deployment Strategies
  + Quiz

Week15-Day 2:

* Configuration Management and IAC
  + Introduction
  + AWS CloudFormation
  + json & yaml introduction
  + CloudFormation Designer
  + Anatomy of CloudFormation template
  + User input
  + Userdata

Week16-Day 1:

* Configuration Management and IAC
  + Cfn-init
  + Intrinsic Functions
  + Change Sets
  + Stack Sets
  + CloudFormation & AWS CLI
  + AWS CloudFormation Lab
  + AWS CloudFormation Wait Conditions

Week16-Day 2:

* Configuration Management and IAC
  + AWS CloudFormation Deletion Policies
  + AWS CloudFormation Stack Updates
  + AWS CloudFormation Custom Resources
  + AWS CloudFormation Drift
* Quiz

**June 15, 2020: 5th Installment Due: $500**

Week17-Day 1:

* Configuration Management and IAC
  + AWS Managed Services
  + AWS Lambda
  + AWS Lambda Step Functions
  + AWS OpsWorks
  + AWS OpsWorks Lab
  + Quiz

Week17-Day 2:

* Monitoring Logging
  + Introduction
  + CloudWatch
  + CloudWatch Lab
  + Monitoring Types
  + More about metrics
  + CLI for metrics

Week18-Day 1:

* Monitoring Logging
  + Creating Alarms
  + Creating Billing Alarms
  + CloudWatch Logs Introduction
  + Streaming Logs to CloudWatch
  + Creating Metric Filters
  + Custom Metrics
  + CloudWatch Events

Week18-Day 2:

* Cloud Trail
  + Introduction
  + Creating a Trail and Viewing CloudTrail Events
  + IAM for CloudTrail
  + CloudTrail  Logs to CloudWatch

Week19-Day 1:

* AutoScaling & ELB
  + Introduction
  + What is AutoScaling
  + Creating a load balancer
  + Creating AutoScaling
  + ELB HealthCheck

Week19-Day 2:

* AutoScaling
  + Scheduled Actions
  + Scaling Policies
  + Life Cycle Hooks
  + Launching Spot Instances in AutoScaling

Week20-Day 1:

* Ansible
  + Setup & Configuration
  + Ansible Adhoc Commands

Week20-Day 2:

* Ansible
  + File Manipulation with Ansible
  + Installing software and Deamon
  + Advance Feature
  + Ansible Roles
  + Variables

**July 15, 2020: 6th Installment Due: $500**

Week21-Day 1:

* Jenkins
  + Introduction to CI/CD
  + Install Jenkins and connect to Dashboard
  + Connect Jenkins to Github

Week21-Day 2:

* Jenkins
  + Configure SMTP AWS Jenkins
  + Create Build Server and Connect Jenkins to Build Servers
  + Run Jobs from Github to AWS

Week22-Day 1:

* AWS ECS
* AWS ECS Lab

Week22-Day 2:

* AWS System Manager
  + Patching using Systems Manager
  + Post Deployment Activities using System Manager
  + Multi-Account multi-region patch management

Week23-Day 1:

* AWS System Manager
  + Patching using Systems Manager
  + Post Deployment Activities using System Manager
  + Multi-Account multi-region patch management
  + Parameter Store

Week23-Day 2:

* Governance and Compliance
  + AWS Config
  + AWS Organization
  + AWS Service Catalog
  + AWS Control Tower

Week24-Day 1: Exam4

Week24-Day 2:

* Application Deployments on Elastic Beanstalk
  + Introduction to Elastic Beanstalk
  + Deployment strategies Deep Dive with Elastic Beanstalk
  + Docker Deployments with Elastic Beanstalk
  + Using Elastic Beanstalk with CloudFormation

**August 15, 2020: 7th Installment Due: $500**

Week25-Day 1:

* DynamoDB
  + What is DynamoDB
  + Partition and Sort Key
  + Boto3 Operations for DynamoDB
  + Scan and GetItem

Week25-Day 2:

* CloudFront Essentials
  + CloudFront Architecture
  + Creating and working with Distributions
  + Working with Custom Origins
  + Optimizing Caching
  + Lambda@Edge
  + Logging, Reporting & Monitoring

Week26-Day 1:

* A/B Testing and Blue/Green Deployment
  + Introduction to Blue/Green Deployment
  + Update DNS Routing and Route53
  + Swap the ASG Behind the ELB
  + Update the ASG Launch Configuration
  + ALB Testing

Week26-Day 2:

* AWS Step Function and Batch
  + What is Step Function
  + Demo Step funtion
  + What is Batch
  + Demo Batch

Bonus:

Docker and Kubenetes

**Remotely Run Commands on an EC2 Instance**

Systems Manager is a Management Tool that enables you gain operational insights and take action on AWS resources safely and at scale. Using the run command, one of the automation features of Systems Manager, you can simplify management tasks by eliminating the need to use bastion hosts, SSH, or remote PowerShell.

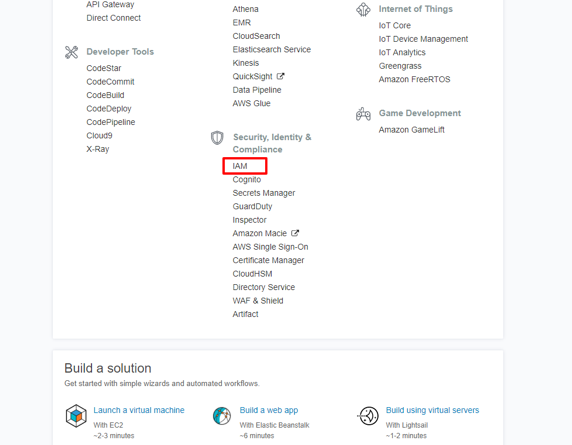
In our example scenario, as a System Administrator, you need to update the packages on your EC2 instances. To complicate this normally simple admin task, your security team does not allow you to direct access production servers via SSH or allow you use bastion hosts. Fortunately, you can use Systems Manager to remotely run commands, like update packages, on your EC2 instances.

To solve this challenging scenario, you will create an Identity and Access Management (IAM) role, enable an agent on your instance that communicates with Systems Manager, then follow best practices by running the AWS-UpdateSSMAgent document to upgrade your Systems Manager Agent, and finally use Systems Manager to run a command on your instance.

**Step 1. Create an Identity and Access Management (IAM) role**

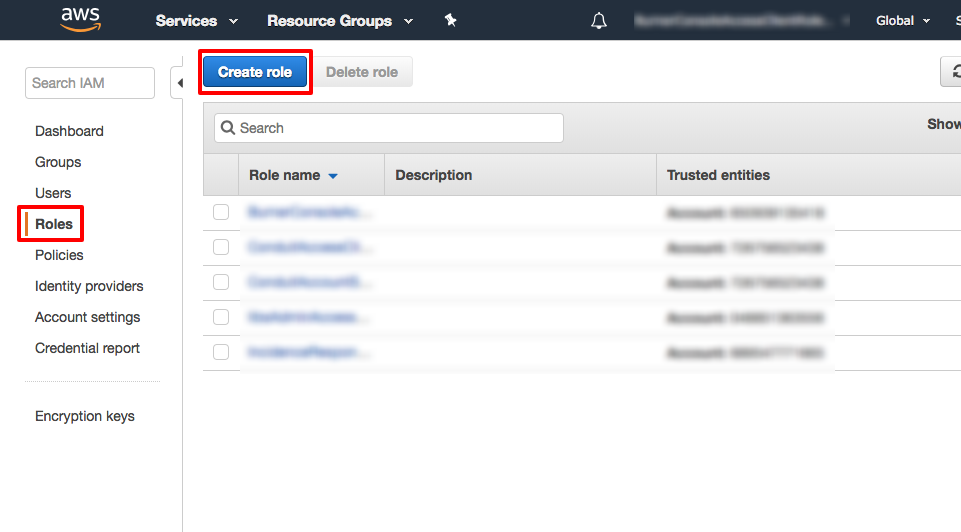
In this step, you will create an IAM role that will be used to give Systems Manager permission to perform actions on your instances.

a.  Open the IAM console at <https://console.aws.amazon.com/iam/>.



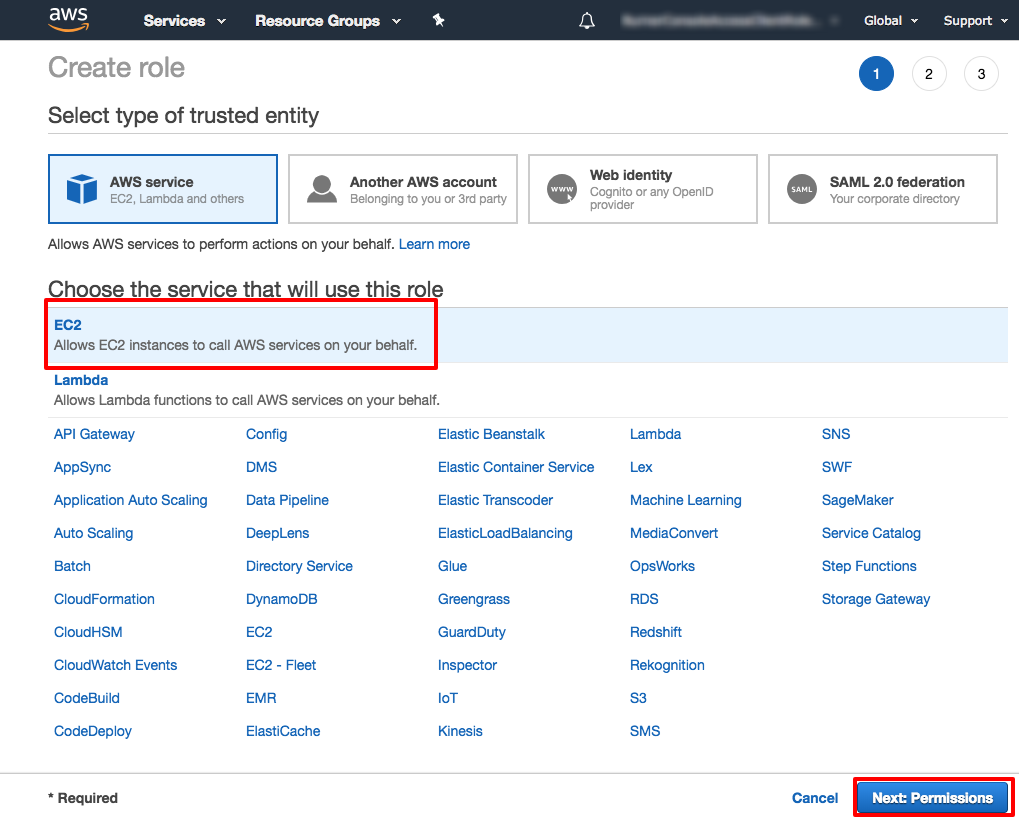
( click to enlarge )

b.  In the left navigation pane, choose Roles, and then choose Create role.



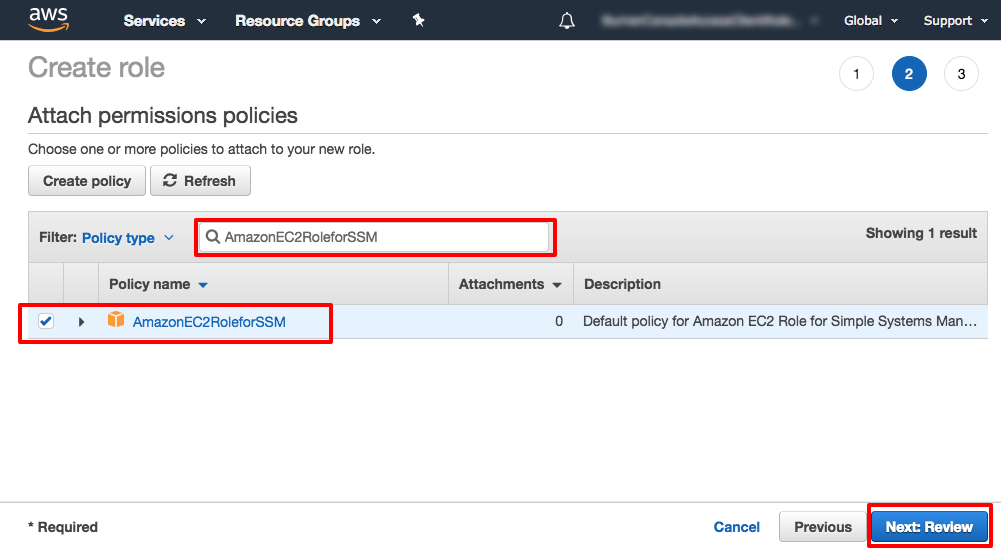
( click to enlarge )

c.  On the Select type of trusted entity page, under AWS Service, choose EC2, and then choose Next: Permissions.



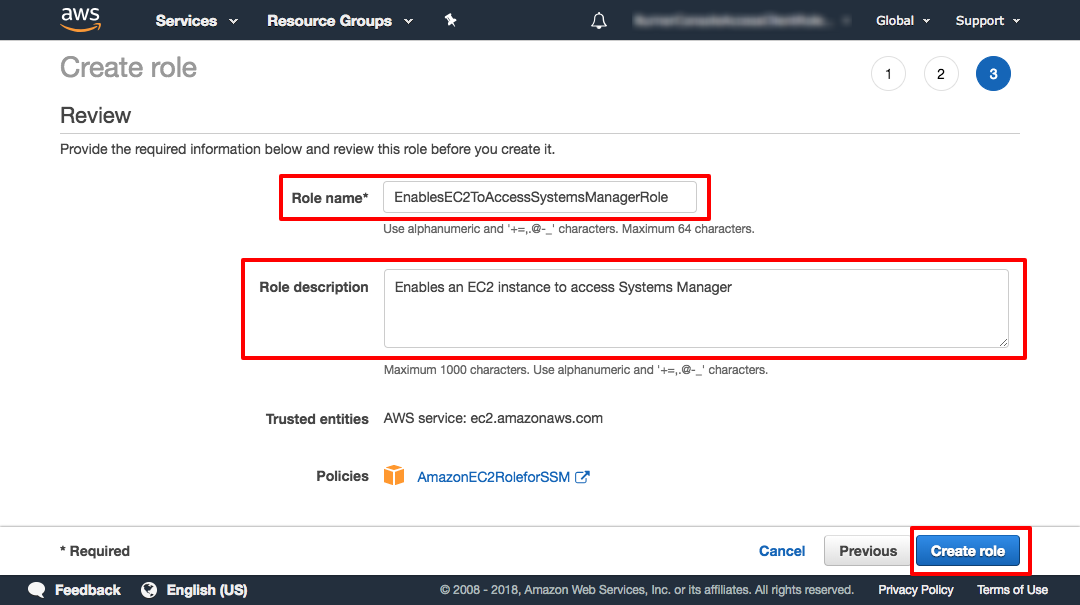
( click to enlarge )

d.  On the Attached permissions policy page, in the search bar type *AmazonEC2RoleforSSM* then from the policy list select AmazonEC2RoleforSSM, and then choose Next: Review.



( click to enlarge )

e.  On the Review page, in the Role name box type in *EnablesEC2ToAccessSystemsManagerRole*. In the Role description box type in *Enables an EC2 instance to access Systems Manager*. Choose Create role.

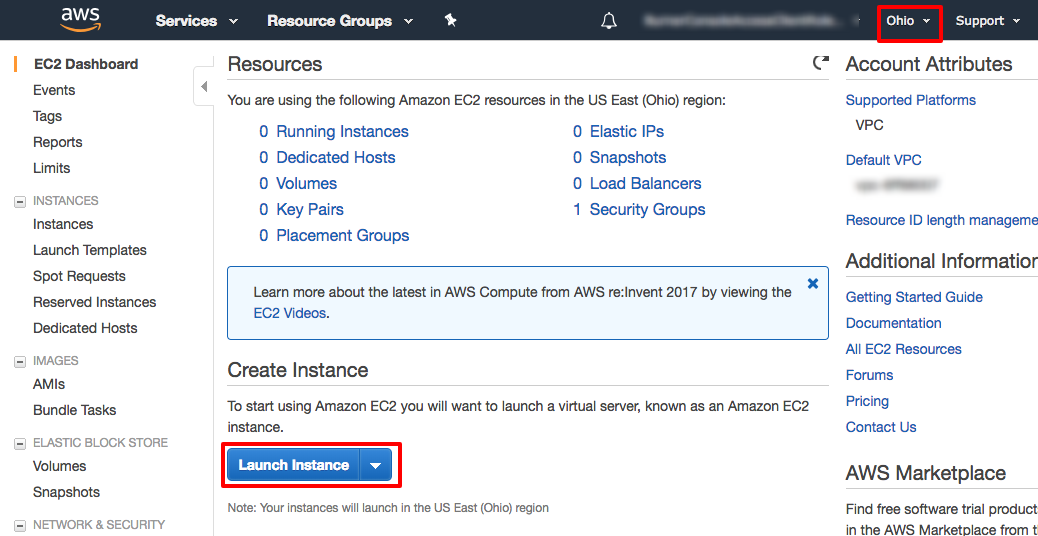


( click to enlarge )

**Step 2. Create an EC2 instance**

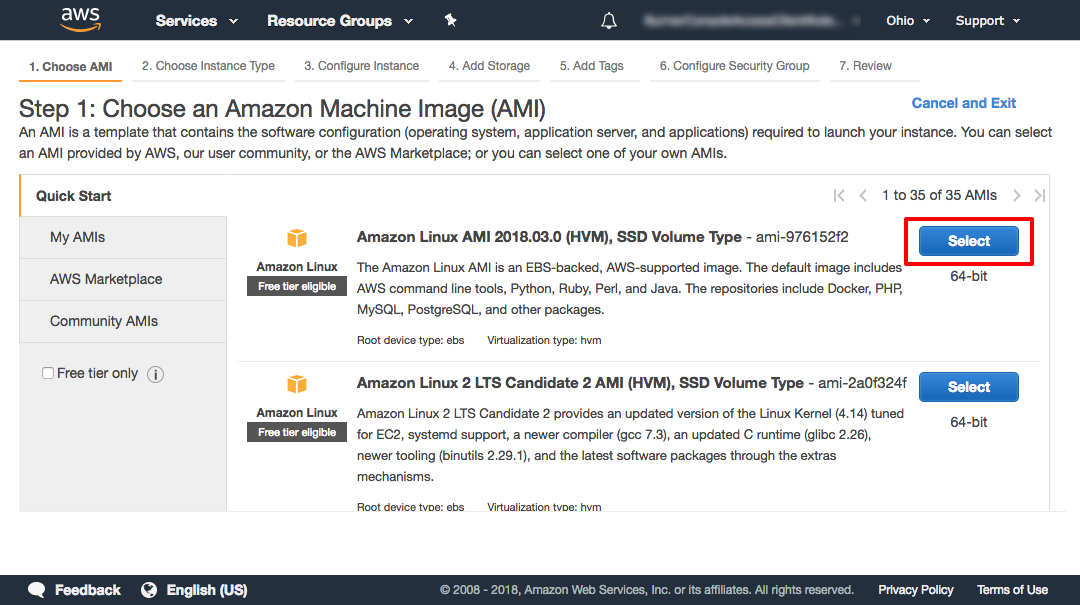
In this step you will you will create an EC2 instance using the EnablesEC2ToAccessSystemsManagerRole role. This will allow the EC2 instance to be managed by Systems Manager.

a.  Open the [Amazon EC2 console](https://console.aws.amazon.com/ec2/). From the EC2 console select your preferred [region](http://docs.aws.amazon.com/general/latest/gr/rande.html#ssm_region). Systems Manager is supported in all AWS Regions. Now choose Launch Instance.



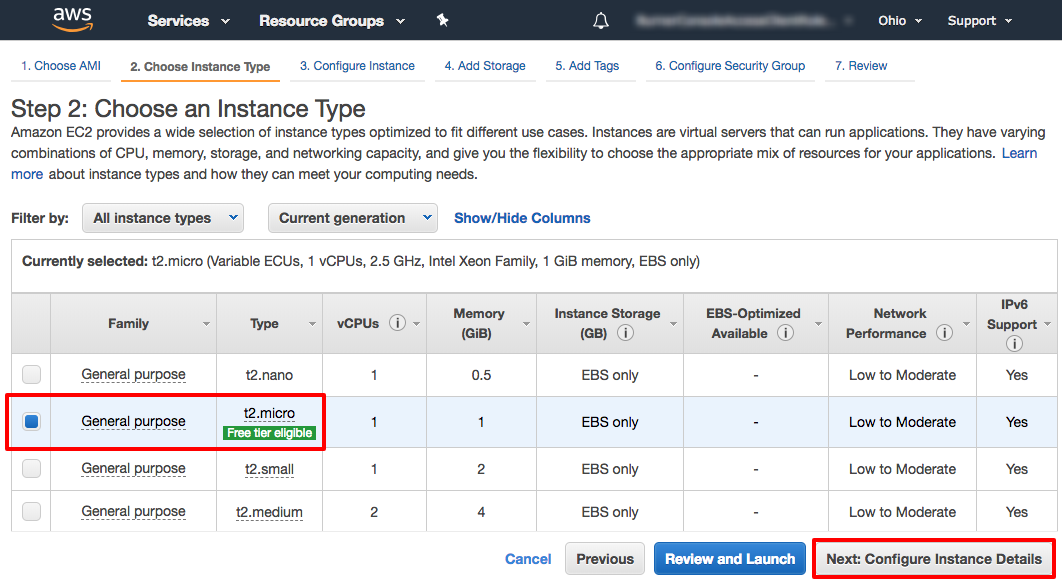
( click to enlarge )

b.  Select the *Amazon Linux AMI*. Make sure you select Amazon Linux base AMI dated 2017.09 or later which includes the Systems Manager Agent by default. You can also install the Systems Manger Agent on your own Windows or Linux system.



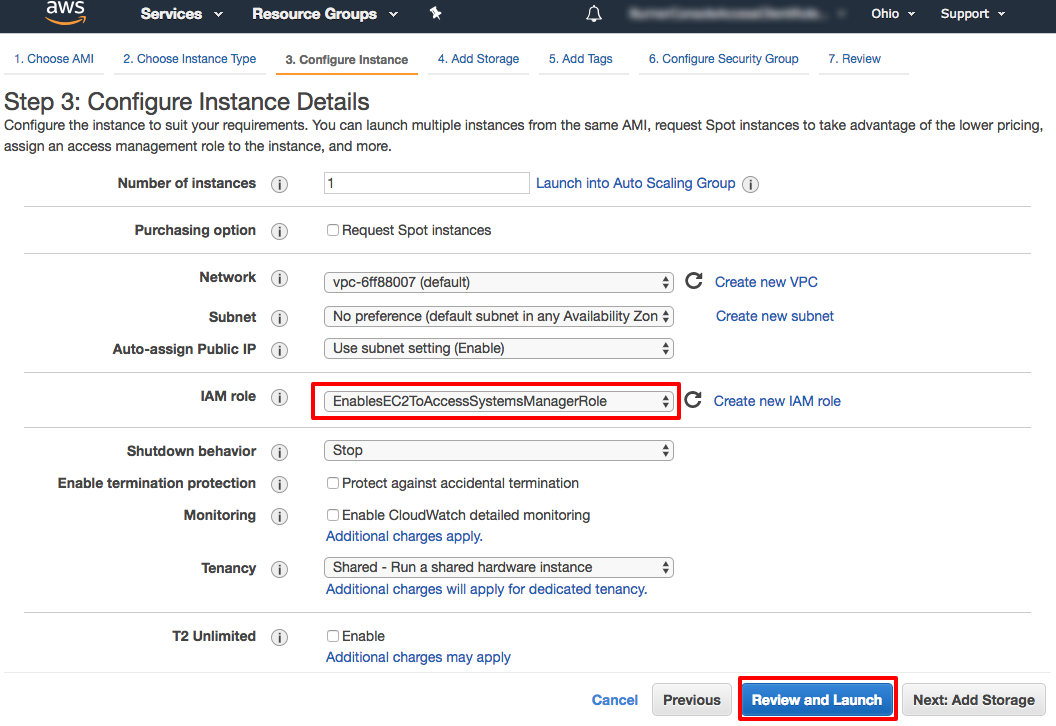
( click to enlarge )

c.  On the Step 2: Choose an Instance Type page, choose the t2.micro instance type and then click Next: Configure Instance Details.



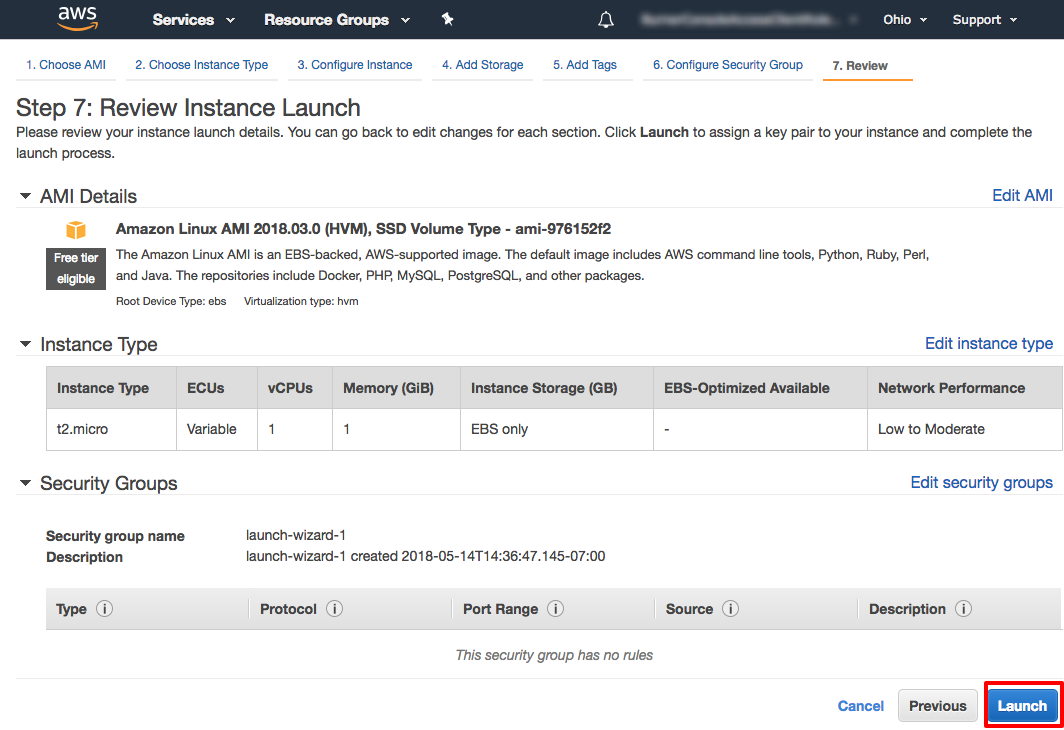
( click to enlarge )

d.  On the Step 3: Configure Instance Details page, In the IAM role dropdown choose the EnablesEC2ToAccessSystemsManagerRole role you created earlier. Leave everything else as default. Choose Review and Launch.



( click to enlarge )

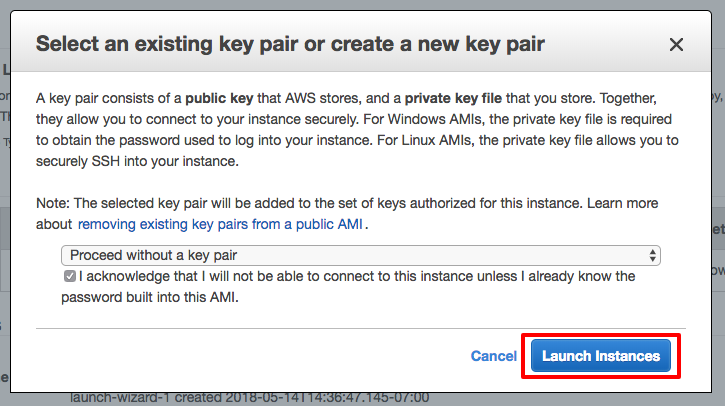
e.  On the Step 7: Review Instance Launch page, choose Launch to launch your instance.



( click to enlarge )

f.  Next the Select an existing keypair or create a new key pair dialog will appear. You will not need a keypair to use Systems Manager to remotely run commands. From the Choose an existing pair dropdown choose Proceed without a key pair and tick the I acknowledge that… checkbox.

Next select Launch Instance.

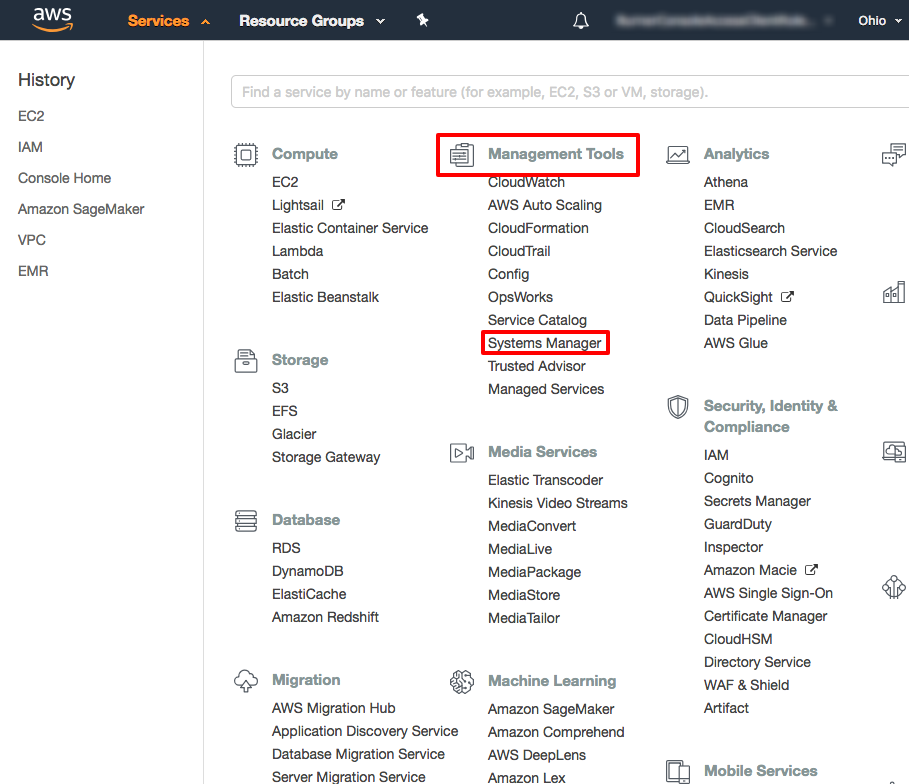


( click to enlarge )

**Step 3. Update the Systems Manager Agent**

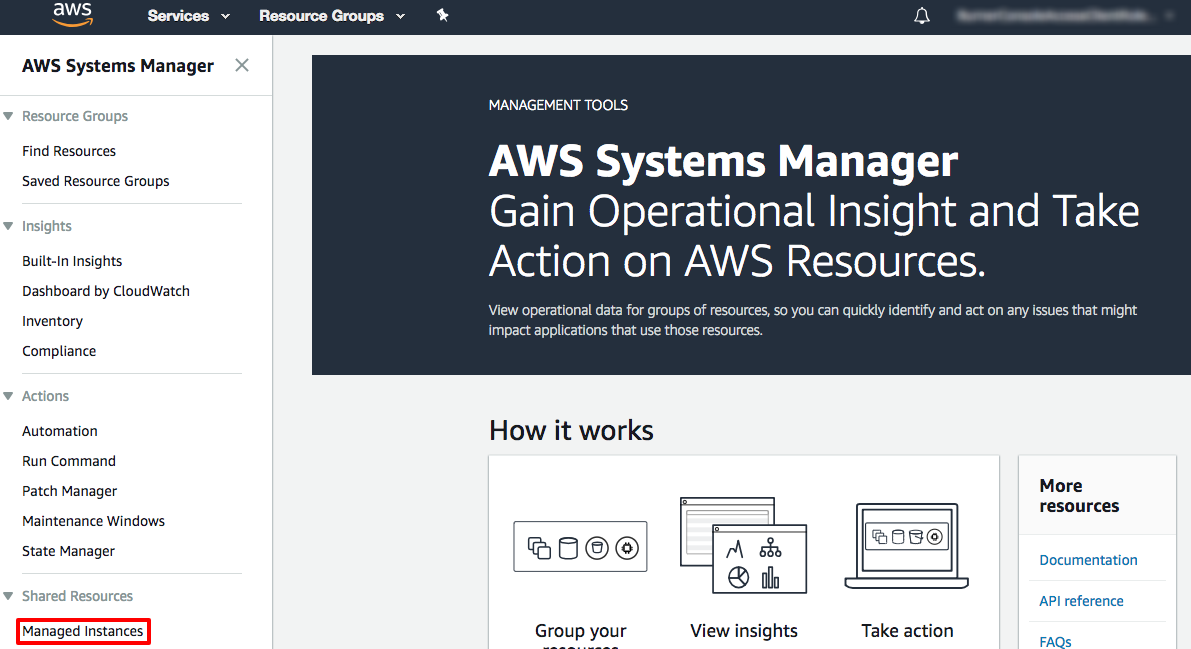
Now that you have an EC2 instance running the Systems Manager agent, you can automate administration tasks and manage the instance. In this step, you run a pre-packaged command, called a document, that will upgrade the agent. It is best practice to update the System Manager Agent when you create a new instance.

a.  In the top menu click on Services. Then, under Management Tools, select *Systems Manager* to open the Systems Manager console.



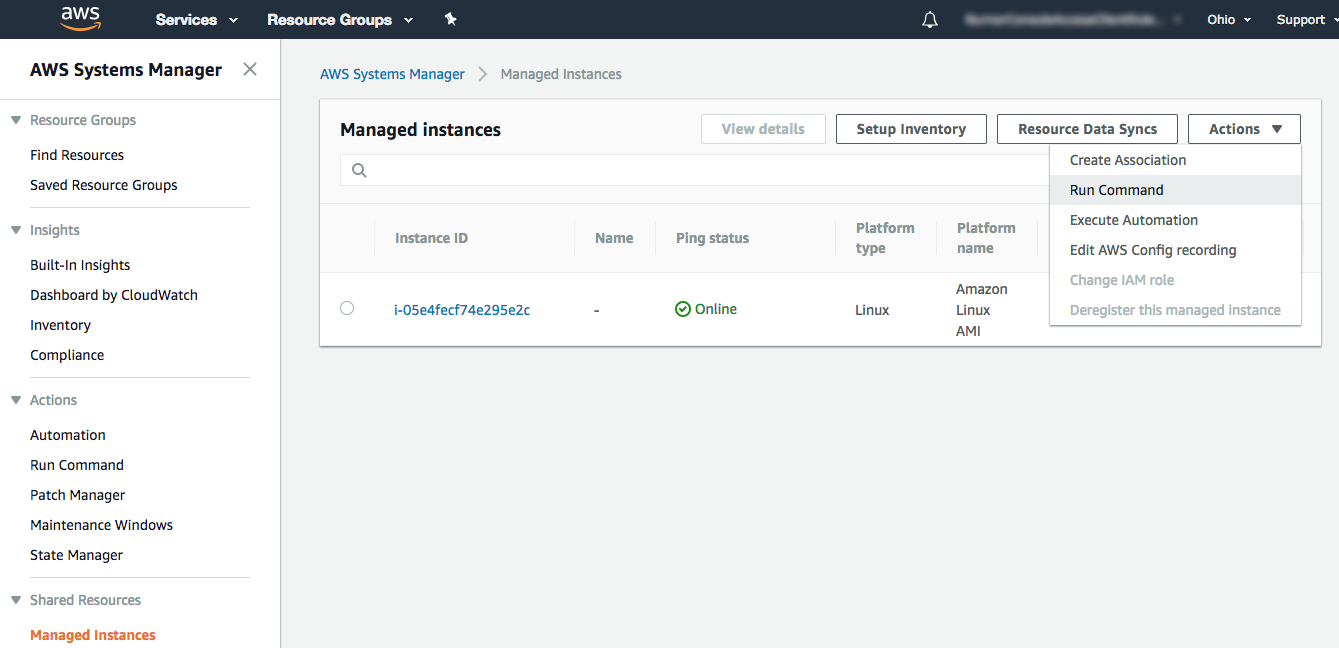
( click to enlarge )

b.  Under the Shared Resources section on the left navigation bar, choose Managed Instances.



( click to enlarge )

c.  On the Managed instances page, in the Actions drop down select Run Command.



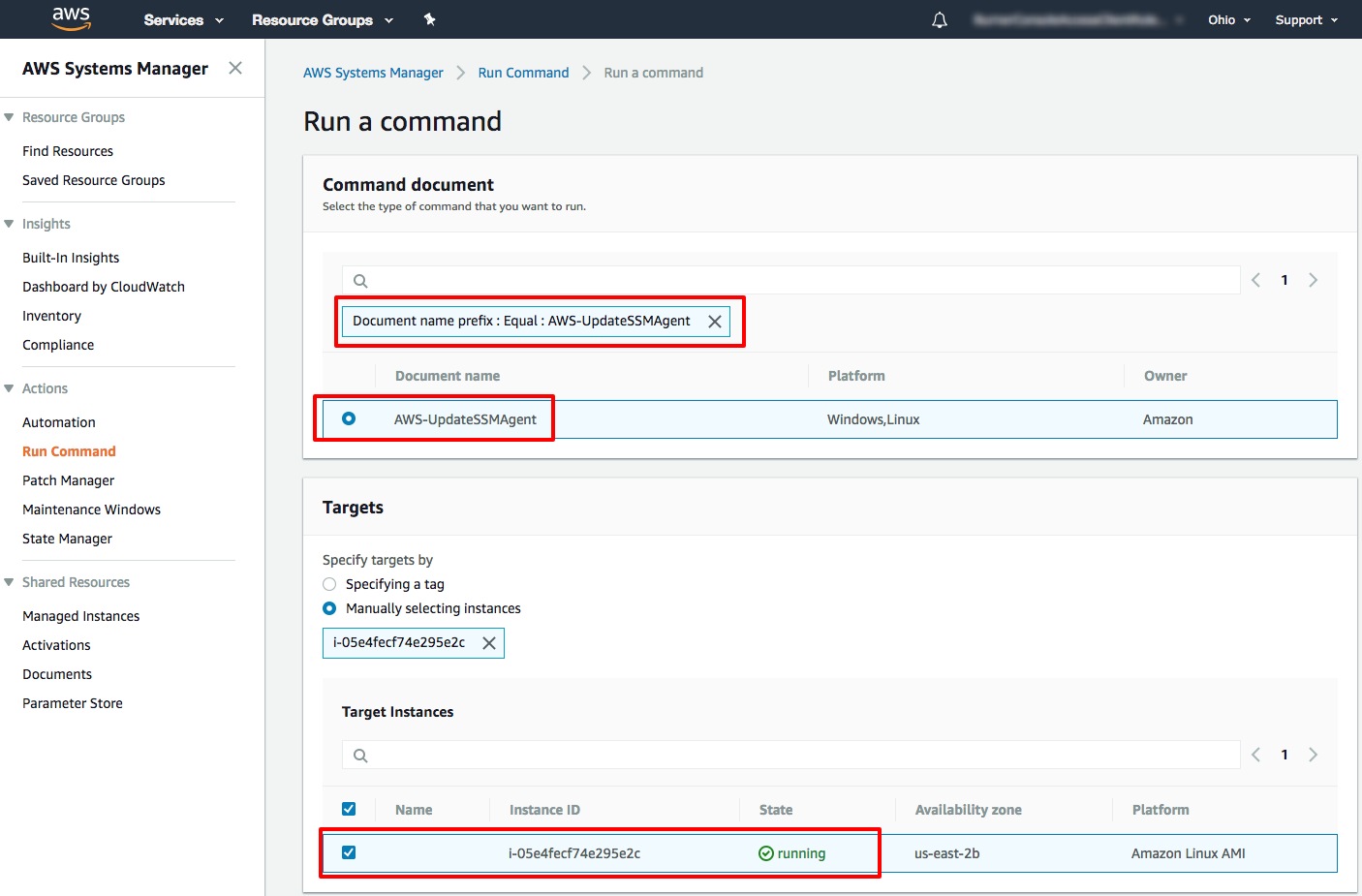
( click to enlarge )

d.  On the Run a command page, click in the search bar and select, Document name prefix, then click on Equal, then type in *AWS-UpdateSSMAgent*.

Now click on the radio button on the left of AWS-UpdateSSMAgent. This document will upgrade Systems Management agent on the instance.

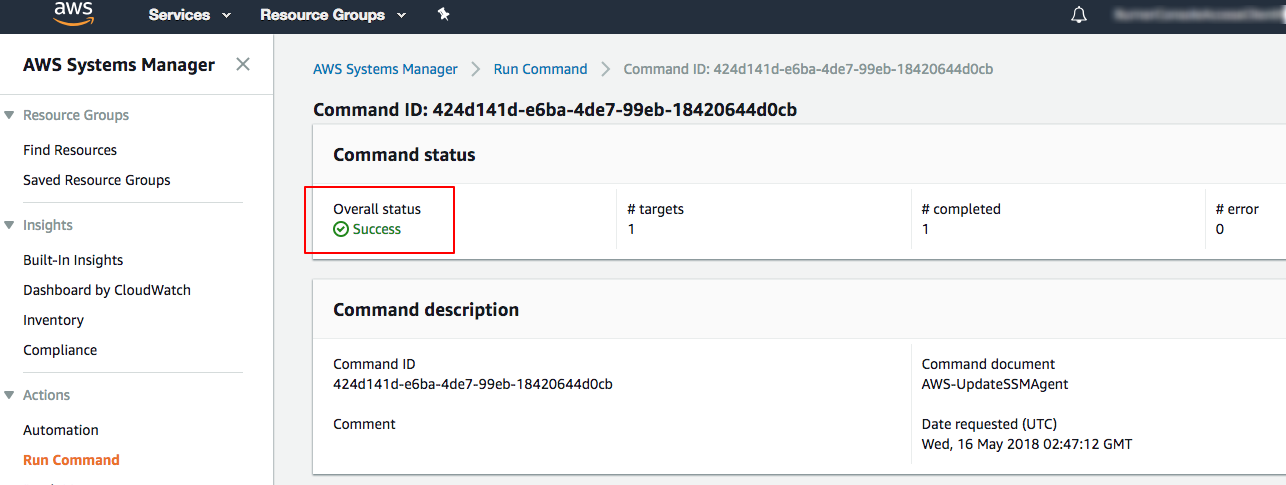
Scroll down to the Targets panel and click the check box next to your managed EC2 instance.

Finally, scroll down and select Run.



( click to enlarge )

e.  Next you will see page documenting your running command then and overall success in green. Congrats, you have just run your first remote command using Systems Manager.

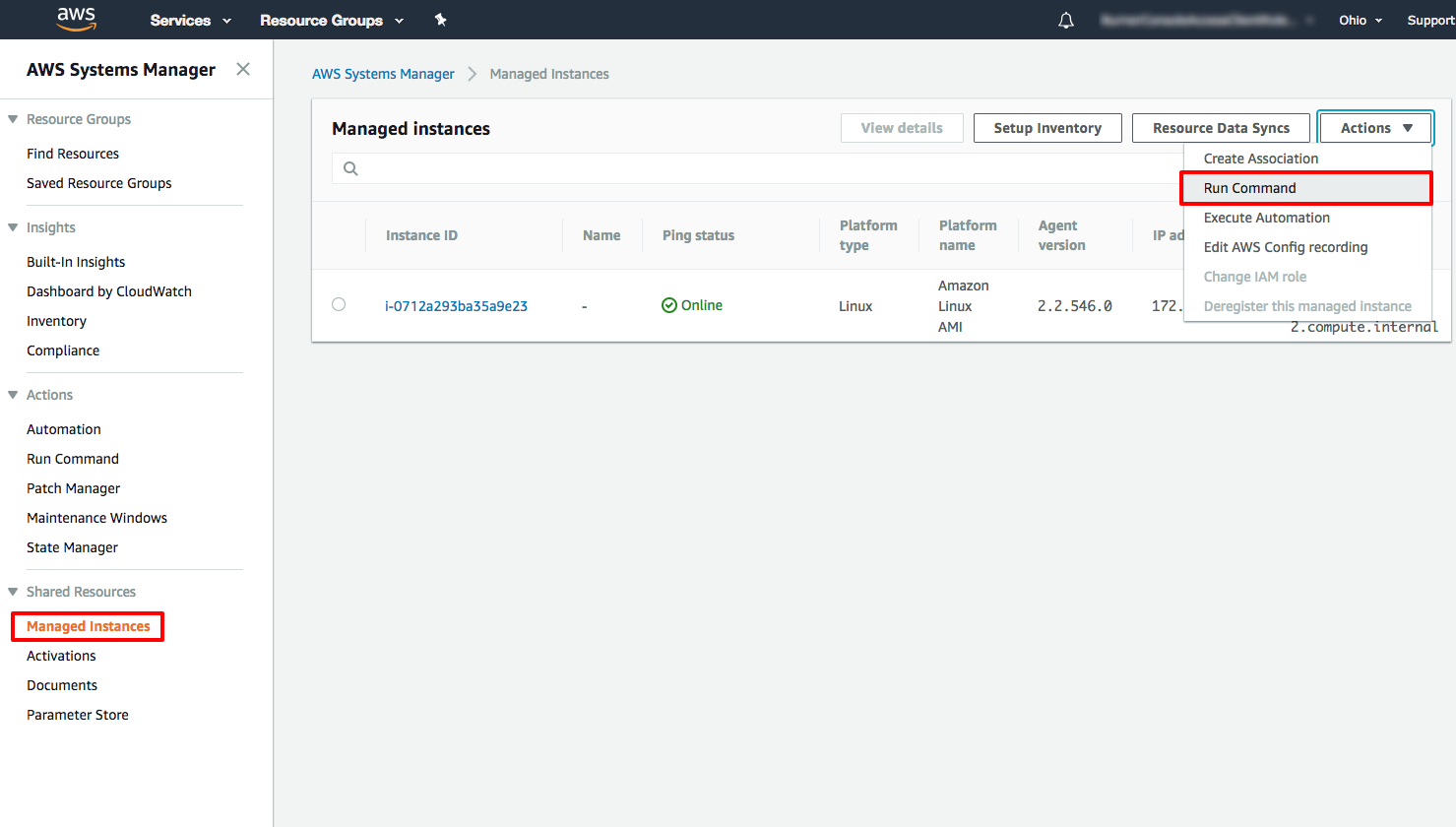


( click to enlarge )

**Step 4. Run a Remote Shell Script**

Now that your EC2 instance has the latest Systems Manager Agent, you can upgrade the packages on the EC2 instance. In this step, you will run a shell script through Run Command.

a.  From the Systems Management console, in the left nav under Shared Resources select Managed instances. Then in the Actions menu, select the Run Command menu item.

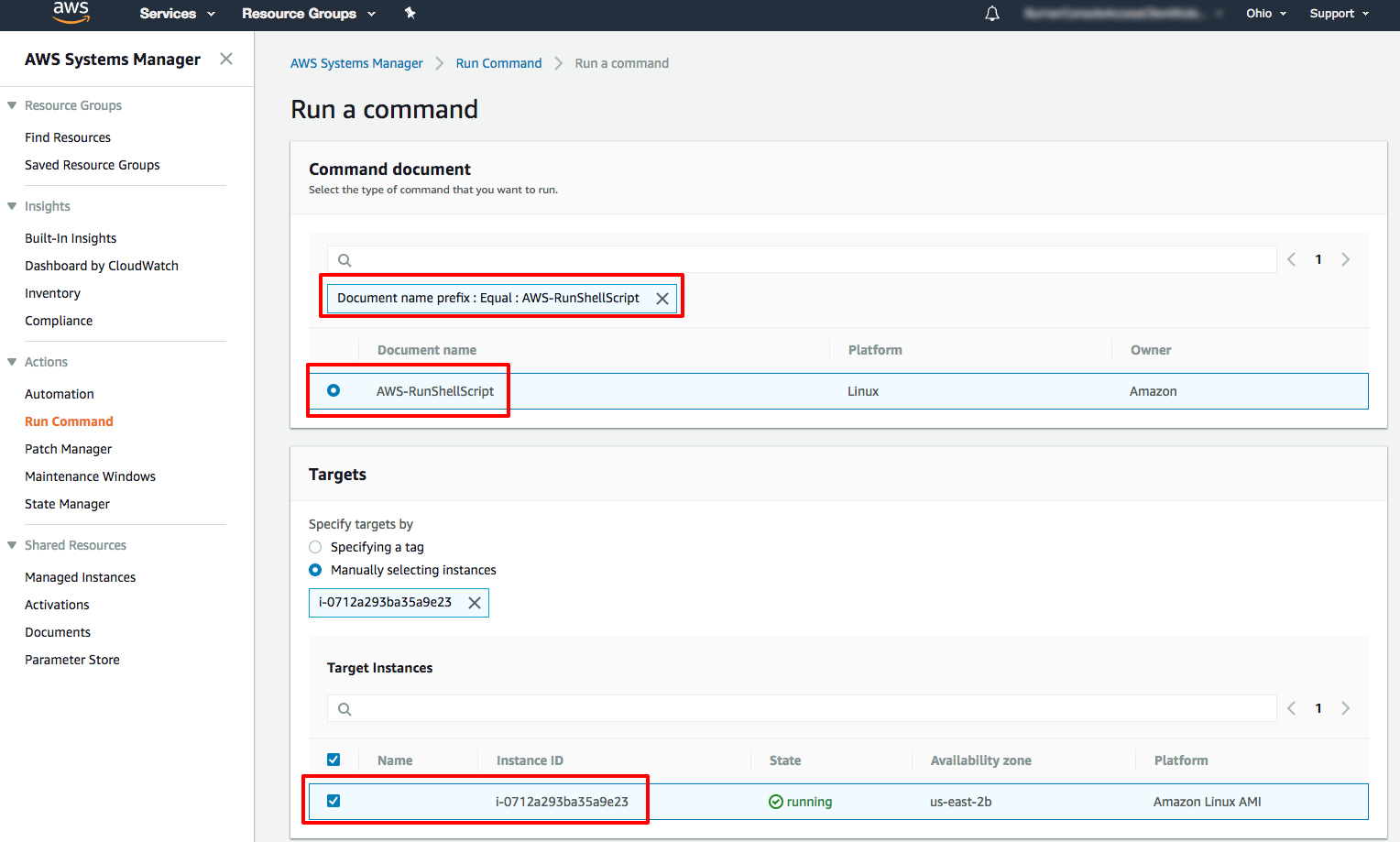


( click to enlarge )

b.  On the Run a command page, click in the search bar and select, Document name prefix, then click on Equal, then type in *AWS-RunShellScript*.

Now click on the radio button on the left of AWS-RunShellScript. This document will upgrade Systems Management agent on the instance.

Scroll down to the Targets panel and click the check box next to your managed EC2 instance.

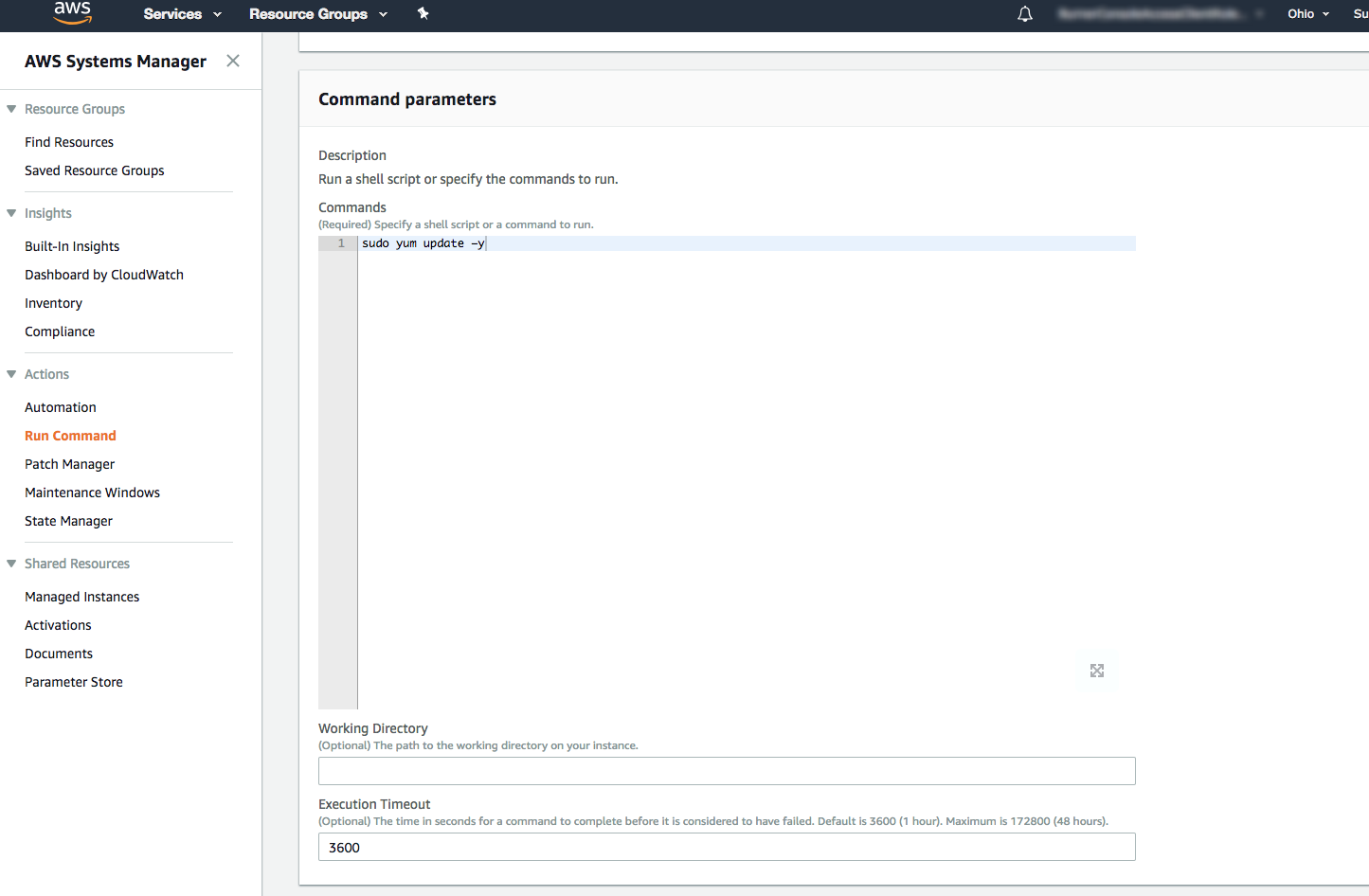


( click to enlarge )

c.  Scroll down to the Command Parameters panel and insert the following command in the Commands text box:

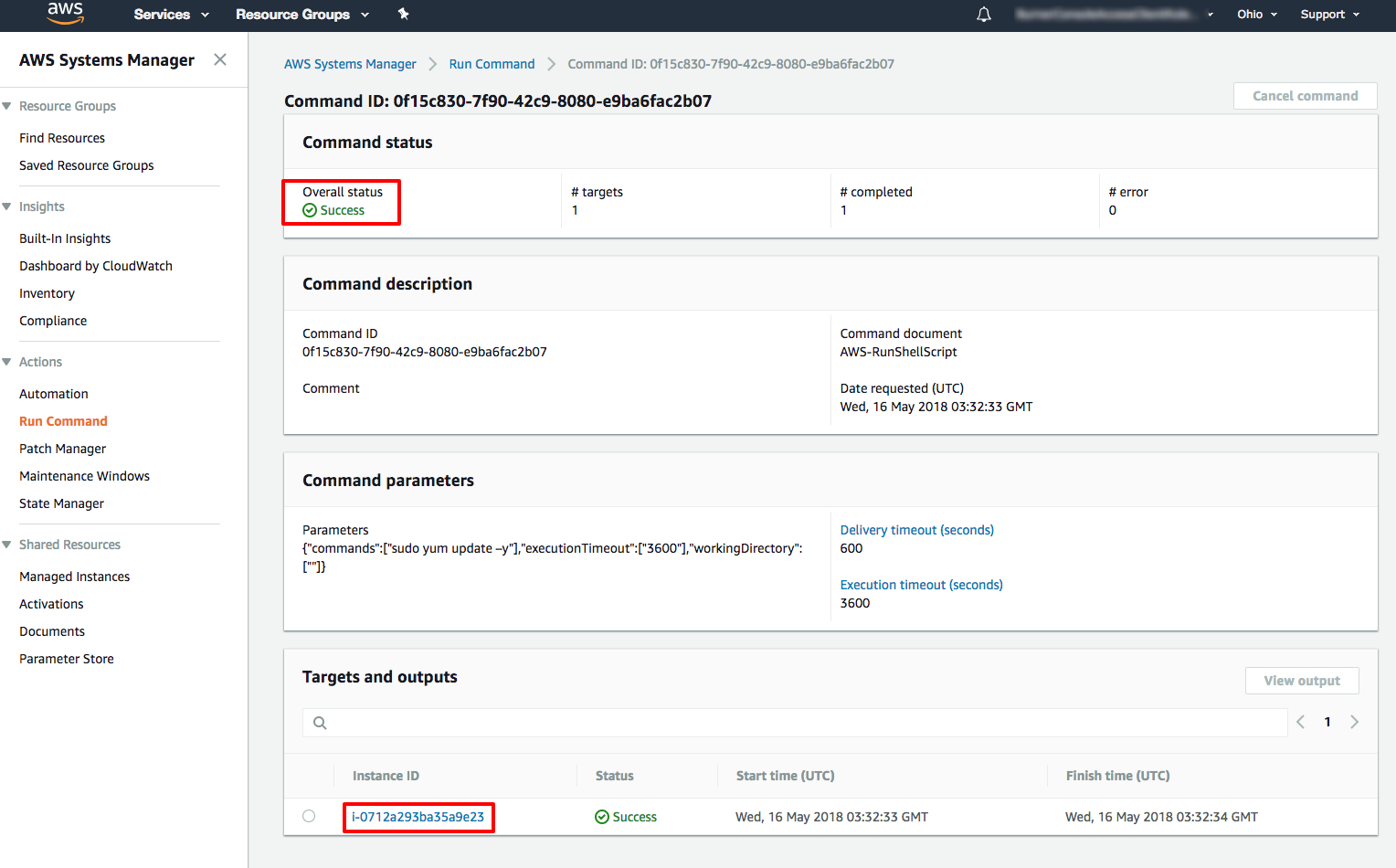
sudo yum update -y

Finally, scroll down and select Run.



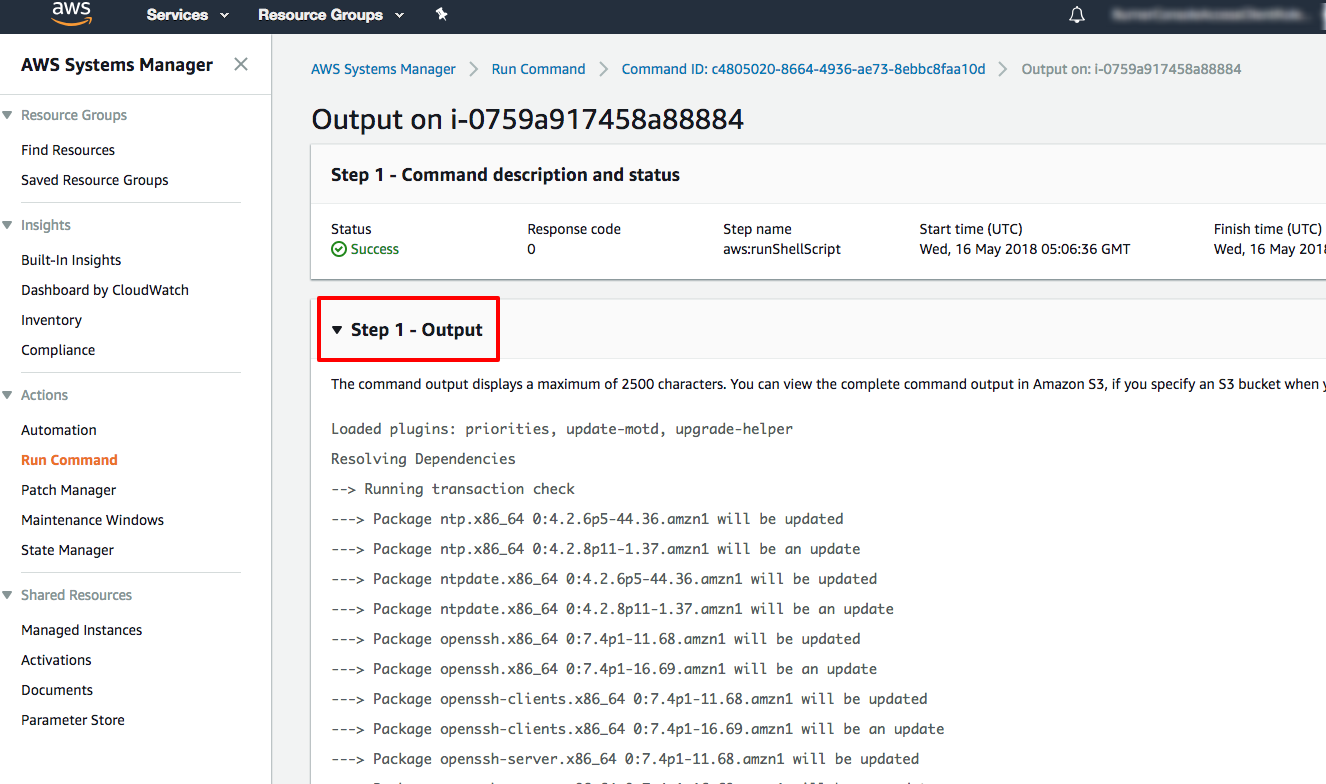
( click to enlarge )

d.  While your script is running remotely on the managed EC2 instance, the Overall status will be In Progress. Soon the Overall status will turn to Success. When it does, scroll down to the Targets and outputs panel and click on the Instance ID of your instance. Your Instance ID will be different than the one pictured.



( click to enlarge )

e.  From the Output on: i-XX page click on the header of the Step 1 - Output panel to view the output of the update command from the instance.



1. **What is DevOps?**

**Background/Problem statement**

The traditional goal of software development and IT operations do not align. Under the traditional separation between Devs & Ops, Dev and Ops have different and opposing goals. The goal of the development team is to deliver features into the hands of customer as quickly as possible (speed) but the goal of the operations team is to maintain the stability of the system and minimize downtime. These two goals end up becoming in opposition to one another because one of the worst thing for stability is change. Every time you change the system, you introduce a risk of instability. In fact, development teams are often measured according to their speed of delivery while operations is measured according to minimizing down time.

* **Software Development**: How many changes and features can we push into production and how quickly can we do it.

* **IT Operation:** We want stable and available systems. Daily updates and pushes to production? No way!

But let's be clear, the business drives this and the business wants **velocity** and **quality**. The business wants the customer to have new features (and bug fixes), while maintaining high availability.

Dev and Ops needs to work together as an integrated team. With DevOps, Dev and Ops work together and share the same goals and are both measured according to those same goals for speed and stability are very important. What DevOps culture has discovered is that when Development and Operations work together and share the same goals and prioritize speed and stability equally, they are able to solution and build tools to allow development to deployment frequently, quickly and to make changes to the system in a way that is robust, standardized without impacting stability negatively.

DevOps = Dev (Development) + Ops (Operations)

Operations Team: Release Management, Configuration Management, Application and Infrastructure Monitoring

DevOps is a software engineering culture and practice that aims at unifying software development (Dev) and software operations (Ops).

DevOps aims at shorter development cycles, increased deployment frequency, more dependable releases, in close alignment with business objectives.

DevOps is first a culture of collaboration between developers and operations people. This culture has given rise a set of practices.

With DevOps, Dev and Ops are playing on the same team. They share the same goals. These goals include things like:

1. Fast Time To Market (TTM) -time it takes to take some work that a development team have created and get it in front of a customer. That is, the time from when code being finished and deploy to production and working. The fastest that time to market is, the more successful and organization will be.
2. Few Production Failure. This goes back to stability. The more production failures, the more instability.

So, if we want production to be stable, we need to make sure that there are few production failures.

1. Immediate recovery from failures: In the real world, failures are going to happen. Mistakes are going to be made, things are going to break. The best thing is to ensure that when those problems happen, you can recover from them quickly and immediately. The faster you can recover from those problems, the less impact they will have. Best case scenario is that you can recover from a problem before customers even notice.
2. Dev write code
3. Code commit trigger automated build, integration and tests
4. QA can get their hands on it almost immediately
5. Once it is ready, kick off an automated deployment to production in a consistent and stable manner.

What’s important to understand about the DevOps culture is the series of automated steps and the series of tools that enables this process. Because these steps are automated, it is more easier to deployment codes in a stable fashion and therefore deployment can happen much frequently, potentially multiple times a day-getting features and bug fixes into the hands of the customers faster.

One of the important aspect of all these automation that goes with DevOps is monitoring. Some automated monitoring to detect problems almost instantly when they occur. The team does a roll back by deploying previous working version, fixing the problem quickly.

**DevOps Concepts and Practices**

1. **Build Automation**

* Automation of the process of preparing code for deployment to a live environment.
* Depending on what languages are used, code needs to be compiled, linted, minified, transformed, unit tested etc. Build automation means taking all of these steps and doing them in a consistent, automated way using a script or tool.
* Why do build automation? Fast, consistent, repeatable, portable, and reliable.

1. **Continuous Integration**

* Continuous Integration (CI) is the practice of frequently merging code changes done by developers.
* Continuous Integration means merging constantly through out the day, usually with the execution of some automated tests such as unit tests to detect any problems caused by the merge.
* Merging all the time could be a lot of work so to avoid that it be **automated.**
* CI is usually done with the help of a **CI server**
* Every time a developer commits a code change, the CI server sees the change and automatically performs a build, also executing automated tests.
* If there is any problem with the build (such as a failure for the code to compile), the CI server immediately and automatically notifies the developers.
* If anyone commits code that breaks the build, they are responsible for fixing the problem or rolling back their changes immediately so that other developers can continue working.
* **Why do CI?**

1. **Early detection**of certain types of bugs. If code does not compile or automated test fails, the developers are notified and can fix immediately. The sooner these bugs are detected the easier they to fix.
2. **Eliminate the scramble** to integrate just before a big a release-The code is constantly merged, so there is no need to do a big merge at the end.
3. Makes**frequently releases**possible-Code is always in a state that can be deployed to production.
4. Makes **continuous testing** possible-Since the code can always be run, QA testers can get their hands on it throughout the development process, not just at the end.
5. Encourage good coding practices- Frequent commits encourages simple modular code, rather than creating a giant complex mess that can only be finished after several months of work.

**Continuous Delivery and Continuous Deployment**

* What is **Continuous Delivery (CD)**? The practice of continuously maintaining code in a deployable state. Regardless of whether or not the decision is made to deploy, the code is always in a state that is able to be deployed.

Some people use the term Continuous deliver and continuous deployment interchangeably, or simply use the abbreviation CD. However, these are two separate concepts even though they are closely related.

* What is **Continuous deployment**? The practice of frequently deploying small code changes to production. CD is keeping the code in a deployable state. Continuous Deployment is actually doing the deployment frequently.

Companies that do continuous deployment deploy to production multiple times a day.

**2. DevOps Phases**

The following are the various phases of the DevOps lifecycle.

1. **Continuous Development**
2. **Continuous Testing**
3. **CI(Continuous Integration)**
4. **CD(Continuous Deployment)**
5. **Continuous Monitoring**



1. **Git and Github/Code Commit**

**Description: Git**

Git plays a crucial role when it comes to managing the code that the collaborators contribute to the shared repository. This code is then pulled for performing continuous integration to create a build and test it on the test server and eventually deploy it on the production.

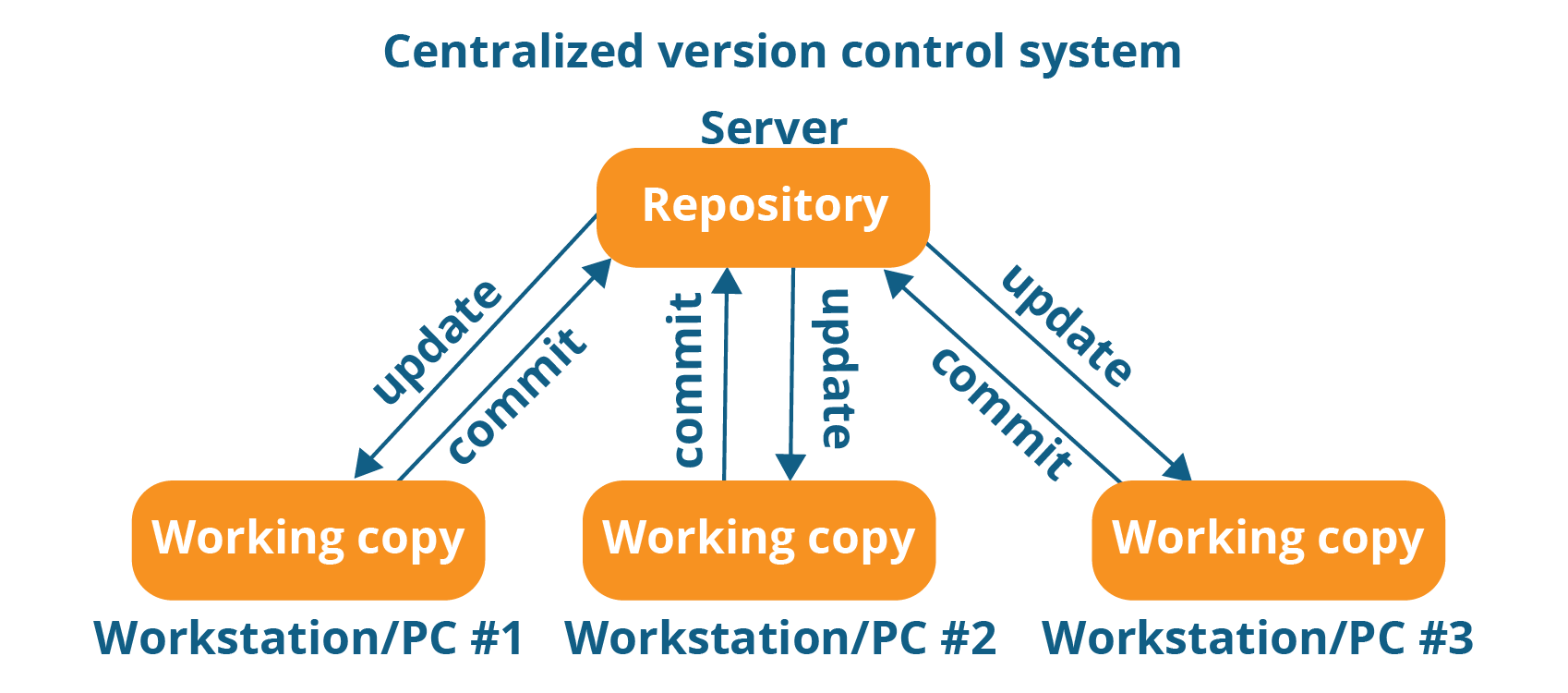
Git enables the communication between the development and the operations team. When you are working on a large project with a huge number of collaborators, it becomes very critical to have communication between the collaborators while making changes in the project.

Commit messages in Git play a vital role in communicating among the team. The bits and pieces that we all deploy lie in the Git. To be successful in DevOps, you need to have all of the communication in Version Control.

**Centralized VCS**

Centralized version control system (CVCS) uses a central server to store all files and enables team collaboration. It works on a single repository to which users can directly access a central server.

Please refer to the diagram below to get a better idea of CVCS:



The repository in the above diagram indicates a central server that could be local or remote which is directly connected to each of the programmer’s workstation.

Every programmer can extract or update their workstations with the data present in the repository or can make changes to the data or commit in the repository. Every operation is performed directly on the repository.

Even though it seems pretty convenient to maintain a single repository, it has some major drawbacks. Some of them are:

It is not locally available; meaning you always need to be connected to a network to perform any action.

Since everything is centralized, in any case of the central server getting crashed or corrupted will result in losing the entire data of the project.

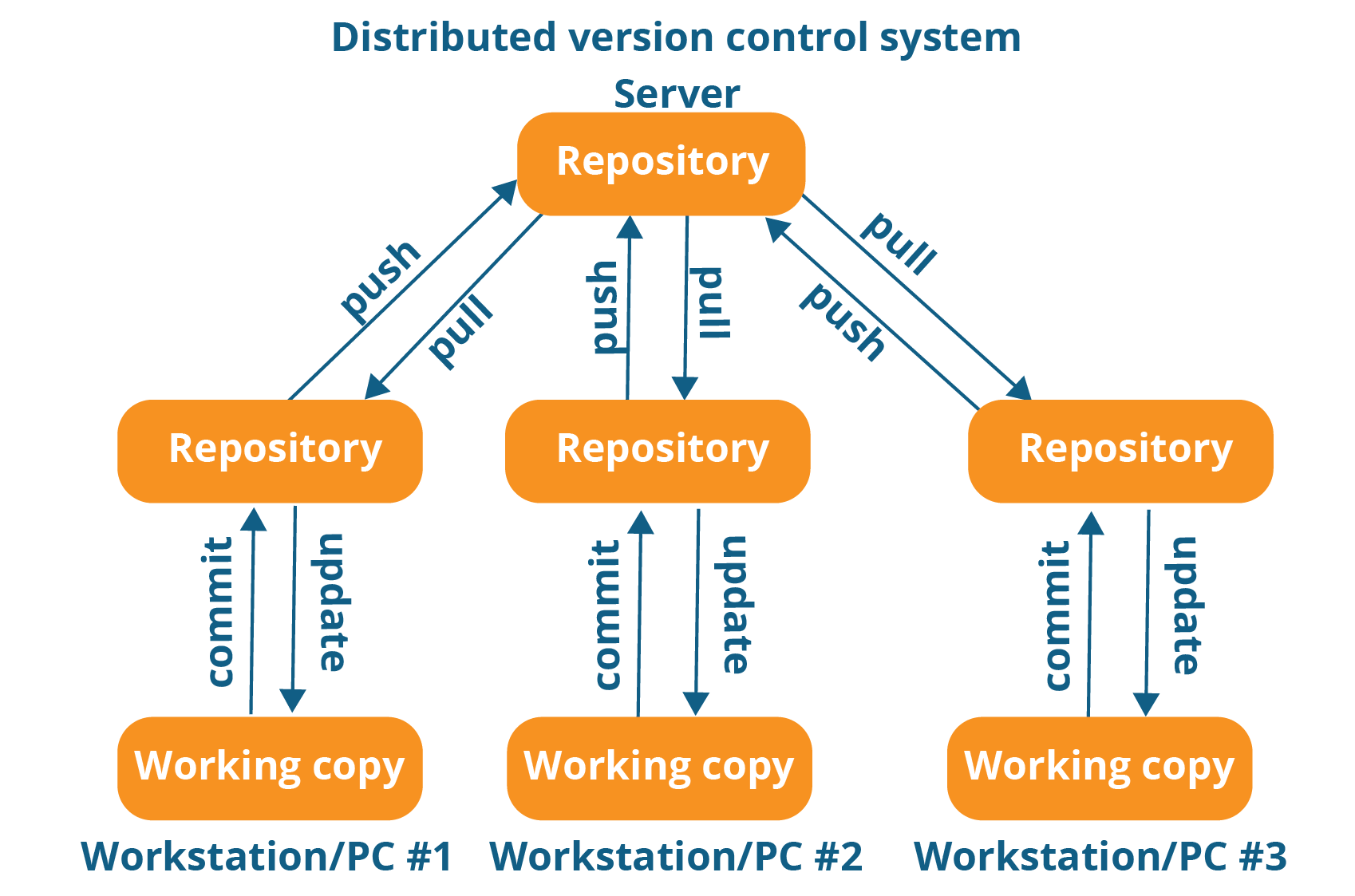
This is when Distributed VCS comes to the rescue.

**Distributed VCS**

These systems do not necessarily rely on a central server to store all the versions of a project file.

In Distributed VCS, every contributor has a local copy or “clone” of the main repository i.e. everyone maintains a local repository of their own which contains all the files and metadata present in the main repository.

You will understand it better by referring to the diagram below:



As you can see in the above diagram, every programmer maintains a local repository on its own, which is actually the copy or clone of the central repository on their hard drive. They can commit and update their local repository without any interference.

They can update their local repositories with new data from the central server by an operation called “pull” and affect changes to the main repository by an operation called “push” from their local repository.

The act of cloning an entire repository into your workstation to get a local repository gives you the following advantages:

**What is version control**

Version control systems are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

Version control protects source code from both catastrophe and the casual degradation of human error and unintended consequences.

Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree.

Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting. Changes made in one part of the software can be incompatible with those made by another developer working at the same time. This problem should be discovered and solved in an orderly manner without blocking the work of the rest of the team. Further, in all software development, any change can introduce new bugs on its own and new software can't be trusted until it's tested. So testing and development proceed together until a new version is ready.

Developing software without using version control is risky, like not having backups. Version control can also enable developers to move faster and it allows software teams to preserve efficiency and agility as the team scales to include more developers.

Branching and merging. Having team members work concurrently is a no-brainer, but even individuals working on their own can benefit from the ability to work on independent streams of changes. Creating a "branch" in VCS tools keeps multiple streams of work independent from each other while also providing the facility to merge that work back together, enabling developers to verify that the changes on each branch do not conflict. Many software teams adopt a practice of branching for each feature or perhaps branching for each release, or both. There are many different workflows that teams can choose from when they decide how to make use of branching and merging facilities in VCS.

Traceability. Being able to trace each change made to the software and connect it to project management and bug tracking software such as [Jira](https://www.atlassian.com/software/jira), and being able to annotate each change with a message describing the purpose and intent of the change can help not only with root cause analysis and other forensics. Having the annotated history of the code at your fingertips when you are reading the code, trying to understand what it is doing and why it is so designed can enable developers to make correct and harmonious changes that are in accord with the intended long-term design of the system. This can be especially important for working effectively with legacy code and is crucial in enabling developers to estimate future work with any accuracy.

While it is possible to develop software without using any version control, doing so subjects the project to a huge risk that no professional team would be advised to accept. So the question is not whether to use version control but which version control system to use.

**Setup for HTTPS Users Using Git Credentials**

The simplest way to set up connections to AWS CodeCommit repositories is to configure Git credentials for CodeCommit in the IAM console, and then use those credentials for HTTPS connections. You can also use these same credentials with any third-party tool or individual development environment (IDE) that supports HTTPS authentication using a static user name and password.

**Note**

If you have previously configured your local computer to use the credential helper for CodeCommit, you must edit your .gitconfig file to remove the credential helper information from the file before you can use Git credentials. If your local computer is running macOS, you might need to clear cached credentials from Keychain Access.

**Step 1: Initial Configuration for CodeCommit**

Follow these steps to set up an AWS account, create an IAM user, and configure access to CodeCommit.

**To create and configure an IAM user for accessing CodeCommit**

1. Create an IAM user, or use an existing one, in your AWS account. Make sure you have an access key ID and a secret access key associated with that IAM user.   
   **Note**   
   CodeCommit requires AWS Key Management Service. If you are using an existing IAM user, make sure there are no policies attached to the user that expressly deny the AWS KMS actions required by CodeCommit.
2. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
3. In the IAM console, in the navigation pane, choose **Users**, and then choose the IAM user you want to configure for CodeCommit access.
4. On the **Permissions** tab, choose **Add Permissions**.
5. In **Grant permissions**, choose **Attach existing policies directly**.
6. From the list of policies, select **AWSCodeCommitFullAccess** or another managed policy for CodeCommit access.

After you have selected the policy you want to attach, choose**Next: Review** to review the list of policies that will be attached to the IAM user. If the list is correct, choose **Add permissions**.   
If you want to use AWS CLI commands with CodeCommit, install the AWS CLI.

**Step 2: Install Git**

To work with files, commits, and other information in CodeCommit repositories, you must install Git on your local machine. CodeCommit supports Git versions 1.7.9 and later.

To install Git, we recommend websites such as [Git Downloads](http://git-scm.com/downloads).

**Note**

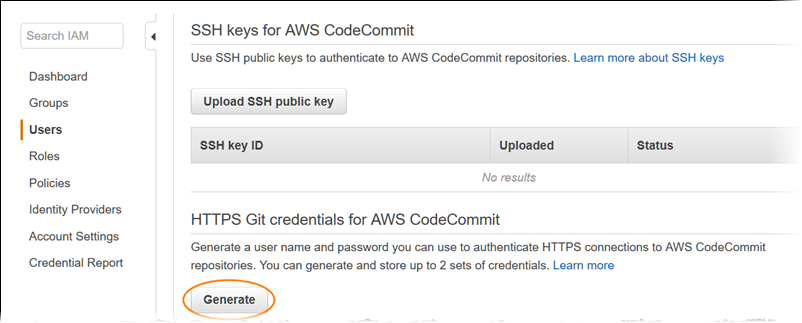
Git is an evolving, regularly updated platform. Occasionally, a feature change might affect the way it works with CodeCommit. If you encounter issues with a specific version of Git and CodeCommit, review the information in [Troubleshooting](https://docs.aws.amazon.com/codecommit/latest/userguide/troubleshooting.html).

**Step 3: Create Git Credentials for HTTPS Connections to CodeCommit**

After you have installed Git, create Git credentials for your IAM user in IAM.

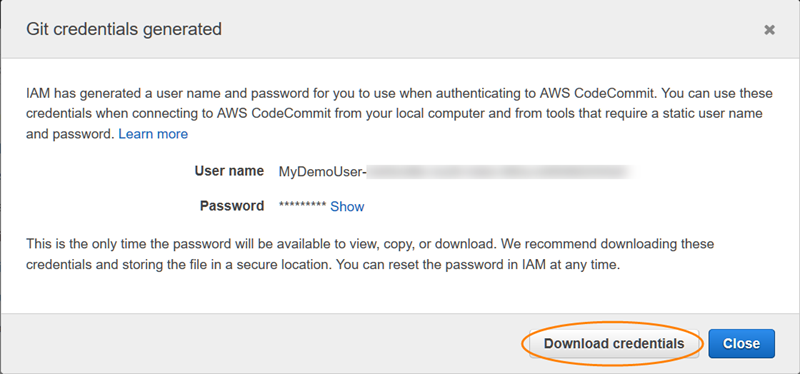
**To set up HTTPS Git credentials for CodeCommit**

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.   
   Make sure to sign in as the IAM user who will create and use the Git credentials for connections to CodeCommit.
2. In the IAM console, in the navigation pane, choose **Users**, and from the list of users, choose your IAM user.   
   **Note**   
   You can directly view and manage your CodeCommit credentials in **My Security Credentials**.
3. On the user details page, choose the **Security Credentials** tab, and in **HTTPS Git credentials for AWS CodeCommit**, choose **Generate**.



**Note**   
You cannot choose your own user name or password for Git credentials.

Copy the user name and password that IAM generated for you, either by showing, copying, and then pasting this information into a secure file on your local computer, or by choosing **Download credentials** to download this information as a .CSV file. You need this information to connect to CodeCommit.



After you have saved your credentials, choose **Close**.   
**Important**   
This is your only chance to save the user name and password. If you do not save them, you can copy the user name from the IAM console, but you cannot look up the password. You must reset the password and then save it.

Step 4: Connect to the CodeCommit Console and Clone the Repository

If an administrator has already sent you the name and connection details for the CodeCommit repository, you can skip this step and clone the repository directly.

**To connect to a CodeCommit repository**

1. Open the CodeCommit console at <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In the region selector, choose the AWS Region where the repository was created. Repositories are specific to an AWS Region.
3. Choose the repository you want to connect to from the list.
4. Choose **Connect**. Review the instructions and copy the URL to use when connecting to the repository.
5. Open a terminal, command line, or Git shell. Using the HTTPS URL you copied, run the **git clone** command to clone the repository. For example, to clone a repository named *MyDemoRepo* to a local repo named *my-demo-repo* in the US East (Ohio) Region:   
      
   git clone <https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo> my-demo-repo   
   The first time you connect, you are prompted for the user name and password for the repository. Depending on the configuration of your local computer, this prompt either originates from a credential management system for the operating system (for example, Keychain Access for macOS), a credential manager utility for your version of Git (for example, the Git Credential Manager included in Git for Windows), your IDE, or Git itself. Enter the user name and password generated for Git credentials in IAM (the ones you created in [Step 3: Create Git Credentials for HTTPS Connections to CodeCommit](https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-gc.html#setting-up-gc-iam)). Depending on your operating system and other software, this information might be saved for you in a credential store or credential management utility. If so, you should not be prompted again unless you change the password, inactivate the Git credentials, or delete the Git credentials in IAM.

**What is Continuous Integration?**

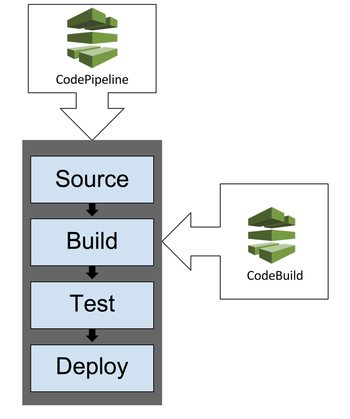
Continuous Integration is a development practice in which the developers are required to commit changes to the source code in a shared repository several times a day or more frequently. Every commit made in the repository is then built. This allows the teams to detect the problems early. Apart from this, depending on the Continuous Integration tool, there are several other functions like deploying the build application on the test server, providing the concerned teams with the build and test results etc.

|  |  |  |
| --- | --- | --- |
| Git task | Notes | Git commands |
| [**Tell Git who you are**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-config) | Configure the author name and email address to be used with your commits.  Note that Git [strips some characters](http://stackoverflow.com/questions/26159274/is-it-possible-to-have-a-trailing-period-in-user-name-in-git/26219423#26219423) (for example trailing periods) from user.name. | git config --global user.name "Sam Smith"  git config --global user.email sam@example.com |
| [**Create a new local repository**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-init) |  | git init |
| [**Check out a repository**](https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone) | Create a working copy of a local repository: | git clone /path/to/repository |
|  | For a remote server, use: | git clone username@host:/path/to/repository |
| [**Add files**](https://www.atlassian.com/git/tutorials/saving-changes#git-add) | Add one or more files to staging (index): | git add <filename>   git add \* |
| [**Commit**](https://www.atlassian.com/git/tutorials/saving-changes#git-commit) | Commit changes to head (but not yet to the remote repository): | git commit -m "Commit message" |
|  | Commit any files you've added with git add, and also commit any files you've changed since then: | git commit -a |
| [**Push**](https://www.atlassian.com/git/tutorials/syncing#git-push) | Send changes to the master branch of your remote repository: | git push origin master |
| [**Status**](https://www.atlassian.com/git/tutorials/inspecting-a-repository#git-status) | List the files you've changed and those you still need to add or commit: | git status |
| [**Connect to a remote repository**](https://www.atlassian.com/git/tutorials/syncing#git-remote) | If you haven't connected your local repository to a remote server, add the server to be able to push to it: | git remote add origin <server> |
|  | List all currently configured remote repositories: | git remote -v |
| [**Branches**](https://www.atlassian.com/git/tutorials/using-branches) | Create a new branch and switch to it: | git checkout -b <branchname> |
|  | Switch from one branch to another: | git checkout <branchname> |
|  | List all the branches in your repo, and also tell you what branch you're currently in: | git branch |
|  | Delete the feature branch: | git branch -d <branchname> |
|  | Push the branch to your remote repository, so others can use it: | git push origin <branchname> |
|  | Push all branches to your remote repository: | git push --all origin |
|  | Delete a branch on your remote repository: | git push origin :<branchname> |
| [**Update from the remote repository**](https://www.atlassian.com/git/tutorials/syncing) | Fetch and merge changes on the remote server to your working directory: | git pull |
|  | To merge a different branch into your active branch: | git merge <branchname> |
|  | View all the merge conflicts:  View the conflicts against the base file:  Preview changes, before merging: | git diff  git diff --base <filename>  git diff <sourcebranch> <targetbranch> |
|  | After you have manually resolved any conflicts, you mark the changed file: | git add <filename> |
| **Tags** | You can use tagging to mark a significant changeset, such as a release: | git tag 1.0.0 <commitID> |
|  | CommitId is the leading characters of the changeset ID, up to 10, but must be unique. Get the ID using: | git log |
|  | Push all tags to remote repository: | git push --tags origin |
| [**Undo local changes**](https://www.atlassian.com/git/tutorials/undoing-changes) | If you mess up, you can replace the changes in your working tree with the last content in head:  Changes already added to the index, as well as new files, will be kept. | git checkout -- <filename> |
|  | Instead, to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it, do this: | git fetch origin   git reset --hard origin/master |
| **Search** | Search the working directory for foo(): | git grep "foo()" |

AWS CodeBuild

[AWS CodeBuild](https://aws.amazon.com/codebuild/) that allows you to build your software. This AWS managed build service can be used to compile your source code, run unit tests, and produce deployable application artifacts. In short, CodeBuild eliminates the need to manage, provision, scale, and maintain build servers. With support for most build languages and tools like Maven and Gradle, CodeBuild is ready to go out of the box. It provides built-in support for programming platforms such as Java, Ruby, Python, Golang, Docker, Node, and Android and can be customized for others. As with most AWS services you pay for only what you use and the service scales automatically based on your demand

In this post we are going to explore the basics of CodeBuild and then learn how to use the service to build a Java application. In Figure 1, you see how CodeBuild fits into the build stage of a typical deployment pipeline. The pipeline itself can be orchestrated with a tool like AWS CodePipeline – which will be discussed later



*Figure 1 – How CodeBuild fits into a deployment pipeline*

Before we start building our own artifacts we need to understand two key CodeBuild concepts:

Build Environment and the Build Spec File.

**Build Environment** – The operating system, programming language runtime, and tools that CodeBuild uses to run a build are specified by a Docker image. A customer can either use their own Docker image or use one of the many optimized images that are provided by AWS.

**Build Spec File** – A required YAML file that contains build commands and related settings that CodeBuild uses for a build.

In addition, CodeBuild relies on the following services and solutions:

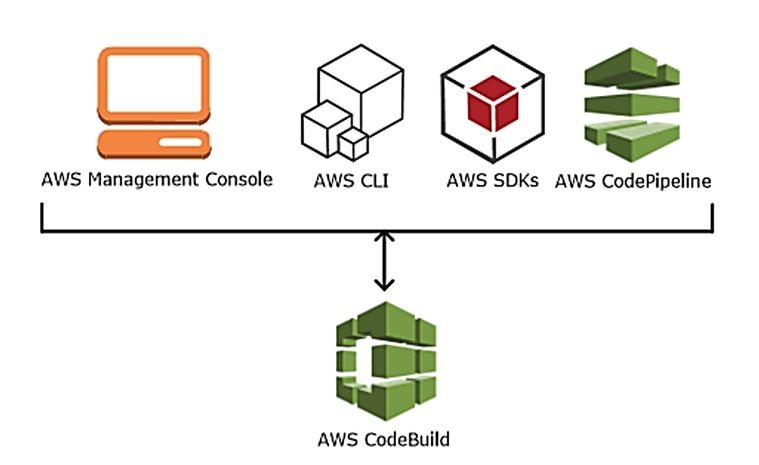
**Amazon S3, GitHub, and CodeCommit** – Used to store source code, build spec, and build environment

**IAM** – Securely provides access to AWS resources

**CloudWatch Logs** – Used to store and access logs for each CodeBuild run

Although CodeBuild is usually used as the build component of AWS CodePipeline, it can also be used to replace the building component of Jenkins and other tools. Cross-tool support is provided through the AWS SDKs for CodeBuild.

By default, CodeBuild allows up to 60 minutes for a build, but this can be set to any amount of time between 5 minutes and 8 hours. Keep in mind that you will also be charged for any other AWS resources that are provisioned as part of CodeBuild. This might include services such as S3, KMS, CloudWatch, and other EC2 resources provisioned by CodeBuild. In figure 2, you see how CodeBuild can be executed via the AWS Management Console, AWS CLI, AWS SDKs, and AWS CodePipeline.



*Figure 2 – Different ways to access CodeBuild*

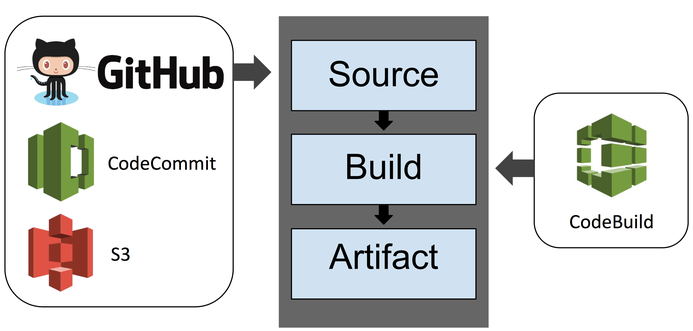
**Who Should Use CodeBuild?**

While CodeBuild can be seamlessly integrated with AWS CodePipeline, you can also integrate it with many other tools in your tooling ecosystem. The following scenarios are ideal candidates for using CodeBuild:

* Are you still building JARs in your IDE?
* Are you looking to remove the need to setup a separate Jenkins environment to build artifacts?
* Do you want to reduce the amount of dedicated build infrastructure you maintain?
* Are you trying to move to a hosted Continuous Integration (CI) server but it’s difficult to get new services approved within your organization?
* Is your organization uncomfortable having a third party run their builds, but comfortable with running those builds in AWS?

**Let’s Get CodeBuild-ing!**

Let’s take a look at what we want to do here. Starting with an AWS provided sample application source code, we are going to use CodeBuild to create a deployable Java artifact. In Figure 3 we can see how this looks as a process.



*Figure 3 – CodeBuild process*

**Step 1 – Setup**

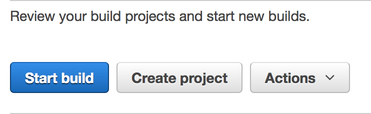
Download this GitHub package: <https://github.com/stelligent/aws-codedeploy-sample-tomcat> (as shown in Figure 3) and upload the zip to an S3 Bucket. **Be sure that S3 versioning is enabled for the bucket.**



Take note of the S3 bucket name and the name of the zip file. You will be needing these later.

**Step 2 – Create Project**

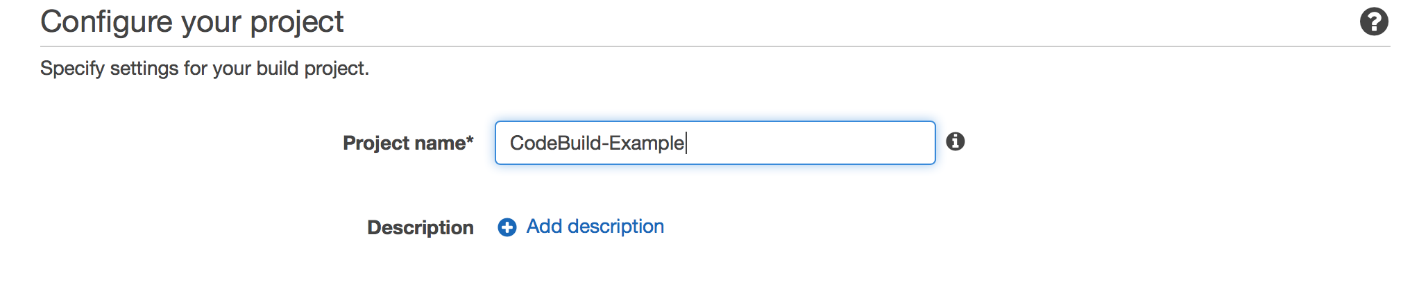
Launch CodeBuild by going to <https://console.aws.amazon.com/codebuild/> and click Create project (or Get Started and then Create project if this is your first time using CodeBuild). Once you’ve clicked this, you will be see a page where you can enter the settings for your build project.



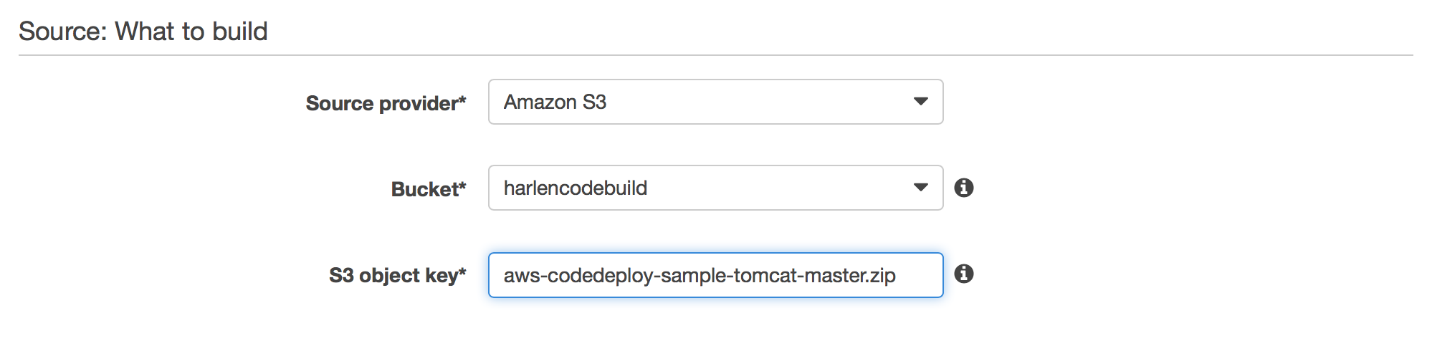
**Step 3 – Configure Project**

1.Let’s begin to configure the project. There are a few steps to this:

*a.* Project Configuration – Add a project name. Pick one that makes sense. This is what you will see on your CodeBuild page and in your logs.



*b.* Source: What to Build – Add the source for your code. Here we select S3 as a source and then enter the name of the S3 bucket that we created earlier in Step 1. You’ll enter aws-codedeploy-sample-tomcat-master.zip as the S3 object key. Although we are using S3 in this example we could also use GitHub or CodeCommit as well.



*c.* Environment: How to Build – Right after the source we want to tell CodeBuild how to build the file. The first step is to select Use an image managed by AWS CodeBuild for the *Environment image\**. This way we do not need to create our own Docker image. Next select the *Operating system\**. Here we use Ubuntu and then we select Java as our run time – since the source we are trying to build is a Java project.

In the version selection, we tell CodeBuild what version of Java we want to use. We are using aws/codebuild/java:openjdk-8.

For the Build specification section, select Insert build commands. Under the Build command\* section, type:

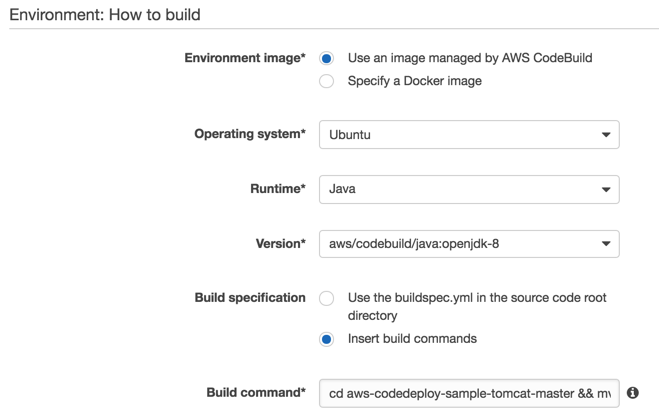
mvn package 

We use this build command because CodeBuild actually extracts the Zip file from S3 and uses the specific folder name. Once in that folder, we want to build from the Maven package in the source code.

**buildspec.yml**

While the above example demonstrates the use of running a simple build command, you can also provide a more expressive description of a build using a buildspec.yml file. In this YAML file you can configure the commands that occur in each of the build phases along with the name and type of the artifact file(s). The buildspec.yml file needs to reside in the root directory of the source repository. A snippet of the buildspec.yml is shown below:

version: 0.1   
phases:   
  build:   
    commands:   
      - mvn package   
artifacts:   
  files:   
    - appspec.yml   
    - target/SampleMavenTomcatApp.war   
    - scripts/\* 

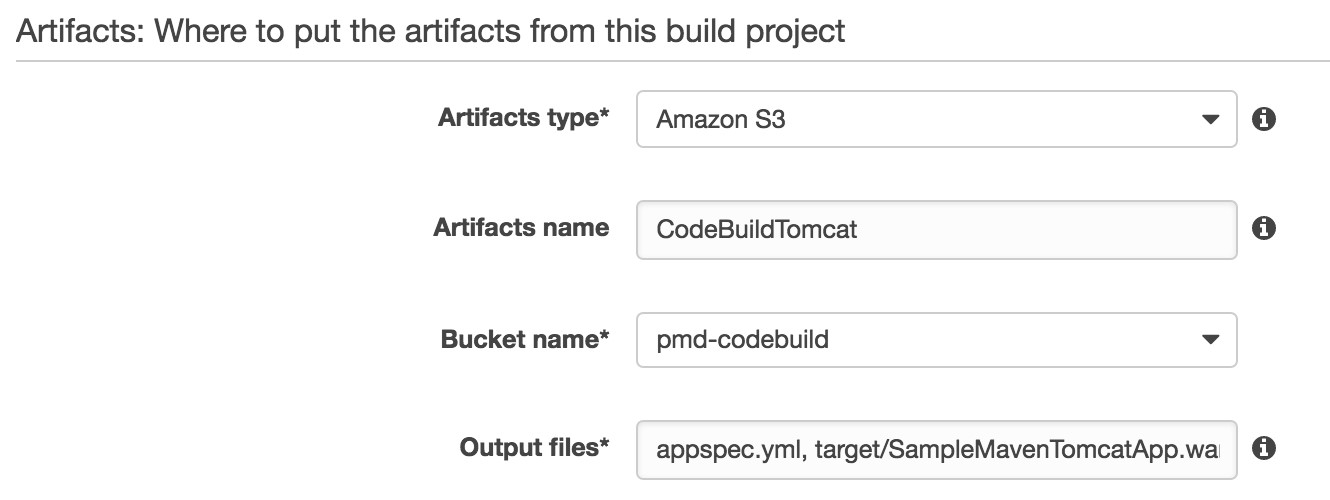


*d.* Artifacts: Where to Put the Artifacts from this Build Project – Now that we have told CodeBuild where to find the source code and how to build the artifact, we are going to tell it where to store the built artifact (i.e. a WAR file, in this case).

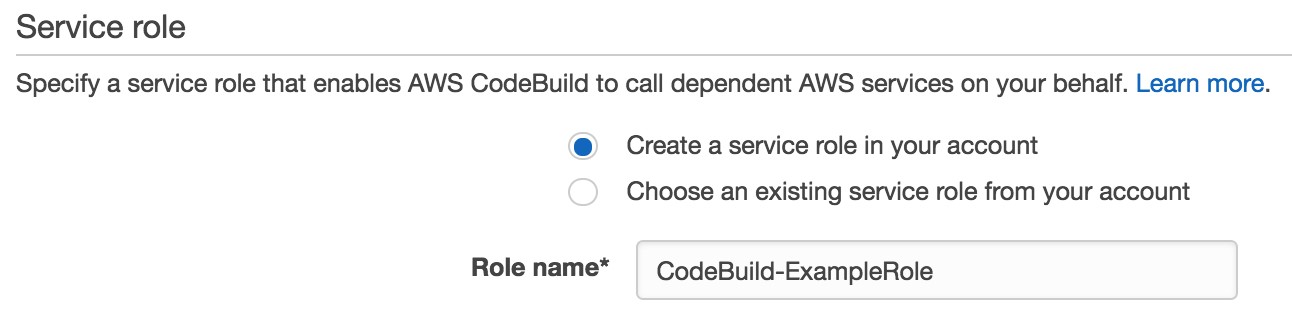
Under the Output Files section, enter:

appspec.yml, target/SampleMavenTomcatApp.war, scripts/\* 

We are going to store the artifact in S3, with the name CodeBuildTomcat, and in the same bucket we used to store the source code.



*e.* Service Role – Create a new Service Role in IAM for this CodeBuild project.



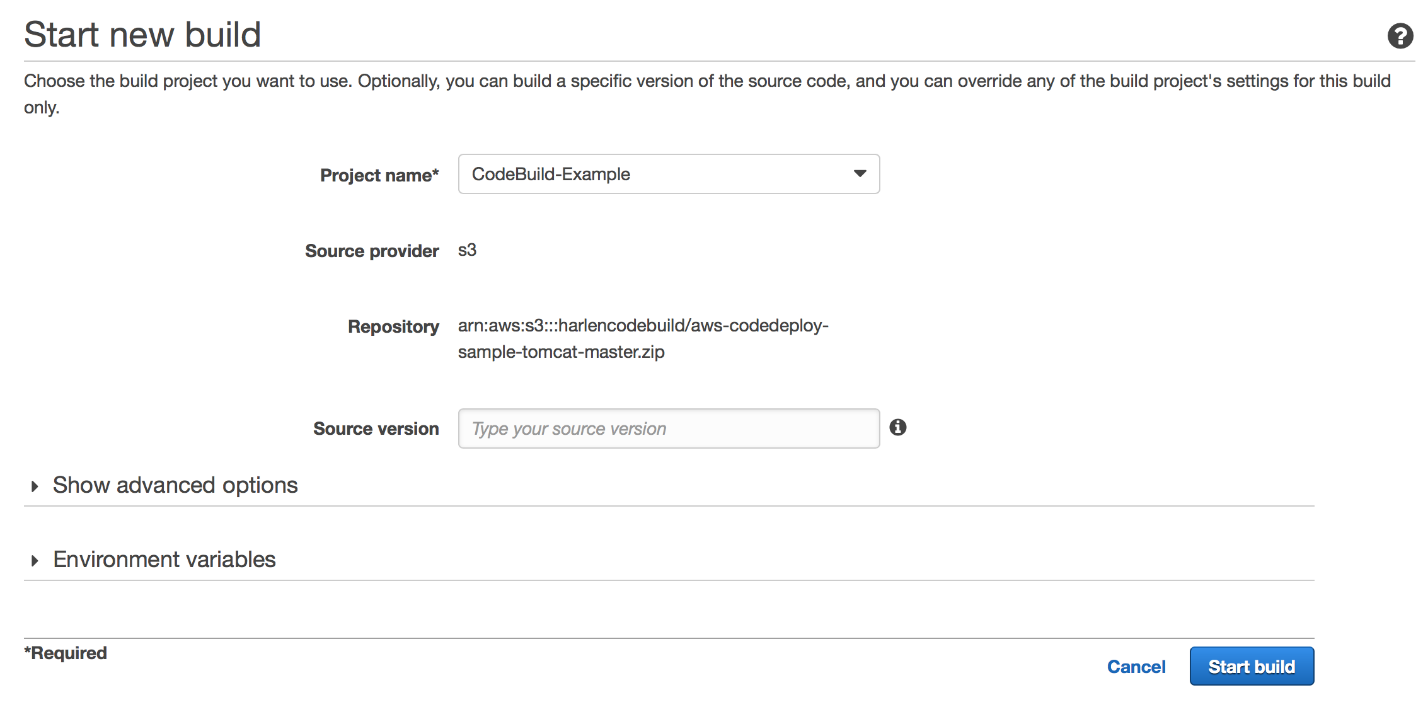
*f.* Click Continue

C:\Users\Zenetex admin\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\FCFEC404.tmp

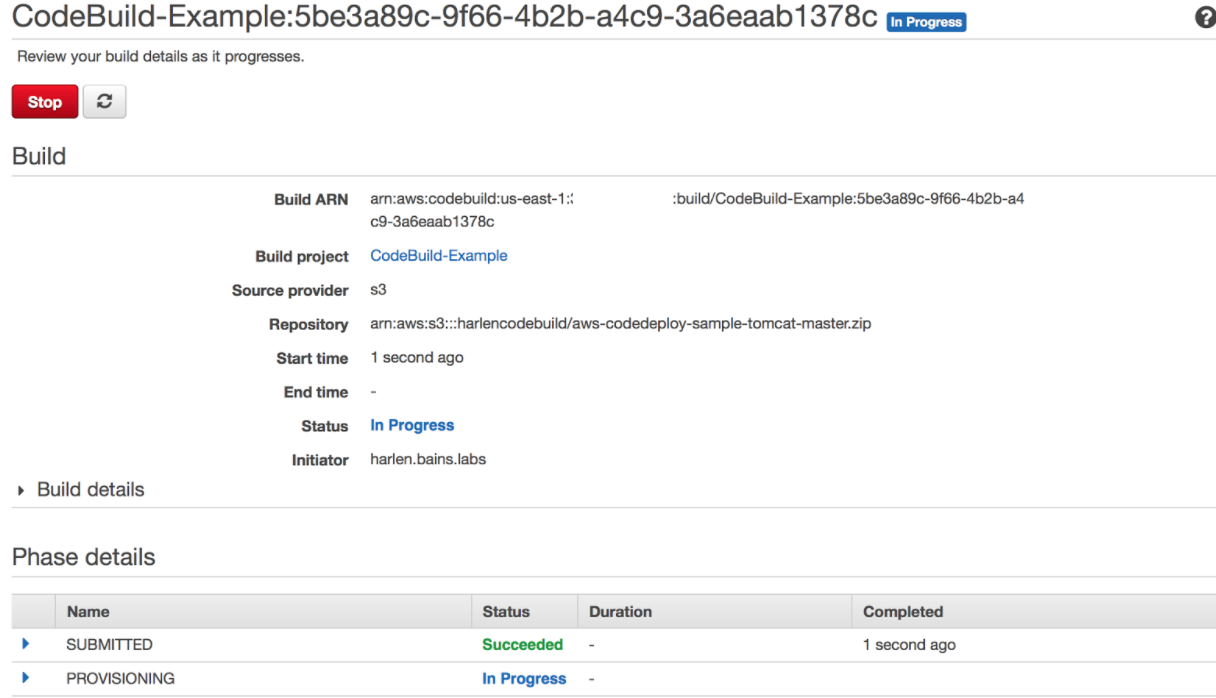
2. View the review page and then click Save and Build.

C:\Users\Zenetex admin\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\CF9A3CB2.tmp

3. Now on the *Start new build* page just click Start build.



4.  On the next page wait for the build to succeed.



5. Once it has completed, you can open the S3 bucket you specified for the artifact and view the S3 bucket with the *SampleMavenTomcatApp.war* file.

**CodeBuild Phases**

As CodeBuild is building artifacts, it goes through several distinct phases that are logged in AWS CloudWatch Logs (and accessible from the CodeBuild console). Each of these phases are described in some more detail below:

* SUBMITTED – This is the initial phase that indicates that the build process initiated and configuration has been received
* PROVISIONING – In this phase, CodeBuild launches a build container using a Docker image as defined
* DOWNLOAD\_SOURCE – Downloads the source from S3, CodeCommit or GitHub
* INSTALL – Installs the source onto the container
* PRE\_BUILD – Any actions that need to occur prior to the build
* BUILD – Executes the commands defined in the build specification
* POST\_BUILD – Runs any cleanup actions
* UPLOAD\_ARTIFACTS – Uploads the build artifacts to S3
* FINALIZING – Completing the build process
* COMPLETED – The build process is complete

6. Congratulations! Now you have used CodeBuild to build a sample application.

Amazon Simple Queue Service

Fully managed message queues for microservices, distributed systems, and serverless applications

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.

SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

**Benefits**

Eliminate administrative overhead

AWS manages all ongoing operations and underlying infrastructure needed to provide a highly available and scalable message queuing service. With SQS, there is no upfront cost, no need to acquire, install, and configure messaging software, and no time-consuming build-out and maintenance of supporting infrastructure. SQS queues are dynamically created and scale automatically so you can build and grow applications quickly and efficiently.

Reliably deliver messages

Use Amazon SQS to transmit any volume of data, at any level of throughput, without losing messages or requiring other services to be available. SQS lets you decouple application components so that they run and fail independently, increasing the overall fault tolerance of the system. Multiple copies of every message are stored redundantly across multiple availability zones so that they are available whenever needed.

Keep sensitive data secure

You can use Amazon SQS to exchange sensitive data between applications using server-side encryption (SSE) to encrypt each message body. Amazon SQS SSE integration with AWS Key Management Service (KMS) allows you to centrally manage the keys that protect SQS messages along with keys that protect your other AWS resources. AWS KMS logs every use of your encryption keys to AWS CloudTrail to help meet your regulatory and compliance needs.

Scale elastically and cost-effectively

Amazon SQS leverages the AWS cloud to dynamically scale based on demand. SQS scales elastically with your application so you don’t have to worry about capacity planning and pre-provisioning. There is no limit to the number of messages per queue, and standard queues provide nearly unlimited throughput. Costs are based on usage which provides significant cost saving versus the “always-on” model of self-managed messaging middleware.

What are Microservices?

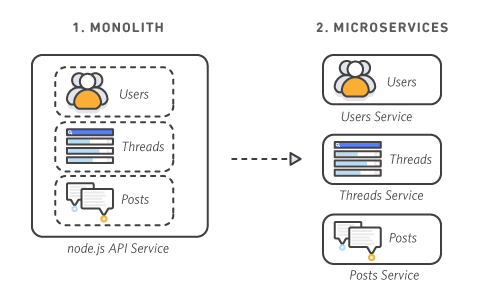
Microservices are an architectural and organizational approach to software development where software is composed of small independent services that communicate over well-defined APIs. These services are owned by small, self-contained teams.

Microservices architectures make applications easier to scale and faster to develop, enabling innovation and accelerating time-to-market for new features.

**Monolithic vs. Microservices Architecture**

With monolithic architectures, all processes are tightly coupled and run as a single service. This means that if one process of the application experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application’s features becomes more complex as the code base grows. This complexity limits experimentation and makes it difficult to implement new ideas. Monolithic architectures add risk for application availability because many dependent and tightly coupled processes increase the impact of a single process failure.

With a microservices architecture, an application is built as independent components that run each application process as a service. These services communicate via a well-defined interface using lightweight APIs. Services are built for business capabilities and each service performs a single function. Because they are independently run, each service can be updated, deployed, and scaled to meet demand for specific functions of an application.



**Characteristics of Microservices**

Autonomous

Each component service in a microservices architecture can be developed, deployed, operated, and scaled without affecting the functioning of other services. Services do not need to share any of their code or implementation with other services. Any communication between individual components happens via well-defined APIs.

Specialized

Each service is designed for a set of capabilities and focuses on solving a specific problem. If developers contribute more code to a service over time and the service becomes complex, it can be broken into smaller services.

**To create a Read Replica from a source MySQL, MariaDB, Oracle, or PostgreSQL DB instance**

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the MySQL, MariaDB, Oracle, or PostgreSQL DB instance that you want to use as the source for a Read Replica.
4. For **Actions**, choose **Create read replica**.
5. Choose the instance specifications that you want to use. We recommend that you use the same DB instance class and storage type as the source DB instance for the Read Replica. For **Multi-AZ deployment**, choose **Yes** to create a standby of your replica in another Availability Zone for failover support for the replica. Creating your Read Replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.
6. Choose the settings that you want to use. For **DB instance identifier**, enter a name for the Read Replica. Adjust other settings as needed.
7. Choose the other settings that you want to use.
8. Choose **Create read replica**.

**Auto Scaling Lifecycle Hooks**

Lifecycle hooks allow you to control what happens when your Amazon EC2 instances are launched and terminated as you scale out and in. For example, you might download and install software when an instance is launching, and archive instance log files in Amazon Simple Storage Service (S3) when an instance is terminating.

Lifecycle hooks enable you to perform custom actions by *pausing* instances as an Auto Scaling group launches or terminates them. When an instance is paused, it remains in a wait state until either you complete the lifecycle action using the **complete-lifecycle-action** command or the CompleteLifecycleAction operation, or the timeout period ends (one hour by default).

For example, your newly launched instance completes its startup sequence and a lifecycle hook pauses the instance. While the instance is in a wait state, you can install or configure software on it, making sure that your instance is fully ready before it starts receiving traffic. For another example of the use of lifecycle hooks, when a scale-in event occurs, the terminating instance is first deregistered from the load balancer (if the Auto Scaling group is being used with Elastic Load Balancing). Then, a lifecycle hook pauses the instance before it is terminated. While the instance is in the wait state, you can, for example, connect to the instance and download logs or other data before the instance is fully terminated.

**How Lifecycle Hooks Work**

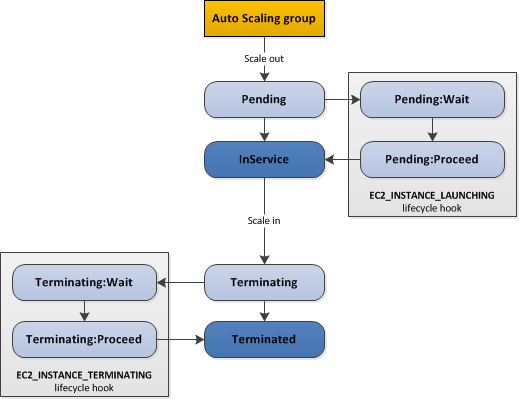
After you add lifecycle hooks to your Auto Scaling group, they work as follows:

1. The Auto Scaling group responds to scale-out events by launching instances and scale-in events by terminating instances.
2. The lifecycle hook puts the instance into a wait state (Pending:Wait or Terminating:Wait). The instance is paused until you continue or the timeout period ends.
3. You can perform a custom action using one or more of the following options:

* Define a CloudWatch Events target to invoke a Lambda function when a lifecycle action occurs. The Lambda function is invoked when Amazon EC2 Auto Scaling submits an event for a lifecycle action to CloudWatch Events. The event contains information about the instance that is launching or terminating, and a token that you can use to control the lifecycle action.
* Define a notification target for the lifecycle hook. Amazon EC2 Auto Scaling sends a message to the notification target. The message contains information about the instance that is launching or terminating, and a token that you can use to control the lifecycle action.
* Create a script that runs on the instance as the instance starts. The script can control the lifecycle action using the ID of the instance on which it runs.

1. By default, the instance remains in a wait state for one hour, and then the Auto Scaling group continues the launch or terminate process (Pending:Proceed or Terminating:Proceed). If you need more time, you can restart the timeout period by recording a heartbeat. If you finish before the timeout period ends, you can complete the lifecycle action, which continues the launch or termination process.

The following illustration shows the transitions between instance states in this process:



**Considerations**

Adding lifecycle hooks to your Auto Scaling group gives you greater control over how instances launch and terminate. The following are things to consider when adding a lifecycle hook to your Auto Scaling group, to help ensure that the group continues to perform as expected.

**Keeping Instances in a Wait State**

Instances can remain in a wait state for a finite period of time. The default is one hour (3600 seconds). You can adjust this time in the following ways:

* Set the heartbeat timeout for the lifecycle hook when you create the lifecycle hook. With the [**put-lifecycle-hook**](https://docs.aws.amazon.com/cli/latest/reference/autoscaling/put-lifecycle-hook.html) command, use the --heartbeat-timeout parameter. With the PutLifecycleHook operation, use the HeartbeatTimeout parameter.
* Continue to the next state if you finish before the timeout period ends, using the [**complete-lifecycle-action**](https://docs.aws.amazon.com/cli/latest/reference/autoscaling/complete-lifecycle-action.html) command or the CompleteLifecycleAction operation.
* Postpone the end of the timeout period by recording a heartbeat, using the [**record-lifecycle-action-heartbeat**](https://docs.aws.amazon.com/cli/latest/reference/autoscaling/record-lifecycle-action-heartbeat.html) command or the RecordLifecycleActionHeartbeat operation. This extends the timeout period by the timeout value specified when you created the lifecycle hook. For example, if the timeout value is one hour, and you call this command after 30 minutes, the instance remains in a wait state for an additional hour, or a total of 90 minutes.

The maximum amount of time that you can keep an instance in a wait state is 48 hours or 100 times the heartbeat timeout, whichever is smaller.

**Add Lifecycle Hooks (Console)**

Follow these steps to add a lifecycle hook to an existing Auto Scaling group. You can specify whether the hook is used when the instances launch or terminate and how long to wait until the lifecycle hook is completed before abandoning or continuing.

**To add a lifecycle hook**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
3. Select your Auto Scaling group.
4. On the **Lifecycle Hooks** tab, choose **Create Lifecycle Hook**.
5. To define a lifecycle hook, do the following:
   1. For **Lifecycle Hook Name**, specify a name for the lifecycle hook.
   2. For **Lifecycle Transition**, choose **Instance Launch** or **Instance Terminate**.
   3. Specify a timeout value for **Heartbeat Timeout**, which allows you to control the amount of time for the instances to remain in a wait state. The value must be from 30 to 7200 seconds. During the timeout period, you can, for example, log on to a newly launched instance, and install applications or perform custom actions.
   4. For **Default Result**, specify the action the Auto Scaling group takes when the lifecycle hook timeout elapses or if an unexpected failure occurs. You can choose to either ABANDON or CONTINUE.   
      If the instance is launching, CONTINUE indicates that your actions were successful, and that the Auto Scaling group can put the instance into service. Otherwise, ABANDON indicates that your custom actions were unsuccessful, and that the instance can be terminated. If the instance is terminating, both ABANDON and CONTINUE allow the instance to terminate. However, ABANDON stops any remaining actions, such as other lifecycle hooks, and CONTINUE allows any other lifecycle hooks to complete.
   5. (Optional) For **Notification Metadata**, specify additional information that you want to include any time Amazon EC2 Auto Scaling sends a message to the notification target.
6. Choose **Create**.

**Add Lifecycle Hooks (AWS CLI)**

Create and update lifecycle hooks using the [**put-lifecycle-hook**](https://docs.aws.amazon.com/cli/latest/reference/autoscaling/put-lifecycle-hook.html) command.

To perform an action on scale out, use the following command.

aws autoscaling put-lifecycle-hook --lifecycle-hook-name *my-hook* --auto-scaling-group-name *my-asg* \ --lifecycle-transition autoscaling:EC2\_INSTANCE\_LAUNCHING

aws autoscaling put-lifecycle-hook --lifecycle-hook-name *my-hook* --auto-scaling-group-name *my-asg* \ --lifecycle-transition autoscaling:EC2\_INSTANCE\_TERMINATING

HIGH AVAILABILITY AND BUSINESS CONTINUITY

DEMONSTRATE ABILITY TO ARCHITECT THE APPROPRIATE LEVEL OF AVAILABILITY BASED ON STAKEHOLDER REQUIREMENTS

A) BACKUP AND RESTORE

Data is backed up to off site regularly. It can take a long time to restore the systems in the event of a disaster. Amazon S3 is ideal for backing up data that might need to be restored quickly. Amazon S3 backup is done through the network and can be accessible from any location

AWS Import/Export can transfer large data sets by shipping storage devices to AWS.

Longer term storage where retrieval times are several hours are adequate, ther is Amazon Glacier. Same durability as Amazon S3. Glacier is a lo coat alternative. S3 and Glacier can be used to produce a tiered backup solutions

AWS Storage Gateway enables snapshots of on premise data volumes to be transparently copied to S3 for backup

Storage Cached volumes allow you to store your primary data on S3 but keep frequently accessed data local.

Gateway VTL is a backup target for existing backup management software. Replace form traditional magnetic tape backup.

– now the storage is made up of

1) Files

Simple solution for presenting one or more S3 buckets and their objects as a mountable NFS to one or more clients

Deployed as a virtual appliance either to VMWare or and EC2 instance.

Use Cases:

Cloud Tiering – on premise environment where storage resources are reaching capacity

Hybrid Cloud Backup – can be used to as a backup volume

2) volumes

a) cached

b) stored

3) Virtual tapes

a) library (S3 storage)

b) shelf ( retrieval not for 24 hours )

Key Steps for backup and restore:

1) Select an appropriate tool or method to backup your data into AWS

2) Ensure that you have an appropriate retention policy for this database

3) Ensure that appropriate security measures are in place for this data, including encryption and access policies

4) Regularly test the recovery of this data and restoration of your system.

B) PILOT LIGHT FOR QUICK RECOVERY INTO AWS

A minimal version of the environment is always running in the cloud. Configure and run the most critical core elements of the system in AWS. When time comes form recovery, you can rapidly provision a full scale production environment around the critical core.

Typically, the system would have some pre-configured servers bundled as AMIs, which are ready to be started up at a moments notice.

Key Steps for preparation phase:

1) Setup Amazon EC2 instances to replicate or mirror database

2) Ensure that you have all supporting custom software packages available in AWS.

3) Create and maintain AMIs of key server where fast recover is required.

4) Regularly run these servers, test them, and apply any software updates and configuration changes

5) Consider automating the provisioning of AWS resources

Key Steps for recovery phase:

1) Start your application EC2 instance from your custom AMIs.

2) Resize existing database/data store instances to process the increased traffic

3) Add additional database/data store instances to give the DR site resilience in the data tier. If you are using Amazon RDS, turn on Multi-AZ to improve resilience

4) Change DNS to point at the Amazon EC2 servers

5) install and configure any non AMI based systems, ideally in an automated way.

C) WARM STANDBY SOLUTION IN AWS

A DR scenario in which a scaled down version of a fully functional environment is always running in the cloud. Fully duplicate business critical systems and have them always on

Key Steps for preparation phase:

1) Setup Amazon EC2 instances to replicate or mirror data

2) Create and maintain AMIs

3) Run your application using a minimal footprint of EC2 instances or AWS infrastructure.

4) Patch and update software and configuration files in line with your live environments.

Key Steps for recovery phase:

1) Increase the size of the Amazon EC2 fleets in service with the load balancer (horizontal scaling)

2) Start applications on larger Amazon EC2 instance types as needed ( vertical scaling)

3) Either manually change the DNS records, or use Amazon Route 53 automated health checks so that all traffic is routed to the AWS environment.

4) Consider using Auto Scaling to right size the fleet or accommodate the increased load

5) Add resilience or scale up your database

D) MULIT SITE SOLUTION DEPLOYED ON AWS AND ON SITE

An Active Active configuration. The data replication method is determined by the RPO.

Key Steps for preparation phase:

1) Set up your AWS environment to duplicate your production environments

2) Set up DNS weighting or similar traffic routing technology to distribute incoming requests to both sites. Configure automated fail over to reroute traffic away from the affected site.

Key Steps for recover phase:

1) Either manually or by using DNS fail over, change the DNS weighting so that all requests are sent to the AWS site

2) Have application logic for fail over to use the local AWS database server for all queries

3) Consider using Auto Scaling to automatically right size the AWS fleet

DEMONSTRATE ABILITY TO IMPLEMENT DR FOR SYSTEMS BASED ON RPO AND RTO

RTO – recovery time objective. The time it takes after a disruption to restore a business process to its service level.

RPO – recovery point objective. The acceptable amount of data loss measured in time.

1. **Deployment Concepts with Auto Scaling**

The following are some of the types of questions that you want to consider when dealing with Auto Scaling and Deployments.

* When we are doing deployments, we are going to want to avoid downtime. How can we assure that our ASG can be deployed in such a way as to avoid downtime? In a general sense, one way to avoid downtime is to make sure that we are not delivering traffic to affected parts of out ASG. We will address the details in this section of the course.

* How long to deploy code and configure an instance? There is a timing component to launch instances in an ASG and to deploy code and configure on  that instance. This is where lifecycle hooks comes into play. One thing we could do is use CloudFormation init scripts to configure our new instances as they launch.

* How do you test new launch configurations?

* How do you phase out older launch configurations?, While phasing in newer configurations

1. **Auto Scaling Termination Policies**

This section provides a detailed explanation of the available termination policies for an Auto Scaling group as well as a detailed walkthrough of the default termination policy. Understanding the default termination policy will give you a solid foundation for understanding termination policies as a whole. We will also learn how to set up termination policies for an Auto Scaling group in the AWS Management Console. Termination policies answers one simple question: which instance should be terminated and why?

Amazon EC2 Auto Scaling supports the following custom termination policies:

**OldestInstance.** Terminate the oldest instance in the group. This option is useful when you're upgrading the instances in the Auto Scaling group to a new EC2 instance type. You can gradually replace instances of the old type with instances of the new type.

**NewestInstance.** Terminate the newest instance in the group. This policy is useful when you're testing a new launch configuration but don't want to keep it in production.

**OldestLaunchConfiguration.** Terminate instances that have the oldest launch configuration. This policy is useful when you're updating a group and phasing out the instances from a previous configuration.

**ClosestToNextInstanceHour.** Terminate instances that are closest to the next billing hour. This policy helps you maximize the use of your instances that have an hourly charge.

Default. Terminate instances according to the default termination policy. This policy is useful when you have more than one scaling policy for the group.

**OldestLaunchTemplate.** Terminate instances that have the oldest launch template. With this policy, instances that use the noncurrent launch template are terminated first, followed by instances that use the oldest version of the current launch template. This policy is useful when you're updating a group and phasing out the instances from a previous configuration.

**AllocationStrategy.** Terminate instances in the Auto Scaling group to align the remaining instances to the allocation strategy for the type of instance that is terminating (either a Spot Instance or an On-Demand Instance). This policy is useful when your preferred instance types have changed. If the Spot allocation strategy is lowest-price, you can gradually rebalance the distribution of Spot Instances across your N lowest priced Spot pools. If the Spot allocation strategy is capacity-optimized, you can gradually rebalance the distribution of Spot Instances across Spot pools where there is more available Spot capacity. You can also gradually replace On-Demand Instances of a lower priority type with On-Demand Instances of a higher priority type.

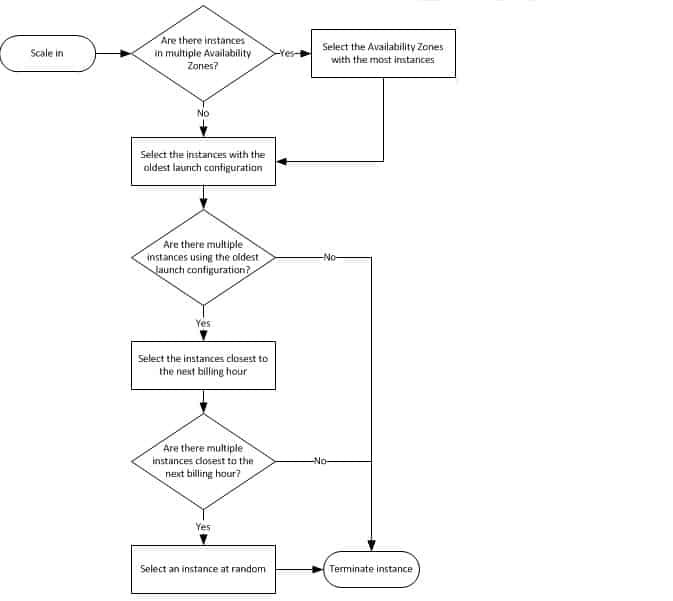
\*Termination policies are evaluated left to right. If our first policy does not find a suitable instance to terminate, our next policy inline gets evaluated. Best practice is to keep default policy the last which will be like the error handling (analogy). Even when we have a custom policy, auto scaling will first of all check the in balances across AZs.

**Default Termination Policy**

The default termination policy is designed to help ensure that your instances span Availability Zones evenly for high availability. The default policy is kept generic and flexible to cover a range of scenarios.

The default termination policy behavior is as follows:

1. Determine which Availability Zones have the most instances, and at least one instance that is not protected from scale in.
2. Determine which instances to terminate so as to align the remaining instances to the allocation strategy for the On-Demand or Spot Instance that is terminating.    
   For example, after your instances launch, you change the priority order of your preferred instance types. When a scale-in event occurs, Amazon EC2 Auto Scaling tries to gradually shift the On-Demand Instances away from instance types that are lower priority.
3. Determine whether any of the instances use the oldest launch template or configuration:
   1. [For Auto Scaling groups that use a launch template]   
      Determine whether any of the instances use the oldest launch template unless there are instances that use a launch configuration. Amazon EC2 Auto Scaling terminates instances that use a launch configuration before instances that use a launch template.
   2. [For Auto Scaling groups that use a launch configuration]   
      Determine whether any of the instances use the oldest launch configuration.
4. After applying all of the above criteria, if there are multiple unprotected instances to terminate, determine which instances are closest to the next billing hour. If there are multiple unprotected instances closest to the next billing hour, terminate one of these instances at random.

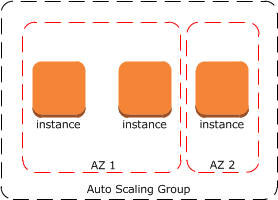


**Example**

Consider an Auto Scaling group that uses a launch configuration. It has one instance type, two Availability Zones, a desired capacity of two instances, and scaling policies that increase and decrease the number of instances by one when certain thresholds are met. The two instances in this group are distributed as follows.



When the threshold for the scale-out policy is met, the policy takes effect and the Auto Scaling group launches a new instance. The Auto Scaling group now has three instances, distributed as follows.



When the threshold for the scale-in policy is met, the policy takes effect and the Auto Scaling group terminates one of the instances. If you did not assign a specific termination policy to the group, it uses the default termination policy. It selects the Availability Zone with two instances, and terminates the instance launched from the oldest launch configuration. If the instances were launched from the same launch configuration, the Auto Scaling group selects the instance that is closest to the next billing hour and terminates it.

1. **Suspending AutoScaling Policies**

These processes are

* Launch
* Terminate
* HealthCheck
* ReplaceUnhealthy
* AZRebalance
* AlarmNotification
* ScheduledActions
* AddToLoadBalancer
* AddToLoadBalancer—Adds instances to the attached load balancer or target group when they are launched.
* AlarmNotification—Accepts notifications from CloudWatch alarms that are associated with the group's scaling policies.
* AZRebalance—Balances the number of EC2 instances in the group evenly across all of the specified Availability Zones when the group becomes unbalanced, for example, a previously unavailable Availability Zone returns to a healthy state.
* HealthCheck—Checks the health of the instances and marks an instance as unhealthy if Amazon EC2 or Elastic Load Balancing tells Amazon EC2 Auto Scaling that the instance is unhealthy. This process can override the health status of an instance that you set manually.
* ReplaceUnhealthy—Terminates instances that are marked as unhealthy and then creates new instances to replace them.
* ScheduledActions—Performs the scheduled scaling actions that you create or that are created by the predictive scaling feature of AWS Auto Scaling.

We can suspend any of these processes and we can suspend more than one of the process.

**Suspending Both Launch and Terminate**

When you suspend the Launch and Terminate process types together, the following happens:

* Your Auto Scaling group cannot initiate scaling activities or maintain its desired capacity.
* If the group becomes unbalanced between Availability Zones, Amazon EC2 Auto Scaling does not attempt to redistribute instances evenly between the Availability Zones that are specified for your Auto Scaling group.
* Your Auto Scaling group cannot replace instances that are marked unhealthy.

When you resume the Launch and Terminate process types, Amazon EC2 Auto Scaling replaces instances that were marked unhealthy while the processes were suspended and may attempt to rebalance the group. Scaling activities will also resume.

But why would we want to suspend AS processes? It's mostly for investigation or troubleshooting; without triggering an AS action. For example, assume we want to resize an instance in an ASG. In order to do that, we will have to stop that instance. It's highly probably that the AS is going to see this an mark the instance as unhealthy and attempt to replace it. In this specific case, we could suspend the launch process while we stop the instance and resize.

Terminate

* Your Auto Scaling group does not scale in for alarms or scheduled actions that occur while the process is suspended. In addition, the following processes are disrupted:
* AZRebalance is still active but does not function properly. It can launch new instances without terminating the old ones. This could cause your Auto Scaling group to grow up to 10 percent larger than its maximum size, because this is allowed temporarily during rebalancing activities. Your Auto Scaling group could remain above its maximum size until you resume the Terminate process. When Terminate resumes, AZRebalance gradually rebalances the Auto Scaling group if the group is no longer balanced between Availability Zones or if different Availability Zones are specified.
* ReplaceUnhealthy is inactive but not HealthCheck. When Terminate resumes, the ReplaceUnhealthy process immediately starts running. If any instances were marked as unhealthy while Terminate was suspended, they will be immediately replaced.

Launch

* Your Auto Scaling group does not scale out for alarms or scheduled actions that occur while the process is suspended. AZRebalance stops rebalancing the group. ReplaceUnhealthy continues to terminate unhealthy instances, but does not launch replacements. When you resume Launch, rebalancing activities and health check replacements are handled in the following way:
  + AZRebalance gradually rebalances the Auto Scaling group if the group is no longer balanced between Availability Zones or if different Availability Zones are specified.
  + ReplaceUnhealthy immediately replaces any instances that it terminated during the time that Launch was suspended.

AddToLoadBalancer

* Amazon EC2 Auto Scaling launches the instances but does not add them to the load balancer or target group. When you resume the AddToLoadBalancer process, it resumes adding instances to the load balancer or target group when they are launched. However, it does not add the instances that were launched while this process was suspended. You must register those instances manually.

AlarmNotification

* Amazon EC2 Auto Scaling does not execute scaling policies when a CloudWatch alarm threshold is in breach. Suspending AlarmNotification allows you to temporarily stop scaling events triggered by the group's scaling policies without deleting the scaling policies or their associated CloudWatch alarms. When you resume AlarmNotification, Amazon EC2 Auto Scaling considers policies with alarm thresholds that are currently in breach.

AZRebalance

* Your Auto Scaling group does not attempt to redistribute instances after certain events. However, if a scale-out or scale-in event occurs, the scaling process still tries to balance the Availability Zones. For example, during scale out, it launches the instance in the Availability Zone with the fewest instances. If the group becomes unbalanced while AZRebalance is suspended and you resume it, Amazon EC2 Auto Scaling attempts to rebalance the group. It first calls Launch and then Terminate.

HealthCheck

* Amazon EC2 Auto Scaling stops marking instances unhealthy as a result of EC2 and Elastic Load Balancing health checks. Your custom health checks continue to function properly, however. After you suspend HealthCheck, if you need to, you can manually set the health state of instances in your group and have ReplaceUnhealthy replace them.

ReplaceUnhealthy

* Amazon EC2 Auto Scaling stops replacing instances that are marked as unhealthy. Instances that fail EC2 or Elastic Load Balancing health checks will still be marked as unhealthy. As soon as you resume the ReplaceUnhealthly process, Amazon EC2 Auto Scaling replaces instances that were marked unhealthy while this process was suspended. The ReplaceUnhealthy process calls both of the primary process types—first Terminate and then Launch.

ScheduledActions

* Amazon EC2 Auto Scaling does not execute scaling actions that are scheduled to run during the suspension period. When you resume ScheduledActions, Amazon EC2 Auto Scaling only considers scheduled actions whose execution time has not yet passed.

CloudWatch Events

Amazon CloudWatch Events enables you to respond quickly to application availability issues or configuration changes that might impact performance or security by notifying you of AWS resource changes in near-real-time. You simply write rules to indicate which events are of interest to your application and what automated action to take when a rule matches an event. You can, for example, invoke AWS Lambda functions or notify an Amazon Simple Notification Service (SNS) topic.

Schedule AWS Lambda Functions Using CloudWatch Events

You can set up a rule to run an AWS Lambda function on a schedule. This tutorial shows how to use the AWS Management Console

Step 1: Create an AWS Lambda Function

Create a Lambda function to log the scheduled events. Specify this function when you create your rule.

To create a Lambda function

Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.

If you are new to Lambda, you see a welcome page. Choose Get Started Now. Otherwise, choose Create a Lambda function.

On the Select blueprint page, type hello for the filter and choose the hello-world blueprint.

On the Configure triggers page, choose Next.

On the Configure function page, do the following:

Type a name and description for the Lambda function. For example, name the function "LogScheduledEvent".

Edit the sample code for the Lambda function. For example:

'use strict'; exports.handler = (event, context, callback) => { console.log('LogScheduledEvent'); console.log('Received event:', JSON.stringify(event, null, 2)); callback(null, 'Finished'); };

For Role, choose Choose an existing role. For Existing role, select your basic execution role. Otherwise, create a new basic execution role.

Choose Next.

On the Review page, choose Create function.

Step 2: Create a Rule

Create a rule to run your Lambda function on a schedule.

To create a rule using the console

Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

In the navigation pane, choose Events, Create rule.

For Event Source, do the following:

Choose Schedule.

Choose Fixed rate of and specify the schedule interval (for example, 5 minutes).

For Targets, choose Add target, Lambda function.

For Function, select the Lambda function that you created.

Choose Configure details.

For Rule definition, type a name and description for the rule.

Choose Create rule.

Set AWS Systems Manager Automation as a CloudWatch Events Target

You can use CloudWatch Events to invoke AWS Systems Manager Automation on a regular timed schedule, or when specified events are detected. This tutorial assumes that you are invoking Systems Manager Automation based on certain events.

To create the CloudWatch Events rule

Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

In the navigation pane, choose Events, Create rule.

For Event source, do the following:

Choose Event Pattern and choose Build event pattern to match events by service.

For Service Name and Event Type, choose the service and event type to use as the trigger.

Depending on the service and event type you choose, you may need to specify additional options under Event Source.

For Targets, choose Add Target, SSM Automation.

For Document, choose the Systems Manager document to run when the target is triggered.

(Optional), To specify a certain version of the document, choose Configure document version.

Under Configure parameter(s), choose No Parameter(s) or Constant.

If you choose Constant, specify the constants to pass to the document execution.

CloudWatch Events can create the IAM role needed for your event to run:

To create an IAM role automatically, choose Create a new role for this specific resource.

To use an IAM role that you created before, choose Use existing role.

Choose Configure details. For Rule definition, type a name and description for the rule.

Choose Create rule.

Log the State of an Amazon EC2 Instance Using CloudWatch Events

You can create an AWS Lambda function that logs the changes in state for an Amazon EC2 instance. You can choose to create a rule that runs the function whenever there is a state transition or a transition to one or more states that are of interest. In this tutorial, you log the launch of any new instance.

Step 1: Create an AWS Lambda Function

Create a Lambda function to log the state change events. You specify this function when you create your rule.

To create a Lambda function

Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.

If you are new to Lambda, you see a welcome page. Choose Get Started Now. Otherwise, choose Create a Lambda function.

On the Select blueprint page, type hello for the filter and choose the hello-world blueprint.

On the Configure triggers page, choose Next.

On the Configure function page, do the following:

Type a name and description for the Lambda function. For example, name the function "LogEC2InstanceStateChange".

Edit the sample code for the Lambda function. For example:

'use strict';

exports.handler = (event, context, callback) => {

console.log('LogEC2InstanceStateChange');

console.log('Received event:', JSON.stringify(event, null, 2));

callback(null, 'Finished');

};

For Role, choose Choose an existing role. For Existing role, select your basic execution role. Otherwise, create a new basic execution role.

Choose Next.

On the Review page, choose Create function.

Step 2: Create a Rule

Create a rule to run your Lambda function whenever you launch an Amazon EC2 instance.

To create a CloudWatch Events rule

Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.

In the navigation pane, choose Events, Create rule.

For Event source, do the following:

Choose Event Pattern.

Choose Build event pattern to match events by service.

Choose EC2, EC2 Instance State-change Notification.

Choose Specific state(s), Running.

By default, the rule matches any instance in the region. To make the rule match a specific instance, choose Specific instance(s) and then select one or more instances.

The Event selector pane

For Targets, choose Add target, Lambda function.

For Function, select the Lambda function that you created.

Choose Configure details.

For Rule definition, type a name and description for the rule.

Choose Create rule.

Step 3: Test the Rule

To test your rule, launch an Amazon EC2 instance. After waiting a few minutes for the instance to launch and initialize, you can verify that your Lambda function was invoked.

Protecting Data Using Encryption

Data protection refers to protecting data while in-transit (as it travels to and from Amazon S3) and at rest (while it is stored on disks in Amazon S3 data centers). You can protect data in transit using Secure Sockets Layer (SSL) or client-side encryption. You have the following options for protecting data at rest in Amazon S3:

Server-Side Encryption – Request Amazon S3 to encrypt your object before saving it on disks in its data centers and then decrypt it when you download the objects.

Client-Side Encryption – Encrypt data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

For more information about server-side encryption and client-side encryption, review the topics listed below.

Amazon EBS Encryption

Amazon EBS encryption offers a straight-forward encryption solution for your EBS resources that doesn't require you to build, maintain, and secure your own key management infrastructure. It uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and snapshots.

Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

How EBS Encryption Works

You can encrypt both the boot and data volumes of an EC2 instance. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

Data at rest inside the volume

All data moving between the volume and the instance

All snapshots created from the volume

All volumes created from those snapshots

EBS encrypts your volume with a data key using the industry-standard AES-256 algorithm. Your data key is stored on-disk with your encrypted data, but not before EBS encrypts it with your CMK. Your data key never appears on disk in plaintext. The same data key is shared by snapshots of the volume and any subsequent volumes created from those snapshots.

AWS Key Management Service (KMS)

AWS KMS provides a web interface to generate and manage cryptographic keys and operates as a cryptographic service provider for protecting data. AWS KMS offers traditional key management services integrated with AWS services to provide a consistent view of customers’ keys across AWS, with centralized management and auditing.

Creating Keys Using AWS Key Management Service

Let’s create a key! I start by opening up the Identity and Access Management Console and clicking on Encryption Keys. At this point I can see my existing keys (KMS created these keys when I opted to encrypt my S3 buckets, EBS volumes, and my Redshift data warehouse):

Services v

Dashboard

Details

Groups

Users

Roles

Identity Providers

Password Policy

Credential Report

Encryption Keys

Edit v

Create Key

Key Actions

Filter: US East (N. Virginia). O

Alias

aws/s3

aws/ebs

aws/redshift

Search

Key ID :

6a4e4ad3-f961400b-9040-21c6beOec632

6cfb2f9749724f86-b3e2-c040ea97fb38

86048d94-02be48c6-8f96-946ff961af86

I can click Create Key to start the process of creating a new key. The first step is to define an alias (display name) for the key and to provide a description:

Services v

Edit v

Create Key in US East (N.

Virginia)

Create Alias and Description

Define an alias and a description for this key. Properties of this key can

Step 1

step 2:

Step 3:

: Create Alias and Description

Define Key Administrators

Define Key Usage Permissions

Alias (required)

Description

AppKey

key for application encryp

Next, I need to decide who is allowed to administer the new key by choosing one or more IAM users and/or roles:

Services v

Edit v

Create Key in US East (N.

Virginia)

Step 1: Create Alias and Description

Step 2 : Define Key Administrators

Step 3: Define Key Usage Permissions

Define Key Administrators

Define which IAM users and roles within this account can administer this

administer this key. Learn More.

Search

Name

awscli

awsroadtrip

cwlogger

nam

In the third and final step, I decide which users, roles, and other AWS accounts are allowed to use the key to encrypt and decrypt data. The permissions defined in this step apply to direct (API-driven) usage and to usage that originates with an AWS service that is integrated in to KMS:

Services v

Edit v

Create Key in US East (N.

Virginia)

Step 1: Create Alias and Description

Step 2: Define Key Administrators

Step 3 : Define Key Usage

Permissions

Define Key Usage Permissions

• This Account

Define which users and roles in this account can use this key to encrypt and

when using AWS semces currently integrated With KMS\_ Users and roles in t

key. Learn More.

Search

Name

awscli

awsroadtrip

cwlogger

dynamic-dynamodb

jeff

Using Keys with EBS and S3

Now that I have created a key, I can easily use it with any of the AWS services that I mentioned above (and others in the future)! Here’s how I would create an EBS volume that is encrypted with my key:

Create Volume

Size (GiB)

lops

Availability Zone

Snapshot ID

Encryption

Master Key

x

General Purpose (SSD)

0)

O)

300 3000

us-east- I a

(Min: IGiB. Max: 1024GiB)

(3000 IOPS bursts and baseline of 3

lops per 6B)

Search (case-insensifve)

Encrypt this volume

default) aws,'ebs

(default) aws/ebs

create

AppKey

And here’s how I would encrypt an S3 object after it is uploaded:

Set Details

All /

Cancel

Details: Set additional details for all of the objects you upload. You can choose between Standard Storage and Reduced Redundancy

You can also choose whether or not to encrvpt vowr files.

Us e Reduced Redundancy Storage

Use server Side Encryption

O

Use the Amazon S3 service master key

S3 Will decrypt the for anyone with permission to access this Object

@ Use an AWS Key Management Service master key

S3 will decrypt the object tor anyone with permission to access this objed and permission to use the master key.

Master Key. aws/s3 (default)

Description:

aws/s3 (default)

A.ppKe•i

Key ID:

Enter a key ARN

Select Files

Set permissions

Start upload

Cancel

Managing and Auditing Key Usage

I can also get full information on a key with one click:

Services

Dashboard

Details

Groups

Users

Roles

Identity Pro•åders

password policy

Credential Report

I Encryption Keys

Edit

Jeff Barr

Global SuOO•ort

Encryption Keys> AppKey

Summary

Region

ARN of key

Al ias

Description

Key Administrators

Key Usage

Key Rotation

Audit

Back to Encryption Keys

us-east-I

arm

AppKey

Key for application encryption

Cancel

Save Changes

You can use CloudTrail to understand when and where your encryption key has been used. Leam More.

All API requests against a key are logged to AWS CloudTrail so that you can track and understand how, when, and by whom the keys where used.

\*What is Envelope encryption

Answer : Envelope encryption is the mechanism to protect encryption keys. It is the practice of encrypting plaintext data with a data key and then encrypting the data key under another key. This ensures that the encryption key remains protected. Envelop encryption ensures various benefits such as the data key can be stored alongside the encrypted data as the key itself is encrypted, saves time by eliminating the need of re encrypting large objects, produces smaller ciphertexts with easier key management features.

What is a CMK in the context of KMS?

CMK stands for Customer Master key. It is the primary resource in AWS KMS. A CMK is a logical representation of a master key. It includes wwwwwwmetadata and contains the key material used to encrypt and decrypt data. CMKs can be symmetric and asymmetric. A symmetric CMK represents a 256 bit key that is used for encryption and decryption. An asymmetric CMK represents an RSA key pair that is used for encryption and decryption or signing and verification. By default AWS KMS creates the key material for a CMK. This cannot be exported, extracted viewed or managed. However, the user may use his own key material, imported to be used as CMK. These are known as customer managed CMKs.

Pass CCD is ea

S chi ikkii

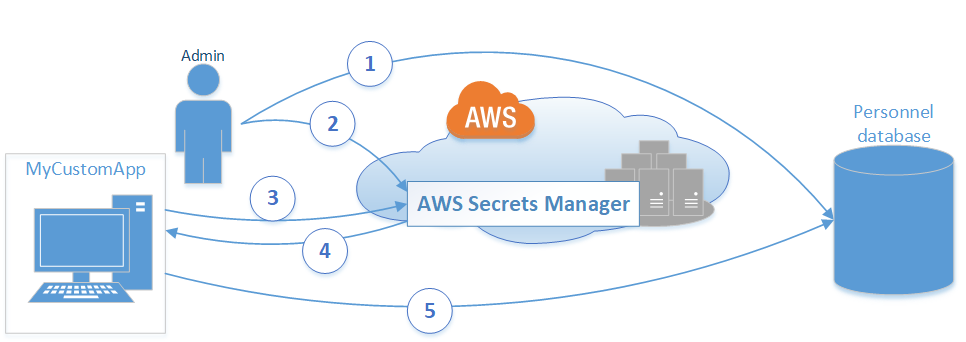
AWS Secrets Manager is an AWS service that makes it easier for you to manage secrets. *Secrets* can be database credentials, passwords, third-party API keys, and even arbitrary text. You can store and control access to these secrets centrally by using the Secrets Manager console, the Secrets Manager command line interface (CLI), or the Secrets Manager API and SDKs.

In the past, when you created a custom application that retrieves information from a database, you typically had to embed the credentials (the secret) for accessing the database directly in the application. When the time came to rotate the credentials, you had to do more than just create new credentials. You had to invest time to update the application to use the new credentials. Then you had to distribute the updated application. If you had multiple applications with shared credentials and you missed updating one of them, the application would break. Because of this risk, many customers have chosen not to regularly rotate their credentials, which effectively substitutes one risk for another.

Secrets Manager enables you to replace hardcoded credentials in your code, including passwords, with an API call to Secrets Manager to retrieve the secret programmatically. This helps ensure the secret can't be compromised by someone examining your code, because the secret isn't there. Also, you can configure Secrets Manager to automatically rotate the secret for you according to a specified schedule. This enables you to replace long-term secrets with short-term ones, which significantly reduces the risk of compromise.

Basic Secrets Manager Scenario

The following diagram illustrates the most basic scenario. It shows how you can store credentials for a database in Secrets Manager, and then use those credentials in an application that needs to access the database.



1. The database administrator creates a set of credentials on the Personnel database for use by an app called MyCustomApp. The administrator also configures those credentials with the permissions required for the app to access the Personnel database.
2. The database administrator stores the credentials as a secret in Secrets Manager named *MyCustomAppCreds*. Then, Secrets Manager encrypts and stores the credentials within the secret as the *protected secret text*.
3. When MyCustomApp needs to access the database, the app queries Secrets Manager for the secret named *MyCustomAppCreds*.
4. Secrets Manager retrieves the secret, decrypts the protected secret text, and returns the secret to the client app over a secured (HTTPS with TLS) channel.
5. The client app parses the credentials, connection string, and any other required information from the response and then uses the information to access the database server.

**Creating a Basic Secret**

1. Sign in to the AWS Secrets Manager console at <https://console.aws.amazon.com/secretsmanager/>.
2. Choose **Store a new secret**.
3. In the **Select secret type** section, specify the kind of secret that you want to create by choosing one of the following options. Then supply the required information.   
   Amazon RDSAmazon RedshiftDocumentDB databaseOther databasesOther type of secret   
   Amazon RDS   
   This secret is for one of the [supported database services](https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html#full-rotation-support) for which Secrets Manager provides full rotation support with a preconfigured Lambda rotation function. You need specify only the authentication credentials because Secrets Manager learns everything else it needs by querying the database instance.
   1. Type the user name and password that allow access to the database. Choose a user that has only the permissions that are required by the customer who will access this secret.
   2. Choose the AWS KMS encryption key that you want to use to encrypt the protected text in the secret. If you don't choose one, Secrets Manager checks to see if there's a default key for the account, and uses it if it exists. If a default key doesn't exist, Secrets Manager creates one for you automatically. You can also choose **Add new key** to create a custom CMK specifically for this secret. To create your own AWS KMS CMK, you must have permissions to create CMKs in your account.
   3. Choose the database instance from the list. Secrets Manager retrieves the connection details about the database by querying the chosen instance.   
      Amazon Redshift   
      This secret is for an Amazon Redshift cluster. You need specify only the authentication credentials because Secrets Manager learns everything else it needs by querying the database instance.
   4. Type the user name and password that allow access to the database.
   5. Choose the AWS KMS encryption key that you want to use to encrypt the protected text in the secret. If you don't choose one, Secrets Manager checks to see if there's a default key for the account, and uses it if it exists. If a default key doesn't exist, Secrets Manager creates one for you automatically. You can also choose **Add new key** to create a custom CMK specifically for this secret. To create your own AWS KMS CMK, you must have permissions to create CMKs in your account.
   6. Choose the correct database engine.
   7. Specify the connection details by typing the database server's IP address, database name, and TCP port number.   
      DocumentDB database   
      This secret is for a DocumentDB database. You need specify only the authentication credentials because Secrets Manager learns everything else it needs by querying the database instance.
   8. Type the user name and password that allow access to the database.
   9. Choose the AWS KMS encryption key that you want to use to encrypt the protected text in the secret. If you don't choose one, Secrets Manager checks to see if there's a default key for the account, and uses it if it exists. If a default key doesn't exist, Secrets Manager creates one for you automatically. You can also choose **Add new key** to create a custom CMK specifically for this secret. To create your own AWS KMS CMK, you must have permissions to create CMKs in your account.
   10. Choose the correct database engine.
   11. Specify the connection details by typing the database server's IP address, database name, and TCP port number.   
       Other databases   
       This secret is for a database service that Secrets Manager knows about and supports. However, Secrets Manager needs you to provide additional information about the database. To rotate this secret, you must write a custom Lambda rotation function that can parse the secret and interact with the service to rotate the secret on your behalf.
   12. Type the user name and password that allow access to the database that you want to protect in this secret.
   13. Choose the AWS KMS encryption key that you want to use to encrypt the protected text in the secret. If you don't choose one, Secrets Manager checks to see if there's a default key for the account, and uses it if it exists. If a default key doesn't exist, Secrets Manager creates one for you automatically. You can also choose **Add new key** to create a custom CMK specifically for this secret. To create your own AWS KMS CMK, you must have permissions to create CMKs in your account.
   14. Choose the type of database engine that runs your database.
   15. Specify the connection details by typing the database server's IP address, database name, and TCP port number.   
       Other type of secret   
       This secret is for a database or service that Secrets Manager doesn't natively know about. You must supply the structure and details of your secret. To rotate this secret, you must write a custom Lambda rotation function that can parse the secret and interact with the service to rotate the secret on your behalf.
   16. Specify the details of your custom secret as **Key** and **Value** pairs. For example, you can specify a key of UserName, and then supply the appropriate user name as its value. Add a second key with the name of Password and the password text as its value. You could also add entries for Database name, Server address, TCP port, and so on. You can add as many pairs as you need to store the information you require.   
       Alternatively, you can choose the **Plaintext** tab and enter the secret value in any way you like.
   17. Choose the AWS KMS encryption key that you want to use to encrypt the protected text in the secret. If you don't choose one, Secrets Manager checks to see if there's a default key for the account, and uses it if it exists. If a default key doesn't exist, Secrets Manager creates one for you automatically. You can also choose **Add new key** to create a custom CMK specifically for this secret. To create your own AWS KMS CMK, you must have permissions to create CMKs in your account.
4. For **Secret name**, type an optional path and name, such as **production/MyAwesomeAppSecret** or **development/TestSecret**. Notice that the use of the slash character enables you to structure your secrets into a hierarchy, such as grouping by the deployment environment, which is useful for organizing and managing your secrets at scale. You can optionally add a description to help you remember the purpose of this secret later on.   
   The secret name must be ASCII letters, digits, or any of the following characters: /\_+=.@-
5. (Optional) In the **Tags** section, you can add one or more tags to your secret. A tag consists of a key and a value that you define. Tags assist with managing your AWS resources. You can create tags that associate resources with your organization's structure, such as Key="Department" and Value="Accounting". This can help with cost allocation and tracking. You can assign tags to group resources together by the application that uses them (Key="AppName" and Value="HRDatabase"). You can create tags for almost any purpose. Each resource, like a secret, can have several tags attached.    
   **Important**   
   Don't store sensitive information about a secret in its tags. Store sensitive information only in the secret value (the SecretString or SecretBinary fields) of the secret where the information is protected with encryption.
6. After you complete the **Name**, **Description**, and any **Tags**, choose **Next**.
7. (Optional) At this point, you can configure rotation for your secret. Because we're working on a "basic" secret without rotation, keep it as **Disable automatic rotation**, and then choose **Next**.
8. Review your settings, and then choose **Store secret** to save everything that you entered as a new secret in Secrets Manager.

**Retrieving the Secret Value**

One of the main purposes of Secrets Manager is to enable you to programmatically and securely retrieve your secrets in your custom applications. However, you can also retrieve them by using the console or the CLI tools.

This section includes procedures and commands that describe how to retrieve the secret value of a secret.

**Using Console:**

1. Open the Secrets Manager console at <https://console.aws.amazon.com/secretsmanager/>.
2. In the list of secrets in your account, choose the name of the secret that you want to view.   
   The **Secret details** page appears. It displays all of the chosen secret's configuration details except for the encrypted secret text.
3. In the **Secret value** section, choose **Retrieve secret value**.
4. Choose **Secret key/value** to see the credentials parsed out as individual keys and values. Choose **Plaintext** to see the original JSON text string that's encrypted and stored.

**Rotating Secrets for Supported Amazon RDS Databases**

You can configure AWS Secrets Manager to automatically rotate the secret for an Amazon RDS database. Secrets Manager uses a Lambda function that Secrets Manager provides.

When you enable rotation for a secret with **Credentials for RDS database** as the secret type, Secrets Manager can automatically create and configure a Lambda rotation function for you.

If your Amazon RDS DB instance is running in a VPC provided by Amazon VPC and that VPC doesn't have public internet access, then Secrets Manager also configures the Lambda function to run within that VPC. The other requirement is that the Lambda rotation function must be able to access a Secrets Manager service endpoint to call the required API operations. If one or more of your resources in the VPC must communicate with the internet, then you can configure the VPC with a NAT gateway to enable the Lambda rotation function to query the public Secrets Manager service endpoint.

* **What is a database?**

An electronic system that allows data to be easily stored, accessed and manipulated.

A database Management System (DBMS) is a type of software that allows you to define, manipulate, retrieve and manage data stored within a database.

* **Database Types:**

1. **Relational Database**

Made up of a set of tables with data that fits into a predefined category.

It uses structured query language (SQL) to manipulate data in a database

Think of a traditional spreadsheet

➢ Database ➢ Tables (work books) ➢ Rows ➢ Fields (Colum) -You need to understand all your field before you create a table. If you have to add any additional fields, you can have to update the entire table.

You need to understand all your need when building the table otherwise you may have to build the table entirely

ID First Name Surname Gender 1

* SQL server (a lot of accounting servers run on SQL)
* Oracle (used by the big enterprise)
* MySQL (open source Relational DB and popular in the development community)
* PostgreSQL
* Amazon Proprietary DB called Amazon Aurora with massive performance and redundancy
* MarioDB

1. **Distributed Databases:**

A distributed database is a database in which portions of the database are stored in multiple physical locations, and in which processing is dispersed or replicated among different points in a network.

1. **Cloud Database:**

A database that has been optimized or built for a virtualized environment, either in a hybrid cloud, public cloud or private cloud.

1. **NoSQL Databases (Non-Relation Database)**
   * A database that does not have a pre-defined schema. Different types of data can be stored in different items (rows) within a table.
   * Effective for big data when relational databases are not ideal.

Non-Relation Database

➢ Collection (Table) o Document (Row) ▪ Key Value pairs (Fields)

In a relational sense, a collection is a table, a document is a row and key value pairs are fields.

With non-relational DB, you don’t need to know how many documents you need. They are very flexible.

Very popular in the development community. It is usually in JSON format:

**Database Types in AWS:**

1. **Amazon Neptune:** Neptune is a fast, scalable graph database service. Neptune stores and navigate highly connected data. Its query processing engine is optimized for leading graph query language. Neptune is fully managed so you no longer need to worry about database management tasks such as hardware provisioning, software patching, setup, configuration, or backups.

1. **Amazon ElasticCache:**  ElasticCache offers a fully managed Redis and Memcached (caching in the cloud).

Deploy, run and scale popular open source compatible in-memory data stores.

Build data-intensive apps or improve the performance of your existing apps by retrieving data from high throughput and low latency in-memory data stores. If your application if reading a lot of data and giving it to the customer, then caching would be a good idea.

1. **Data warehousing (Amazon Redshift)**

Used for business intelligence. Used to pull very large and complex data sets. Usually used by management to do queries on data such as current performance versus targets. Business intelligence comprise of data, technology, analytics and human intelligent to provide insights that results in more successful business outcome.

**Use case:**

➢ Understanding number of page views on my website JJTECH INC

➢ Age and gender of the users who are visiting ➢ Session duration of the users.  ➢ Find out session based on country of origin

* Business systems based on data warehouse extract information from various organizational sources. The data is then transformed, cleaned and loaded into data warehouse. This data is then used to gain business insights. Data from various sources is integrated into one platform so it becomes easier for analysis. Various sources such log files, operating system, Flat files, database, CRM etc.
* These sources will send the data to ETL (Extraction, transformation and loading) takes place. Transform according to your need eg you don’t need certain columns or rows.
* While in the data warehouse, you then use certain tools to do BI

So what is the difference between relational database and data warehouse

* Redshift is a fast, scalable data warehouse that makes it simple and cost effective to analyze all your data across your data warehouse and data lake. Redshift delivers ten times faster performance than other data warehouse by using machine learning, massive parallel query execution, and columnar storage on high performance disks.

You can setup and deploy a new data warehouse in minutes and run queries across petabytes of data in your Redshift data warehouse, and exabytes of data in your data lake built on Amazon S3.

1. **Amazon DynamoDB (NoSQL)**

* DynamoDB is a nonrelational database that delivers reliable performance at any scale.
* It’s fully managed, multi-region, multi-master database that provides consistent single-digit millisecond latency, and offers built-in security, backup and restore, and in memory caching.

1. **Amazon RDS:**
   * RDS makes it easy to setup, operate, and scale a relational database in cloud. It provides cost efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so that you can give them the fast performance, high availability, security and compatibility they need.
   * RDS are available for several database instance types-optimized for memory, performance or I/O
   * You can use the AWS DB Migration Service to easily migrate or replicate your existing database to Amazon RDS.

Database Types in AWS- Amazon RDS (six familiar engines to choose from)

1. PostgreSQL

2. MySQL

3. MariaDB

4. Oracle

5. Microsoft SQL Server

6. Amazon Aurora

* RDS provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Why do you want a managed database server?

1. Because Amazon RDS takes over many of the difficult or tedious management tasks of a relational database.
2. When you buy a server, you get CPU, memory, storage and IOPS, all bundled together. With Amazon RDS, these are split apart so that you can scale them independently. If you need more CPU, less IOPS, or more storage, you can easily allocate them -based on the instance types.
3. Amazon RDS manages backups, software patching, automatic failover detection, and recovery.
4. Amazon RDS does not provide shell access to the database instances and restrict access to certain system procedures and tables that require advanced privileges.
5. You can have automated backups performed or manually create your own backup snapshots. You can use these backups to restore a database. The amazon RDS restore process works reliable and efficiently.
6. You can use MySQL, MariaDB or PostgreSQL Read Replicas to increase read scaling.
7. You can get high availability with a primary instance and a synchronous secondary instance that you can failover when problem occur.

Database Instance

* You can create and modify a DB instance by using AWS CLI, the amazon RDS API, or the AWS Management Console.
* Each DB instance runs a DB engine.

**Database instance class**

* The computation and memory capacity of a DB instance is determined by its DB instance class.
* The DB instance class you need depends on your processing power and memory requirements.
* You can select a DB instance that best meets your needs.
* If your needs change over time, you can modify the DB instance.
* RDS supports three types of instance classes:
  1. **Standard:** db.m1 -db.m5 (db.m5 is latest generation instance classes that provides more computing capacity than the previous db.m4 classes.
  2. **Burstable Performance:** db.t2 -instance classes that provides a baseline performance level, with the ability to burst to full CPU usage. It is recommended to use these instance classes for development and test servers or other non-production servers.
  3. **Memory Optimized**: Latest and current generation of instance classes optimized for memory-intensive applications.  a. db.x1e -latest b. db.x1-current generation – offers one of the lowest price per GiB of RAM c. db.r4 -current generation – offer a better price per GiB of RAM than db.r3 d. db.r3-previous generation -more computing capacity than the db.m2 e. db.m2 -previous generation

Things to consider when choosing an instance class:

* vCPU- The number of virtual central processing units (CPUs)
* Memory (GiB) -The RAM memory in gibibytes, allocated to the DB instance
* EBS Optimized – The DB instance uses an optimized configuration stack and provides additional, dedicated capacity for I/O
* Max. Bandwith (Mbps)-The maximum bandwidth in megabits per second.  •
* Network Performance- the network speed relative to other DB instance classes.

**Database Storage**:

* DB instances for RDS for MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft servers use Amazon Elastic Block Store (Amazon EBS) volumes for database and log storage.
* Depending on the amount of storage requested, amazon RDS automatically stripes across multiple EBS volumes to enhance performance.
* Amazon RDS provides three storage types: o General Purpose SSD (also known as gp2) o Provisioned IOPS (also known as io1) o Magnetic

They differ in performance characteristics and price, which means that you can tailor your storage performance and cost to the needs of your database workloads.

1. **General Purpose SSD (also known as gp2):** offer cost-effective storage that is ideal for a broad range of workloads. Volumes deliver single-digit millisecond latencies and the ability to burst to 3000 IOPS for extended periods of time. Baseline performance for this volume is determined by the volume’s size.

1. **Provisioned IOPS (also known as io1):** Designed to meet the needs of i/o-intensive workloads. Particularly database workloads, that require low I/O latency and consistent I/O throughput.

1. **Magnetic:** supported for backward compatibility. Not recommended to be used in AWS anymore; use general purpose SDD instead.

**Security in AWS for RDS**

* **Security Group** (up to 5 security groups associated with an instance)-add rules that control the inbound traffic to instances and separate set of rules that controls the outbound traffic. You can not specify deny rule and that’s where NACL comes in.

* **NACls** -optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. Rules are evaluated in numerical order, 10 before 200. By default, NACL is set to deny both in and out traffic.

**RDS Backups, Multi AZ and Read Replicas**

**Two types of backups:** automated backups and database snapshots.

* **Automated Backups** allow you to recover your database to any point in time within a retention period. The retention period can be between 1 and 35days.
  + Automated Backups will take a full daily snapshot and will also store transaction logs through out the day. When you do recover, AWS will first choose the most recent daily backup, and then apply transaction logs relevant to that day. This allows you to do a point in time recovery down to a second, with the retention period.
  + Automated Backups are enabled by default. The backup data is stored in S3 and you get free storage space equal to the size of your database. Backups are taken with a defined window. During the backup within, storage i/o may be suspended while your data is being backed up and you may experience elevated latency.

* **DB Snapshots are done manually** (i.e. they are user initiated). They are stored even after you delete the original RDS instance, unlike automated backups.
  + Whenever you restore either an automated snapshot or a manual snapshot, the restored version of the database will be a new RDS instance with a new DNS endpoint.

**Encryption:**

* Encryption at rest is supported for MySql, Oracle, SQL server, PostgreSQL, MariaDB and Aurora. Encryption is done using AWS KMS service.
* Once your RDS is encrypted, the data stored at rest in the underlying storage is encrypted, as are its automated backups, read replicas, and snapshots.

At the moment, encrypted an existing DB instance is not supported. To use Amazon encryption for an existing database, you must first create a snapshot, make a copy of that snapshot and encrypt the copy.

What is Multi-AZ deployment of your RDS?

* Multi-AZ allows you to have an exact copy of your production database in another Availability Zone. AWS handles the replication for you, so when your production database is written to, this write will automatically synchronize to the stand by database.
* In the event of planned database maintenance, DB instance failure, or an AZ failure, Amazon RDS will automatically failover to the standby so that database operations can resume quickly without administrative intervention.

**Multi-AZ is for Disaster recovery only.**

* It is not primarily used for improving performance. For performance improvement, you need read replicas. Multi-AZ databases include:
* SQL server (a lot of accounting servers run on SQL)
* Oracle (used by the big enterprise)
* MySQL (open source Relational DB and popular in the development community)
* PostgreSQL
* MarioDB

What is a read replica?

* Read Replicas are read-only database instances you can setup from a production database.
* AWS will automatically handle the replication between primary database and the read replica (s)
* Read replica operates as a DB instance that allows ready-only connections.
* Applications connect to a read replica the same way they do for any DB instance.
* Amazon RDS sets up a secure communications channel between the source DB instance and a Read Replica if that Read Replica is in a different AWS Region from the DB instance.
* Amazon RDS establishes any AWS security configurations needed to enable the secure channel, such as adding security group entries.
* Read replicas allow you to have a read-only copy of your production database.
* You use read replicas primarily for very heavy database workloads.
* Read replicas are available for  1. MySQL (open source Relational DB and popular in the development community) 2. PostgreSQL 3. Aurora 4. MarioDB
* Read replicas are used for scaling, not for DR.
* You must have automated backups turned on in order to deploy a read replica.
* You can have up to 5 read replica copies of any database.
* You can read replicas that have multi-AZ (released in 2018)
* Each read replica will have its own DNS end point.
* Read replicas can be promoted to be their own database. This breaks the replication.
* You can have a read replica in a second region.
* Read replica is created with the same storage type as the source DB instance. However, you can create a Read Replica that has a different storage type from the source DB instance.
* Read Replicas can also be promoted when needed to become standalone DB instances.
* You can create read replica of a read replica; latency may become an issue.

Deploying one or more read replicas might make sense in a variety of scenarios:

✓ Scaling beyond the compute or I/O capacity of a single DB instance for read-heavy database workloads.

You can direct this excess read traffic to one or more Read Replicas.

E.g view an application or add it to the cache or checkout time.

 ✓ Serving read traffic while the source DB instance is unavailable.

✓ Business reporting or data warehousing scenarios where you might want business reporting queries to run against a Read Replica, rather than your primary, production DB instance.

✓ Implementing Disaster Recovery. You can promote a Read Replica to a standalone instance as a disaster recover solution if the source DB instance fails. This process is not automatic.

✓ Scale read operations into a regions close to your users.

Data Basic Concepts:

* When you have to store data in a database in any environment, the traditional way is using a relational DB like Oracle, Postgre, SQL server. Relational DB technology is decades old which was really designed to run on a single machine. This is why RDB isn't really a good solution for a modern web scaled application which may have very high transactional volume or higher storage requirement than a single DB can withstand.

* Typically, in order to scale out a relational DB, you are forced to vertically scale, which means adding more CPU, memory and storage. With this, there is always a limit you will reach that  the DB engine can sustain.

That is why developers are looking for alternatives to RDB such as No SQL DB.

* With a No SQL DB, you can horizontally scale the cluster (these are not single instances)  and when you need more storage or transaction capacity, you can scale out the cluster by adding more nodes.
* Note that RDB have some features  which no SQL DB don’t have:
  + such as a very **common query language** is used in an R DB to query your data  where as with a no SQL DB, you have to use what ever the proprietary API is for the DB engine.
  + R DB are designed for the data to be normalized across a tables and then the tables join together, where as No SQL DB don’t have the concept of joining tables together. When you use a No SQL DB, you are storing the DB in a flat table structure instead of multiple tables that join together.
  + RBD are ACID compliant, (Atomicity, Consistency, Isolation and Durability). This means there is the capability to issue your commands against them inside a transaction. If the commands inside the transaction  fails then the DB can roll back to the state before the transaction. The No SQ DB don’t have that.
* One thing you can do with an RDB to kind of scale horizontally is by adding **read replicas**. These are read only copies of your data. If your application is making a lot more select statements than writes or updates or deletes, then this will help scale your DB horizontally but limited to a certain number of replicas.

**Types:**

* There are 02 main types of relational DB; OLTP and OLAP.
  + Online transactional processing OLTP:  This is a RDB optimized for transactions such as inserts, updates etc
  + Online analytical processing OLAP: This is a RBD optimized for queries and reporting.

A RTB can be both an OLTP and an OLAP but they are usually optimized for one use case over the other.  For EX a RDB such as oracle or SQL server are OLTP but a data ware house is an OLAP.

* There are 4 main types of no SQL DB; Column, Key Value, Document and Graph.
  + With Column, instead of storing the data in rows, it stores it by column. Column DB is really fast in terms of querying  because queries mostly looks at certain columns in the table. Amazon's column **DB is the Hbase on EMR.**
  + Key value is a very simple DB because it is a key and a value. Amazon's **Elastic Cache is the key value DB (this runs either Reddis or Memcache)**
  + Document DB is optimized for storing JSON or XML document Example of document **DB is Dynamo DB**
  + Graph DB is used by social media applications used because it is optimized for analyzing things like friends. EX is **Amazon Neptune**

* Examples of No SQL DB includes MongoDB, CouchDB, Hbase, Cassandra and Riak.

When it comes to deploying a DB on AWS, there are several choices we have to make. Such as if you should use an AWS managed service to deploy your DB or do you need to deploy your own DB on EC2 instances. This is a significant decision, because by using an AWS managed services, you are offloading the overhead work to them. You won't have to manage servers or do any security patches to the platform because AWS is doing that for you. Most of the managed DB are serverless which means you have no idea of what is happening underneath and you are just doing the API calls.

* Another reason a customer may not use the managed DB service is because the DB engine is designed for a particular DB service.

**Amazon's Relational Database Service (RDS):**

* When you use Amazon RDS (managed service), amazon's engineers are configuring and deploying the instance for you and provisioning the DB engine on the instance. In this case, they are responsible for all the operating functions of the instance such as security patching, installing and maintaining the DB engine, upgrades etc. For major updates you can decide if and when they should do so on your behalf.
* The only access you have is at the DB level, such as a DB connection through SQL tools or a command line interface but you cant SSH to the instance.
* Licensing is one big advantage of using RDS. If you had to provision and run this your self, you will have to buy a license from Microsoft which includes upfront cost. But when you use RDS which is a managed service, your license fee is included in the monthly fee you pay.
* Typically with relational DB you are limited to vertical scaling. RDS has push button scaling available;
  + If you want to increase the storage you can just go to the console and increase the storage
  + If you need more CPU capacity and more RAM, you can just change the DB instance to a larger size. You will need to restart the instance.
* A really powerful tool of RDS is the multi-AZ  option. This allows you to run 02 identical copies of your DB (Primary and hot standby) so in the case there is a failure in the primary, the hot standby takes over.
* Although you cant horizontally scale RDS, you can increase capacity by adding multiple read replicas. This will make a huge difference if your application is doing all its select statement from read replicas and use the master for inserts, updates and deletes.
* When you launch an RDS instance you can choose which DB engine you want. My SQL, MariaDB, Postgre, Oracle, & SQL server are all supported.

**Aurora:**

* Amazon has its own in-house home grown DB engine called Amazon  Aurora which is mysql or postgresql compatible:
  + It is 5X faster than MySQL. 3X faster than PostgreSQL.
  + Features of Aurora include continuous backup, 15 read replicas across 3 AZ, replication Lag of single digit millisecond, Backtrack in seconds to any point in time you want ( because of the continuous backup), Multi-Master Option.
  + It therefor combines the speed and availability of commercial databases with the simplicity and cost effectiveness of open source databases.
* Amazon has taken its home grown DB to the next level with **Amazon Aurora severless**. With the Aurora Severless, you have no concept of instances whatsoever, you just use the aurora DB engine and don’t worry about masters, read replicas etc. Amazon takes care of everything for you.
  + Another cool feature with the severless is that if you are not using your DB, they will scale it down to zero and you won't have to pay for any compute usage (pay as you go).

* **RDS Back ups & snapshot:** There are 02 types, automated and manual.
  + AWS provides automated point in time backups against the RDS database instance.
    - Automated backups are deleted once the DB instance is deleted and cannot be recovered.
    - The maximum retention period for automated snap shots is 35 days.
  + Manual snapshots can be retained for as long as you want
    - Snap shots can be copied to other regions for DR
    - RDS Encryption - snapshots are automatically encrypted.

RDS also allows server side encryption.

* **RDS Read Replicas:**
  + Read Replicas are asynchronous copies of the primary DB that are used for read only purpose (only allow read connections)
  + When you write new data to the primary DB, AWS copies it for you to the read replica.
  + You can create and have multiple read replicas for a primary DB
  + Cross region read replicas are supported
  + MySQL, MariaDB, PostgreSQL and Aurora currently support read replicas.
  + You can monitor replication lag by using cloud watch services.
  + If your primary DB has a disaster, you can promote the read replica to a primary.

* **RDS Multi AZ Failover:**
  + Multi AZ fail over (Automatic AZ Fail over) synchronously replicates data to a backup (stand by) DB instance located in another AZ but in the same region. The secondary instance is always kept in sync with the primary instance.
  + In the event of:
    - Service outage in an AZ
    - Primary DB failure
    - Instance server type is changed
    - Manual failover initiated
    - Updating software version

AWS will automatically switch the DNS record from the primary instance to the stand by instance.

* + RDS back ups are taken against the stand by instance to reduce I/O freeze and slow down IF multi-az is enabled.
  + In order for multi AZ to work, your primary DB instance must be launched into a **subnet group**( 02 or more subnets in different AZ).

When you connect to your RDS instance, you use a DNS name instead of an IP address. That is because if the primary DB fails, the secondary will be come the primary and you wont have to remap any IP address. The failover is pretty much automatic.

**AMAZON'S NO SQL DB**

**DynamoDB**

* This is a No SQL DB services offered by Amazon (managed service). It is very widely used to help make application stateless so they can be more easily scaled behind a load balancer. (similar to MongoDB except that it is home grown)
* Since this is a NoSQL DB, it runs on a cluster of nodes which can scale out/in depending on your storage needs and also how many reads per second and writes per second that the table needs (reads/writes capacity units). For EX if your table needs more read capacity units or write capacity units, then AWS adds more nodes to the cluster & vs.
* Since this is a fully managed service, the customer does not have full knowledge on how it functions. Interaction with Dynamo DB is mainly through API calls. The APIs include; putting data into a table, querying the table, scanning the table etc.
* **It is schemaless:** An RDS will require you to define columns in a table. Every row in that table has to have an entry in each column or a null value. If you need to insert a row into a table that has a new column, you have to alter the table to accommodate the row's new column. That is not the case with DynamoDB, its schemaless which means every item in the table can have different numbers of attributes and different attributes. (every attribute is itself a key and value). An item is just a collection of keys and values and a table is just a collection of items.
* The amount of capacity that AWS provisions for your DB table depends on your storage capacity and also your read capacity (reads and writes per second).  When you create a table you can define values for these numbers and DynamoDB will maintain the capacity based on what you have defined. You could also choose auto scaling for DynamoDB and AWS will adjust the cluster for you.
* Dynamo DB makes multiple copies of your data and spread it across multiple AZ. It is a very fault tolerance service.
* This is a Document Type of No SQL which makes it easy to store JSON document (which will become key value). Easily used when you have a flat table structure.
* Most RDS has an upper limit on how the storage can be but not with DynamoDB where you can just keep adding data and not worry about scaling.
* DynamoDB can easily integrate with other AWS services such as Elastic MapReduce. Can easily move data to a hadoop cluster in Elastic MapReduce.

You don’t have to store data in one solution. Many smart developers are using both RDS and DynamoDB at the same time.

To set up DynamoDB in the console.

1. Go to DynamoDb and click create a table
2. Name your table and primary key (partition) which has to be a unique item on the table. If it is not unique you will need to add a sort key.
3. Select the data type (string, Binary or number)
4. Complete your table settings. You can use the default settings or enable auto scaling instead.
5. Click create to create your table.

To load data to DynamoDB

1. You can load data from the CLI by using a simple comman such as below

Go back and check the table and you will see that your items are organized in your table. You can now scan or query your data.

**Elastic Cache:**

* This is a fully managed, in-memory data store cache services.
* It is used to improve Db performance by caching results of queries that are made to a DB.
* Elastic Cache provisions a node cluster which you can scale.
* This is generally used to offload other DB services such as RDS and DynamoDB so that you don’t have to keep asking the DB services the same questions (select statement). Instead of doing that, you get the result from the DB set once and you store it in cache such that your application will get it from there instead of asking the DB again.
* It improves application performance
* Memcached compliant cache cluster on demand
* Managed patching cache node failure detection and recovery of the instance
* Simple API calls to grow and shrink the cach cluster
* Seamlessly cache infront of Amazon RDS instance.
* The available engines to power ElasticCache include:
  + **Memcached**: Simple model, easy to scale and multi threaded.
  + **Redis**: Complex data types, Multi-AZ failover, Data Persistence, Snapshots for backup & restore, Encryption.

**Red Shift:**

* Red shift is an OLAP petabyte data base ware housing service.
* This DB service is very different because it is optimized for analytics. It is a data ware house, which means the other services are used to ingest the data and for reporting purposes the data is copied to red shift where you run your queries for your report.
* It is not paticularly good for loading txn one at a time. It is good for batch loading.
* Generally used for big data analytics and it can intergrate with most popular business intelligence tools including Tableau, Jaspersoft, ext.
* Redshift delivers ten times faster performance than other data warehouse by using machine learning, massive parallel query execution, and columnar storage on high performance disks.
* You can setup and deploy a new data warehouse in minutes and run queries across petabytes of data in your Redshift data warehouse, and exabytes of data in your data lake built on Amazon S3.
* Redshift is used to query structured, semi-structured, or unstructured data stored in S3 using SQL and joining that with your Redshift tables

**Key Benefits of Database**

* Managed: AWS database manage time consuming data base task such as back ups, patch management  and replication, allowing you to pursue higher value application development
* Flexibility: Choice of DB - SQL, MySQL Oracle and Postgre.

Amazon RDS is tightly integrated with other AWS products and services. For example, an application running on amazon EC2 will experience  low latency database access to amazon RDS DB instance in the same region.

* Secure: DB security groups SSL connections automated back ups, DB snapshots and multi AZ deployments.

|  |  |  |
| --- | --- | --- |
| **Monitoring RDS**  **RDS Limits**    **Sample Questions: Amazon Database**     1. If I launch a standby RDS instance, will it be in the same Availability Zone as my primary?   A. Only for Oracle RDS types  B. Yes  C. Only if it is configured at launch  D. No  Answer D.  Explanation: No, since the purpose of having a standby instance is to avoid an infrastructure failure (if it happens), therefore the standby instance is stored in a different availability zone, which is a physically different independent infrastructure.     1. When would I prefer Provisioned IOPS over Standard RDS storage?   A. If you have batch-oriented workloads  B. If you use production online transaction processing (OLTP) workloads.  C. If you have workloads that are not sensitive to consistent performance  D. All of the above  Answer A.  Explanation:  Provisioned IOPS deliver high IO rates but on the other hand it is expensive as well. Batch processing workloads do not require manual intervention they enable full utilization of systems, therefore a provisioned IOPS will be preferred for batch oriented workload.     1. How is Amazon RDS, DynamoDB and Redshift different?   • Amazon RDS is a database management service for relational databases,  it manages patching, upgrading, backing up of data etc. of databases for you without your intervention. RDS  is a Db management service for structured data only.  • DynamoDB, on the other hand, is a NoSQL database service, NoSQL deals with unstructured data.  • Redshift, is an entirely different service, it is a data warehouse product and is used in data analysis. OLAP     1. If I am running my DB Instance as a Multi-AZ deployment, can I use the standby DB Instance for read or write operations along with primary DB instance?   A. Yes  B. Only with MySQL based RDS  C. Only for Oracle RDS instances  D. No  Answer D.  Explanation: No, Standby DB instance cannot be used with primary DB instance in parallel, as the former is solely used for standby purposes, it cannot be used unless the primary instance goes down.     1. Your company’s branch offices are all over the world, they use a software with a multi-regional deployment on AWS, they use MySQL 5.6 for data persistence.   The task is to run an hourly batch process and read data from every region to compute cross regional reports which will be distributed to all the branches. This should be done in the shortest time possible. How will you build the DB architecture in order to meet the requirements?  A. For each regional deployment, use RDS MySQL with a master in the region and a read replica in the HQ region  B. For each regional deployment, use MySQL on EC2 with a master in the region and send hourly EBS snapshots to the HQ region  C. For each regional deployment, use RDS MySQL with a master in the region and send hourly RDS snapshots to the HQ region  D. For each regional deployment, use MySQL on EC2 with a master in the region and use S3 to copy data files hourly to the HQ region  Answer A.  Explanation: For this we will take an RDS instance as a master, because it will manage our database for us and since we have to read from every region, we’ll put a read replica of this instance in every region where the data has to be read from. Option C is not correct since putting a read replica would be more efficient than putting a snapshot, a read replica can be promoted if needed  to an independent DB instance, but with a Db snapshot it becomes mandatory to launch a separate DB Instance.     1. Can I run more than one DB instance for Amazon RDS for free?   Yes. You can run more than one Single-AZ Micro database instance, that too for free! However, any use exceeding 750 instance hours, across all Amazon RDS Single-AZ Micro DB instances, across all eligible database engines and regions, will be billed at standard Amazon RDS prices. For example: if you run two Single-AZ Micro DB instances for 400 hours each in a single month, you will accumulate 800 instance hours of usage, of which 750 hours will be free. You will be billed for the remaining 50 hours at the standard Amazon RDS price.     1. Which AWS services will you use to collect and process e-commerce data for near real-time analysis?   A. Amazon ElastiCache  B. Amazon DynamoDB  C. Amazon Redshift  D. Amazon Elastic MapReduce  Answer B,C.  Explanation: DynamoDB is a fully managed NoSQL database service. DynamoDB, therefore can be fed any type of unstructured data, which can be data from e-commerce websites as well, and later, an analysis can be done on them using Amazon Redshift. We are not using Elastic MapReduce, since a near real time analyses is needed.     1. A company is deploying a new two-tier web application in AWS. The company has limited staff and requires high availability, and the application requires complex queries and table joins. Which configuration provides the solution for the company’s requirements?   A. MySQL Installed on two Amazon EC2 Instances in a single Availability Zone  B. Amazon RDS for MySQL with Multi-AZ  C. Amazon ElastiCache  D. Amazon DynamoDB  Answer D.  Explanation: DynamoDB has the ability to scale more than RDS or any other relational database service, therefore DynamoDB would be the apt choice.     1. What happens to my backups and DB Snapshots if I delete my DB Instance?   When you delete a DB instance, you have an option of creating a final DB snapshot, if you do that you can restore your database from that snapshot. RDS retains this user-created DB snapshot along with all other manually created DB snapshots after the instance is deleted, also automated backups are deleted and only manually created DB Snapshots are retained.     1. Which of the following use cases are suitable for Amazon DynamoDB? Choose 2 answers   A. Managing web sessions.  B. Storing JSON documents.  C. Storing metadata for Amazon S3 objects.  D. Running relational joins and complex updates.  Answer C,D.  Explanation: If all your JSON data have the same fields eg [id,name,age] then it would be better to store it in a relational database, the metadata on the other hand is unstructured, also running relational joins or complex updates would work on DynamoDB as well.     1. How can I load my data to Amazon Redshift from different data sources like Amazon RDS, Amazon DynamoDB and Amazon EC2?   You can load the data in the following two ways:  • You can use the COPY command to load data in parallel directly to Amazon Redshift from Amazon EMR, Amazon DynamoDB, or any SSH-enabled host.  • AWS Data Pipeline provides a high performance, reliable, fault tolerant solution to load data from a variety of AWS data sources. You can use AWS Data Pipeline to specify the data source, desired data transformations, and then execute a pre-written import script to load your data into Amazon Redshift.     1. You are running a website on EC2 instances deployed across multiple Availability Zones with a Multi-AZ RDS MySQL Extra Large DB Instance. The site performs a high number of small reads and writes per second and relies on an eventual consistency model. After comprehensive tests you discover that there is read contention on RDS MySQL. Which are the best approaches to meet these requirements? (Choose 2 answers)   A. Deploy ElastiCache in-memory cache running in each availability zone  B. Implement sharding to distribute load to multiple RDS MySQL instances  C. Increase the RDS MySQL Instance size and Implement provisioned IOPS  D. Add an RDS MySQL read replica in each availability zone  Answer A,C.  Explanation:  Since it does a lot of read writes, provisioned IO may become expensive. But we need high performance as well, therefore the data can be cached using ElastiCache which can be used for frequently reading the data. As for RDS since read contention is happening, the instance size should be increased and provisioned IO should be introduced to increase the performance.     1. A startup is running a pilot deployment of around 100 sensors to measure street noise and air quality in urban areas for 3 months. It was noted that every month around 4GB of sensor data is generated. The company uses a load balanced auto scaled layer of EC2 instances and a RDS database with 500 GB standard storage. The pilot was a success and now they want to deploy at least  100K sensors which need to be supported by the backend. You need to store the data for at least 2 years to analyze it. Which setup of the following would you prefer?   A. Add an SQS queue to the ingestion layer to buffer writes to the RDS instance  B. Ingest data into a DynamoDB table and move old data to a Redshift cluster  C. Replace the RDS instance with a 6 node Redshift cluster with 96TB of storage  D. Keep the current architecture but upgrade RDS storage to 3TB and 10K provisioned IOPS  Answer C.  Explanation: A Redshift cluster would be preferred because it easy to scale, also the work would be done in parallel through the nodes, therefore is perfect for a bigger workload like our use case. Since each month 4 GB of data is generated, therefore in 2 year, it should be around 96 GB. And since the servers will be increased to 100K in number, 96 GB will approximately become 96TB. Hence option C is the right answer.     1. What is the difference between Scalability and Elasticity?   Scalability is the ability of a system to increase its hardware resources to handle the increase in demand. It can be done by increasing the hardware specifications or increasing the processing nodes.  Elasticity is the ability of a system to handle increase in the workload by adding additional hardware resources when the demand increases(same as scaling) but also rolling back the scaled resources, when the resources are no longer needed. This is particularly helpful in Cloud environments, where a pay per use model is followed. |  |  |

|  |  |
| --- | --- |
|  | <https://aws.amazon.com/rds/faqs/> |

**Step 1: Create an RDS DB Instance**

[**PDF**](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-ug.pdf#CHAP_Tutorials.WebServerDB.CreateDBInstance)

[**Kindle**](http://www.amazon.com/dp/B07647QNMN)

[**RSS**](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rdsupdates.rss)

Filter View

All

In this step, you create an Amazon RDS MySQL DB instance that maintains the data used by a web application.

**Important**

Before you begin this step, you must have a VPC with both public and private subnets, and corresponding security groups. If you don't have these, see [Tutorial: Create an Amazon VPC for Use with a DB Instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html). Complete the steps in [Create a VPC with Private and Public Subnets](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.VPCAndSubnets), [Create Additional Subnets](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.AdditionalSubnets), [Create a VPC Security Group for a Public Web Server](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.SecurityGroupEC2), and [Create a VPC Security Group for a Private DB Instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.SecurityGroupDB).

**Note**

A new console interface is available for database creation. Choose either the **New Console** or the **Original Console** instructions based on the console that you are using. The **New Console** instructions are open by default.

**New Console**

**To launch a MySQL DB instance**

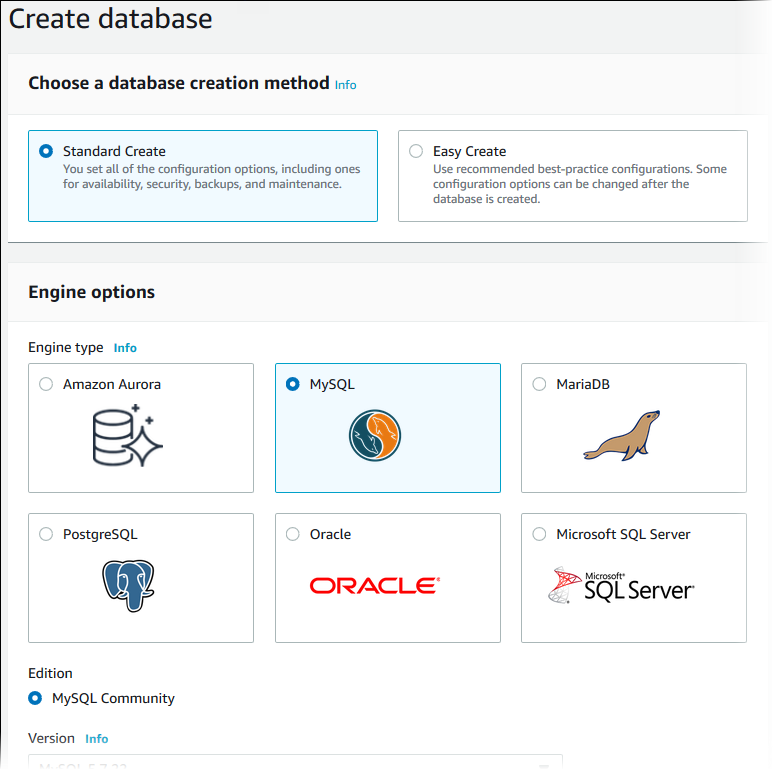
Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

In the upper-right corner of the AWS Management Console, choose the AWS Region in which you want to create the DB instance. This example uses the US West (Oregon) Region.

In the navigation pane, choose **Databases**.

Choose **Create database**.

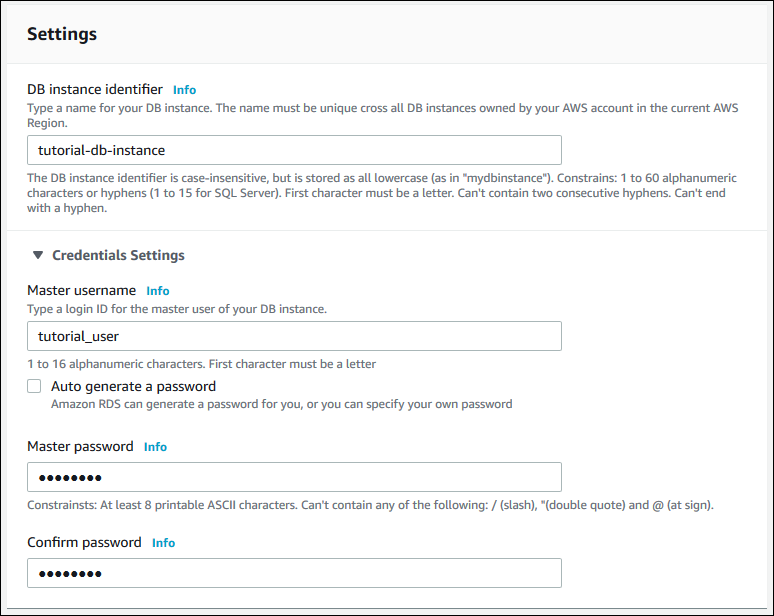
On the **Create database** page, shown following, make sure that the **Standard Create** option is chosen, and then choose **MySQL**.



In the **Templates** section, choose **Dev/Test**.

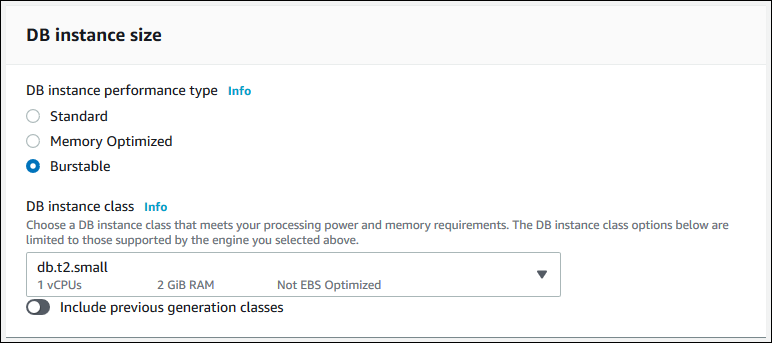
In the **Settings** section, set these values:

* **DB instance identifier** – **tutorial-db-instance**
* **Master username** – **tutorial\_user**
* **Auto generate a password** – Disable the option
* **Master password** – Choose a password.
* **Confirm password** – Retype the password.



In the **DB instance size** section, set these values:

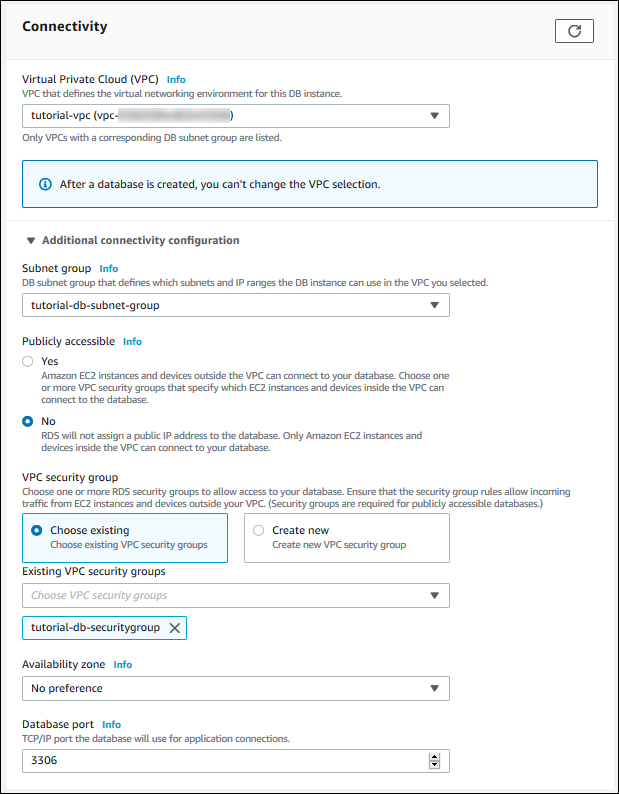
* **DB instance performance type** – **Burstable**
* **DB instance class** – **db.t2.small**



In the **Storage** and **Availability & durability** sections, use the default values.

In the **Connectivity** section, open **Additional connectivity configuration** and set these values:

* **Virtual Private Cloud (VPC)** – Choose an existing VPC with both public and private subnets, such as the tutorial-vpc (vpc-*identifier*) created in [Create a VPC with Private and Public Subnets](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.VPCAndSubnets)**Note**   
  The VPC must have subnets in different Availability Zones.
* **Subnet group** – The DB subnet group for the VPC, such as the tutorial-db-subnet-group created in [Create a DB Subnet Group](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.DBSubnetGroup)
* **Publicly accessible** – **No**
* **VPC security groups** – Choose an existing VPC security group that is configured for private access, such as the tutorial-db-securitygroup created in [Create a VPC Security Group for a Private DB Instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html#CHAP_Tutorials.WebServerDB.CreateVPC.SecurityGroupDB).   
  Remove other security groups, such as the default security group, by choosing the **X** associated with each.
* **Availability zone** – **No Preference**
* **Database port** – **3306**

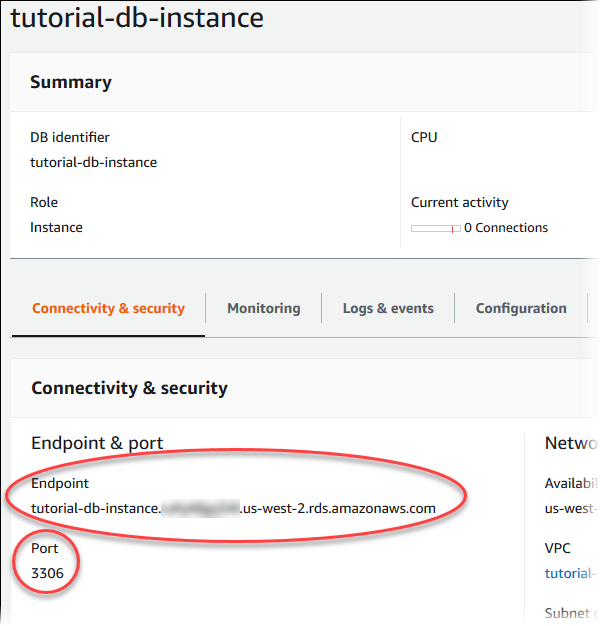


Open the **Additional configuration** section, and enter **sample** for **Initial database name**. Keep the default settings for the other options.

To create your Amazon RDS MySQL DB instance, choose **Create database**.   
Your new DB instance appears in the **Databases** list with the status **Creating**.

Wait for the **Status** of your new DB instance to show as **Available**. Then choose the DB instance name to show its details.

In the **Connectivity & security** section, view the **Endpoint** and **Port** of the DB instance.



Note the endpoint and port for your DB instance. You use this information to connect your web server to your RDS DB instance.

To make sure your RDS MySQL DB instance is as secure as possible, verify that sources outside of the VPC can't connect to your RDS MySQL DB instance.

**Understanding Federation**

**The need for LDAP and AD**

Let's assume there are 1000 users within an organization; and the organization is using 3 services:

1. AWS (Infrastructure)
2. Jenkins (CI/CD)
3. HR Activator (Payroll)

You have been assigned a role to give users access to all 3 services.

Rather than going to add users manually in each of this service, we can have a central directory (such as LDAP-Light Weight Directory Protocol or AD) where all the users will be stored. So the administrator only have to store the users in LDAP. Once the users are stored in LDAP and depending on the federated settings that you have done, users will be able to login to Jenkins, AWS and HR Activator. This removes the burden of creating users in each of this individual systems.

Just create a user in LDAP and establish a trust relationship between LDAP and all three services and users will be able to seamlessly connect to all three services.

There are various solutions available that can store users centrally:

1. Microsoft Active Directory
2. Red Hat Identify management or freeIPA

Note that Federation allows external identities (federated users) to have secure access to your AWS account without having to create any IAM users.

These external identities can come from:

1. Corporate identity provider (AD, IPA)
2. Web Identity Provider (Facebook, Google, Amazon, Cognito or OpenID)

**Web Identity Federation:**

It let's you give users access to AWS resources after they have successfully authenticated with web base identity providers like Amazon, Facebook, Google. After successful authentication, the user receives an authentication code from the web provider, which they can trade for temporary AWS security credentials.

**Amazon Cognito:**Amazon cognito provides web identity federation with the following features:

* Sign up and sign in to your apps
* Access for guest users
* Acts as an identity broker between your application and web ID provider, so you don't need to write any additional code
* Synchronizes user data for multiple devices
* Recommended for all mobile applications AWS services

How do I minimize the impact of a DDoS attack on my application?

A Distributed Denial of Service (DDoS) attack is a malicious attempt to make a targeted system, such as a website or application, unavailable to end users. To achieve this, attackers use a variety of techniques that consume network or other resources, interrupting access for legitimate end users.

AWS provides flexible infrastructure and services that help customers implement strong DDoS mitigations and create highly available application architectures that follow AWS best practices for DDOS resiliency. These include services such as [Amazon Route 53](https://aws.amazon.com/route53/), [Amazon CloudFront](https://aws.amazon.com/cloudfront/), [Elastic Load Balancing](https://aws.amazon.com/elasticloadbalancing/), and [AWS WAF](https://aws.amazon.com/waf/) to control and absorb traffic, and deflect unwanted requests. These services integrate with [AWS Shield](https://aws.amazon.com/shield/), a managed DDoS protection service that provides always-on detection and automatic inline mitigations to safeguard web applications running on AWS. This document describes common DDoS attack types and provides AWS customers with best practices and strategies for protecting applications from a DDoS attack.

**General Best Practices**

At its core, DDoS protection and mitigation involves establishing a secure perimeter around your infrastructure and allowing or denying certain traffic based on filters or rules. AWS customers can take advantage of the flexible nature of the cloud and adapt their infrastructure defensively in the event of an attack. When thinking about DDoS security, keep the following best practices in mind:

* **Be ready to scale.** Building infrastructure for scale is fundamental to a well-architected system, however it is also an effective DDoS mitigation technique. Scaling to meet the additional traffic volumes, whether valid or from a DDoS attack, will increase your application’s ability to keep running. AWS Shield Advanced includes some cost protection against Amazon EC2, Elastic Load Balancing, Amazon CloudFront, and Amazon Route 53 usage spikes that could result from scaling during a DDoS attack.
* **Minimize the attack surface area and safeguard exposed resources.** In other words: decouple your infrastructure. For example, when running public websites, separate the application from the database and, if possible, the media and static content as well. Decoupled applications limit internet access to critical system components, protecting them from an attack and enabling teams to focus DDoS mitigation efforts on resources that are publicly accessible.
* **Know what is normal; alert on what is not**. In order to get the best help from AWS Support, it is critical that you identify traffic correctly. For example, if a website gets media attention and is suddenly overwhelmed with traffic, blocking that traffic can cause more harm than good. Constant logging and monitoring of infrastructure can help you quickly identify a legitimate attack and engage AWS. For extended visibility into attacks on your Amazon EC2, CloudFront, and Elastic Load Balancing resources, use AWS Shield Advanced to gain exclusive access to advanced, real-time Amazon CloudWatch metrics and reports.
* **Architect for resilience.** Understand the [AWS shared responsibility model](https://aws.amazon.com/compliance/shared-responsibility-model/) and take advantage of services and features that can help protect your application. AWS Shield Standard, when used with Amazon CloudFront and Amazon Route 53, provides comprehensive protection against all known infrastructure layer (layer 3 and layer 4) attacks. For additional protection against application layer (layer 7) attacks, use AWS WAF to apply custom mitigation rules.
* Create a Plan for Attacks:
  + Check from which country increased traffic is coming from
  + Check whther the source IP is the same
  + Nature of attacks (SYN Floods, Application Levels)
  + Can it be blocked with NACL or Security Groups if it's server level atacks
  + WAF for application level DDOS

**AWS Services for DDoS Attack Mitigation**

AWS offers globally distributed, high network bandwidth and resilient services that, when used in conjunction with application-specific strategies, are key to mitigating DDoS attacks.

**What Is AWS Organizations?**

AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an *organization* that you create and centrally manage. AWS Organizations includes account management and consolidated billing capabilities that enable you to better meet the budgetary, security, and compliance needs of your business. As an administrator of an organization, you can create accounts in your organization and invite existing accounts to join the organization.

**AWS Organizations Features**

AWS Organizations offers the following features:

**Centralized management of all of your AWS accounts**

You can combine your existing accounts into an organization that enables you to manage the accounts centrally. You can create accounts that automatically are a part of your organization, and you can invite other accounts to join your organization. You also can attach policies that affect some or all of your accounts.

**Consolidated billing for all member accounts**

Consolidated billing is a feature of AWS Organizations. You can use the master account of your organization to consolidate and pay for all member accounts.

**Hierarchical grouping of your accounts to meet your budgetary, security, or compliance needs**

You can group your accounts into organizational units (OUs) and attach different access policies to each OU. For example, if you have accounts that must access only the AWS services that meet certain regulatory requirements, you can put those accounts into one OU. You then can attach a policy to that OU that blocks access to services that do not meet those regulatory requirements. You can nest OUs within other OUs to a depth of five levels, providing flexibility in how you structure your account groups.

**Control over the AWS services and API actions that each account can access**

As an administrator of the master account of an organization, you can use service control policies (SCPs) to specify the maximum permissions for member accounts in the organization. In SCPs, you can restrict which AWS services, resources, and individual API actions the users and roles in each member account can access. You can also define conditions for when to restrict access to AWS services, resources, and API actions. These restrictions even override the administrators of member accounts in the organization. When AWS Organizations blocks access to a service, resource, or API action for a member account, a user or role in that account can't access it. This block remains in effect even if an administrator of a member account explicitly grants such permissions in an IAM policy.

The following diagram shows a basic organization that consists of seven accounts that are organized into four organizational units (OUs) under the root. The organization also has several policies that are attached to some of the OUs or directly to accounts. For a description of each of these items, refer to the definitions in this topic.



**Organization**

An entity that you create to consolidate your AWS accounts.  An organization has one master account along with zero or more member accounts. You can organize the accounts in a hierarchical, tree-like structure with a root at the top and organizational nested under the root. Each account can be directly in the root, or placed in one of the OUs in the hierarchy.

**Root**

The parent container for all the accounts for your organization. If you apply a policy to the root, it applies to all Ous and accounts in the organization.

**Note**

Currently, you can have only one root. AWS Organizations automatically creates it for you when you create an organization.

**Organization unit (OU)**

A container for accounts within a root. An OU also can contain other OUs, enabling you to create a hierarchy that resembles an upside-down tree, with a root at the top and branches of OUs that reach down, ending in accounts that are the leaves of the tree. When you attach a policy to one of the nodes in the hierarchy, it flows down and affects all the branches (OUs) and leaves (accounts) beneath it. An OU can have exactly one parent, and currently each account can be a member of exactly one OU.

**Account**

A standard AWS account that contains your AWS resources. You can attach a policy to an account to apply controls to only that one account.

There are two types of accounts in an organization: a single account that is designated as the master account, and member accounts.

* The **master account** is the account that creates the organization. From the organization's master account, you can do the following:
* Create accounts in the organization
* Invite other existing accounts to the organization
* Remove accounts from the organization
* Manage invitations
* Apply policies to entities (roots, OUs, or accounts) within the organization   
  The master account has the responsibilities of a *payer account* and is responsible for paying all charges that are accrued by the member accounts. You can't change an organization's master account.
* The rest of the accounts that belong to an organization are called **member accounts**. An account can be a member of only one organization at a time.

**Invitation**

The process of asking another account to join your organization. An invitation can be issued only by the organization's master account. The invitation is extended to either the account ID or the email address that is associated with the invited account. After the invited account accepts an invitation, it becomes a member account in the organization.

**Service control policy (SCP)**

A policy that specifies the services and actions that users and roles can use in the accounts that the SCP affects. SCPs are similar to IAM permissions policies except that they don't grant any permissions. Instead, SCPs specify the maximum permissions for an organization, organizational unit (OU), or account. When you attach an SCP to your organization root or an OU, the SCP limits permissions for entities in member accounts.

**To enable SCPs (console)**

1. Sign in to the Organizations console at <https://console.aws.amazon.com/organizations/>. You must sign in as an IAM user, assume an IAM role, or sign in as the root user ([not recommended](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#lock-away-credentials)) in the organization's master account.
2. On the **Organize accounts** tab, choose **Root** in the left navigation pane.
3. In the details pane on the right side, next to **Service control policies**, choose **Enable**.
4. Under **What is a service control policy?**, choose **Enable service control polices**.

<https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_example-scps.html#example_scp_3>

**To attach an SCP to a root, OU, or account (console)**

1. Sign in to the Organizations console at <https://console.aws.amazon.com/organizations/>. You must sign in as an IAM user, assume an IAM role, or sign in as the root user ([not recommended](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#lock-away-credentials)) in the organization's master account.
2. On the **Organize accounts** tab, [navigate to](https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_ous.html#navigate_tree) and select the check box for the root, OU, or account you want to attach the SCP to.
3. In the **Details** pane on the right, expand the **Service control policies** section to see the list of the currently attached SCPs.
4. On the list of available SCPs, find the one that you want and choose **Attach**. The list of attached SCPs is updated with the new addition. The SCP goes into effect immediately. For example, an SCP immediately affects the permissions of IAM users and roles in the attached account or all accounts under the attached root or OU.

**AWS CloudTrail**

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting. In addition, you can use CloudTrail to detect unusual activity in your AWS accounts. These capabilities help simplify operational analysis and troubleshooting.

CloudTrail is enabled by default for your AWS account. You can use **Event history** in the CloudTrail console to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure. This includes activity made through the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

For an ongoing record of events in your AWS account, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs.

If you have created an organization in AWS Organizations, you can create a trail that will log all events for all AWS accounts in that organization.

**Creating a Trail For Your AWS Account**

When you create a trail, you enable ongoing delivery of events as log files to an Amazon S3 bucket that you specify. Creating a trail has many benefits, including:

* A record of events that extends past 90 days.
* The option to automatically monitor and alarm on specified events by sending log events to Amazon CloudWatch Logs.
* The option to query logs and analyze AWS service activity with Amazon Athena.

You can configure the following settings when you create or update a trail with the CloudTrail console or the AWS Command Line Interface (AWS CLI). Both methods follow the same steps:

1. Create a trail. By default, when you create a trail in a region in the CloudTrail console, the trail applies to all regions.
2. Create an Amazon S3 bucket or specify an existing bucket where you want the log files delivered. By default, log files from all regions in your account are delivered to the bucket that you specify.
3. Configure your trail to log read-only, write-only, or all management events, all Insights events, and all or a subset of data events. By default, trails log all management events and no data or Insights events.
4. Create an Amazon SNS topic to receive notifications when log files are delivered. Delivery notifications from all regions are sent to the topic that you specify.
5. Configure CloudWatch Logs to receive your logs from CloudTrail so that you can monitor for specific log events.
6. Change the encryption method for your log files from server-side encryption with Amazon S3-managed encryption keys (SSE-S3) to server-side encryption with AWS KMS–managed keys (SSE-KMS).
7. Turn on integrity validation for log files. This enables the delivery of digest files that you can use to validate the integrity of log files after CloudTrail has delivered them.
8. Add tags (custom key-value pairs) to your trail.

**To create a CloudTrail trail with the AWS Management Console**

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose the AWS Region where you want the trail to be created.
3. Choose **Get Started Now**.   
   **Tip**   
   If you do not see **Get Started Now**, choose **Trails**, and then choose **Create trail**.
4. On the **Create Trail** page, for **Trail name**, type a name for your trail.
5. For **Apply trail to all regions**, choose **Yes** to receive log files from all Regions. This is the default and recommended setting. If you choose **No**, the trail logs files only from the Region in which you create the trail.
6. For **Management events**, do the following.
7. For **Read/Write events**, choose if you want your trail to log **All**, **Read-only**, **Write-only**, or **None**, and then choose **Save**. By default, trails log all management events.
8. For **Log AWS KMS events**, choose **Yes** to log AWS Key Management Service (AWS KMS) events in your trail. Choose **No** to filter AWS KMS events out of your trail. The default setting is **Yes**.
9. In **Insights events**, for **Log Insights events**, choose **Yes** if you want your trail to log Insights events. By default, trails don't log Insights events.    
   Insights events are delivered to a different folder named /CloudTrail-Insightof the same S3 bucket that is specified in the **Storage location** area of the trail details page. CloudTrail creates the new prefix for you. For example, if your current destination S3 bucket is named S3bucketName/AWSLogs/CloudTrail/, the S3 bucket name with a new prefix is named S3bucketName/AWSLogs/CloudTrail-Insight/.
10. For **Data events**, you can specify logging data events for Amazon S3 buckets, for AWS Lambda functions, or both. By default, trails don't log data events. Additional charges apply for logging data events.    
    You can select the option to log all S3 buckets and Lambda functions, or you can specify individual buckets or functions.   
    For Amazon S3 buckets:

* Choose the **S3** tab.
* To specify a bucket, choose **Add S3 bucket**. Type the S3 bucket name and prefix (optional) for which you want to log data events. For each bucket, specify whether you want to log **Read** events, such as GetObject, **Write** events, such as PutObject, or both.
* To log data events for all S3 buckets in your AWS account, select **Select all S3 buckets in your account**. Then choose whether you want to log **Read** events, such as GetObject, **Write** events, such as PutObject, or both. This setting takes precedence over individual settings you configure for individual buckets. For example, if you specify logging **Read** events for all S3 buckets, and then choose to add a specific bucket for data event logging, **Read** is already selected for the bucket you added. You cannot clear the selection. You can only configure the option for **Write**.   
  **Note**   
  For Lambda functions:
* Choose the **Lambda** tab.
* To specify logging individual functions, select them from the list.   
  To log data events for all Lambda functions in your AWS account, select **Log all current and future functions**. This setting takes precedence over individual settings you configure for individual functions. All functions are logged, even if all functions are not displayed.   
  Logging data events for all functions also enables logging of data event activity performed by any user or role in your AWS account, even if that activity is performed on a function that belongs to another AWS account.

1. For **Storage location**, for **Create a new S3 bucket**, choose **Yes** to create a bucket. When you create a bucket, CloudTrail creates and applies the required bucket policies.   
   **Note**   
   If you chose **No**, choose an existing S3 bucket. The bucket policy must grant CloudTrail permission to write to it. For **S3 bucket**, type a name for the bucket you want to designate for log file storage.
2. For **Tags**, add one or more custom tags (key-value pairs) to your trail. Tags can help you identify both your CloudTrail trails and the Amazon S3 buckets that contain CloudTrail log files. You can then use resource groups for your CloudTrail resources.
3. The new trail appears on the **Trails** page. The **Trails** page shows the trails in your account from all Regions. In about 15 minutes, CloudTrail publishes log files that show the AWS API calls made in your account. You can see the log files in the S3 bucket that you specified.

**What is DNS?**

Domain Name System (DNS):

Worldwide network of servers that maintains domain names to IP Addresses

DNS is used to convert human friendly domain names such as aws.amazon.com into an internet protocol address such [http://54.239.31.69](http://54.239.31.69/). Essentially mapping friendly domain names to resources that can process the request.

IP addresses are used by computers to identify each other on the network. Remember the analogy of a phone book.

If you look at common domain names such google.com, bbc.co.uk, etc., you notice a string of characters separated by dots (periods). The last word in a domain name represents the “top level domain”. The second word in a domain name is known as second level domain name (this is optional and depends on the domain name)

.com

.edu

.gov

.co.uk

.com.au

**Top Level Domains:**

Last part of a domain name (.com, .org, .cloud).

These top-level domain names are controlled by the Internet Assigned Numbers Authority (IANA) in a root zone database which is essentially a database of all available top level domains. You can view this database by visiting: <http://www.iana.org/domain/root/db>

You can trace domain by going to <http://dimpledns.com/lookup-dg.aspx>

Because all of the names in a given domain name have to be unique, there needs to be a way to organize this to avoid duplicated domains. This is where domain registrars comes in. A registrar is an authority that can assign domain names directly under one or more top level domains. These domain re registered with InterNIC, a service of ICANN which enforces uniqueness of domain names across the internet.  Each domain name becomes registered in a central database known as WhoIS database.

Popular domain registrars include GoDaddy.com

**Geographic Top-Level Domain:** Domains associated with geographic areas. (.uk, .fr, .io, .in)

**Start of Authority Record:**Start Of (a Zone) Authority Record. Specifies authoritative information about a DNS zone including Primary Name Server, Email of the domain administrator, domain serial number and several timers related to refreshing the zone

**NS Records:** stands for Name Server records: Used by top level domain servers to direct traffic to the content DNS server. Name Servers (NS) are Servers in DNS that respond to DNS queries

**A Records:** The A stands for Address. The A record is used by a computer to translate the name of the domain to an IP address. For example [http://aws.amazon.to](http://aws.amazon.to/) [http://54.239.31.69](http://54.239.31.69/).

TTL: The length of time that DNS record is cached on either the resolving server or the user’s own local PC is equal to the value of the “Time To Live” TTL in seconds. The lower the time to live the faster changes to DNS records take to propagate throughout the internet.

CNAMES: A Canonical Name (CName) can be used to resolve one domain to another.

Alias Records: Alias records are used to map resource sets in your hosted zone to ELB, CloudFront distributions, S3 buckets that are configured as websites. Alias records work like CName record in that you can map one DNS name to another target DNS name (elb12345elb.amazonaws.com)

Note:

ELBs do not have pre-defined IPV4 addresses. You resolve to them using a DNS name

**ROUTE 53**

**What is Route 53:**

* Route 53 is a Domain management service provided by AWS.  This means that if customers put a url or domain name in their web browser and the hosted zone  is set up pointing to amazon's name  servers,  then route 53 response to the domain name with an IP address so the web browser can connect to a server.  Key features of R53 include:
  + **Domain Registration**: register domain names such as juzzyk.com
  + **DNS service**:
    - Translate friendly domain names such as [www.juzzyk.com](http://www.juzzyk.com/) into ip addresses likes 192.0.2.1
    - R53 response to DNS queries using a global network of authoritative DNS servers which reduces latency.
  + **Health Checking**:
    - R53 sends automated requests over the internet to your application to verify that its reachable, available and functional.

* The main purpose of a DNS server is to provide an IP address of the web server.
* The hosted zone set up in R53 determines the IP address that will be returned to the web browser. It could be the Ip of the ELB, the EC2 instance Ip an s3 bucket configured as a static website, or a cloud front distribution.

 Note that the ELB, S3 bucket and cloud front do not have a single IP address associated with them because there are a number of Ip addresses for those end points. Route 53 has a modification to take care of this. You can create an alias record which allows you to use an alias instead of an IP address for your zone Apex record.

* Route 53 can mange external DNS for domain routing  to the proper AWS resource such as cloud front distribution, ELB, EC2 instance or RDS server.
* R53 is commonly used with ELB to direct traffic from the domain to the ELB ( and thus have traffic evenly distributed among servers running your application)
* It can also be used to manage internal DNS for custom internal host names within a VPC as long as the VPC is configured for it.
* Latency, GEO, basic and fail over routing policies allow for region to region fault tolerant architecture design.
* You can easily configure for fail over to S3 ( if website bucket hosting is enabled).

**Route 53 Core Capabilities**

• Domain Registration & Renewals •

Domain Name Service to map domain names to resource

• Health Check – Route traffic to healthy end points & independently monitor endpoints

 • Private DNS – Visible only in attached VPCs

 • Global Traffic Management – Route traffic to your globally distributed end points using variety of routing criteria

**Routing Policies:**

1. **Simple Routing** If you specify multiple values in a record, Route 53 returns all values to the user in random order. You can only create a single record set for a domain using the simple routing policy.

Simple routing lets you configure standard DNS records, with no special Route 53 routing such as weighted or latency. With simple routing, you typically route traffic to a single resource, for example, to a web server for your website.

If you choose the simple routing policy in the Route 53 console, you can't create multiple records that have the same name and type, but you can specify multiple values in the same record, such as multiple IP addresses. (If you choose the simple routing policy for an alias record, you can specify only one AWS resource or one record in the current hosted zone.) If you specify multiple values in a record, Route 53 returns all values to the recursive resolver in random order, and the resolver returns the values to the client (such as a web browser) that submitted the DNS query. The client then chooses a value and resubmits the query.

1. **Weighted Routing:** Used when you have multiple resources performing similar function and you want to route traffic to resources in proportions that you specify. For example: Several web servers serving content, A/B testing. The weighted routing policy lets you split your traffic base on different weights assigned. You set 20% of traffic to go to US-east 1 and 80% to go to us-west-1.
2. **Latency Routing:** Used when you have deployed your application across multiple regions and want to route customers to resources that offer best possible latency (lowest network latency for end uers); ie *which region will give them the fastest response time.* In this case, when AWS Route 53 receives for a query for your site, it selects the latency resource record set for the region that gives the user the lowest latency. For example, a Cameroon user wants to come to our website. EU-WEST-2 offers 54ms latency and US-east-1 offers 300ms latency, in this case route 53 will send the traffic to eu-west-2 because it has a much lower latency.
3. **Failover Routing** Active-Passive failover support.  All traffic is routed to Primary endpoint (known as Active). If primary is down, then all traffic is send to Second endpoint (known as Passive).
4. **Geo location Routing** Used when you want to route traffic to resources in the same geography as your users.  Can be used for compliance requirements.  You can support a default record set to handle requests where you don’t have resources.  Otherwise, Route 53 will return a “No Answer” response
5. **Multivalue Answer Routing:** Allows you to have multiple record sets and can have an individual health check for each one so that if one of those goes down, you can have that automatically removed from the dns record.

Registering Domain:

1. **Start by registering your domain.**
   1. Go tp R53 & click get started now
   2. Enter your new unique domain name & check to make sure it is available.
   3. Enter the detail & done. It takes about 15 minutes to be completed & you should see it under registered domain.
2. **Next is to set up hosted zones:**

A hosted zone is a container that holds information about how you want to route traffic on the internet for a specific domain name.

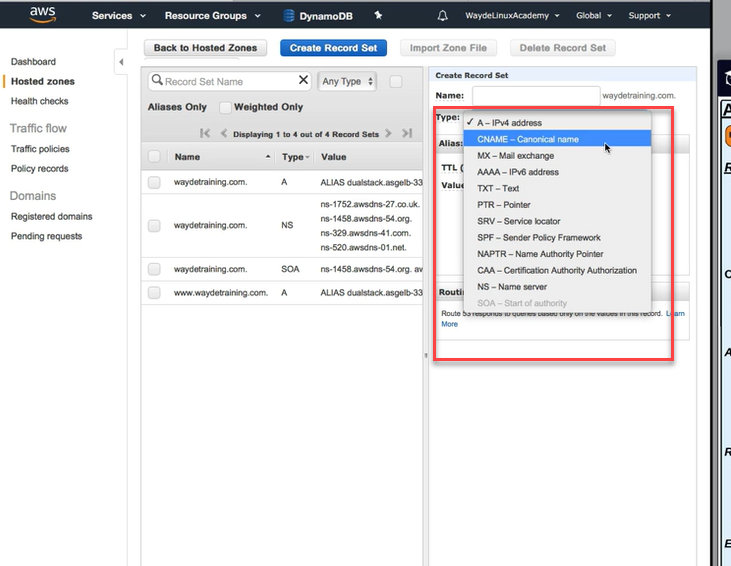
1. Go to hosted zone and choose the domain name you created and  start creating record sets for your domain. By default AWS route 53 has already created  a hosted Zone with some records for the name servers. These servers are amazon's servers which are going to respond to the DNS request that comes in.

Once your domain name and hosted zone is created, you can query that Ip address in your browser and you should be able to see it, http:// ip address

1. If you would like your domain to respond with an Ip address of the ELB, you can set that up by clicking on record set, - create record set, & enter the details.
   1. Select Alias,
   2. copy and paste the DNS name of the load balancer you created
   3. Set the routing policy (**simple, weighted, Latency, fail over or Geo location**)
   4. Select health check if you have more than one endpoint and you want R53 to evaluate the health of the primary end point.
   5. Click create. You can create as many record sets as you want depending on the routing policy. Note that the simple routing policy only has one record set.

R53 is global and it takes almost a minute to update.

There are different types of record types used for different purposes (see below):



* A records for IPV4
* CNAME - When you wan to have a domain name point to another domain name. Such as a sub domain that should point to your domain
* MX rrecords for sending emails
* AAA is for IPV6

**How to set up a Failover Routing policy for R53:**

We use a fail over policy when we want to send users to another place when our website is down. This could be an S3 bucket for DNS fail over:

* By using a fail over routing policy in a Route 53 DNS record set, an s3 bucket can be used as the fail over end point.
* This can provide an extremely reliable back up solution if your primary end point fails.
* And even though S3 should only be used for static web hosting, it gives you the opportunity to provide your users with some type of information until the primary end point is working again.
* An S3 bucket can also be used as a primary endpoint, if you just want to host a simple static site.

For a DNS record to use an S3 bucket as the endpoint, the bucket name MUST be the same as the domain name.

Steps

* 1. Step up an S3 bucket with the same name as your domain name. The content of the s3 bucket should be a simple static html website.
  2. Go to my hosted zone in R53 and select the record set you want to use as the fail over.
  3. Change the routing policy to "fail over" and mark the current record set as the **primary endpoint**
  4. Check the box for "Evaluate health target"
  5. Create another record set for this same domain name, click S3 bucket as the Alias target, set the routing policy to fail over and mark this record set as**Secondary**. This means  it will not see any request until the primary is down.  No need to evaluate the health of the secondary end point.

You can test this by disabling the primary end point and see if you will be redirected to the secondary.

**Choosing a Routing Policy**

When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

* **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
* **Failover routing policy** – Use when you want to configure active-passive failover.
* **Geolocation routing policy** – Use when you want to route traffic based on the location of your users.
* **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
* **Latency routing policy** – Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency.
* **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
* **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

**DNS: Focus on Route53**

**What is DNS?**

Domain Name System (DNS):

Worldwide network of servers that maintains domain names to IP Addresses

DNS is used to convert human friendly domain names such as aws.amazon.com into an internet protocol address such [http://54.239.31.69](http://54.239.31.69/). Essentially mapping friendly domain names to resources that can process the request.

IP addresses are used by computers to identify each other on the network. Remember the analogy of a phone book.

If you look at common domain names such google.com, bbc.co.uk, etc., you notice a string of characters separated by dots (periods). The last word in a domain name represents the “top level domain”. The second word in a domain name is known as second level domain name (this is optional and depends on the domain name)

.com

.edu

.gov

.co.uk

.com.au

**Top Level Domains:**

Last part of a domain name (.com, .org, .cloud).

* These top-level domain names are controlled by the**Internet Assigned Numbers Authority** (IANA) in a root zone database which is essentially a database of all available top level domains. You can view this database by visiting: <http://www.iana.org/domain/root/db>
* You can trace domain by going to <http://dimpledns.com/lookup-dg.asp> Because all of the names in a given domain name have to be unique, there needs to be a way to organize this to avoid duplicated domains. This is where domain registrars comes in. A registrar is an authority that can assign domain names directly under one or more top level domains. These domain are registered with InterNIC, a service of ICANN which enforces uniqueness of domain names across the internet.  Each domain name becomes registered in a central database known as WhoIS database.

**Popular domain registrars include GoDaddy.com**

**Geographic Top-Level Domain**:

Domains associated with geographic areas. (.uk, .fr, .io, .in)

**Start of Authority Record:** Start Of (a Zone) Authority Record Specifies authoritative information about a DNS zone including Primary Name Server, Email of the domain administrator, domain serial number and several timers related to refreshing the zone

**NS Records:** stands for Name Server records: Used by top level domain servers to direct traffic to the content DNS server. Name Servers (NS) are Servers in DNS that respond to DNS queries

**A Records:** The A stands for Address. The A record is used by a computer to translate the name of the domain to an IP address. For example [http://aws.amazon](http://aws.amazon/). to http://54.239.31.69.

**TTL:** The length of time that DNS record is cached on either the resolving server or the user’s own local PC is equal to the value of the**“Time To Live” TTL in seconds.** The lower the time to live the faster changes to DNS records take to propagate throughout the internet.

**CNAMES:** A Canonical Name (CName) can be used to resolve one domain to another.

**Alias Records:** Alias records are used to map resource sets in your hosted zone to ELB, CloudFront distributions, S3 buckets that are configured as websites. Alias records work like CName record in that you can map one DNS name to another target DNS name (elb12345elb.amazonaws.com)

**Note:**

ELBs do not have pre-defined IPV4 addresses. You resolve to them using a DNS name

Route 53 Core Capabilities

• Domain Registration & Renewals •

Domain Name Service to map domain names to resource

• Health Check – Route traffic to healthy end points & independently monitor endpoints

 • Private DNS – Visible only in attached VPCs

 • Global Traffic Management – Route traffic to your globally distributed end points using variety of routing criteria

**Simple Routing** If you specify multiple values in a record, Route 53 returns all values to the user in random order.

**Weighted Routing** Used when you have multiple resources performing similar function and you want to route traffic to resources in proportions that you specify.  For example: Several web servers serving content, A/B testing. WR lets you split your traffic base on different weights assigned. You set 20% of traffic to go to US-east 1 and 80% to go to eu-west-1

**Latency Routing** Used when you have deployed your application across multiple regions and want to route customers to resources that offer best possible latency (lowest network latency for end uers); ie which region will give them the fastest response time. In this case, when AWS Route 53 receives for a query for your site, it selects the latency resource record set for the region that gives the user the lowest latency. For example, a Cameroon user wants to come to our website. EU-WEST-2 offers 54ms latency and US-east-1 offers 300ms latency, in this case route 53 will send the traffic to eu-west-2 because it has a much lower latency.

**Failover Routing** Active-Passive failover support.  All traffic is routed to Primary endpoint (known as Active). If primary is down, then all traffic is send to Second endpoint (known as Passive).

Geolocation Routing Used when you want to route traffic to resources in the same geography as your users.  Can be used for compliance requirements.  You can support a default record set to handle requests where you don’t have resources.  Otherwise, Route 53 will return a “No Answer” response

**Multivalue Answer Routing**: Allows you to have multiple record sets and can have an individual health check for each one so that if one of those goes down, you can have that automatically removed from the dns record.

Demos:

A. Register a domain

B. Simple Routing Policy

C. Weight Routing Policy

D. latency based routing policy

**Running Commands on Your Linux Instance at Launch**

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives.

In the following examples, the commands from the [Install a LAMP Web Server on Amazon Linux 2](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html) are converted to a shell script and a set of cloud-init directives that executes when the instance launches. In each example, the following tasks are executed by the user data:

* The distribution software packages are updated.
* The necessary web server, php, and mariadb packages are installed.
* The httpd service is started and turned on via **systemctl**.
* The ec2-user is added to the apache group.
* The appropriate ownership and file permissions are set for the web directory and the files contained within it.
* A simple web page is created to test the web server and PHP engine.

**Prerequisites**

The following examples assume that your instance has a public DNS name that is reachable from the Internet. For more information, see

Step1: Launch an instance. You must also configure your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections.

Also, these instructions are intended for use with Amazon Linux 2, and the commands and directives may not work for other Linux distributions.

In the example script below, the script creates and configures our web server.

**#!/bin/bash**

**yum update -y**

**amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2**

**yum install -y httpd mariadb-server**

**systemctl start httpd**

**systemctl enable httpd**

**usermod -a -G apache ec2-user**

**chown -R ec2-user:apache /var/www**

**chmod 2775 /var/www**

**find /var/www -type d -exec chmod 2775 {} \;**

**find /var/www -type f -exec chmod 0664 {} \;**

**echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php**

Allow enough time for the instance to launch and execute the commands in your script, and then check to see that your script has completed the tasks that you intended.

For our example, in a web browser, enter the URL of the PHP test file the script created. This URL is the public DNS address of your instance followed by a forward slash and the file name.

[http://*my.public.dns.amazonaws.com*/phpinfo.php](http://my.public.dns.amazonaws.com/phpinfo.php)

You should see the PHP information page. If you are unable to see the PHP information page, check that the security group you are using contains a rule to allow HTTP (port 80) traffic.

#!/bin/bash   
yum update -y   
yum install httpd -y   
cd /var/www/html   
echo "JJTECH students are now studying Route53, which is one of several self healing based architectures in AWS. Other include AS, ELB, RDS & CW. Current Batch include Armelle, Ayuk, Chrintine, Clifford, Edmond, Elie, Elvis,Estevia, Eva, Kelly, lister, Mbandi, Mike, Ransom, Paul, Ntap, Roland, Ruddy, Samuel, Senge,Suzzy,Terence, Yvonne, Yves....." > index.html   
service httpd start   
chkconfig httpd on

**IAM Roles for Amazon EC2**

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting your credentials from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalf, such as Spot Instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We designed IAM roles so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles as follows:

1. Create an IAM role.
2. Define which accounts or AWS services can assume the role.
3. Define which API actions and resources the application can use after assuming the role.
4. Specify the role when you launch your instance, or attach the role to an existing instance.
5. Have the application retrieve a set of temporary credentials and use them.

For example, you can use IAM roles to grant permissions to applications running on your instances that need to use a bucket in Amazon S3. You can specify permissions for IAM roles by creating a policy in JSON format. These are similar to the policies that you create for IAM users. If you change a role, the change is propagated to all instances.

You cannot attach multiple IAM roles to a single instance, but you can attach a single IAM role to multiple instances.

**Creating an IAM Role**

You must create an IAM role before you can launch an instance with that role or attach it to an instance.

**To create an IAM role using the IAM console**

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**, **Create role**.
3. On the **Select role type** page, choose **EC2** and the **EC2** use case. Choose **Next: Permissions**.
4. On the **Attach permissions policy** page, select an AWS managed policy that grants your instances access to the resources that they need.
5. On the **Review** page, enter a name for the role and choose **Create role**.

**Elastic File Storage (EFS)**

Many applications require access to shared file storage

These shared files are generally stored in NAS (network attached storage)

However, there are certain limitations to this storage including high availability, security and scalability (it takes quite amount time to grow your storage from say 500 MB to 1 GB)  limitations.

All these challenges can be solved with the use of EFS (Elastic File System)

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, elastic file system for Linux-based workloads for use with AWS Cloud services and on-premises resources.

It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, so your applications have the storage they need – when they need it.

It is designed to provide massively parallel shared access to thousands of Amazon EC2 instances, enabling your applications to achieve high levels of aggregate throughput and IOPS with consistent low latencies.

NAT Gateways

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

NAT Gateway Basics

To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside. You must also specify an Elastic IP address to associate with the NAT gateway when you create it. The Elastic IP address cannot be changed once you associate it with the NAT Gateway. After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point Internet-bound traffic to the NAT gateway. This enables instances in your private subnets to communicate with the internet.

Each NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone. You have a limit on the number of NAT gateways you can create in an Availability Zone.

Note

If you have resources in multiple Availability Zones and they share one NAT gateway, in the event that the NAT gateway’s Availability Zone is down, resources in the other Availability Zones lose internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

The following diagram illustrates the architecture of a VPC with a NAT gateway. The main route table sends internet traffic from the instances in the private subnet to the NAT gateway. The NAT gateway sends the traffic to the internet gateway using the NAT gateway’s Elastic IP address as the source IP address.

A VPC with public and private subnets and a NAT gateway

Note: An Elastic IP address is a static, public IPv4 address designed for dynamic cloud computing. You can associate an Elastic IP address with any instance or network interface for any VPC in your account. With an Elastic IP address, you can mask the failure of an instance by rapidly remapping the address to another instance in your VPC.

Testing the Internet Connection

The following example demonstrates how to test if your instance in a private subnet can connect to the internet.

Launch an instance in your public subnet (you use this as a bastion host). In the launch wizard, ensure that you select an Amazon Linux AMI, and assign a public IP address to your instance. Ensure that your security group rules allow inbound SSH traffic from the range of IP addresses for your local network, and outbound SSH traffic to the IP address range of your private subnet (you can also use 0.0.0.0/0 for both inbound and outbound SSH traffic for this test).

Launch an instance in your private subnet. In the launch wizard, ensure that you select an Amazon Linux AMI. Do not assign a public IP address to your instance. Ensure that your security group rules allow inbound SSH traffic from the private IP address of your instance that you launched in the public subnet, and all outbound ICMP traffic. You must choose the same key pair that you used to launch your instance in the public subnet.

Configure SSH agent forwarding on your local computer, and connect to your bastion host in the public subnet.

From your bastion host, connect to your instance in the private subnet, and then test the internet connection from your instance in the private subnet.

To configure SSH agent forwarding for Linux or macOS

From your local machine, add your private key to the authentication agent.

For Linux, use the following command:

ssh-add -c mykeypair.pem

For macOS, use the following command:

ssh-add -K mykeypair.pem

Connect to your instance in the public subnet using the -A option to enable SSH agent forwarding, and use the instance's public address; for example:

ssh -A ec2-user@54.0.0.123

To configure SSH agent forwarding for Windows (PuTTY)

Download and install Pageant from the PuTTY download page, if not already installed.

Convert your private key to .ppk format.

Start Pageant, right-click the Pageant icon on the taskbar (it may be hidden), and choose Add Key. Select the .ppk file that you created, type the passphrase if necessary, and choose Open.

Start a PuTTY session and connect to your instance in the public subnet using its public IP address. In the Auth category, ensure that you select the Allow agent forwarding option, and leave the Private key file for authentication box blank.

To test the internet connection

From your instance in the public subnet, connect to your instance in your private subnet by using its private IP address, for example:

ssh ec2-user@10.0.1.123

From your private instance, test that you can connect to the internet by running the ping command for a website that has ICMP enabled, for example:

ping ietf.org

Network Access Control Lists

A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.

Network ACL Basics

The following are the basic things that you need to know about network ACLs:

Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.

You can create a custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.

Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.

You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.

A network ACL contains a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. The highest number that you can use for a rule is 32766. We recommend that you start by creating rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need to later on.

A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.

Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

Network ACL Rules

You can add or remove rules from the default network ACL, or create additional network ACLs for your VPC. When you add or remove rules from a network ACL, the changes are automatically applied to the subnets it's associated with.

The following are the parts of a network ACL rule:

Rule number. Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.

Type. The type of traffic; for example, SSH. You can also specify all traffic or a custom range.

Protocol. You can specify any protocol that has a standard protocol number. If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.

Port range. The listening port or port range for the traffic. For example, 80 for HTTP traffic.

Source. [Inbound rules only] The source of the traffic (CIDR range).

Destination. [Outbound rules only] The destination for the traffic (CIDR range).

Allow/Deny. Whether to ALLOW or DENY the specified traffic.

Default Network ACL

The default network ACL is configured to allow all traffic to flow in and out of the subnets with which it is associated. Each network ACL also includes a rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

The following is an example default network ACL for a VPC that supports IPv4 only.

Inbound

Rule #

Type

Protocol

Port Range

Source

Allow/Deny

100

All IPv4 traffic

All

All

0.0.0.0/0

ALLOW

\*

All IPv4 traffic

All

All

0.0.0.0/0

DENY

Outbound

Rule #

Type

Protocol

Port Range

Destination

Allow/Deny

100

All IPv4 traffic

All

All

0.0.0.0/0

ALLOW

\*

All IPv4 traffic

All

All

0.0.0.0/0

DENY

Custom Network ACL

The following table shows an example of a custom network ACL for a VPC that supports IPv4 only. It includes rules that allow HTTP and HTTPS traffic in (inbound rules 100 and 110). There's a corresponding outbound rule that enables responses to that inbound traffic (outbound rule 120, which covers ephemeral ports 32768-65535).

The network ACL also includes inbound rules that allow SSH and RDP traffic into the subnet. The outbound rule 120 enables responses to egress the subnet.

The network ACL has outbound rules (100 and 110) that allow outbound HTTP and HTTPS traffic out of the subnet. There's a corresponding inbound rule that enables responses to that outbound traffic (inbound rule 140, which covers ephemeral ports 32768-65535).

Note

Each network ACL includes a default rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other rules, it's denied. You can't modify or remove this rule.

Inbound

Rule #

Type

Protocol

Port Range

Source

Allow/Deny

Comments

100

HTTP

TCP

80

0.0.0.0/0

ALLOW

Allows inbound HTTP traffic from any IPv4 address.

110

HTTPS

TCP

443

0.0.0.0/0

ALLOW

Allows inbound HTTPS traffic from any IPv4 address.

120

SSH

TCP

22

192.0.2.0/24

ALLOW

Allows inbound SSH traffic from your home network's public IPv4 address range (over the Internet gateway).

130

RDP

TCP

3389

192.0.2.0/24

ALLOW

Allows inbound RDP traffic to the web servers from your home network's public IPv4 address range (over the Internet gateway).

140

Custom TCP

TCP

32768-65535

0.0.0.0/0

ALLOW

Allows inbound return IPv4 traffic from the Internet (that is, for requests that originate in the subnet).

This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports.

\*

All traffic

All

All

0.0.0.0/0

DENY

Denies all inbound IPv4 traffic not already handled by a preceding rule (not modifiable).

Outbound

Rule #

Type

Protocol

Port Range

Destination

Allow/Deny

Comments

100

HTTP

TCP

80

0.0.0.0/0

ALLOW

Allows outbound IPv4 HTTP traffic from the subnet to the Internet.

110

HTTPS

TCP

443

0.0.0.0/0

ALLOW

Allows outbound IPv4 HTTPS traffic from the subnet to the Internet.

120

Custom TCP

TCP

32768-65535

0.0.0.0/0

ALLOW

Allows outbound IPv4 responses to clients on the Internet (for example, serving web pages to people visiting the web servers in the subnet).

This range is an example only. For more information about how to select the appropriate ephemeral port range, see Ephemeral Ports.

\*

All traffic

All

All

0.0.0.0/0

DENY

Denies all outbound IPv4 traffic not already handled by a preceding rule (not modifiable).

As a packet comes to the subnet, we evaluate it against the ingress rules of the ACL the subnet is associated with (starting at the top of the list of rules, and moving to the bottom). Here's how the evaluation goes if the packet is destined for the SSL port (443). The packet doesn't match the first rule evaluated (rule 100). It does match the second rule (110), which allows the packet into the subnet. If the packet had been destined for port 139 (NetBIOS), it doesn't match any of the rules, and the \* rule ultimately denies the packet.

You might want to add a DENY rule in a situation where you legitimately need to open a wide range of ports, but there are certain ports within that range you want to deny. Just make sure to place the DENY rule earlier in the table than the rule that allows the wide range of port traffic.

You add ALLOW rules depending on your use case; for example, a rule that allows outbound TCP and UDP access on port 53 for DNS resolution. For every rule that you add, ensure that there is a corresponding inbound or outbound rule that allows response traffic.

Security Groups

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign up to five security groups to the instance. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups. If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

The following are the basic characteristics of security groups for your VPC:

You can specify allow rules, but not deny rules.

You can specify separate rules for inbound and outbound traffic.

When you create a security group, it has no inbound rules. Therefore, no inbound traffic originating from another host to your instance is allowed until you add inbound rules to the security group.

By default, a security group includes an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that allow specific outbound traffic only. If your security group has no outbound rules, no outbound traffic originating from your instance is allowed.

Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.

Instances associated with a security group can't talk to each other unless you add rules allowing it (exception: the default security group has these rules by default).

AWS NACL vs Security group comparison

Security Groups function at the EC2 instance level. Where as NACLs reside at the subnet level. This is the most obvious of differences and is likely to be brought up in the exam.

NACL’s allow for rules to be defined for both inbound and outbound traffic individually. Security groups only allow to define rules that apply to both inbound and outbound. You can’t defined difference behaviors for each.

Security Groups are stateful! This is something that has a good chance of showing up in the exam. Because NACLs are not stateful. This means that return traffic must be allowed. I.e. you need an inbound and an outbound rule.

NACL rules are evaluated in numerical order. Where as Security Group rules are all processed before deciding whether or not to allow traffic into the instance.

Finally, nacl rules apply to all the instances within a subnet. So if traffic is blocked at the nacl level but allowed at the instance level. It will never reach the instance as it was blocked before the security group rules could be applied.

Create a new NACL

Navigate to the VPC dashboard:

Networking & Content

Delivery

VPC

CloudFront

Route 53

API Gateway

Direct Connect

AWS Cloud Map

Global Accelerator

Select Network ACL’s from the security tab:

Security

Network ACLs

Security Groups

Click Create new ACL:

Create network ACL

Actions

Enter a name tag and a VPC to associate the ACL with:

Create network ACL

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Name tag

VPC\*

o

o

On the dashbaord for the ACL’s you can see your newly created ACL and the subnets it is associated with (currently 0).

From here we can configure rules for the ACL:

Details

Inbound Rules

Outbound Rules

Subnet associations

Tags

Edit inbound rules

Rule #

100

View

Type

ALL Traffic

ALL Traffic

All rules

Protocol

All

Port Range

Source

0.0.0.0/0

0.0.0.0/0

Allow / Deny

ALLOW

DENY

Here we can see the inbound rules for this ACL. In this case it will allow all traffic from all IP addresses.

Details

Inbound Rules

Outbound Rules

Subnet associations

Tags

Edit outbound rules

Rule #

100

View

Type

ALL Traffic

ALL Traffic

All rules

Protocol

All

Port Range

Destination

0.0.0.0/0

0.0.0.0/0

Allow / Deny

ALLOW

DENY

VPC Flow Logs

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs and Amazon S3. After you've created a flow log, you can retrieve and view its data in the chosen destination.

Flow logs can help you with a number of tasks, such as:

Diagnosing overly restrictive security group rules

Monitoring the traffic that is reaching your instance

Determining the direction of the traffic to and from the network interfaces

Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency. You can create or delete flow logs without any risk of impact to network performance.

Flow Logs Basics

You can create a flow log for a VPC, a subnet, or a network interface. If you create a flow log for a subnet or VPC, each network interface in that subnet or VPC is monitored.

Flow log data for a monitored network interface is recorded as flow log records, which are log events consisting of fields that describe the traffic flow.

To create a flow log, you specify:

The resource for which to create the flow log

The type of traffic to capture (accepted traffic, rejected traffic, or all traffic)

The destinations to which you want to publish the flow log data

After you've created a flow log, it can take several minutes to begin collecting and publishing data to the chosen destinations.

AWS Transit Gateway

What is Transit Gateway?

A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

Transit Gateway Concepts

The following are the key concepts for transit gateways:

attachment — You can attach a VPC, an AWS Direct Connect gateway, or a VPN connection to a transit gateway.

transit gateway route table — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be a VPC or a VPN connection. By default, the VPCs and VPN connections that you attach to a transit gateway are associated with the default transit gateway route table.

associations — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.

route propagation — A VPC or VPN connection can dynamically propagate routes to a transit gateway route table. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP).

Simplify how you connect Amazon VPCs and VPNs

Without AWS Transit Gateway

tgw-before

You needed to peer each Amazon VPC to each other and to each onsite location using a VPN connection which can be complex as its scales.

With AWS Transit Gateway

tgw-after

You simply connect each Amazon VPC or VPN to the AWS Transit Gateway and it will route traffic to and from each VPC or VPN.

**NAT Gateways**

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

**NAT Gateway Basics**

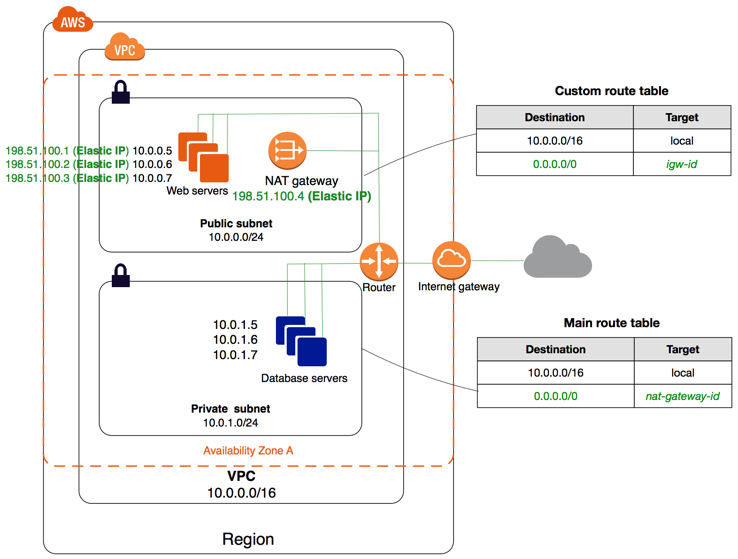
To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside. You must also specify an Elastic IP address to associate with the NAT gateway when you create it. The Elastic IP address cannot be changed once you associate it with the NAT Gateway. After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point Internet-bound traffic to the NAT gateway. This enables instances in your private subnets to communicate with the internet.

Each NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone. You have a limit on the number of NAT gateways you can create in an Availability Zone.

**Note**

If you have resources in multiple Availability Zones and they share one NAT gateway, in the event that the NAT gateway’s Availability Zone is down, resources in the other Availability Zones lose internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

The following diagram illustrates the architecture of a VPC with a NAT gateway. The main route table sends internet traffic from the instances in the private subnet to the NAT gateway. The NAT gateway sends the traffic to the internet gateway using the NAT gateway’s Elastic IP address as the source IP address.



**Note:**An *Elastic IP address* is a static, public IPv4 address designed for dynamic cloud computing. You can associate an Elastic IP address with any instance or network interface for any VPC in your account. With an Elastic IP address, you can mask the failure of an instance by rapidly remapping the address to another instance in your VPC.

***Testing the Internet Connection***

The following example demonstrates how to test if your instance in a private subnet can connect to the internet.

1. Launch an instance in your public subnet (you use this as a bastion host). In the launch wizard, ensure that you select an Amazon Linux AMI, and assign a public IP address to your instance. Ensure that your security group rules allow inbound SSH traffic from the range of IP addresses for your local network, and outbound SSH traffic to the IP address range of your private subnet (you can also use 0.0.0.0/0 for both inbound and outbound SSH traffic for this test).
2. Launch an instance in your private subnet. In the launch wizard, ensure that you select an Amazon Linux AMI. Do not assign a public IP address to your instance. Ensure that your security group rules allow inbound SSH traffic from the private IP address of your instance that you launched in the public subnet, and all outbound ICMP traffic. You must choose the same key pair that you used to launch your instance in the public subnet.
3. Configure SSH agent forwarding on your local computer, and connect to your bastion host in the public subnet.

1. From your bastion host, connect to your instance in the private subnet, and then test the internet connection from your instance in the private subnet.

**To configure SSH agent forwarding for Linux or macOS**

1. From your local machine, add your private key to the authentication agent.

For Linux, use the following command:   
ssh-add -c mykeypair.pem

For macOS, use the following command:   
ssh-add -K mykeypair.pem

1. Connect to your instance in the public subnet using the -A option to enable SSH agent forwarding, and use the instance's public address; for example:   
      
   ssh -A ec2-user@*54.0.0.123*

**To configure SSH agent forwarding for Windows (PuTTY)**

1. Download and install Pageant from the [PuTTY download page](http://www.chiark.greenend.org.uk/~sgtatham/putty/), if not already installed.
2. Convert your private key to .ppk format.

1. Start Pageant, right-click the Pageant icon on the taskbar (it may be hidden), and choose **Add Key**. Select the .ppk file that you created, type the passphrase if necessary, and choose **Open**.
2. Start a PuTTY session and connect to your instance in the public subnet using its public IP address.  In the **Auth** category, ensure that you select the **Allow agent forwarding** option, and leave the **Private key file for authentication** box blank.

**To test the internet connection**

1. From your instance in the public subnet, connect to your instance in your private subnet by using its private IP address, for example:   
      
   ssh ec2-user@*10.0.1.123*
2. From your private instance, test that you can connect to the internet by running the ping command for a website that has ICMP enabled, for example:   
      
   ping ietf.org

**Network Access Control Lists**

A *network access control list (ACL)* is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.

**Network ACL Basics**

The following are the basic things that you need to know about network ACLs:

* Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
* You can create a custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
* Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
* You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.
* A network ACL contains a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. The highest number that you can use for a rule is 32766. We recommend that you start by creating rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need to later on.
* A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
* Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

**Network ACL Rules**

You can add or remove rules from the default network ACL, or create additional network ACLs for your VPC. When you add or remove rules from a network ACL, the changes are automatically applied to the subnets it's associated with.

The following are the parts of a network ACL rule:

* **Rule number**. Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.
* **Type**. The type of traffic; for example, SSH. You can also specify all traffic or a custom range.
* **Protocol**. You can specify any protocol that has a standard protocol number. If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.
* **Port range**. The listening port or port range for the traffic. For example, 80 for HTTP traffic.
* **Source**. [Inbound rules only] The source of the traffic (CIDR range).
* **Destination**. [Outbound rules only] The destination for the traffic (CIDR range).
* **Allow/Deny**. Whether to ALLOW or DENY the specified traffic.

**Default Network ACL**

The default network ACL is configured to allow all traffic to flow in and out of the subnets with which it is associated. Each network ACL also includes a rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

The following is an example default network ACL for a VPC that supports IPv4 only.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Inbound** |  |  |  |  |  |
| **Rule #** | **Type** | **Protocol** | **Port Range** | **Source** | **Allow/Deny** |
| 100 | All IPv4 traffic | All | All | 0.0.0.0/0 | ALLOW |
| \* | All IPv4 traffic | All | All | 0.0.0.0/0 | DENY |
| **Outbound** |  |  |  |  |  |
| **Rule #** | **Type** | **Protocol** | **Port Range** | **Destination** | **Allow/Deny** |
| 100 | All IPv4 traffic | All | All | 0.0.0.0/0 | ALLOW |
| \* | All IPv4 traffic | All | All | 0.0.0.0/0 | DENY |

**Custom Network ACL**

The following table shows an example of a custom network ACL for a VPC that supports IPv4 only. It includes rules that allow HTTP and HTTPS traffic in (inbound rules 100 and 110). There's a corresponding outbound rule that enables responses to that inbound traffic (outbound rule 120, which covers ephemeral ports 32768-65535).

The network ACL also includes inbound rules that allow SSH and RDP traffic into the subnet. The outbound rule 120 enables responses to egress the subnet.

The network ACL has outbound rules (100 and 110) that allow outbound HTTP and HTTPS traffic out of the subnet. There's a corresponding inbound rule that enables responses to that outbound traffic (inbound rule 140, which covers ephemeral ports 32768-65535).

**Note**

Each network ACL includes a default rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other rules, it's denied. You can't modify or remove this rule.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inbound** |  |  |  |  |  |  |
| **Rule #** | **Type** | **Protocol** | **Port Range** | **Source** | **Allow/Deny** | **Comments** |
| 100 | HTTP | TCP | 80 | 0.0.0.0/0 | ALLOW | Allows inbound HTTP traffic from any IPv4 address. |
| 110 | HTTPS | TCP | 443 | 0.0.0.0/0 | ALLOW | Allows inbound HTTPS traffic from any IPv4 address. |
| 120 | SSH | TCP | 22 | 192.0.2.0/24 | ALLOW | Allows inbound SSH traffic from your home network's public IPv4 address range (over the Internet gateway). |
| 130 | RDP | TCP | 3389 | 192.0.2.0/24 | ALLOW | Allows inbound RDP traffic to the web servers from your home network's public IPv4 address range (over the Internet gateway). |
| 140 | Custom TCP | TCP | 32768-65535 | 0.0.0.0/0 | ALLOW | Allows inbound return IPv4 traffic from the Internet (that is, for requests that originate in the subnet).  This range is an example only. For more information about how to select the appropriate ephemeral port range, see [Ephemeral Ports](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports). |
| \* | All traffic | All | All | 0.0.0.0/0 | DENY | Denies all inbound IPv4 traffic not already handled by a preceding rule (not modifiable). |
| **Outbound** |  |  |  |  |  |  |
| **Rule #** | **Type** | **Protocol** | **Port Range** | **Destination** | **Allow/Deny** | **Comments** |
| 100 | HTTP | TCP | 80 | 0.0.0.0/0 | ALLOW | Allows outbound IPv4 HTTP traffic from the subnet to the Internet. |
| 110 | HTTPS | TCP | 443 | 0.0.0.0/0 | ALLOW | Allows outbound IPv4 HTTPS traffic from the subnet to the Internet. |
| 120 | Custom TCP | TCP | 32768-65535 | 0.0.0.0/0 | ALLOW | Allows outbound IPv4 responses to clients on the Internet (for example, serving web pages to people visiting the web servers in the subnet).  This range is an example only. For more information about how to select the appropriate ephemeral port range, see [Ephemeral Ports](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports). |
| \* | All traffic | All | All | 0.0.0.0/0 | DENY | Denies all outbound IPv4 traffic not already handled by a preceding rule (not modifiable). |

As a packet comes to the subnet, we evaluate it against the ingress rules of the ACL the subnet is associated with (starting at the top of the list of rules, and moving to the bottom). Here's how the evaluation goes if the packet is destined for the SSL port (443). The packet doesn't match the first rule evaluated (rule 100). It does match the second rule (110), which allows the packet into the subnet. If the packet had been destined for port 139 (NetBIOS), it doesn't match any of the rules, and the \* rule ultimately denies the packet.

You might want to add a DENY rule in a situation where you legitimately need to open a wide range of ports, but there are certain ports within that range you want to deny. Just make sure to place the DENY rule earlier in the table than the rule that allows the wide range of port traffic.

You add ALLOW rules depending on your use case; for example, a rule that allows outbound TCP and UDP access on port 53 for DNS resolution. For every rule that you add, ensure that there is a corresponding inbound or outbound rule that allows response traffic.

**Security Groups**

A *security group* acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign up to five security groups to the instance. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups. If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

The following are the basic characteristics of security groups for your VPC:

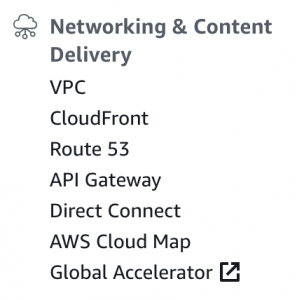
* You can specify allow rules, but not deny rules.
* You can specify separate rules for inbound and outbound traffic.
* When you create a security group, it has no inbound rules. Therefore, no inbound traffic originating from another host to your instance is allowed until you add inbound rules to the security group.
* By default, a security group includes an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that allow specific outbound traffic only. If your security group has no outbound rules, no outbound traffic originating from your instance is allowed.
* Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.
* Instances associated with a security group can't talk to each other unless you add rules allowing it (exception: the default security group has these rules by default).

**AWS NACL vs Security group comparison**

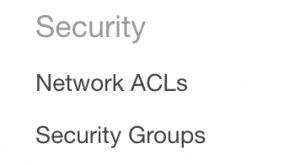
* Security Groups function at the EC2 instance level. Where as NACLs reside at the subnet level. This is the most obvious of differences and is likely to be brought up in the exam.
* NACL’s allow for rules to be defined for both inbound and outbound traffic individually. Security groups only allow to define rules that apply to both inbound and outbound. You can’t defined difference behaviors for each.
* Security Groups are stateful! This is something that has a good chance of showing up in the exam. Because NACLs are not stateful. This means that return traffic must be allowed. I.e. you need an inbound and an outbound rule.
* NACL rules are evaluated in numerical order. Where as Security Group rules are all processed before deciding whether or not to allow traffic into the instance.
* Finally, nacl rules apply to all the instances within a subnet. So if traffic is blocked at the nacl level but allowed at the instance level. It will never reach the instance as it was blocked before the security group rules could be applied.

**Create a new NACL**

Navigate to the VPC dashboard:



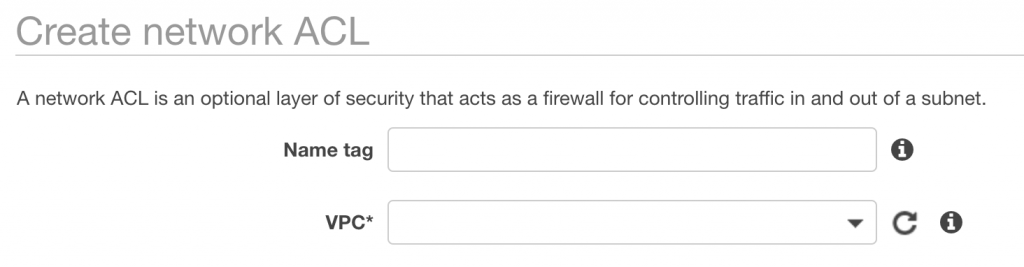
Select Network ACL’s from the security tab:



Click Create new ACL:

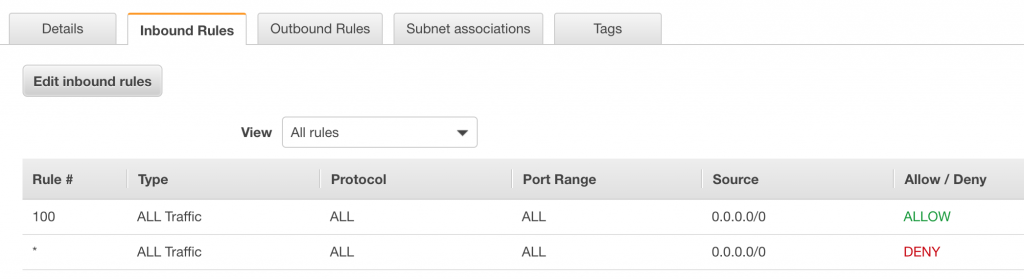
C:\Users\Zenetex admin\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\D93EA33C.tmp

Enter a name tag and a VPC to associate the ACL with:

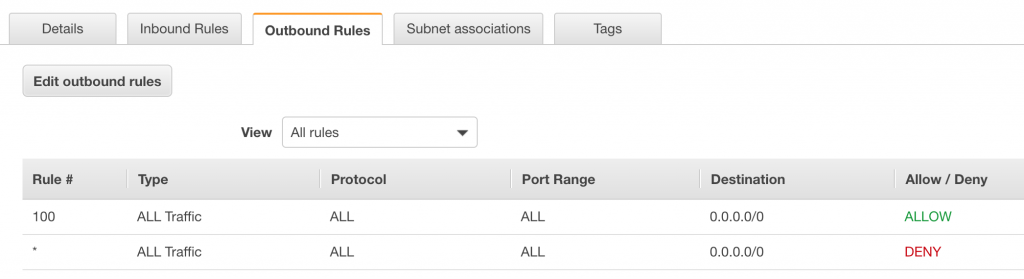


On the dashbaord for the ACL’s you can see your newly created ACL and the subnets it is associated with (currently 0).

From here we can configure rules for the ACL:



Here we can see the inbound rules for this ACL. In this case it will allow all traffic from all IP addresses.



**VPC Flow Logs**

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs and Amazon S3. After you've created a flow log, you can retrieve and view its data in the chosen destination.

Flow logs can help you with a number of tasks, such as:

* Diagnosing overly restrictive security group rules
* Monitoring the traffic that is reaching your instance
* Determining the direction of the traffic to and from the network interfaces

Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency. You can create or delete flow logs without any risk of impact to network performance.

**Flow Logs Basics**

You can create a flow log for a VPC, a subnet, or a network interface. If you create a flow log for a subnet or VPC, each network interface in that subnet or VPC is monitored.

Flow log data for a monitored network interface is recorded as *flow log records*, which are log events consisting of fields that describe the traffic flow.

To create a flow log, you specify:

* The resource for which to create the flow log
* The type of traffic to capture (accepted traffic, rejected traffic, or all traffic)
* The destinations to which you want to publish the flow log data

After you've created a flow log, it can take several minutes to begin collecting and publishing data to the chosen destinations.

**AWS Transit Gateway**

What is Transit Gateway?

A *transit gateway* is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

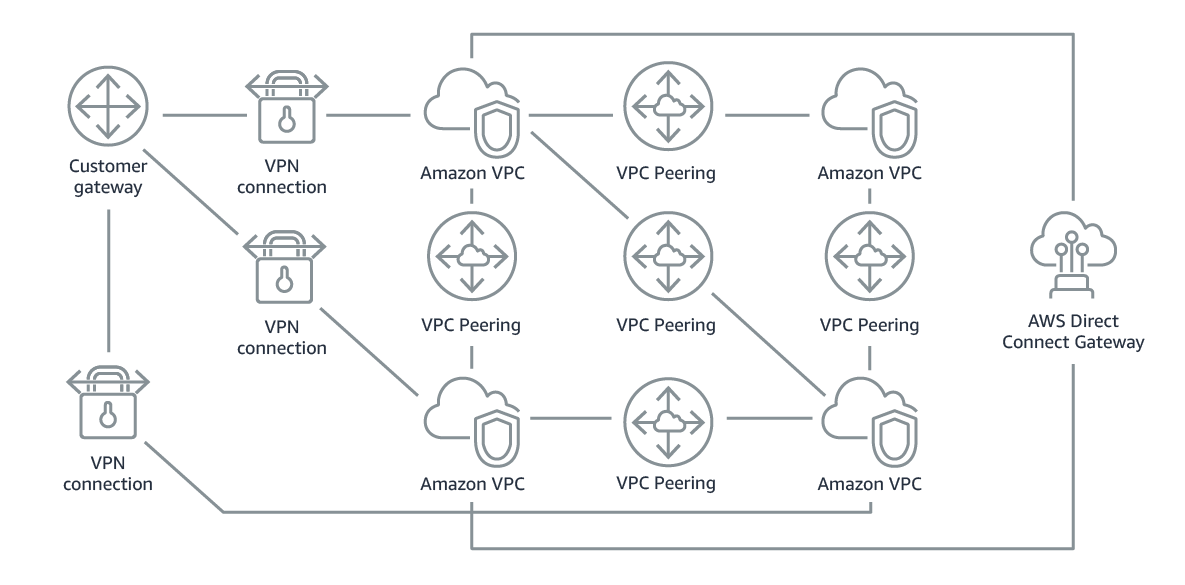
**Transit Gateway Concepts**

The following are the key concepts for transit gateways:

* **attachment** — You can attach a VPC, an AWS Direct Connect gateway, or a VPN connection to a transit gateway.
* **transit gateway route table** — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be a VPC or a VPN connection. By default, the VPCs and VPN connections that you attach to a transit gateway are associated with the default transit gateway route table.
* **associations** — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.
* **route propagation** — A VPC or VPN connection can dynamically propagate routes to a transit gateway route table. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP).

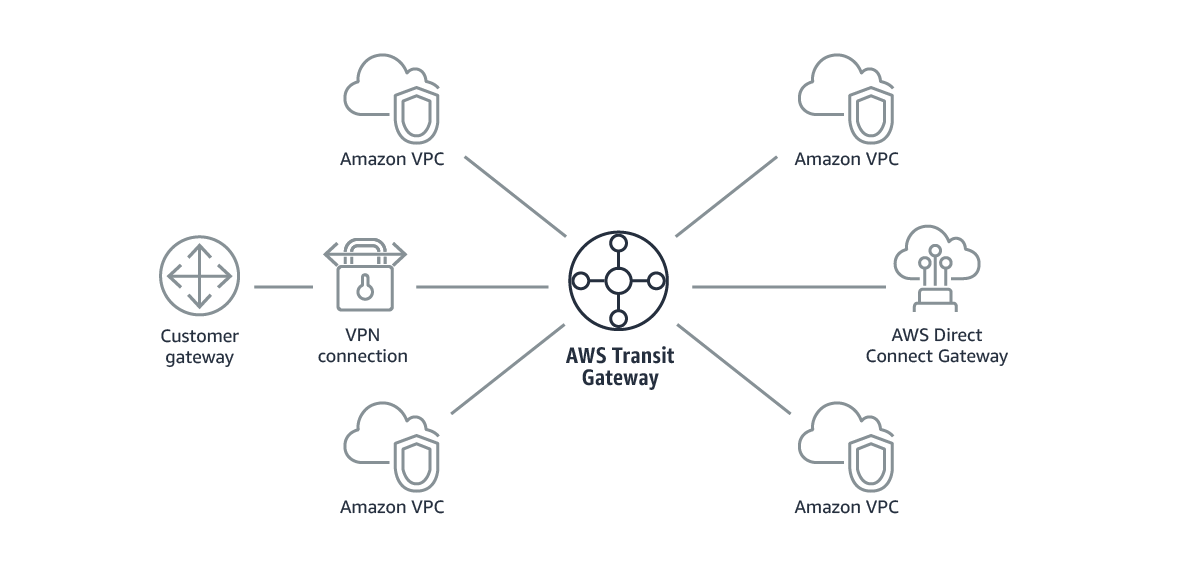
1. Simplify how you connect Amazon VPCs and VPNs

Without AWS Transit Gateway



You needed to peer each Amazon VPC to each other and to each onsite location using a VPN connection which can be complex as its scales.

With AWS Transit Gateway



You simply connect each Amazon VPC or VPN to the AWS Transit Gateway and it will route traffic to and from each VPC or VPN.

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS 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 own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

You can easily customize the network configuration for your Amazon VPC. For example, you can create a public-facing subnet for your web servers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

Benefits

SECURE

Amazon VPC provides advanced security features, such as security groups and network access control lists, to enable inbound and outbound filtering at the instance level and subnet level. In addition, you can store data in Amazon S3 and restrict access so that it’s only accessible from instances in your VPC. Optionally, you can also choose to launch [Dedicated Instances](http://aws.amazon.com/dedicated-instances) which run on hardware dedicated to a single customer for additional isolation.

The following are the key concepts for VPCs:

* A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account.
* A *subnet* is a range of IP addresses in your VPC.
* A *route table* contains a set of rules, called routes, that are used to determine where network traffic is directed.
* An *internet gateway* is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.
* A *VPC endpoint* enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

Features

MULTIPLE CONNECTIVITY OPTIONS

A variety of connectivity options exist for your Amazon VPC. You can connect your VPC to the Internet, to your data center, or other VPCs, based on the AWS resources that you want to expose publicly and those that you want to keep private.

* Connect directly to the Internet (public subnets)– You can launch instances into a publicly accessible subnet where they can send and receive traffic from the Internet.
* Connect to the Internet using Network Address Translation (private subnets) – Private subnets can be used for instances that you do not want to be directly addressable from the Internet. Instances in a private subnet can access the Internet without exposing their private IP address by routing their traffic through a Network Address Translation (NAT) gateway in a public subnet.
* Connect securely to your corporate datacenter– All traffic to and from instances in your VPC can be routed to your corporate datacenter over an industry standard, encrypted IPsec hardware VPN connection.
* Connect privately to other VPCs- Peer VPCs together to share resources across multiple virtual networks owned by your or other AWS accounts.
* Privately connect to AWS Services without using an Internet gateway, NAT or firewall proxy through a VPC Endpoint. Available AWS services include S3, DynamoDB, Kinesis Streams, Service Catalog, AWS Systems Manager, Elastic Load Balancing (ELB) API, Amazon Elastic Compute Cloud (EC2) API, and SNS.
* Privately connect to SaaS solutions supported by [AWS PrivateLink](https://aws.amazon.com/privatelink/).
* Privately connect your internal services across different accounts and VPCs within your own organizations, significantly simplifying your internal network architecture.
* Use Amazon VPC traffic mirroring to capture and mirror network traffic for Amazon EC2 instances

Use cases

HOST A SIMPLE, PUBLIC-FACING WEBSITE

You can host a basic web application, such as a blog or simple website in a VPC, and gain the additional layers of privacy and security afforded by Amazon VPC. You can help secure the website by creating security group rules which allow the webserver to respond to inbound HTTP and SSL requests from the Internet while simultaneously prohibiting the webserver from initiating outbound connections to the Internet. You can create a VPC that supports this use case by selecting "VPC with a Single Public Subnet Only" from the Amazon VPC console wizard.

HOST MULTI-TIER WEB APPLICATIONS

You can use Amazon VPC to host multi-tier web applications and strictly enforce access and security restrictions between your webservers, application servers, and databases. You can launch webservers in a publicly accessible subnet and application servers and databases in non-publically accessible subnets. The application servers and databases can’t be directly accessed from the Internet, but they can still access the Internet via a NAT gateway to download patches, for example. You can control access between the servers and subnets using inbound and outbound packet filtering provided by network access control lists and security groups. To create a VPC that supports this use case, you can select "VPC with Public and Private Subnets" in the Amazon VPC console wizard.

HOST SCALABLE WEB APPLICATIONS IN THE AWS CLOUD THAT ARE CONNECTED TO YOUR DATACENTER

You can create a VPC where instances in one subnet, such as web servers, communicate with the Internet while instances in another subnet, such as application servers, communicate with databases on your corporate network. An IPsec VPN connection between your VPC and your corporate network helps secure all communication between the application servers in the cloud and databases in your data center. Web servers and application servers in your VPC can leverage Amazon EC2 elasticity and Auto Scaling features to grow and shrink as needed. You can create a VPC to support this use case by selecting "VPC with Public and Private Subnets and Hardware VPN Access" in the Amazon VPC console wizard.

EXTEND YOUR CORPORATE NETWORK INTO THE CLOUD

You can move corporate applications to the cloud, launch additional web servers, or add more compute capacity to your network by connecting your VPC to your corporate network. Because your VPC can be hosted behind your corporate firewall, you can seamlessly move your IT resources into the cloud without changing how your users access these applications. You can select "VPC with a Private Subnet Only and Hardware VPN Access" from the Amazon VPC console wizard to create a VPC that supports this use case.

DISASTER RECOVERY

You can periodically backup your mission critical data from your datacenter to a small number of Amazon EC2 instances with Amazon Elastic Block Store (EBS) volumes, or import your virtual machine images to Amazon EC2. In the event of a disaster in your own datacenter, you can quickly launch replacement compute capacity in AWS to ensure business continuity. When the disaster is over, you can send your mission critical data back to your datacenter and terminate the Amazon EC2 instances that you no longer need. By using Amazon VPC for disaster recovery, you can have all the benefits of a disaster recovery site at a fraction of the normal cost.

**Pricing for Amazon VPC**

There's no additional charge for using Amazon VPC. You pay the standard rates for the instances and other Amazon EC2 features that you use. There are charges for using an Site-to-Site VPN connection and using a NAT gateway.

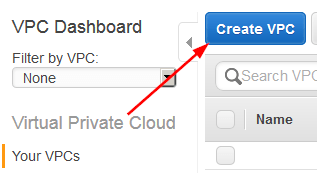
Create a custom VPC with private and public subnets on AWS

**Create a custom VPC in AWS**

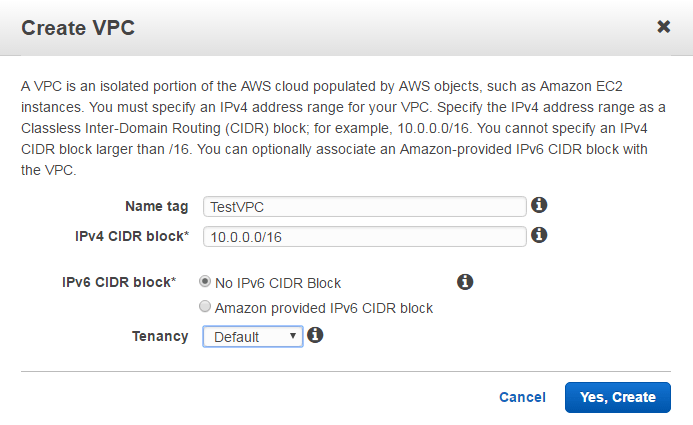
1. First, log into your AWS Console and click on **VPC** under *Network & Content Delivery*“



1. [Click on “Create VPC“.](https://miketabor.com/wp-content/uploads/2017/03/create-vpc-aws-console.png)



1. [Enter a Name and CIDR block address, for this VPC I’ll be using 10.0.0.0/16.](https://miketabor.com/wp-content/uploads/2017/03/create-vpc-create-vpc.png)



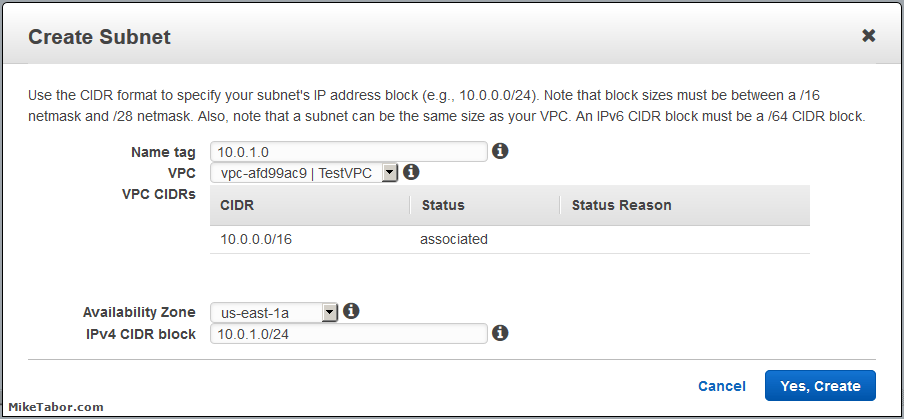
[When you create a VPC a Route Table, Network ACL, and Security Group are automatically created.](https://miketabor.com/wp-content/uploads/2017/03/create-vpc-vpc-name.png)

[Subnets or Internet Gateways are NOT automatically created, so we’ll create those below.](https://miketabor.com/wp-content/uploads/2017/03/create-vpc-vpc-name.png)

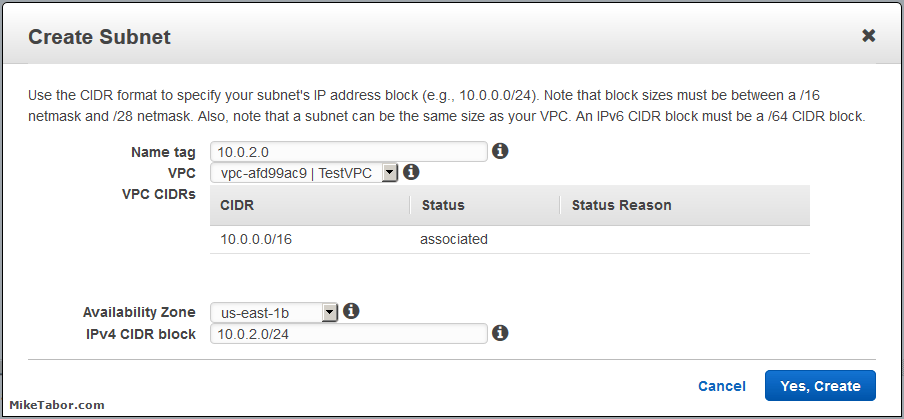
1. [To create Subnets for our new VPC we need to click on Subnets, then click on “Create Subnet“.](https://miketabor.com/wp-content/uploads/2017/03/create-vpc-vpc-name.png)



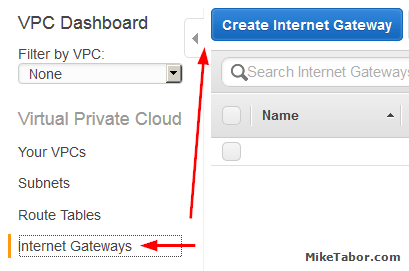
1. [Enter a name for your subnet then select the VPC and Availability Zone then finally the CIDR block for the subnet. Keep in mind your subnet CIDR blocks need to be smaller than the VPC CIDR block.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-subnets.png)



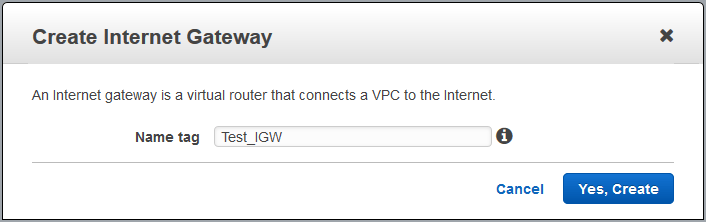
1. [Repeat the process to create a second subnet:](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-subnets-02.png)



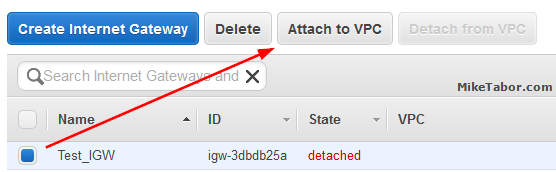
1. [Once you’ve created your two subnets you should have something that looks like this:](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-subnets-03.png)
2. [Next we need to create an Internet Gateway to make this VPC Internet accessible. So click on Internet Gateways from the menu on the left and then click “Create Internet Gateway“](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-subnets-overview.png)



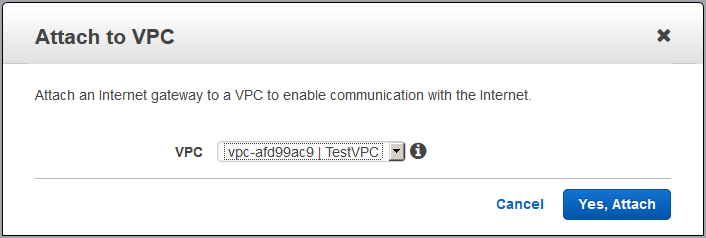
1. [Give the new Internet Gateway a name, then click on “Yes, Create“.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-internet-gateway.png)



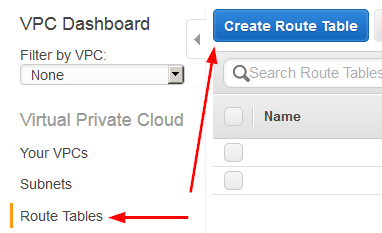
1. [Attach the newly created Internet Gateway to a VPC by selecting your Internet Gateway and click on “Attach to VPC“.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-internet-gateway-name.png)



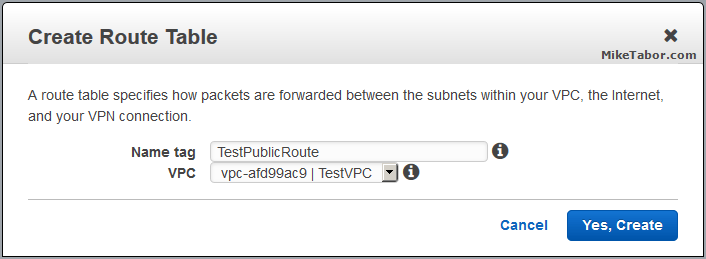
1. [Note: You can only attach ONE Internet Gateway to a VPC.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-attach-internet-gateway-vpc.png)
2. [Then select the VPC you want to connect the Internet Gateway to and click “Yes, Attach“.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-attach-internet-gateway-vpc.png)



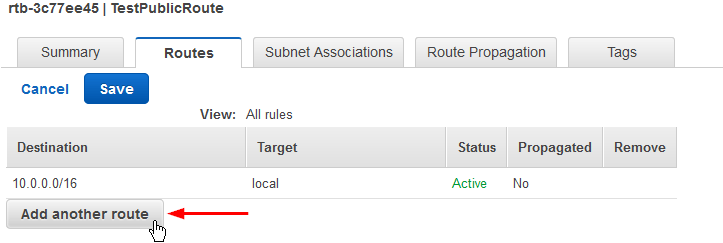
1. [Now we need to create a new route table by selecting Route Tables and clicking on “Create Route Table“.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-attach-internet-gateway-vpc-02.png)



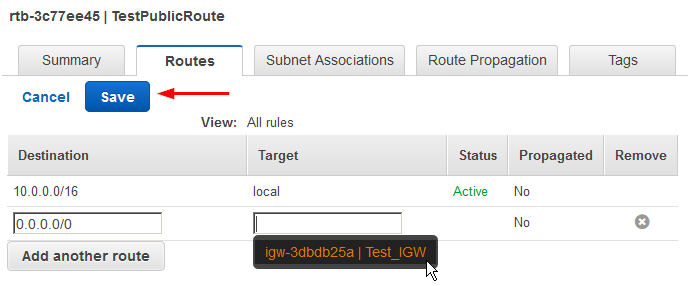
1. [Give the new Route Table a name, then click on “Yes, Create“.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-route-table.png)



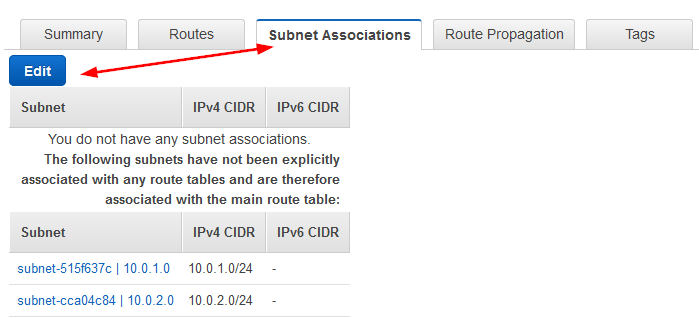
1. [We now need to give the Route Table (TestPublicRoute) a route to the internet. Select the Route Table and click on Edit.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-route-table-name.png)
2. [Lets add a route out to the Internet by clicking on “Add another route“](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-route-table-edit.png)



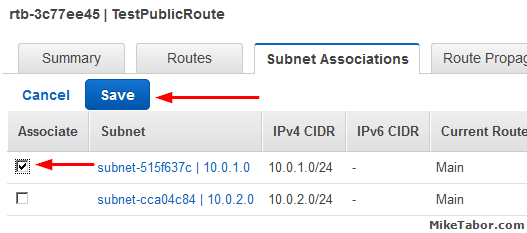
[Then enter 0.0.0.0/0 for the Destination and select your Internet Gateway for the Target and click on “Save“:](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-route-table-add-another-route.png)



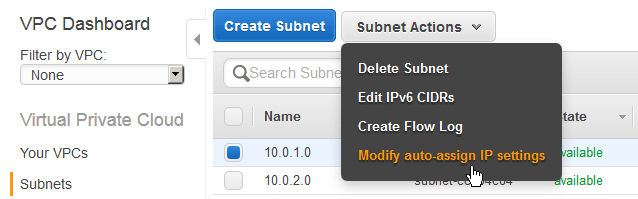
[Lets associate a subnet we want to have internet access by going to the Subnet Associations and clicking on Edit.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-create-route-table-internet-route.png)



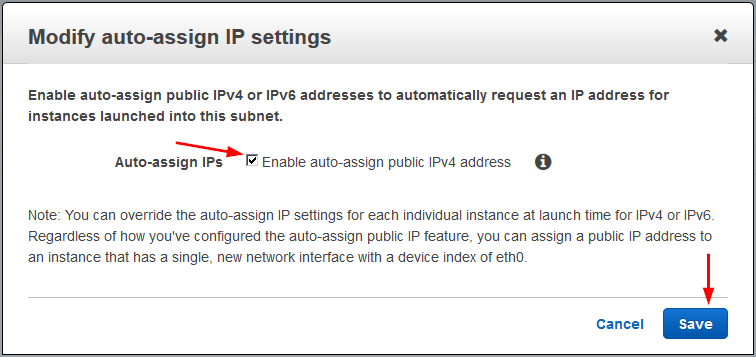
[Then select your subnet, in this case 10.0.1.0, and click on “Save“.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-route-table-subnet-associations.png)



1. [Finally, since this subnet (10.0.1.0) is an internet facing subnet – lets go ahead and change the auto-assign an IP settings so that when EC2 instances are deployed they get a public IP.   
   So lets click on Subnets then select your public subnet and then click on “Modify auto-assign IP settings” from the Subnet Actions drop down menu.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-route-table-subnet-associations-subnet.png)



[Then check the box and click on Save.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet.png)



[And with that, we have now created a custom VPC in AWS with a public (10.0.1.0) subnet and a private (10.0.2.0) subnet!](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png)

[VPC Sizing   
Some additional VPC information regarding subnets. The smallest subnet you can create is a /28 and the largest subnet is a /16. AWS also reserves 5 IP addresses in each subnet.   
For example, in a subnet with a CIDR block 10.0.0.0/24 (like we created above), the following 5 IP addresses are reserved:](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png)

[10.0.0.0 – Network address.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png)

[10.0.0.1 – Reserved for the VPC router.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png)

[10.0.0.2 – Reserved for mapping to the Amazon DNS Server.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png)

[10.0.0.3 – Reserved for future use.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png)

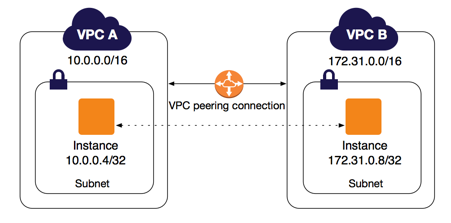
[10.0.0.255 – Network broadcast address. AWS does not support broadcast in a VPC, therefore AWS reserves this address.](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png)

|  |  |  |
| --- | --- | --- |
| [**CIDR Block**](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [**Addresses**](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [**Usable**](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/28](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [16](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [11](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/27](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [32](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [27](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/26](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [64](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [59](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/25](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [128](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [123](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/24](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [256](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [251](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/23](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [512](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [507](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/22](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [1024](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [1019](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/21](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [2048](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [2043](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/20](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [4096](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [4091](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/19](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [8192](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [8187](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/18](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [16384](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [16379](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/17](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [32768](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [32763](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |
| [/16](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [65536](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) | [65531](https://miketabor.com/wp-content/uploads/2017/03/create-custom-vpc-auto-assign-ip-subnet-enable.png) |

**What is VPC Peering?**

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined.

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection).



AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.

A VPC peering connection helps you to facilitate the transfer of data. For example, if you have more than one AWS account, you can peer the VPCs across those accounts to create a file sharing network. You can also use a VPC peering connection to allow other VPCs to access resources you have in one of your VPCs.

You can establish peering relationships between VPCs across different AWS Regions (also called Inter-Region VPC Peering). This allows VPC resources including EC2 instances, Amazon RDS databases and Lambda functions that run in different AWS Regions to communicate with each other using private IP addresses, without requiring gateways, VPN connections, or separate network appliances. The traffic remains in the private IP space. All inter-region traffic is encrypted with no single point of failure, or bandwidth bottleneck. Traffic always stays on the global AWS backbone, and never traverses the public internet, which reduces threats, such as common exploits, and DDoS attacks. Inter-Region VPC Peering provides a simple and cost-effective way to share resources between regions or replicate data for geographic redundancy.

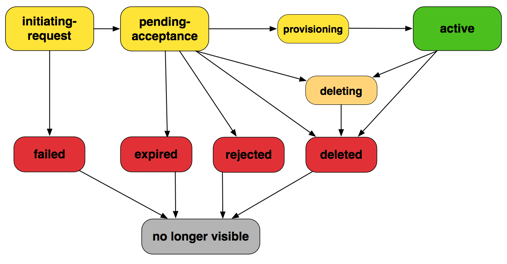
**VPC Peering Basics**

To establish a VPC peering connection, you do the following:

1. The owner of the *requester VPC* sends a request to the owner of the *accepter VPC* to create the VPC peering connection. The accepter VPC can be owned by you, or another AWS account, and cannot have a CIDR block that overlaps with the requester VPC's CIDR block.
2. The owner of the accepter VPC accepts the VPC peering connection request to activate the VPC peering connection.
3. To enable the flow of traffic between the VPCs using private IP addresses, the owner of each VPC in the VPC peering connection must manually add a route to one or more of their VPC route tables that points to the IP address range of the other VPC (the peer VPC).
4. If required, update the security group rules that are associated with your instance to ensure that traffic to and from the peer VPC is not restricted. If both VPCs are in the same region, you can reference a security group from the peer VPC as a source or destination for ingress or egress rules in your security group rules.
5. By default, if instances on either side of a VPC peering connection address each other using a public DNS hostname, the hostname resolves to the instance's public IP address. To change this behavior, enable DNS hostname resolution for your VPC connection. After enabling DNS hostname resolution, if instances on either side of the VPC peering connection address each other using a public DNS hostname, the hostname resolves to the private IP address of the instance.

**VPC Peering Connection Lifecycle**

A VPC peering connection goes through various stages starting from when the request is initiated. At each stage, there may be actions that you can take, and at the end of its lifecycle, the VPC peering connection remains visible in the Amazon VPC console and API or command line output for a period of time.

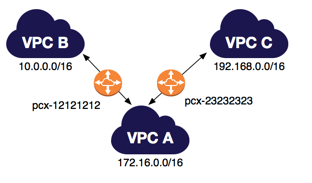


* **Initiating-request**: A request for a VPC peering connection has been initiated. At this stage, the peering connection may fail or may go to pending-acceptance.
* **Failed**: The request for the VPC peering connection has failed. During this state, it cannot be accepted, rejected, or deleted. The failed VPC peering connection remains visible to the requester for 2 hours.
* **Pending-acceptance**: The VPC peering connection request is awaiting acceptance from the owner of the accepter VPC. During this state, the owner of the requester VPC can delete the request, and the owner of the accepter VPC can accept or reject the request. If no action is taken on the request, it expires after 7 days.
* **Expired**: The VPC peering connection request has expired, and no action can be taken on it by either VPC owner. The expired VPC peering connection remains visible to both VPC owners for 2 days.
* **Rejected**: The owner of the accepter VPC has rejected a pending-acceptance VPC peering connection request. During this state, the request cannot be accepted. The rejected VPC peering connection remains visible to the owner of the requester VPC for 2 days, and visible to the owner of the accepter VPC for 2 hours. If the request was created within the same AWS account, the rejected request remains visible for 2 hours.
* **Provisioning**: The VPC peering connection request has been accepted, and will soon be in the activestate.
* **Active**: The VPC peering connection is active, and traffic can flow between the VPCs (provided that your security groups and route tables allow the flow of traffic). During this state, either of the VPC owners can delete the VPC peering connection, but cannot reject it.    
  **Note**   
  If an event in a region in which a VPC resides prevents the flow of traffic, the status of the VPC peering connection remains Active.
* **Deleting**: Applies to an inter-region VPC peering connection that is in the process of being deleted. The owner of either VPC has submitted a request to delete an active VPC peering connection, or the owner of the requester VPC has submitted a request to delete a pending-acceptance VPC peering connection request.
* **Deleted**: An active VPC peering connection has been deleted by either of the VPC owners, or apending-acceptance VPC peering connection request has been deleted by the owner of the requester VPC. During this state, the VPC peering connection cannot be accepted or rejected. The VPC peering connection remains visible to the party that deleted it for 2 hours, and visible to the other party for 2 days. If the VPC peering connection was created within the same AWS account, the deleted request remains visible for 2 hours.

**Multiple VPC Peering Connections**

A VPC peering connection is a one to one relationship between two VPCs. You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported. You do not have any peering relationship with VPCs that your VPC is not directly peered with.

The following diagram is an example of one VPC peered to two different VPCs. There are two VPC peering connections: VPC A is peered with both VPC B and VPC C. VPC B and VPC C are not peered, and you cannot use VPC A as a transit point for peering between VPC B and VPC C. If you want to enable routing of traffic between VPC B and VPC C, you must create a unique VPC peering connection between them.



**Pricing for a VPC Peering Connection**

If the VPCs in the VPC peering connection are within the same region, the charges for transferring data within the VPC peering connection are the same as the charges for transferring data across Availability Zones. If the VPCs are in different regions, inter-region data transfer costs apply.

**VPC Peering Limitations**

To create a VPC peering connection with another VPC, be aware of the following limitations and rules:

* You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks. Amazon always assigns your VPC a unique IPv6 CIDR block. If your IPv6 CIDR blocks are unique but your IPv4 blocks are not, you cannot create the peering connection.
* You have a limit on the number of active and pending VPC peering connections that you can have per VPC. In a VPC peering connection, your VPC does not have access to any other VPCs with which the peer VPC may be peered. This includes VPC peering connections that are established entirely within your own AWS account.
* You cannot have more than one VPC peering connection between the same two VPCs at the same time.
* Unicast reverse path forwarding in VPC peering connections is not supported.
* If the VPCs are in the same region, you can enable the resources on either side of a VPC peering connection to communicate with each other over IPv6. IPv6 communication is not automatic. You must associate an IPv6 CIDR block with each VPC, enable the instances in the VPCs for IPv6 communication, and add routes to your route tables that route IPv6 traffic intended for the peer VPC to the VPC peering connection.
* Any tags that you create for your VPC peering connection are only applied in the account or region in which you create them.
* If the IPv4 CIDR block of a VPC in a VPC peering connection falls outside of the private IPv4 address ranges specified by [RFC 1918](http://www.faqs.org/rfcs/rfc1918.html), private DNS hostnames for that VPC cannot be resolved to private IP addresses. To resolve private DNS hostnames to private IP addresses, you can enable DNS resolution support for the VPC peering connection.
* You cannot connect to or query the Amazon DNS server in a peer VPC.

An inter-region VPC peering connection has additional limitations:

* You cannot create a security group rule that references a peer VPC security group.
* You cannot enable support for an EC2-Classic instance that's linked to a VPC via ClassicLink to communicate with the peer VPC.
* Communication over IPv6 is not supported.
* The Maximum Transmission Unit (MTU) across the VPC peering connection is 1500 bytes (jumbo frames are not supported).
* You must enable DNS resolution support for the VPC peering connection to resolve private DNS hostnames of the peered VPC to private IP addresses, even if the IPv4 CIDR for the VPC falls into the private IPv4 address ranges specified by RFC 1918.
* Inter-region peering in China is only allowed between the China (Beijing) Region, operated by SINNET and the China (Ningxia) Region, operated by NWCD.

**(AWS CLI)-Create VPC Peering connections**

**To create a VPC peering connection between your VPCs**

This example requests a peering connection between your VPCs vpc-1a2b3c4d and vpc-11122233.

Command:

aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233

**To create a VPC peering connection with a VPC in another account**

This example requests a peering connection between your VPC (vpc-1a2b3c4d), and a VPC (vpc-11122233) that belongs AWS account 123456789012.

Command:

aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-owner-id 123456789012

**To create a VPC peering connection with a VPC in a different region**

This example requests a peering connection between your VPC in the current region (vpc-1a2b3c4d), and a VPC (vpc-11122233) in your account in the us-west-2 region.

Command:

aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-region us-west-2

This example requests a peering connection between your VPC in the current region (vpc-1a2b3c4d), and a VPC (vpc-11122233) that belongs AWS account 123456789012 that's in the us-west-2 region.

Command:

aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-owner-id 123456789012 --peer-region us-west-2

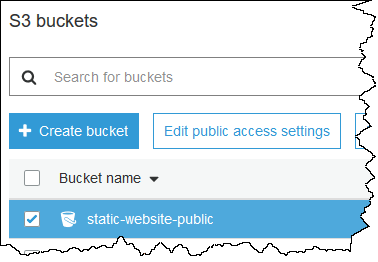
**Step 1: Creating a Bucket and Configuring It as a Website**

1. Sign in to the AWS Management Console and open the Amazon S3 console at [https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3).
2. Create a bucket. Name jjtechinc-your-last-name
3. Open the bucket **Properties** pane, choose **Static Website Hosting**, and do the following:
   1. Choose **Use this bucket to host a website**.
   2. In the **Index Document** box, type the name of your index document. The name is typicallyindex.html.
   3. Choose **Save** to save the website configuration.
   4. Write down the **Endpoint**.   
      This is the Amazon S3-provided website endpoint for your bucket. You use this endpoint in the following steps to test your website.

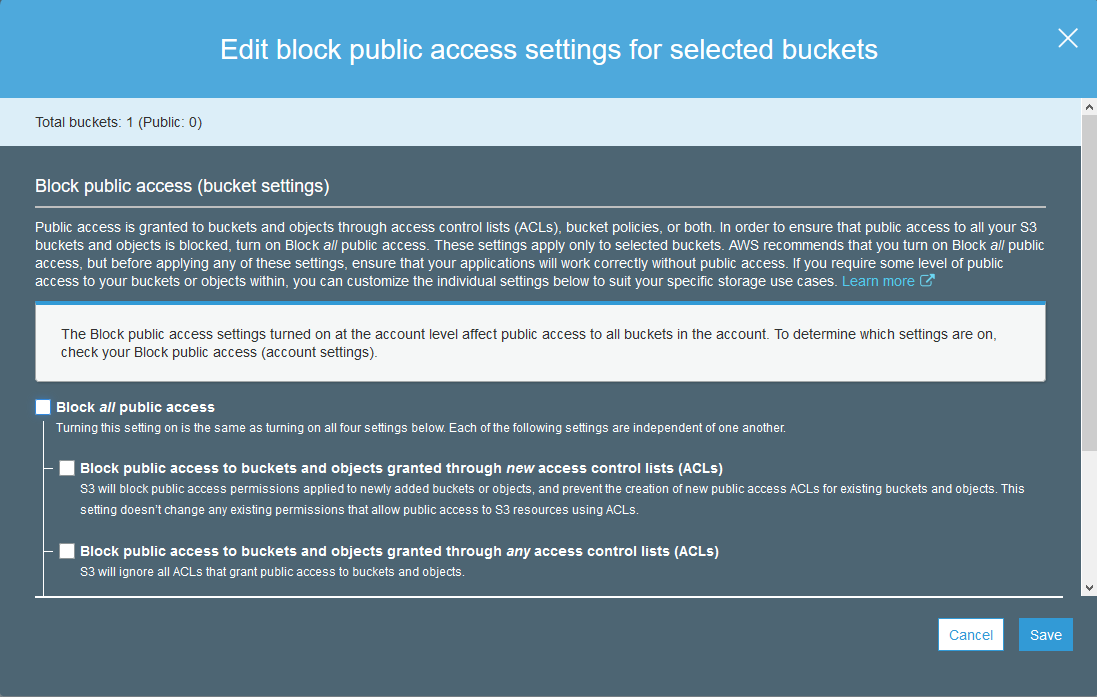
**Editing Block Public Access Settings**

By default, Amazon S3 blocks public access to your account and buckets. If you want to use a bucket to host a static website, you can use these steps to edit block public access settings:

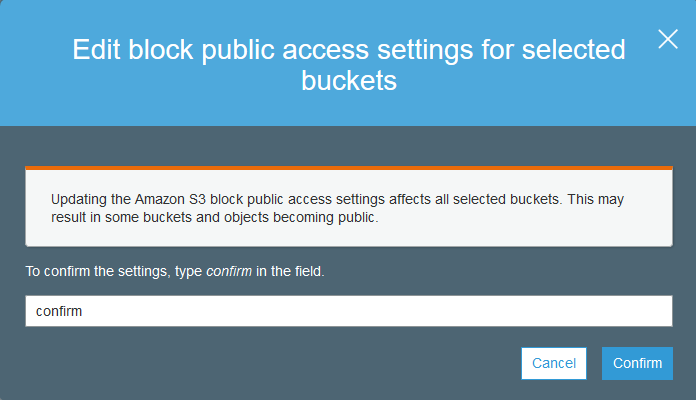
1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the bucket that you have configured as a static website, and choose **Edit public access settings.**



1. Clear **Block *all* public access**, and choose **Save**.



1. In the confirmation text box, enter **confirm**, and then choose **Confirm**.



In your Amazon S3 bucket listing, the **Access** for your bucket updates to **Objects can be public**. You can now add a bucket policy to make the objects in the bucket publicly readable.

**Step 2: Adding a Bucket Policy That Makes Your Bucket Content Publicly Available**

1. In the **Properties** pane for the bucket, choose **Permissions**.
2. Choose **Bucket Policy**.
3. To grant public read access for your website, copy the following bucket policy, paste it in the **Bucket policy editor**, and choose **Save**.    
      
   {   
      "Version":"2012-10-17",   
      "Statement":[{   
   "Sid":"PublicReadForGetBucketObjects",   
            "Effect":"Allow",   
     "Principal": "\*",   
          "Action":["s3:GetObject"],   
          "Resource":["arn:aws:s3:::*example-bucket*/\*"   
          ]   
        }   
      ]   
   }
4. In the policy, replace *example-bucket* with the name of your bucket.
5. Choose **Save**.   
   In your Amazon S3 bucket listing, the **Access** for your bucket updates to **Public**.

**Step 3: Uploading an Index Document**

1. Create a document. Give it the same name that you gave the index document earlier.
2. Using the console, upload the index document to your bucket.

**Step 4: Testing Your Website**

Enter the following URL in the browser, replacing *example-bucket* with the name of your bucket and *website-region* with the name of the AWS Region where you deployed your bucket.

The two general forms of an Amazon S3 website endpoint are as follows:

[http://*example-bucket*.s3-website-*region*.amazonaws.com](http://example-bucket.s3-website-region.amazonaws.com/)

[http://*example-bucket*.s3-website.*region*.amazonaws.com](http://example-bucket.s3-website.region.amazonaws.com/)

If your browser displays your index.html page, the website was successfully deployed.

**Index.html**

<!DOCTYPE html>

<html>

  <body>

    This is JJTech batch 2 hosting it's first static website! The students include Charles, Christian, Cliford, Collins, Elie, Elvis, Emmanuel, Eva, Lister, Mbangi, Ransom, Ntap, Roland, Ruddy, Senge, Suzzy, Viban, Yves and Yvonne! They are being trained as full stack Engineers with average earnings of $150k per annum!!

  </body>

</html>

Q1) What is AWS?

Answer:AWS stands for Amazon Web Services. AWS is a platform that provides on-demand resources for hosting web services, storage, networking, databases and other resources over the internet with a pay-as-you-go pricing.

Q2)  What are the components of AWS?

Answer:EC2 – Elastic Compute Cloud, S3 – Simple Storage Service, Route53, EBS – Elastic Block Store, Cloudwatch, management services, developer tools, etc.

Q3)  What are key-pairs?

Answer:Key-pairs are secure login information for your instances/virtual machines. To connect to the instances we use key-pairs that contain a public-key and private-key.

Q4)  What is S3?

Answer:S3 stands for Simple Storage Service. It is a storage service that provides an interface that you can use to store any amount of data, at any time, from anywhere in the world. With S3 you pay only for what you use and the payment model is pay-as-you-go.

Q5)  What are the pricing models for EC2instances?

Answer:The different pricing model for EC2 instances are as below,

* On-demand
* Reserved
* Spot
* Scheduled
* Dedicated

Q6) What are the types of volumes for EC2 instances?

Answer:

* There are two types of volumes,
* Instance store volumes
* EBS – Elastic Block Stores

Q7) What are EBS volumes?

Answer:EBS stands for Elastic Block Stores. They are persistent volumes that you can attach to the instances. With EBS volumes, your data will be preserved even when you stop your instances, unlike your instance store volumes where the data is deleted when you stop the instances.

Q8) What are the types of volumes in EBS?

Answer:Following are the types of volumes in EBS,

* General purpose
* Provisioned IOPS
* Magnetic
* Cold HDD
* Throughput optimized

Q9) What are the different types of instances?

Answer: Following are the types of instances,

* General purpose
* Computer Optimized
* Storage Optimized
* Memory Optimized
* Accelerated Computing

Q10) What is an auto-scaling and what are the components?

Answer: Auto scaling allows you to automatically scale-up and scale-down the number of instances depending on the CPU utilization or memory utilization. There are 2 components in Auto scaling, they are Auto-scaling groups and Launch Configuration.

Q11) What are reserved instances?

Answer: Reserved instances are the instance that you can reserve a fixed capacity of EC2 instances. In reserved instances you will have to get into a contract of 1 year or 3 years.

Q12)What is an AMI?

Answer: AMI stands for Amazon Machine Image. AMI is a template that contains the software configurations, launch permission and a block device mapping that specifies the volume to attach to the instance when it is launched.

Q13) What is an EIP?

Answer: EIP stands for Elastic IP address. It is designed for dynamic cloud computing. When you want to have a static IP address for your instances when you stop and restart your instances, you will be using EIP address.

Q14) What is Cloudwatch?

Answer: Cloudwatch is a monitoring tool that you can use to monitor your various AWS resources. Like health check, network, Application, etc.

Q15) What are the types in cloudwatch?

Answer: There are 2 types in cloudwatch. Basic monitoring and detailed monitoring. Basic monitoring is free and detailed monitoring is chargeable.

Q16) What are the cloudwatch metrics that are available for EC2 instances?

Answer: Diskreads, Diskwrites, CPU utilization, networkpacketsIn, networkpacketsOut, networkIn, networkOut, CPUCreditUsage, CPUCreditBalance.

Q17) What is the minimum and maximum size of individual objects that you can store in S3

Answer: The minimum size of individual objects that you can store in S3 is 0 bytes and the maximum bytes that you can store for individual objects is 5TB.

Q18) What are the different storage classes in S3?

Answer: Following are the types of storage classes in S3,

* Standard frequently accessed
* Standard infrequently accessed
* One-zone infrequently accessed.
* Glacier
* RRS – reduced redundancy storage

 Q19) What is the default storage class in S3?

Answer: The default storage class in S3 in Standard frequently accessed.

 Q20) What is glacier?

Answer: Glacier is the back up or archival tool that you use to back up your data in S3.

 Q21) How can you secure the access to your S3 bucket?

Answer: There are two ways that you can control the access to your S3 buckets,

* ACL – Access Control List
* Bucket polices

 Q22) How can you encrypt data in S3?

Answer: You can encrypt the data by using the below methods,

* Server Side Encryption – S3 (AES 256 encryption)
* Server Side Encryption – KMS (Key management Service)
* Server Side Encryption – C (Client Side)

 Q23) What are the parameters for S3 pricing?

Answer: The pricing model for S3 is as below,

* Storage used
* Number of requests you make
* Storage management
* Data transfer
* Transfer acceleration

 Q24) What is the pre-requisite to work with Cross region replication in S3?

Answer: You need to enable versioning on both source bucket and destination to work with cross region replication. Also both the source and destination bucket should be in different region.

 Q25) What are roles?

Answer: Roles are used to provide permissions to entities that you trust within your AWS account. Roles are users in another account. Roles are similar to users but with roles you do not need to create any username and password to work with the resources.

 Q26) What are policies and what are the types of policies?

Answer: Policies are permissions that you can attach to the users that you create. These policies will contain that access that you have provided to the users that you have created. There are 2 types of policies.

* Managed policies
* Inline policies

 Q27) What is cloudfront?

Answer: Cloudfront is an AWS web service that provided businesses and application developers an easy and efficient way to distribute their content with low latency and high data transfer speeds. Cloudfront is content delivery network of AWS.

 Q28) What are edge locations?

Answer: Edge location is the place where the contents will be cached. When a user tries to access some content, the content will be searched in the edge location. If it is not available then the content will be made available from the origin location and a copy will be stored in the edge location.

Q29) What is the maximum individual archive that you can store in glacier?

Answer: You can store a maximum individual archive of upto 40 TB.

 Q30) What is VPC?

Answer: VPC stands for Virtual Private Cloud. VPC allows you to easily customize your networking configuration. VPC is a network that is logically isolated from other network in the cloud. It allows you to have your own IP address range, subnets, internet gateways, NAT gateways and security groups.

 Q31) What is VPC peering connection?

Answer: VPC peering connection allows you to connect 1 VPC with another VPC. Instances in these VPC behave as if they are in the same network.

 Q32) What are NAT gateways?

Answer: NAT stands for Network Address Translation. NAT gateways enables instances in a private subnet to connect to the internet but prevent the internet from initiating a connection with those instances.

 Q33) How can you control the security to your VPC?

Answer: You can use security groups and NACL (Network Access Control List) to control the security to your

VPC.

 Q34) What are the different types of storage gateway?

Answer: Following are the types of storage gateway.

* File gateway
* Volume gateway
* Tape gateway

 Q35) What is a snowball?

Answer: Snowball is a data transport solution that used source appliances to transfer large amounts of data into and out of AWS. Using snowball, you can move huge amount of data from one place to another which reduces your network costs, long transfer times and also provides better security.

 Q36) What are the database types in RDS?

Answer: Following are the types of databases in RDS,

* Aurora
* Oracle
* MYSQL server
* Postgresql
* MariaDB
* SQL server

 Q37) What is a redshift?

Answer: Amazon redshift is a data warehouse product. It is a fast and powerful, fully managed, petabyte scale data warehouse service in the cloud.

 Q38) What is SNS?

Answer: SNS stands for Simple Notification Service. SNS is a web service that makes it easy to notifications from the cloud. You can set up SNS to receive email notification or message notification.

 Q39) What are the types of routing polices in route53?

Answer: Following are the types of routing policies in route53,

* Simple routing
* Latency routing
* Failover routing
* Geolocation routing
* Weighted routing
* Multivalue answer

 Q40) What is the maximum size of messages in SQS?

Answer: The maximum size of messages in SQS is 256 KB.

Q41) What are the types of queues in SQS?

Answer: There are 2 types of queues in SQS.

* Standard queue
* FIFO (First In First Out)

 Q42) What is multi-AZ RDS?

Answer: Multi-AZ (Availability Zone) RDS allows you to have a replica of your production database in another availability zone. Multi-AZ (Availability Zone) database is used for disaster recovery. You will have an exact copy of your database. So when your primary database goes down, your application will automatically failover to the standby database.

Q43) What are the types of backups in RDS database?

Answer: There are 2 types of backups in RDS database.

* Automated backups
* Manual backups which are known as snapshots.

 Q44) What is the difference between security groups and network access control list?

Answer:

|  |  |
| --- | --- |
| Security Groups | Network access control list |
| Can control the access at the instance level | Can control access at the subnet level |
| Can add rules for “allow” only | Can add rules for both “allow” and “deny” |
| Evaluates all rules before allowing the traffic | Rules are processed in order number when allowing traffic. |
| Can assign unlimited number of security groups | Can assign upto 5 security groups. |
| Statefull filtering | Stateless filtering |

 Q45) What are the types of load balancers in EC2?

Answer: There are 3 types of load balancers,

* Application load balancer
* Network load balancer
* Classic load balancer

 Q46) What is and ELB?

Answer: ELB stands for Elastic Load balancing. ELB automatically distributes the incoming application traffic or network traffic across multiple targets like EC2, containers, IP addresses.

 Q47) What are the two types of access that you can provide when you are creating users?

Answer: Following are the two types of access that you can create.

* Programmatic access
* Console access

 Q48) What are the benefits of auto scaling?

Answer: Following are the benefits of auto scaling

* Better fault tolerance
* Better availability
* Better cost management

 Q49) What are security groups?

Answer: Security groups acts as a firewall that contains the traffic for one or more instances. You can associate one or more security groups to your instances when you launch then. You can add rules to each security group that allow traffic to and from its associated instances. You can modify the rules of a security group at any time, the new rules are automatically  and immediately applied to all the instances that are associated with the security group

Q50) What are shared AMI’s?

Answer: Shared AMI’s are the AMI that are created by other developed and made available for other developed to use.

 Q51)What is the difference between the classic load balancer and application load balancer?

Answer: Dynamic port mapping, multiple port multiple listeners is used in Application Load Balancer, One port one listener is achieved via Classic Load Balancer

 Q52) By default how many Ip address does aws reserve in a subnet?

Answer: 5

 Q53) What is meant by subnet?

Answer: A large section of IP Address divided in to chunks are known as subnets

 Q54) How can you convert a public subnet to private subnet?

Answer: Remove IGW & add NAT Gateway, Associate subnet in Private route table

 Q55) Is it possible to reduce a ebs volume?

Answer: no it’s not possible, we can increase it but not reduce them

 Q56) What is the use of elastic ip are they charged by AWS?

Answer: These are ipv4 address which are used to connect the instance from internet, they are charged if the instances are not attached to it

 Q57) One of my s3 in a bucket is deleted but i need to restore is there any possible way?

Answer: If versioning is enabled we can easily restore them

 Q58) When I try to launch an ec2 instance i am getting Service limit exceed, how to fix the issue?

Answer: By default AWS offer service limit of 20 running instances per region, to fix the issue we need to contact AWS support to increase the limit based on the requirement

 Q59) I need to modify the ebs volumes in Linux and windows is it possible

Answer: yes its possible from console use modify volumes in section give the size u need then for windows go to disk management for Linux mount it to achieve the modification

 Q60) Is it possible to stop a RDS instance, how can I do that?

Answer: Yes it’s possible to stop rds. Instance which are non-production and non multi AZ’s

 Q61) What is meant by parameter groups in rds. And what is the use of it?

Answer: Since RDS is a managed service AWS offers a wide set of parameter in RDS as parameter group which is modified as per requirement

 Q62) What is the use of tags and how they are useful?

Answer: Tags are used for identification and grouping AWS Resources

 Q63) I am viewing an AWS Console but unable to launch the instance, I receive an IAM Error how can I rectify it?

Answer: As AWS user I don’t have access to use it, I need to have permissions to use it further

 Q64) I don’t want my AWS Account id to be exposed to users how can I avoid it?

Answer: In IAM console there is option as sign in url where I can rename my own account name with AWS account

 Q65) By default how many Elastic Ip address does AWS Offer?

Answer: 5 elastic ip per region

 Q66) You are enabled sticky session with ELB. What does it do with your instance?

Answer: Binds the user session with a specific instance

 Q67) Which type of load balancer makes routing decisions at either the transport layer or the

Application layer and supports either EC2 or VPC.

Answer: Classic Load Balancer

 Q68) Which is virtual network interface that you can attach to an instance in a VPC?

Answer: Elastic Network Interface

 Q69) You have launched a Linux instance in AWS EC2. While configuring security group, you

Have selected SSH, HTTP, HTTPS protocol. Why do we need to select SSH?

Answer: To verify that there is a rule that allows traffic from EC2 Instance to your computer

 Q70) You have chosen a windows instance with Classic and you want to make some change to the

Security group. How will these changes be effective?

Answer: Changes are automatically applied to windows instances

 Q71) Load Balancer and DNS service comes under which type of cloud service?

Answer: IAAS-Storage

 Q72) You have an EC2 instance that has an unencrypted volume. You want to create another

 Encrypted volume from this unencrypted volume. Which of the following steps can achieve this?

Answer: Create a snapshot of the unencrypted volume (applying encryption parameters), copy the. Snapshot and create a volume from the copied snapshot

 Q73) Where does the user specify the maximum number of instances with the auto scaling Commands?

Answer: Auto scaling Launch Config

 Q74) Which are the types of AMI provided by AWS?

Answer: Instance Store backed, EBS Backed

 Q75) After configuring ELB, you need to ensure that the user requests are always attached to a Single instance. What setting can you use?

Answer:  Sticky session

 Q76) When do I prefer to Provisioned IOPS over the Standard RDS storage?

Answer:If you have do batch-oriented is workloads.

Q77) If I am running on my DB Instance a Multi-AZ deployments, can I use to the stand by the DB Instance for read or write a operation along with to primary DB instance?

Answer: Primary db instance does not working.

Q78) Which the AWS services will you use to the collect and the process e-commerce data for the near by real-time analysis?

Answer:  Good of Amazon DynamoDB.

Q79) A company is deploying the new two-tier an web application in AWS. The company has to limited on staff and the requires high availability, and the application requires to complex queries and table joins. Which configuration provides to the solution for company’s requirements?

Answer: An web application provide on Amazon DynamoDB solution.

Q80) Which the statement use to cases are suitable for Amazon DynamoDB?

Answer:The storing metadata for the Amazon S3 objects& The Running of relational joins and complex an updates.

Q81) Your application has to the retrieve on data from your user’s mobile take every 5 minutes and then data is stored in the DynamoDB, later every day at the particular time the data is an extracted into S3 on a per user basis and then your application is later on used to visualize the data to user. You are the asked to the optimize the architecture of the backend system can to lower cost, what would you recommend do?

Answer: Introduce Amazon Elasticache to the cache reads from the Amazon DynamoDB table and to reduce the provisioned read throughput.

Q82) You are running to website on EC2 instances can deployed across multiple Availability Zones with an Multi-AZ RDS MySQL Extra Large DB Instance etc. Then site performs a high number of the small reads and the write per second and the relies on the eventual consistency model. After the comprehensive tests you discover to that there is read contention on RDS MySQL. Which is the best approaches to the meet these requirements?

Answer:The Deploy Elasti Cache in-memory cache is  running in each availability zone and Then Increase the RDS MySQL Instance size and the Implement provisioned IOPS.

Q83) An startup is running to a pilot deployment of around 100 sensors to the measure street noise and The air quality is urban areas for the 3 months. It was noted that every month to around the 4GB of sensor data are generated. The company uses to a load balanced take auto scaled layer of the EC2 instances and a RDS database with a 500 GB standard storage. The pilot was success and now they want to the deploy take atleast 100K sensors.let which to need the supported by backend. You need to the stored data for at least 2 years to an analyze it. Which setup of  following would you be prefer?

Answer: The Replace the RDS instance with an 6 node Redshift cluster with take 96TB of storage.

Q84) Let to Suppose you have an application where do you have to render images and also do some of general computing. which service will be best fit your need?

Answer:Used on Application Load Balancer.

Q85) How will change the instance give type for the instances, which are the running in your applications tier and Then using Auto Scaling. Where will you change it from areas?

Answer: Changed to Auto Scaling launch configuration areas.

Q86) You have an content management system running on the Amazon EC2 instance that is the approaching 100% CPU of utilization. Which option will be reduce load on the Amazon EC2 instance?

Answer: Let Create a load balancer, and Give register the Amazon EC2 instance with it.

Q87) What does the Connection of draining do?

Answer: The re-routes traffic from the instances which are to be updated (or) failed an health to check.

Q88) When the instance is an unhealthy, it is do terminated and replaced with an new ones, which of the services does that?

Answer: The survice make a fault tolerance.

Q89) What are the life cycle to hooks used for the AutoScaling?

Answer: They are used to the  put an additional taken wait time to the scale in or scale out events.

Q90) An user has to setup an Auto Scaling group. Due to some issue the group has to failed for launch a single instance for the more than 24 hours. What will be happen to the Auto Scaling in the condition?

Answer: The auto Scaling will be suspend to the scaling process.

Q91) You have an the EC2 Security Group with a several running to EC2 instances. You changed to the Security of Group rules to allow the inbound traffic on a new port and protocol, and then the launched a several new instances in the same of Security Group.Such the new rules apply?

Answer:The Immediately to all the instances in security groups.

Q92) To create an mirror make a image of your environment in another region for the disaster recoverys, which of the following AWS is resources do not need to be recreated in second region?

Answer: May be the selected on Route 53 Record Sets.

Q93) An customers wants to the captures all client connections to get information from his load balancers at an interval of 5 minutes only, which cal select option  should he choose for his application?

Answer: The condition should be Enable to AWS CloudTrail for the loadbalancers.

Q94) Which of the services to you would not use to deploy an app?

Answer: Lambda app not used on deploy.

Q95) How do the Elastic Beanstalk can apply to updates?

Answer: By a duplicate ready with a updates prepare before swapping.

Q96) An created a key in the oregon region to encrypt of my data in North Virginia region for security purposes. I added to two users to the key and the external AWS accounts. I wanted to encrypt an the object in S3, so when I was tried, then key that I just created is not listed.What could be reason&solution?

Answer:The Key should be working in the same region.

Q97) As a company needs to monitor a read and write IOPS for the AWS MySQL RDS instances and then send real-time alerts to the operations of team. Which AWS services to can accomplish this?

Answer:The monitoring on Amazon CloudWatch

Q98) The organization that is currently using the consolidated billing has to recently acquired to another company that already has a number of the AWS accounts. How could an Administrator to ensure that all the AWS accounts, from the both existing company and then acquired company, is billed to the single account?

Answer: All Invites take acquired the company’s AWS account to join  existing the company’s of organization by using AWS Organizations.

Q99) The user has created an the applications, which will be hosted on the EC2. The application makes calls to the Dynamo DB to fetch on certain data. The application using the DynamoDB SDK to connect with  the EC2 instance. Which of  respect to  best practice for the security in this scenario?

Answer: The user should be attach an IAM roles with the DynamoDB access to  EC2 instance.

Q100) You have an application are running on EC2 Instance, which will allow users to download the files from a private S3 bucket using the pre-assigned URL. Before generating to URL the Q101) application should be verify the existence of file in S3. How do the application use the AWS credentials to access  S3 bucket securely?

Answer:An  Create an IAM role for the EC2 that allows list access to objects in  S3 buckets. Launch to instance with this role, and retrieve an role’s credentials from  EC2 Instance make metadata.

Q101) You use the Amazon CloudWatch as your primary monitoring system for web application. After a recent to software deployment, your users are to getting Intermittent the 500 Internal Server to the Errors, when you using web application. You want to create the CloudWatch alarm, and notify the on-call engineer let when these occur. How can you accomplish the using the AWS services?

Answer: An Create a CloudWatch get Logs to group and A define metric filters that assure capture 500 Internal Servers should  be  Errors. Set a CloudWatch alarm on the metric and By Use of  Amazon Simple to create a Notification Service to notify an the on-call engineers when prepare CloudWatch alarm is triggered.

Q102) You are designing a multi-platform of web application for the AWS. The application will run on the EC2 instances and Till will be accessed from PCs, tablets and smart phones.Then Supported accessing a platforms are Windows, MACOS, IOS and Android. They Separate sticky sessions and SSL certificate took setups are required for the different platform types. Which do describes the most cost effective and Like performance efficient the architecture setup?

Answer:Assign to multiple ELBs  an EC2 instance or group of EC2 take instances running to common component  of the web application, one ELB change  for each platform type.Take Session will be stickiness and SSL termination are done for the ELBs.

Q103) You are migrating to legacy client-server application for AWS. The application responds to a specific DNS visible domain (e.g. [www.example.com](http://www.example.com/)) and server 2-tier architecture, with multiple application for the servers and the database server. Remote clients use to TCP to connect to the application of servers. The application servers need to know the IP address of clients in order to  the function of properly and are currently taking of that information from  TCP socket. A Multi-AZ RDS MySQL instance to will be used for database. During the migration you  change the application code but you have file a change request. How do would you implement the architecture on the AWS in order to maximize scalability and high availability?

Answer: File a change request to get implement of Proxy Protocol support in the application. Use of ELB with TCP Listener and A Proxy Protocol enabled to distribute the  load on two application servers in the different AZs.

Q104) Your application currently is leverages AWS Auto Scaling to the grow and shrink as a load Increases/decreases and has been performing as well. Your marketing a team expects and steady ramp up in traffic to follow an upcoming campaign that will result in 20x growth in the traffic over 4 weeks. Your forecast for approximate number of the Amazon EC2 instances necessary to meet  peak demand is 175. What should be you do  avoid potential service disruptions during the ramp up traffic?

Answer: Check the service limits in the Trusted Advisors and adjust as necessary, so that forecasted count remains within  the limits.

Q105) You have a web application running on the six Amazon EC2 instances, consuming about 45% of resources on the each instance. You are using the auto-scaling to make sure that a six instances are running at all times. The number of requests this application processes to consistent and does not experience to spikes. Then application are critical to your business and you want to high availability for at all times. You want to the load be distributed evenly has between all instances. You also want to between use same Amazon Machine Image (AMI) for all instances. Which are  architectural choices should you make?

Answer: Deploy  to 3 EC2 instances in one  of availability zone and 3 in another availability of zones and to use of Amazon Elastic is Load Balancer.

Q106) You are the designing an application that a contains protected health information. Security and Then compliance requirements for your application mandate that all protected to health information in application use to encryption at rest and in the transit module. The application to uses an three-tier architecture. where should data flows through the load balancers and is stored on the Amazon EBS volumes for the processing, and the results are stored in the Amazon S3 using a AWS SDK. Which of the options satisfy the security requirements?

Answer: Use TCP load balancing on load balancer system, SSL termination on Amazon to create EC2 instances, OS-level disk  take encryption on Amazon EBS volumes, and The amazon S3 with server-side to encryption and Use the SSL termination on load balancers, an SSL listener on the Amazon to create EC2 instances, Amazon EBS encryption on the EBS volumes containing the PHI, and Amazon S3 with a server-side of encryption.

Q107) An startup deploys its create photo-sharing site in a VPC. An elastic load balancer distributes to web traffic across two the subnets. Then the load balancer session to stickiness is configured to use of AWS-generated session cookie, with a session TTL of the 5 minutes. The web server to change Auto Scaling group is configured as like min-size=4, max-size=4. The startup is the preparing for a public launchs, by running the load-testing software installed on the single Amazon Elastic Compute Cloud (EC2) instance to running in us-west-2a. After 60 minutes of load-testing, the web server logs of show the following:WEBSERVER LOGS | # of HTTP requests to from load-tester system | # of HTTP requests  to from private on beta users || webserver #1 (subnet an us-west-2a): | 19,210 | 434 | webserver #2 (subnet an us-west-2a): | 21,790 | 490 || webserver #3 (subnet an us-west-2b): | 0 | 410 || webserver #4 (subnet an us-west-2b): | 0 | 428 |Which as recommendations can be help of  ensure that load-testing HTTP requests are will evenly distributed across to four web servers?

Answer:Result of cloud is re-configure the load-testing software to the re-resolve DNS for each web request.

Q108) To serve the Web traffic for a popular product to your chief financial officer and IT director have purchased 10 m1.large heavy utilization of Reserved Instances (RIs) evenly put spread across two availability zones: Route 53 are used to deliver the traffic to on Elastic Load Balancer (ELB). After the several months, the product grows to even more popular and you need to additional capacity As a result, your company that purchases two c3.2xlarge medium utilization RIs You take register the two c3.2xlarge instances on with your ELB and quickly find that the ml of large instances at 100% of capacity and the c3.2xlarge instances have significant to capacity that’s can unused Which option is the most of cost effective and uses EC2 capacity most of effectively?

Answer: To use a separate ELB for the each instance type and the distribute load to ELBs with a Route 53 weighted round of  robin.

Q109) An AWS customer are deploying an web application that is the composed of a front-end running on the Amazon EC2 and confidential data that are stored on the Amazon S3. The customer security policy is that all accessing operations to this sensitive data must authenticated and authorized by centralized access to management system that is operated by separate security team. In addition, the web application team that be owns and administers the EC2 web front-end instances are prohibited from having the any ability to access data that circumvents this centralized access to management system. Which are configurations will support these requirements?

Answer:The configure to the web application get authenticate end-users against the centralized access on the  management system. Have a web application provision trusted to users STS tokens an entitling the download of the approved data directly from a Amazon S3.

Q110) A Enterprise customer is starting on their migration to the cloud, their main reason for the migrating is agility and they want to the make their internal Microsoft active directory available to the many applications running on AWS, this is so internal users for only have to remember one set of the credentials and as a central point of user take control for the leavers and joiners. How could they make their actions the directory secures and the highly available with minimal on-premises on infrastructure changes in the most cost and the time-efficient way?

Answer: By Using a VPC, they could be create an the extension to their data center and to  make use of resilient hardware IPSEC on tunnels, they could then have two domain consider to controller instances that are joined to the existing domain and reside within the different subnets in the different availability zones.

Q111)What is Cloud Computing?

Answer:Cloud computing means it provides services to access programs, application, storage, network, server over the internet through browser or client side application on your PC, Laptop, Mobile by the end user without installing, updating and maintaining them.

Q112)Why we go for Cloud Computing?

Answer:

* Lower computing cost
* Improved Performance
* No IT Maintenance
* Business connectivity
* Easily upgraded
* Device Independent

Q113)What are the deployment models using in Cloud?

Answer:

* Private Cloud
* Public Cloud
* Hybrid cloud
* Community cloud 4

Q114)Explain Cloud Service Models?

Answer: SAAS (Software as a Service): It is software distribution model in which application are hosted by a vendor over the internet for the end user freeing from complex software and hardware management. (Ex: Google drive, drop box)

PAAS (Platform as a Service): It provides platform and environment to allow developers to build applications. It frees developers without going into the complexity of building and maintaining the infrastructure. (Ex: AWS Elastic Beanstalk, Windows Azure)

IAAS (Infrastructure as a Service): It provides virtualized computing resources over the internet like cpu, memory, switches, routers, firewall, Dns, Load balancer (Ex: Azure, AWS)

Q115)What are the advantage of Cloud Computing?

Answer:

* Pay per use
* Scalability
* Elasticity
* High Availability
* Increase speed and Agility
* Go global in Minutes

Q116)What is AWS?

Answer: Amazon web service is a secure cloud services platform offering compute, power, database, storage, content delivery and other functionality to help business scale and grow.

AWS is fully on-demand

AWS is Flexibility, availability and Scalability

AWS is Elasticity: scale up and scale down as needed.

Q117)What is mean by Region, Availability Zone and Edge Location?

Answer: Region: An independent collection of AWS resources in a defined geography. A collection of Data centers (Availability zones). All availability zones in a region connected by high bandwidth.

Availability Zones: An Availability zone is a simply a data center. Designed as independent failure zone. High speed connectivity, Low latency.

Edge Locations: Edge location are the important part of AWS Infrastructure. Edge locations are CDN endpoints for cloud front to deliver content to end user with low latency

Q118)How to access AWS Platform?

Answer:

* AWS Console
* AWS CLI (Command line interface)
* AWS SDK (Software Development Kit)

Q119)What is EC2? What are the benefits in EC2?

Amazon Elastic compute cloud is a web service that provides resizable compute capacity in the cloud.AWS EC2 provides scalable computing capacity in the AWS Cloud. These are the virtual servers also called as an instances. We can use the instances pay per use basis.

Benefits:

* Easier and Faster
* Elastic and Scalable
* High Availability
* Cost-Effective

Q120)What are the pricing models available in AWS EC2?

Answer:

* On-Demand Instances
* Reserved Instances
* Spot Instances
* Dedicated Host

Q121)What are the types using in AWS EC2?

Answer:

* General Purpose
* Compute Optimized
* Memory optimized
* Storage Optimized
* Accelerated Computing (GPU Based)

Q122)What is AMI? What are the types in AMI?

Answer:

Amazon machine image is a special type of virtual appliance that is used to create a virtual machine within the amazon Elastic compute cloud. AMI defines the initial software that will be in an instance when it is launched.

Types of AMI:

* Published by AWS
* AWS Marketplace
* Generated from existing instances
* Uploaded virtual server

Q123)How to Addressing AWS EC2 instances?

Answer:

* Public Domain name system (DNS) name: When you launch an instance AWS creates a DNS name that can be used to access the
* Public IP: A launched instance may also have a public ip address This IP address assigned from the address reserved by AWS and cannot be specified.
* Elastic IP: An Elastic IP Address is an address unique on the internet that you reserve independently and associate with Amazon EC2 instance. This IP Address persists until the customer release it and is not tried to

Q124)What is Security Group?

Answer: AWS allows you to control traffic in and out of your instance through virtual firewall called Security groups. Security groups allow you to control traffic based on port, protocol and source/Destination.

Q125)When your instance show retired state?

Answer:Retired state only available in Reserved instances. Once the reserved instance reserving time (1 yr/3 yr) ends it shows Retired state.

Q126)Scenario: My EC2 instance IP address change automatically while instance stop and start. What is the reason for that and explain solution?

Answer:AWS assigned Public IP automatically but it’s change dynamically while stop and start. In that case we need to assign Elastic IP for that instance, once assigned it doesn’t change automatically.

Q127)What is Elastic Beanstalk?

Answer:AWS Elastic Beanstalk is the fastest and simplest way to get an application up and running on AWS.Developers can simply upload their code and the service automatically handle all the details such as resource provisioning, load balancing, Auto scaling and Monitoring.

Q128)What is Amazon Lightsail?

Answer:Lightsail designed to be the easiest way to launch and manage a virtual private server with AWS.Lightsail plans include everything you need to jumpstart your project a virtual machine, ssd based storage, data transfer, DNS Management and a static ip.

Q129)What is EBS?

Answer:Amazon EBS Provides persistent block level storage volumes for use with Amazon EC2 instances. Amazon EBS volume is automatically replicated with its availability zone to protect component failure offering high availability and durability. Amazon EBS volumes are available in a variety of types that differ in performance characteristics and Price.

Q130)How to compare EBS Volumes?

Answer: Magnetic Volume: Magnetic volumes have the lowest performance characteristics of all Amazon EBS volume types.

EBS Volume size: 1 GB to 1 TB Average IOPS: 100 IOPS Maximum throughput: 40-90 MB

General-Purpose SSD: General purpose SSD volumes offers cost-effective storage that is ideal for a broad range of workloads. General purpose SSD volumes are billed based on the amount of data space provisioned regardless of how much of data you actually store on the volume.

EBS Volume size: 1 GB to 16 TB Maximum IOPS: upto 10000 IOPS Maximum throughput: 160 MB

Provisioned IOPS SSD: Provisioned IOPS SSD volumes are designed to meet the needs of I/O intensive workloads, particularly database workloads that are sensitive to storage performance and consistency in random access I/O throughput. Provisioned IOPS SSD Volumes provide predictable, High performance.

EBS Volume size: 4 GB to 16 TB Maximum IOPS: upto 20000 IOPS Maximum throughput: 320 MB

Q131)What is cold HDD and Throughput-optimized HDD?

Answer: Cold HDD: Cold HDD volumes are designed for less frequently accessed workloads. These volumes are significantly less expensive than throughput-optimized HDD volumes.

EBS Volume size: 500 GB to 16 TB Maximum IOPS: 200 IOPS Maximum throughput: 250 MB

Throughput-Optimized HDD: Throughput-optimized HDD volumes are low cost HDD volumes designed for frequent access, throughput-intensive workloads such as big data, data warehouse.

EBS Volume size: 500 GB to 16 TB Maximum IOPS: 500 IOPS Maximum throughput: 500 MB

Q132)What is Amazon EBS-Optimized instances?

Answer: Amazon EBS optimized instances to ensure that the Amazon EC2 instance is prepared to take advantage of the I/O of the Amazon EBS Volume. An amazon EBS-optimized instance uses an optimized configuration stack and provide additional dedicated capacity for Amazon EBS I/When you select Amazon EBS-optimized for an instance you pay an additional hourly charge for that instance.

Q133)What is EBS Snapshot?

Answer:

* It can back up the data on the EBS Volume. Snapshots are incremental backups.
* If this is your first snapshot it may take some time to create. Snapshots are point in time copies of volumes.

Q134)How to connect EBS volume to multiple instance?

Answer: We can’t able to connect EBS volume to multiple instance, but we can able to connect multiple EBS Volume to single instance.

Q135)What are the virtualization types available in AWS?

Answer: Hardware assisted Virtualization: HVM instances are presented with a fully virtualized set of hardware and they executing boot by executing master boot record of the root block device of the image. It is default Virtualization.

Para virtualization: This AMI boot with a special boot loader called PV-GRUB. The ability of the guest kernel to communicate directly with the hypervisor results in greater performance levels than other  virtualization approaches but they cannot take advantage of hardware extensions such as networking,  GPU etc. Its customized Virtualization image. Virtualization image can be used only for particular service.

Q136)Differentiate Block storage and File storage?

Answer:

Block Storage: Block storage operates at lower level, raw storage device level and manages data as a set of numbered, fixed size blocks.

File Storage: File storage operates at a higher level, the operating system level and manage data as a named hierarchy of files and folders.

Q137)What are the advantage and disadvantage of EFS? Advantages:

Answer:

* Fully managed service
* File system grows and shrinks automatically to petabytes
* Can support thousands of concurrent connections
* Multi AZ replication
* Throughput scales automatically to ensure consistent low latency Disadvantages:
* Not available in all region
* Cross region capability not available
* More complicated to provision compared to S3 and EBS

Q138)what are the things we need to remember while creating s3 bucket?

Answer:

* Amazon S3 and Bucket names are
* This means bucket names must be unique across all AWS
* Bucket names can contain upto 63 lowercase letters, numbers, hyphens and
* You can create and use multiple buckets
* You can have upto 100 per account by

Q139)What are the storage class available in Amazon s3?

Answer:

* Amazon S3 Standard
* Amazon S3 Standard-Infrequent Access
* Amazon S3 Reduced Redundancy Storage
* Amazon Glacier

Q140)Explain Amazon s3 lifecycle rules?

Answer: Amazon S3 lifecycle configuration rules, you can significantly reduce your storage costs by automatically transitioning data from one storage class to another or even automatically delete data after  a period of time.

* Store backup data initially in Amazon S3 Standard
* After 30 days, transition to Amazon Standard IA
* After 90 days, transition to Amazon Glacier
* After 3 years, delete

Q141)What is the relation between Amazon S3 and AWS KMS?

Answer: To encrypt Amazon S3 data at rest, you can use several variations of Server-Side Encryption. Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypt it for you when you access it’ll SSE performed by Amazon S3 and AWS Key Management Service (AWS KMS) uses the 256-bit Advanced Encryption Standard (AES).

Q142)What is the function of cross region replication in Amazon S3?

Answer: Cross region replication is a feature allows you asynchronously replicate all new objects in the source bucket in one AWS region to a target bucket in another region. To enable cross-region replication, versioning must be turned on for both source and destination buckets. Cross region replication is commonly used to reduce the latency required to access objects in Amazon S3

Q143)How to create Encrypted EBS volume?

Answer: You need to select Encrypt this volume option in Volume creation page. While creation a new master key will be created unless you select a master key that you created separately in the service. Amazon uses the AWS key management service (KMS) to handle key management.

Q144)Explain stateful and Stateless firewall.

Answer:

Stateful Firewall: A Security group is a virtual stateful firewall that controls inbound and outbound network traffic to AWS resources and Amazon EC2 instances. Operates at the instance level. It supports allow rules only. Return traffic is automatically allowed, regardless of any rules.

Stateless Firewall: A Network access control List (ACL) is a virtual stateless firewall on a subnet level. Supports allow rules and deny rules. Return traffic must be explicitly allowed by rules.

Q145)What is NAT Instance and NAT Gateway?

Answer:

NAT instance: A network address translation (NAT) instance is an Amazon Linux machine Image (AMI) that is designed to accept traffic from instances within a private subnet, translate the source IP address to the Public IP address of the NAT instance and forward the traffic to IWG.

NAT Gateway: A NAT gateway is an Amazon managed resources that is designed to operate just like a NAT instance but it is simpler to manage and highly available within an availability Zone. To allow instance within a private subnet to access internet resources through the IGW via a NAT gateway.

Q146)What is VPC Peering?

Answer: Amazon VPC peering connection is a networking connection between two amazon vpc’s that enables instances in either Amazon VPC to communicate with each other as if they are within the same network. You can create amazon VPC peering connection between your own Amazon VPC’s or Amazon VPC in another AWS account within a single region.

Q147)What is MFA in AWS?

Answer: Multi factor Authentication can add an extra layer of security to your infrastructure by adding a second method of authentication beyond just password or access key.

Q148)What are the Authentication in AWS?

Answer:

* User Name/Password
* Access Key
* Access Key/ Session Token

Q149)What is Data warehouse in AWS?

Data ware house is a central repository for data that can come from one or more sources. Organization typically use data warehouse to compile reports and search the database using highly complex queries. Data warehouse also typically updated on a batch schedule multiple times per day or per hour compared to an OLTP (Online Transaction Processing) relational database that can be updated thousands of times per second.

Q150)What is mean by Multi-AZ in RDS?

Answer: Multi AZ allows you to place a secondary copy of your database in another availability zone for disaster recovery purpose. Multi AZ deployments are available for all types of Amazon RDS Database engines. When you create s Multi-AZ DB instance a primary instance is created in one Availability Zone and a secondary instance is created by another Availability zone.

Q151)What is Amazon Dynamo DB?

Answer: Amazon Dynamo DB is fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. Dynamo DB makes it simple and Cost effective to store and retrieve any amount of data.

Q152)What is cloud formation?

Answer: Cloud formation is a service which creates the AWS infrastructure using code. It helps to reduce time to manage resources. We can able to create our resources Quickly and faster.

Q153)How to plan Auto scaling?

Answer:

* Manual Scaling
* Scheduled Scaling
* Dynamic Scaling

Q154)What is Auto Scaling group?

Answer: Auto Scaling group is a collection of Amazon EC2 instances managed by the Auto scaling service. Each auto scaling group contains configuration options that control when auto scaling should launch new instance or terminate existing instance.

Q155)Differentiate Basic and Detailed monitoring in cloud watch?

Answer:

Basic Monitoring: Basic monitoring sends data points to Amazon cloud watch every five minutes for a limited number of preselected metrics at no charge.

Detailed Monitoring: Detailed monitoring sends data points to amazon CloudWatch every minute and allows data aggregation for an additional charge.

 Q156)What is the relationship between Route53 and Cloud front?

Answer: In Cloud front we will deliver content to edge location wise so here we can use Route 53 for Content Delivery Network. Additionally, if you are using Amazon CloudFront you can configure Route 53 to route Internet traffic to those resources.

 Q157)What are the routing policies available in Amazon Route53?

Answer:

* Simple
* Weighted
* Latency Based
* Failover
* Geolocation

Q158)What is Amazon ElastiCache?

Answer: Amazon ElastiCache is a web services that simplifies the setup and management of distributed in memory caching environment.

* Cost Effective
* High Performance
* Scalable Caching Environment
* Using Memcached or Redis Cache Engine

Q159)What is SES, SQS and SNS?

Answer: SES (Simple Email Service): SES is SMTP server provided by Amazon which is designed to send bulk mails to customers in a quick and cost-effective manner.SES does not allows to configure mail server.

SQS (Simple Queue Service): SQS is a fast, reliable and scalable, fully managed message queuing service. Amazon SQS makes it simple and cost Effective. It’s temporary repository for messages to waiting for processing and acts as a buffer between the component producer and the consumer.

SNS (Simple Notification Service): SNS is a web service that coordinates and manages the delivery or sending of messages to recipients.

Q160)How To Use Amazon Sqs? What Is Aws?

Answer:Amazon Web Services is a secure cloud services stage, offering compute power, database storage, content delivery and other functionality to help industries scale and grow.

Q161) What is the importance of buffer in AWS?

Answer:low price – Consume only the amount of calculating, storage and other IT devices needed. No long-term assignation, minimum spend or up-front expenditure is required.

Elastic and Scalable – Quickly Rise and decrease resources to applications to satisfy customer demand and control costs. Avoid provisioning maintenance up-front for plans with variable consumption speeds or low lifetimes.

Q162)What is the way to secure data for resounding in the cloud?

Answer:

* Avoid storage sensitive material in the cloud. …
* Read the user contract to find out how your cloud service storing works. …
* Be serious about passwords. …
* Encrypt. …
* Use an encrypted cloud service.

Q163) Name The Several Layers Of Cloud Computing?

Answer:Cloud computing can be damaged up into three main services: Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS). PaaS in the middle, and IaaS on the lowest

Q164) What Is Lambda edge In Aws?

Answer:Lambda Edge lets you run Lambda functions to modify satisfied that Cloud Front delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to Cloud Front events, without provisioning or managing server.

Q165) Distinguish Between Scalability And Flexibility?

Answer:Cloud computing offers industries flexibility and scalability when it comes to computing needs:

Flexibility. Cloud computing agrees your workers to be more flexible – both in and out of the workplace. Workers can access files using web-enabled devices such as smartphones, laptops and notebooks. In this way, cloud computing empowers the use of mobile technology.

One of the key assistances of using cloud computing is its scalability. Cloud computing allows your business to easily expensive or downscale your IT requests as and when required. For example, most cloud service workers will allow you to increase your existing resources to accommodate increased business needs or changes. This will allow you to support your commercial growth without exclusive changes to your present IT systems.

Q166) What is IaaS?

Answer:IaaS is a cloud service that runs services on “pay-for-what-you-use” basis

IaaS workers include Amazon Web Services, Microsoft Azure and Google Compute Engine

Users: IT Administrators

Q167) What is PaaS?

Answer:PaaS runs cloud platforms and runtime environments to develop, test and manage software

Users: Software Developers

Q168)**What is SaaS?**

Answer:In SaaS, cloud workers host and manage the software application on a pay-as-you-go pricing model

Users: End Customers

Q169) Which Automation Gears Can Help With Spinup Services?

Answer:The API tools can be used for spin up services and also for the written scripts. Persons scripts could be coded in Perl, bash or other languages of your preference. There is one more option that is flowery management and stipulating tools such as a dummy or improved descendant. A tool called Scalar can also be used and finally we can go with a controlled explanation like a Right scale. Which automation gears can help with pinup service.

Q170) What Is an Ami? How Do I Build One?

Answer:An Amazon Machine Image (AMI) explains the programs and settings that will be applied when you launch an EC2 instance. Once you have finished organizing the data, services, and submissions on your ArcGIS Server instance, you can save your work as a custom AMI stored in Amazon EC2. You can scale out your site by using this institution AMI to launch added instances

Use the following process to create your own AMI using the AWS Administration Console:

\*Configure an EC2 example and its attached EBS volumes in the exact way you want them created in the custom AMI.

1. Log out of your instance, but do not stop or terminate it.
2. Log in to the AWS Management Console, display the EC2 page for your region, then click Instances.
3. Choose the instance from which you want to create a custom AMI.
4. Click Actions and click Create Image.
5. Type a name for Image Name that is easily identifiable to you and, optionally, input text for Image Description.
6. Click Create Image.

Read the message box that appears. To view the AMI standing, go to the AMIs page. Here you can see your AMI being created. It can take a though to create the AMI. Plan for at least 20 minutes, or slower if you’ve connected a lot of additional applications or data.

**Q171)What Are The Main Features Of Amazon Cloud Front?**

Answer:Amazon Cloud Front is a web service that speeds up delivery of your static and dynamic web content, such as .html, .css, .js, and image files, to your users.CloudFront delivers your content through a universal network of data centers called edge locations

**Q172)What Are The Features Of The Amazon Ec2 Service?**

Answer:Amazon Elastic Calculate Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud calculating easier for designers. Amazon EC2’s simple web serviceinterface allows you to obtain and configure capacity with minimal friction.

**Q173)Explain Storage For Amazon Ec2 Instance.?**

Answer:An instance store is a provisional storing type located on disks that are physically attached to a host machine. … This article will present you to the AWS instance store storage type, compare it to AWS Elastic Block Storage (AWS EBS), and show you how to backup data stored on instance stores to AWS EBS

Amazon SQS is a message queue service used by scattered requests to exchange messages through a polling model, and can be used to decouple sending and receiving components

**Q174)When attached to an Amazon VPC which two components provide connectivity with external networks?**

Answer:

* Internet Gateway {IGW)
* Virtual Private Gateway (VGW)

**Q175)Which of the following are characteristics of Amazon VPC subnets?**

Answer:

* Each subnet maps to a single Availability Zone.
* By defaulting, all subnets can route between each other, whether they are private or public.

**Q176)How can you send request to Amazon S3?**

Answer:Every communication with Amazon S3 is either genuine or anonymous. Authentication is a process of validating the individuality of the requester trying to access an Amazon Web Services (AWS) product. Genuine requests must include a autograph value that authenticates the request sender. The autograph value is, in part, created from the requester’s AWS access keys (access key identification and secret access key).

**Q177)What is the best approach to anchor information for conveying in the cloud ?**

Answer:Backup Data Locally. A standout amongst the most vital interesting points while overseeing information is to guarantee that you have reinforcements for your information,

* Avoid Storing Sensitive Information. …
* Use Cloud Services that Encrypt Data. …
* Encrypt Your Data. …
* Install Anti-infection Software. …
* Make Passwords Stronger. …
* Test the Security Measures in Place.

**Q178)What is AWS Certificate Manager ?**

Answer:AWS Certificate Manager is an administration that lets you effortlessly arrangement, oversee, and send open and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) endorsements for use with AWS administrations and your inward associated assets. SSL/TLS declarations are utilized to anchor arrange interchanges and set up the character of sites over the Internet and additionally assets on private systems. AWS Certificate Manager expels the tedious manual procedure of obtaining, transferring, and reestablishing SSL/TLS endorsements.

**Q179)What is the AWS Key Management Service**

Answer:AWS Key Management Service (AWS KMS) is an overseen benefit that makes it simple for you to make and control the encryption keys used to scramble your information. … AWS KMS is additionally coordinated with AWS CloudTrail to give encryption key use logs to help meet your inspecting, administrative and consistence needs.

**Q180)**

**What is Amazon EMR ?**

Answer:Amazon Elastic MapReduce (EMR) is one such administration that gives completely oversaw facilitated Hadoop system over Amazon Elastic Compute Cloud (EC2).

**Q181)What is Amazon Kinesis Firehose ?**

Answer:Amazon Kinesis Data Firehose is the least demanding approach to dependably stack gushing information into information stores and examination devices. … It is a completely overseen benefit that consequently scales to coordinate the throughput of your information and requires no continuous organization

**Q182)What Is Amazon CloudSearch and its highlights ?**

Answer:Amazon CloudSearch is a versatile cloud-based hunt benefit that frames some portion of Amazon Web Services (AWS). CloudSearch is normally used to incorporate tweaked seek abilities into different applications. As indicated by Amazon, engineers can set a pursuit application up and send it completely in under 60 minutes.

**Q183)Is it feasible for an EC2 exemplary occurrence to wind up an individual from a virtual private cloud?**

Answer:Amazon Virtual Private Cloud (Amazon VPC) empowers you to characterize a virtual system in your very own consistently disengaged zone inside the AWS cloud, known as a virtual private cloud (VPC). You can dispatch your Amazon EC2 assets, for example, occasions, into the subnets of your VPC. Your VPC nearly looks like a conventional system that you may work in your very own server farm, with the advantages of utilizing adaptable foundation from AWS. You can design your VPC; you can choose its IP address extend, make subnets, and arrange course tables, organize portals, and security settings. You can interface occurrences in your VPC to the web or to your own server farm

**Q184)Mention crafted by an Amazon VPC switch.**

Answer:VPCs and Subnets. A virtual private cloud (VPC) is a virtual system committed to your AWS account. It is consistently segregated from other virtual systems in the AWS Cloud. You can dispatch your AWS assets, for example, Amazon EC2 cases, into your VPC.

**Q185)How would one be able to associate a VPC to corporate server farm?**

Answer:AWS Direct Connect empowers you to safely associate your AWS condition to your on-premises server farm or office area over a standard 1 gigabit or 10 gigabit Ethernet fiber-optic association. AWS Direct Connect offers committed fast, low dormancy association, which sidesteps web access suppliers in your system way. An AWS Direct Connect area gives access to Amazon Web Services in the locale it is related with, and also access to different US areas. AWS Direct Connect enables you to consistently parcel the fiber-optic associations into numerous intelligent associations called Virtual Local Area Networks (VLAN). You can exploit these intelligent associations with enhance security, separate traffic, and accomplish consistence necessities.

**Q186)Is it conceivable to push off S3 with EC2 examples ?**

Answer:Truly, it very well may be pushed off for examples with root approaches upheld by local event stockpiling. By utilizing Amazon S3, engineers approach the comparative to a great degree versatile, reliable, quick, low-valued information stockpiling substructure that Amazon uses to follow its own overall system of sites. So as to perform frameworks in the Amazon EC2 air, engineers utilize the instruments giving to stack their Amazon Machine Images (AMIs) into Amazon S3 and to exchange them between Amazon S3 and Amazon EC2. Extra use case may be for sites facilitated on EC2 to stack their stationary substance from S3.

**Q187)What is the distinction between Amazon S3 and EBS ?**

Answer:EBS is for mounting straightforwardly onto EC2 server examples. S3 is Object Oriented Storage that isn’t continually waiting be gotten to (and is subsequently less expensive). There is then much less expensive AWS Glacier which is for long haul stockpiling where you don’t generally hope to need to get to it, however wouldn’t have any desire to lose it.

There are then two principle kinds of EBS – HDD (Hard Disk Drives, i.e. attractive turning circles), which are genuinely ease back to access, and SSD, which are strong state drives which are excessively quick to get to, yet increasingly costly.

* Finally, EBS can be purchased with or without Provisioned IOPS.
* Obviously these distinctions accompany related estimating contrasts, so it merits focusing on the distinctions and utilize the least expensive that conveys the execution you require.

**Q188)What do you comprehend by AWS?**

Answer:This is one of the generally asked AWS engineer inquiries questions. This inquiry checks your essential AWS learning so the appropriate response ought to be clear. Amazon Web Services (AWS) is a cloud benefit stage which offers figuring power, investigation, content conveyance, database stockpiling, sending and some different administrations to help you in your business development. These administrations are profoundly versatile, solid, secure, and cheap distributed computing administrations which are plot to cooperate and, applications in this manner made are further developed and escalade.

**Q189)Clarify the principle components of AWS?**

Answer:The principle components of AWS are:

Highway 53: Route53 is an exceptionally versatile DNS web benefit.

Basic Storage Service (S3): S3 is most generally utilized AWS stockpiling web benefit.

Straightforward E-mail Service (SES): SES is a facilitated value-based email benefit and enables one to smoothly send deliverable messages utilizing a RESTFUL API call or through an ordinary SMTP.

Personality and Access Management (IAM): IAM gives enhanced character and security the board for AWS account.

Versatile Compute Cloud (EC2): EC2 is an AWS biological community focal piece. It is in charge of giving on-request and adaptable processing assets with a “pay as you go” estimating model.

Flexible Block Store (EBS): EBS offers consistent capacity arrangement that can be found in occurrences as a customary hard drive.

CloudWatch: CloudWatch enables the controller to viewpoint and accumulate key measurements and furthermore set a progression of cautions to be advised if there is any inconvenience.

This is among habitually asked AWS engineer inquiries questions. Simply find the questioner psyche and solution appropriately either with parts name or with the portrayal alongside.

**Q190)I’m not catching your meaning by AMI? What does it incorporate?**

Answer:You may run over at least one AMI related AWS engineer inquiries amid your AWS designer meet. Along these lines, set yourself up with a decent learning of AMI.

AMI represents the term Amazon Machine Image. It’s an AWS format which gives the data (an application server, and working framework, and applications) required to play out the dispatch of an occasion. This AMI is the duplicate of the AMI that is running in the cloud as a virtual server. You can dispatch occurrences from the same number of various AMIs as you require. AMI comprises of the followings:

A pull volume format for a current example

Launch authorizations to figure out which AWS records will inspire the AMI so as to dispatch the occasions

Mapping for square gadget to compute the aggregate volume that will be appended to the example at the season of dispatch

**Q191) Is vertically scale is conceivable on Amazon occurrence?**

Answer:Indeed, vertically scale is conceivable on Amazon example.

This is one of the normal AWS engineer inquiries questions. In the event that the questioner is hoping to find a definite solution from you, clarify the system for vertical scaling.

**Q192)What is the association among AMI and Instance?**

Answer:Various sorts of examples can be propelled from one AMI. The sort of an occasion for the most part manages the equipment segments of the host PC that is utilized for the case. Each kind of occurrence has unmistakable registering and memory adequacy.

When an example is propelled, it gives a role as host and the client cooperation with it is same likewise with some other PC however we have a totally controlled access to our occurrences. AWS engineer inquiries questions may contain at least one AMI based inquiries, so set yourself up for the AMI theme exceptionally well.

**Q193)What is the distinction between Amazon S3 and EC2?**

Answer:The contrast between Amazon S3 and EC2 is given beneath:

Amazon S3

Amazon EC2

The significance of S3 is Simple Storage Service. The importance of EC2 is Elastic Compute Cloud.

It is only an information stockpiling administration which is utilized to store huge paired files. It is a cloud web benefit which is utilized to have the application made.

It isn’t required to run a server. It is sufficient to run a server.

It has a REST interface and utilizations secure HMAC-SHA1 validation keys. It is much the same as a tremendous PC machine which can deal with application like Python, PHP, Apache and some other database.

When you are going for an AWS designer meet, set yourself up with the ideas of Amazon S3 and EC2, and the distinction between them.

**Q194)What number of capacity alternatives are there for EC2 Instance?**

Answer:There are four stockpiling choices for Amazon EC2 Instance:

* Amazon EBS
* Amazon EC2 Instance Store
* Amazon S3
* Adding Storage

Amazon EC2 is the basic subject you may run over while experiencing AWS engineer inquiries questions. Get a careful learning of the EC2 occurrence and all the capacity alternatives for the EC2 case.

**Q195)What are the security best practices for Amazon Ec2 examples?**

Answer:There are various accepted procedures for anchoring Amazon EC2 occurrences that are pertinent whether occasions are running on-preface server farms or on virtual machines. How about we view some broad prescribed procedures:

Minimum Access: Make beyond any doubt that your EC2 example has controlled access to the case and in addition to the system. Offer access specialists just to the confided in substances.

Slightest Privilege: Follow the vital guideline of minimum benefit for cases and clients to play out the capacities. Produce jobs with confined access for the occurrences.

Setup Management: Consider each EC2 occasion a design thing and use AWS arrangement the executives administrations to have a pattern for the setup of the occurrences as these administrations incorporate refreshed enemy of infection programming, security highlights and so forth.

Whatever be the activity job, you may go over security based AWS inquiries questions. Along these lines, motivate arranged with this inquiry to break the AWS designer meet.

**Q196)Clarify the highlights of Amazon EC2 administrations.**

Answer:Amazon EC2 administrations have following highlights:

* Virtual Computing Environments
* Proffers Persistent capacity volumes
* Firewall approving you to indicate the convention
* Pre-designed layouts
* Static IP address for dynamic Cloud Computing

**Q197)What is the system to send a demand to Amazon S3?**

Answer: Reply: There are 2 different ways to send a demand to Amazon S3 –

* Using REST API
* Using AWS SDK Wrapper Libraries, these wrapper libraries wrap the REST APIs for Amazon

**Q198)What is the default number of basins made in AWS?**

Answer**:**This is an extremely straightforward inquiry yet positions high among AWS engineer inquiries questions. Answer this inquiry straightforwardly as the default number of pails made in each AWS account is 100.

**Q199)What is the motivation behind T2 examples?**

Answer:T2 cases are intended for

Providing moderate gauge execution

Higher execution as required by outstanding task at hand

**Q200)What is the utilization of the cradle in AWS?**

Answer:This is among habitually asked AWS designer inquiries questions. Give the appropriate response in straightforward terms, the cradle is primarily used to oversee stack with the synchronization of different parts i.e. to make framework blame tolerant. Without support, segments don’t utilize any reasonable technique to get and process demands. Be that as it may, the cushion makes segments to work in a decent way and at a similar speed, hence results in quicker administrations.

**Q201)What happens when an Amazon EC2 occurrence is halted or ended?**

Answer:At the season of ceasing an Amazon EC2 case, a shutdown is performed in a typical way. From that point onward, the changes to the ceased state happen. Amid this, the majority of the Amazon EBS volumes are stayed joined to the case and the case can be begun whenever. The occurrence hours are not included when the occasion is the ceased state.

At the season of ending an Amazon EC2 case, a shutdown is performed in an ordinary way. Amid this, the erasure of the majority of the Amazon EBS volumes is performed. To stay away from this, the estimation of credit deleteOnTermination is set to false. On end, the occurrence additionally experiences cancellation, so the case can’t be begun once more.

**Q202)What are the mainstream DevOps devices?**

Answer:In an AWS DevOps Engineer talk with, this is the most widely recognized AWS inquiries for DevOps. To answer this inquiry, notice the well known DevOps apparatuses with the kind of hardware –

* Jenkins – Continuous Integration Tool
* Git – Version Control System Tool
* Nagios – Continuous Monitoring Tool
* Selenium – Continuous Testing Tool
* Docker – Containerization Tool
* Puppet, Chef, Ansible – Deployment and Configuration Administration Tools.

Q203)What are IAM Roles and Policies, What is the difference between IAM Roles and Policies.

Answer:Roles are for AWS services, Where we can assign permission of some AWS service to other Service.

Example – Giving S3 permission to EC2 to access S3 Bucket Contents.

Policies are for users and groups, Where we can assign permission to user’s and groups.

Example – Giving permission to user to access the S3 Buckets.

Q204)What are the Defaults services we get when we create custom AWS VPC?

Answer:

* Route Table
* Network ACL
* Security Group

Q205)What is the Difference Between Public Subnet and Private Subnet ?

Answer:Public Subnet will have Internet Gateway Attached to its associated Route Table and Subnet, Private Subnet will not have the Internet Gateway Attached to its associated Route Table and Subnet

Public Subnet will have internet access  and Private subnet will not have the internet access directly.

Q206) How do you access the Ec2 which has private IP which is in private Subnet ?

Answer: We can access using VPN if the VPN is configured into that Particular VPC where Ec2 is assigned to that VPC in the Subnet. We can access using other Ec2 which has the Public access.

Q207)We have a custom VPC Configured and MYSQL Database server which is in Private Subnet and      we need to update the MYSQL Database Server, What are the Option to do so.

Answer:By using NAT Gateway in the VPC or Launch a NAT Instance ( Ec2) Configure or Attach the NAT Gateway in Public Subnet ( Which has Route Table attached to IGW) and attach it to the Route Table which is Already attached to the Private Subnet.

Q208) What are the Difference Between Security Groups and  Network ACL

Answer:

|  |  |
| --- | --- |
| **Security Groups** | **Network ACL** |
| Attached to Ec2 instance | Attached to a subnet. |
| Stateful – Changes made in incoming rules is automatically applied to the outgoing rule | Stateless – Changes made in incoming rules is not applied to the outgoing rule |
| Blocking IP Address can’t be done | IP Address can be Blocked |
| Allow rules only, by default all rules are denied | Allow and Deny can be Used. |

 Q209)What are the Difference Between Route53 and ELB?

Answer:Amazon Route 53 will handle DNS servers. Route 53 give you web interface through which the DNS can be managed using Route 53, it is possible to direct and failover traffic. This can be achieved by using DNS Routing Policy.

One more routing policy is Failover Routing policy. we set up a health check to monitor your application endpoints. If one of the endpoints is not available, Route 53 will automatically forward the traffic to other  endpoint.

Elastic Load Balancing

ELB automatically scales depends on the demand, so sizing of the load balancers to handle more traffic effectively when it is not required.

Q210)What are the DB engines which can be used in AWS RDS?

Answer:

* MariaDB
* MYSQL DB
* MS SQL DB
* Postgre DB
* Oracle DB

Q211)What is Status Checks in AWS Ec2?

Answer: System Status Checks – System Status checks will look into problems with instance which needs AWS help to resolve the issue. When we see system status check failure, you can wait for AWS to resolve the issue, or do it by our self.

* Network connectivity
* System power
* Software issues Data Centre’s
* Hardware issues
* Instance Status Checks – Instance Status checks will look into issues which need our involvement to fix the issue. if status check fails, we can reboot that particular instance.
* Failed system status checks
* Memory Full
* Corrupted file system
* Kernel issues

Q212)To establish a peering connections between two VPC’s What condition must be met?

Answer:

* CIDR Block should overlap
* CIDR Block should not overlap
* VPC should be in the same region
* VPC must belong to same account.
* CIDR block should not overlap between vpc setting up a peering connection . peering connection is allowed within a region , across region, across different account.

Q213) Troubleshooting with EC2 Instances:

Answer: Instance States

* If the instance state is 0/2- there might be some hardware issue
* If the instance state is ½-there might be issue with OS.   
  Workaround-Need to restart the instance, if still that is not working logs will help to fix the issue.

Q214) How EC2instances can be resized.

Answer: EC2 instances can be resizable(scale up or scale down) based on requirement

Q215) EBS: its block-level storage volume which we can use after mounting with EC2 instances.

Answer:For types please refer AWS Solution Architect book.

Q216) Difference between EBS,EFS and S3

Answer:

* We can access EBS only if its mounted with instance, at a time EBS can be mounted only with one instance.
* EFS can be shared at a time with multiple instances
* S3 can be accessed without mounting with instances

Q217) Maximum number of bucket which can be crated in AWS.

Answer:100 buckets can be created by default in AWS account.To get more buckets additionally you have to request Amazon for that.

Q218)Maximum number of EC2 which can be created in VPC.

Answer:Maximum 20 instances can be created in a VPC. we can create 20 reserve instances and request for spot instance as per demand.

Q219) How EBS can be accessed?

Answer:**EBS**provides high performance block-level storage which can be attached with running EC2 instance. Storage can be formatted and mounted with EC2 instance, then it can be accessed.

Q220)Process to mount EBS to EC2 instance

Answer:

* Df –k
* mkfs.ext4 /dev/xvdf
* Fdisk –l
* Mkdir /my5gbdata
* Mount /dev/xvdf /my5gbdata

Q221)How to add volume permanently with instance.

Answer:With each restart volume will get unmounted from instance, to keep this attached need to perform below step

Cd /etc/fstab

/dev/xvdf /data ext4  defaults  0

0 <edit the file system name accordingly>

Q222) What is the Difference between the Service Role and SAML Federated Role**.**

Answer: Service Role are meant for usage of AWS Services and based upon the policies attached to it,it will have the scope to do its task. Example : In case of automation we can create a service role and attached to it.

Federated Roles are meant for User Access and getting access to AWS as per designed role. Example  : We can have a federated role created for our office employee and corresponding to that a Group will be created in the AD and user will be added to it.

 Q223)How many Policies can be attached to a role.

Answer: 10 (Soft limit), We can have till 20.

 Q224) What are the different ways to access AWS.

Answer:3 Different ways (CLI, Console, SDK)

Q225)How a Root AWS user is different from in IAM User.

Answer: Root User will have acces to entire AWS environment and it will not have any policy attached to it. While IAM User will be able to do its task on the basis of policies attached to it.

Q226)What do you mean by Principal of least privilege in term of IAM.

Answer: Principal of least privilege means to provide the same or equivalent permission to the user/role.

Q227)What is the meaning of non-explicit deny for an IAM User.

Answer: When an IAM user is created and it is not having any policy attached to it,in that case he will not be able to access any of the AWS Service until a policy has been attached to it.

Q228) What is the precedence level between explicit allow and explicit deny.

Answer: Explicit deny will always override Explicit Allow.

Q229) What is the benefit of creating a group in IAM.

Answer:Creation of Group makes the user management process much simpler and user with the same kind of permission can be added in a group and at last addition of a policy will be much simpler to the group in comparison to doing the same thing manually.

Q230)What is the difference between the Administrative Access and Power User Access in term of pre-build policy.

Answer: Administrative Access will have the Full access to AWS resources. While Power User Access will have the Admin access except the user/group management permission.

 Q231)What is the purpose of Identity Provider.

Answer: Identity Provider helps in building the trust between the AWS and the Corporate AD environment while we create the Federated role.

Q232) What are the benefits of STS (Security Token Service).

Answer: It help in securing the AWS environment as we need not to embed or distributed the AWS Security credentials in the application. As the credentials are temporary we need not to rotate them and revoke them.

Q233)What is the benefit of creating the AWS Organization.

Answer: It helps in managing the IAM Policies, creating the AWS Accounts programmatically, helps in managing the payment methods and consolidated billing.

Q234)What is the maximum file length in S3?

Answer: utf-8 1024 bytes

Q235)which activity cannot be done using autoscaling?

Answer:Maintain fixed running of ec2

Q236)How will you secure data at rest in EBS?

Answer: EBS data is always secure

Q237)What is the maximum size of S3 Bucket?

Answer: 5TB

Q238)Can objects in Amazon s3 be delivered through amazon cloud front?

Answer:Yes

Q239)which service is used to distribute content to end user service using global network of edge location?

Answer: Virtual Private Cloud

Q240)What is ephemaral storage?

Answer: Temporary storage

Q241)What are shards in kinesis aws services?

Answer: Shards are used to store data in Kinesis.

Q242)Where can you find the ephemeral storage?

Answer: In Instance store service.

Q243)I have some private servers on my premises also i have distributed some of My workload on the public cloud,what is the architecture called?

Answer:Virtual private cloud

Q244)Route 53 can be used to route users to infrastructure outside of  aws.True/false?

Answer: False

Q245)Is simple workflow service one of the valid Simple Notification Service  subscribers?

Answer: No

Q246)which cloud model do Developers and organizations all around the world leverage extensively?

Answer: IAAS-Infrastructure as a service.

Q247)Can cloud front serve content from a non AWS origin server?

Answer: No

Q248)Is EFS a centralised storage service in AWS?

Answer: Yes

Q249)Which AWS service will you use to collect and process ecommerce data for near real time analysis?

Answer: Both Dynamo DB & Redshift

Q250)An high demand of IOPS performance is expected around 15000.Which EBS volume type would you recommend?

Answer:  Provisioned IOPS.

Q251) How are Spot Instance, On-demand Instance, and Reserved Instance different from one another?

Both Spot Instance and On-demand Instance are models for pricing.

|  |  |
| --- | --- |
| Spot Instance | On-demand Instance |
| With Spot Instance, customers can purchase compute capacity with no upfront commitment at all. | With On-demand Instance, users can launch instances at any time based on the demand. |
| Spot Instances are spare Amazon instances that you can bid for. | On-demand Instances are suitable for high-availability needs of applications. |
| When the bidding price exceeds the spot price, the instance is automatically launched, and the spot price fluctuates based on supply and demand for instances. | On-demand Instances are launched by users only with the pay-as-you-go model. |
| When the bidding price is less than the spot price, the instance is immediately taken away by Amazon. | On-demand Instances will remain persistent without any automatic termination from Amazon. |
| Spot Instances are charged on an hourly basis. | On-demand Instances are charged on a per-second basis |

Q252)Your organization has decided to have all their workload on the public cloud. But, due to certain security concerns, your organization decides to distribute some of the workload on private servers. You are asked to suggest a cloud architecture for your organization. What will be your suggestion?

A hybrid cloud. The hybrid cloud architecture is where an organization can use the public cloud for shared resources and the private cloud for its confidential workloads.

Q254)The data on the root volumes of store-backed and EBS-backed instances get deleted by default when they are terminated. If you want to prevent that from happening, which instance would you use?

EBS-backed instances. EBS-backed instances use EBS volume as their root volume. EBS volume consists of virtual drives that can be easily backed up and duplicated by snapshots. The biggest advantage of EBS-backed volumes is that the data can be configured to be stored for later retrieval even if the virtual machine or the instances are shut down.

Q255) You have connected four instances to ELB. To automatically terminate your unhealthy instances and replace them with new ones, which functionality would you use?

Auto-scaling groups

1. How will you configure an Amazon S3 bucket to serve static assets for your public web application?

By configuring the bucket policy to provide public read access to all objects

1. Your organization wants to send and receive compliance emails to its clients using its own email address and domain. What service would you suggest for achieving the same in an easy and cost-effective way?

Amazon Simple Email Service (Amazon SES), which is a cloud-based email sending service, can be used for this purpose.

1. Can you launch Amazon Elastic Compute Cloud (EC2) instances with predetermined private IP addresses? If yes, then with which Amazon service it is possible?

Yes. It is possible by using VPC (Virtual Private Cloud).

1. Why do we make subnets?

Creating subnets means dividing a large network into smaller ones. These subnets can be created for several reasons. For example, creating and using subnets can help reduce congestion by making sure that the traffic destined for a subnet stays in that subnet. This helps in efficiently routing the traffic coming to the network that reduces the network’s load.

1. If you launched a standby RDS, will it be launched in the same availability zone as your primary?

No, standby instances are automatically launched in different availability zones than the primary, making them physically independent infrastructures. This is because the whole purpose of standby instances is to prevent infrastructure failure. So, in case the primary goes down, the standby instance will help recover all of the data.

1. Which of the following is a global Content Delivery Network service that securely delivers data to users with low latency and high transfer speed.

Amazon CloudFront

1. Which Amazon solution will you use if you want to accelerate moving petabytes of data in and out of AWS, using storage devices that are designed to be secure for data transfer?

Amazon Snowball. AWS Snowball is the data transport solution for large amounts of data that need to be moved into and out of AWS using physical storage devices.

1. If you are running your DB instance as Multi-AZ deployment, can you use standby DB instances along with your primary DB instance?

No, the standby DB instance cannot be used along with the primary DB instances since the standby DB instances are supposed to be used only if the primary instance goes down.

1. Your organization is developing a new multi-tier web application in AWS. Being a fairly new and small organization, there’s limited staff. But, the organization requires high availability. This new application comprises complex queries and table joins. Which Amazon service will be the best solution for your organization’s requirements?

DynamoDB will be the right choice here since it is designed to be highly scalable, more than RDS or any other relational database services.

1. Your organization is using DynamoDB for its application. This application collects data from its users every 10 minutes and stores it in DynamoDB. Then every day, after a particular time interval, the data (respective to each user) is extracted from DynamoDB and sent to S3. Then, the application visualizes this data to the users. You are asked to propose a solution to help optimize the backend of the application for latency at lower cost. What would you recommend?

ElastiCache. Amazon ElastiCache is a caching solution offered by Amazon. It can be used to store a cached version of the application in a region closer to users so that when requests are made by the users the cached version of the application can respond, and hence latency will be reduced

1. You accidently stopped an EC2 instance in a VPC with an associated Elastic IP. If you start the instance again, what will be the result?

The data stored on the instance will be lost. Elastic IP is disassociated from the instance only if the instance is terminated.

1. Your organization has around 50 IAM users. Now, it wants to introduce a new policy that will affect the access permissions of an IAM user. How can it implement this without having to apply the policy at the individual user level?

It is possible using IAM groups, by adding users in the groups as per their roles and by simply applying the policy to the groups.

1. I created a web application with autoscaling. I observed that the traffic on my application is the highest on Wednesdays and Fridays between 9 AM and 7 PM. What would be the best solution for me to handle the scaling?

Configure a policy in autoscaling to scale as per the predictable traffic patterns.

1. How would you handle a situation where the relational database engine crashes often whenever the traffic to your RDS instances increases, given that the replica of RDS instance is not promoted as the master instance?

A bigger RDS instance type needs to be opted for handling large amounts of traffic, creating manual or automated snapshots to recover data in case the RDS instance goes down.

1. Is there a way to upload a file that is greater than 100 megabytes in Amazon S3?

Yes, it is possible by using multipart upload utility from AWS. With multipart upload utility, larger files can be uploaded in multiple parts that are uploaded independently. You can also decrease upload time by uploading these parts in parallel. After the upload is done, the parts will be merged into a single object or file to create the original file from which the parts were created.

1. Suppose, you hosted an application on AWS that lets the users render images and do some general computing. Which of the below listed services can you use to route the incoming user traffic?

* Classic Load Balancer
* Application Load Balancer
* Network Load balancer

**Application Load Balancer:**It supports path-based routing of the traffic and hence helps in enhancing the performance of the application structured as smaller services. Using application load balancer, the traffic can be routed based on the requests made. In this case scenario, the traffic where requests are made for rendering images can be directed to the servers only deployed for rendering images and the traffic where the requests are made for computing can be directed to the servers deployed only for general computing purposes.

1. You have an application running on your Amazon EC2 instance. You want to reduce the load on your instance as soon as the CPU utilization reaches 100 percent. How will you do that?

It can be done by creating an autoscaling group to deploy more instances when the CPU utilization exceeds 100 percent and distributing traffic among instances by creating a load balancer and registering the Amazon EC2 instances with it.

1. What would I have to do if I want to access Amazon Simple Storage buckets and use the information for access audits?

AWS CloudTrail can be used in this case as it is designed for logging and tracking API calls, and it has also been made available for storage solutions.

1. I created a key in North Virginia region to encrypt my data in Oregon region. I also added three users to the key and an external AWS account. Then, to encrypt an object in S3, when I tried to use the same key, it was not listed. Where did I go wrong?

The data and the key should be in the same region. That is, the data that has to be encrypted should be in the same region as the one in which the key was created. In this case, the data is in Oregon region, whereas the key is created in North Virginia region.

1. Suppose, I created a subnet and launched an EC2 instance in the subnet with default settings. Which of the following options will be ready to use on the EC2 instance as soon as it is launched?

* Elastic IP
* Private IP
* Public IP
* Internet Gateway

Private IP. Private IP is automatically assigned to the instance as soon as it is launched. While elastic IP has to be set manually, Public IP needs an Internet Gateway which again has to be created since it’s a new VPC.

1. If you launched a standby RDS, will it be launched in the same availability zone as your primary?

No, standby instances are automatically launched in different availability zones than the primary, making them physically independent infrastructures. This is because the whole purpose of standby instances is to prevent infrastructure failure. So, in case the primary goes down, the standby instance will help recover all of the data.

1. Which of the following is a global Content Delivery Network service that securely delivers data to users with low latency and high transfer speed.

Amazon CloudFront

1. Which Amazon solution will you use if you want to accelerate moving petabytes of data in and out of AWS, using storage devices that are designed to be secure for data transfer?

Amazon Snowball. AWS Snowball is the data transport solution for large amounts of data that need to be moved into and out of AWS using physical storage devices.

1. If you are running your DB instance as Multi-AZ deployment, can you use standby DB instances along with your primary DB instance?

No, the standby DB instance cannot be used along with the primary DB instances since the standby DB instances are supposed to be used only if the primary instance goes down.

1. Your organization is developing a new multi-tier web application in AWS. Being a fairly new and small organization, there’s limited staff. But, the organization requires high availability. This new application comprises complex queries and table joins. Which Amazon service will be the best solution for your organization’s requirements?

DynamoDB will be the right choice here since it is designed to be highly scalable, more than RDS or any other relational database services.

1. Your organization is using DynamoDB for its application. This application collects data from its users every 10 minutes and stores it in DynamoDB. Then every day, after a particular time interval, the data (respective to each user) is extracted from DynamoDB and sent to S3. Then, the application visualizes this data to the users. You are asked to propose a solution to help optimize the backend of the application for latency at lower cost. What would you recommend?

ElastiCache. Amazon ElastiCache is a caching solution offered by Amazon. It can be used to store a cached version of the application in a region closer to users so that when requests are made by the users the cached version of the application can respond, and hence latency will be reduced.

1. Your organization has around 50 IAM users. Now, it wants to introduce a new policy that will affect the access permissions of an IAM user. How can it implement this without having to apply the policy at the individual user level?

It is possible using IAM groups, by adding users in the groups as per their roles and by simply applying the policy to the groups.

1. I created a web application with autoscaling. I observed that the traffic on my application is the highest on Wednesdays and Fridays between 9 AM and 7 PM. What would be the best solution for me to handle the scaling?

Configure a policy in autoscaling to scale as per the predictable traffic patterns.

1. How would you handle a situation where the relational database engine crashes often whenever the traffic to your RDS instances increases, given that the replica of RDS instance is not promoted as the master instance?

A bigger RDS instance type needs to be opted for handling large amounts of traffic, creating manual or automated snapshots to recover data in case the RDS instance goes down.

1. Is there a way to upload a file that is greater than 100 megabytes in Amazon S3?

Yes, it is possible by using multipart upload utility from AWS. With multipart upload utility, larger files can be uploaded in multiple parts that are uploaded independently. You can also decrease upload time by uploading these parts in parallel. After the upload is done, the parts will be merged into a single object or file to create the original file from which the parts were created.

1. Suppose, you hosted an application on AWS that lets the users render images and do some general computing. Which of the below listed services can you use to route the incoming user traffic?

* Classic Load Balancer
* Application Load Balancer
* Network Load balancer

**Application Load Balancer:**It supports path-based routing of the traffic and hence helps in enhancing the performance of the application structured as smaller services. Using application load balancer, the traffic can be routed based on the requests made. In this case scenario, the traffic where requests are made for rendering images can be directed to the servers only deployed for rendering images and the traffic where the requests are made for computing can be directed to the servers deployed only for general computing purposes.

1. You have an application running on your Amazon EC2 instance. You want to reduce the load on your instance as soon as the CPU utilization reaches 100 percent. How will you do that?

It can be done by creating an autoscaling group to deploy more instances when the CPU utilization exceeds 100 percent and distributing traffic among instances by creating a load balancer and registering the Amazon EC2 instances with it.

1. What would I have to do if I want to access Amazon Simple Storage buckets and use the information for access audits?

AWS CloudTrail can be used in this case as it is designed for logging and tracking API calls, and it has also been made available for storage solutions.

1. Your organization has four instances for production and another four for testing. You are asked to set up a group of IAM users that can only access the four production instances and not the other four testing instances. How will you achieve this?

We can achieve this by defining tags on the test and production instances and then adding a condition to the IAM policy that allows access to specific tags.

1. What is the maximum number of S3 buckets you can create?

* 50
* 20
* 70
* 100

100

1. Your organization wants to monitor the read and write IOPS for its AWS MySQL RDS instance and then send real-time alerts to its internal operations team. Which service offered by Amazon can help your organization achieve this scenario?

Amazon CloudWatch would help us achieve this. Since Amazon CloudWatch is a monitoring tool offered by Amazon, it’s the right service to use in the above-mentioned scenario.

1. Which of the following services can be used if you want to capture client connection information from your load balancer at a particular time interval?

* Enabling access logs on your load balancer
* Enabling CloudTrail for your load balancer
* Enabling CloudWatch metrics for your load balancer

Enabling CloudTrail for your load balancer. AWS CloudTrail is an inexpensive log monitoring solution provided by Amazon. It can provide logging information for load balancer or any other AWS resources. The provided information can further be used for analysis.

1. You have created a VPC with private and public subnets. In what kind of subnet would you launch the database servers?

Database servers should be ideally launched in private subnets. Private subnets are ideal for the backend services and databases of all applications since they are not meant to be accessed by the users of the applications, and private subnets are not routable from the Internet.

1. Is it possible to switch from an Instance-backed root volume to an EBS-backed root volume at any time?

No, it is not possible.

1. When should you use the classic load balancer and the application load balancer?

The classic load balancer is used for simple load balancing of traffic across multiple EC2 instances. While, the application load balancing is used for more intelligent load balancing, based on the multi-tier architecture or container-based architecture of the application. Application load balancing is mostly used when there is a need to route traffic to multiple services.

1. Which service provided by AWS can you use to transfer objects from your data center, when you are using Amazon CloudFront?

Amazon Direct Connect. It is a network service that acts as an alternative to using the Internet to connect customers in on-premise sites with AWS.

1. You have deployed multiple EC2 instances across multiple availability zones to run your website. You have also deployed a Multi-AZ RDS MySQL Extra Large DB Instance. The site performs a high number of small read and write operations per second. After some time, you observed that there is read contention on RDS MySQL. What would be your approach to resolve the contention and optimize your website?

We can deploy ElastiCache in memory cache running in every availability zone. This will help in creating a cached version of the website for faster access in each availability zone. We can also add RDS MySQL read replica in each availability zone that can help in efficient and better performance for read operations. So, there will not be any increased workload on RDS MySQL instance, hence resolving the contention issue.

1. Your company wants you to propose a solution so that the company’s data center can be connected to Amazon cloud network. What would be your proposal?

The data center can be connected to Amazon cloud network by establishing a virtual private network (VPN) between the VPC and the data center. Virtual private network lets you establish a secure pathway or tunnel from your premise or device to AWS global network.

1. Which of the following Amazon Services would you choose if you want complex querying capabilities but not a whole data warehouse?

* RDS
* Redshift
* ElastiCache
* DynamoDB

Amazon RDS

1. You want to modify the security group rules while it is being used by multiple EC2 instances. Will you be able to do that? If yes, will the new rules be implemented on all previously running EC2 instances that were using that security group?

Yes, the security group that is being used by multiple EC2 instances can be modified. The changes will be implemented immediately and be applied to all the previously running EC2 instances without restarting the instances

1. Which one of the following is a structured data store that supports indexing and data queries to both EC2 and S3?

* DynamoDB
* MySQL
* Aurora
* SimpleDB

SimpleDB

1. How many total VPCs per account/region and subnets per VPC can you have?

* 4, 100
* 7, 40
* 5, 200
* 3, 150

5, 200

1. Which service offered by Amazon will you choose if you want to collect and process e-commerce data for near real-time analysis? (Choose any two)

* DynamoDB
* Redshift
* Aurora
* SimpleDB

DynamoDB. DynamoDB is a fully managed NoSQL database service that can be fed any type of unstructured data. Hence, DynamoDB is the most apt choice for collecting data from e-commerce websites.

For near real-time analysis, we can use Amazon Redshift.

1. If in CloudFront the content is not present at an edge location, what will happen when a request is made for that content?

CloudFront will deliver the content directly from the origin server. It will also store the content in the cache of the edge location where the content was missing.

1. Can you change the private IP address of an EC2 instance while it is in running or in a stopped state?

No, it cannot be changed. When an EC2 instance is launched, a private IP address is assigned to that instance at the boot time. This private IP address is attached to the instance for its entire lifetime and can never be changed.

1. Which of the following options will you use if you have to move data over long distances using the Internet, from instances that are spread across countries to your Amazon S3 bucket?

* Amazon CloudFront
* Amazon Transfer Acceleration
* Amazon Snowball
* Amazon Glacier

Amazon Transfer Acceleration. It throttles the data transfer up to 300 percent using optimized network paths and Amazon Content Delivery Network. Snowball cannot be used here as this service does not support cross-region data transfer.

1. Which of the following services is a data storage system that also has REST API interface and uses secure HMAC-SHA1 authentication keys?

* Amazon Elastic Block Store
* Amazon Snapshot
* Amazon S3

Amazon S3. It gets various requests from applications, and it has to identify which requests are to be allowed and which to be denied. Amazon S3 REST API uses a custom HTTP scheme based on a keyed HMAC for authentication of requests.

1. What kind of IP address can you use for your customer gateway (CGW) address?

We can use the Internet routable IP address, which is a public IP address of your NAT device.

1. Which of the following is not an option in security groups?

* List of users
* Posts
* IP addresses
* List of protocols

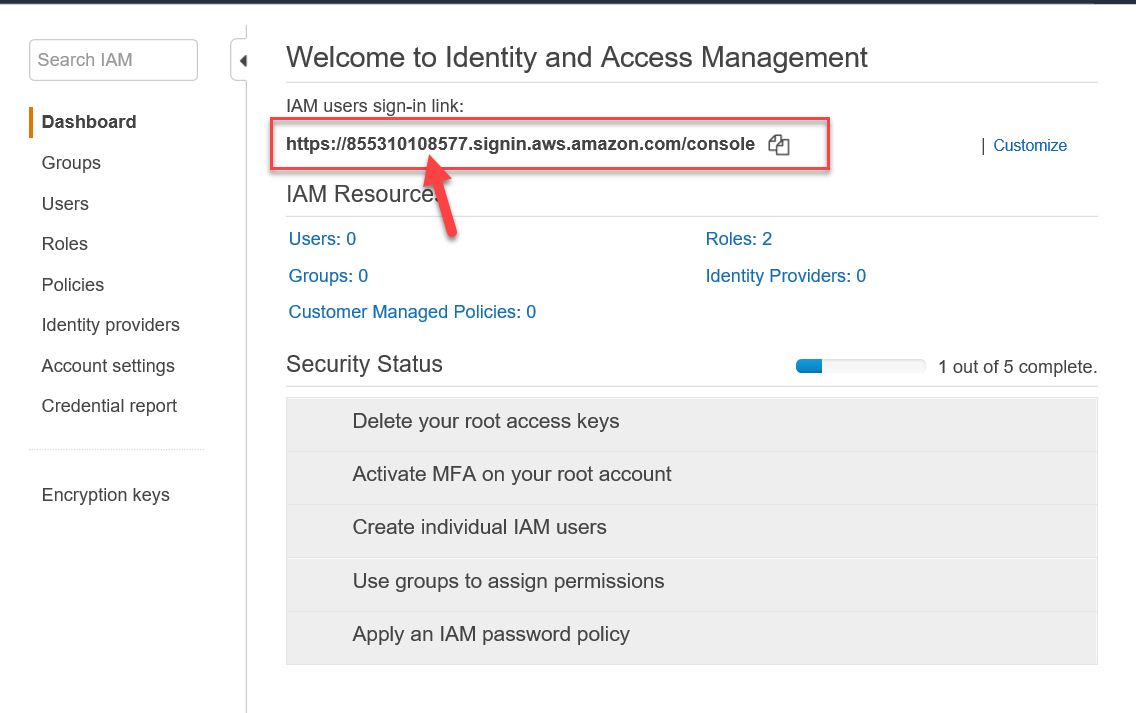
List of users

* IAM is where you manage your AWS users, groups and roles and their access to AWS accounts and services.
  + IAM provides access and access permission to AWS resources such as EC2, S3 and DynamoDB
  + IAM is **global** to all AWS regions, creating a user account will apply to all the regions. This means there is no particular region associated with IAM.
* The common use of IAM is to manage users, groups, roles, IAM access policies , API keys and specify a password policy as well as manage MFA requirements on a per user basis.

By default, any new IAM user you create in an AWS account is created with**NO access** to any AWS services. This is an **IMPLICIT DENY** rule set on all new IAM users.

* Ur umm
* Best practices include:
  + Delete your root access keys
  + Activate MFA on your root account
  + Create and use an IAM  user with admin privileges instead of using your root account.
  + Create individual IAM users
  + Use groups to assign permissions
  + Follow the "**PRINCIPLE OF LEAST PRIVILEDGE**"
  + Apply an IAM password policy.

* Start by going to IAM under services. You will notice that the sign in link uses your account ID as a domain prefix (see below). How ever, you can change it to something you can remember by clicking on customize.

* 

* The left navigation bar has access to groups, users, roles, policies, etc.
  + Identity providers are used for Identity federation.
  + Use account settings to set a strong password policy for IAM users.
  + Use credential report to see what access users have.

* IAM is the key service in the security pillar of AWS's well architected framework.
* IAM is generally used to create user and assign permission policies to those users.
* Users also belong to groups. Every user assigned to a group inherits the permission of the group.
* IAM roles are used to grant applications access to AWS services. Such as giving EC2 an IAM role so they can access an S3 bucket.
* The Command line interface (CLI) and SDKs make API calls that are authenticated by an IAM API key. These access key are associated with a user, role or policy.
* There is no cost for using IAM.

**IAM Permissions Policies:**

* A policy is a document that formally states one or more permissions
* By default, all permissions are**implicitly denied**
* An **explicit deny** always overrides an **explicit allow**. This happens in cases where a user has multiple IAM policies. If one policy grants them an access and another policy denies the access, then the Deny will override the allow.
* IAM provides pre-built policy templates to assign to users and groups, examples include:
  + **Administrator access:**Full access to ALL AWS resources
  + **Power user access:**Admin access except it does not allow user/group management.
  + **Read only access:**Only view AWS resources (i.e user can only view what is in an S3 bucket)
* You can also create custom IAM permission policies using the **visual editor** or written from scratch.
* Sometimes instead of starting from scratch, you can import an existing policy and use the visual editor to edit it.
* You can also create a condition on a permission which must be met to allow a particular action.
* A created policy is called a **Customer Managed policy**.
* More than one policy can be attached to a user or group at the same time
* Policies cannot be directly attached to AWS resources (such as EC2 instance)
* The below image is a case where one policy Allows access and the other denies access. (The asterisk used for action and resource are just wild cards). If both policies are attached to an IAM user, then the deny policy will override the allow.

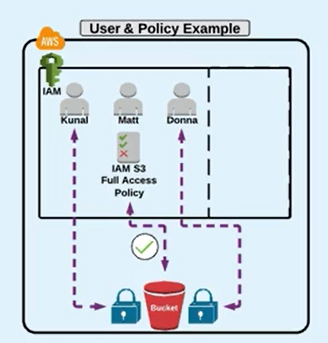


**IAM Users:**

* When first created by default, an IAM user has an **Implicit deny** for all AWS services and does not have access to use them until a policy granting allow access has been applied to the user or to the group that the user belongs to.
* When a user is created, it is important to determine what the user is going to be accessing. If they will be accessing the management console they will need a console password. If they will be accessing a  command line interface (CLI)  or SDKs then they need access keys.
* IAM users receive unique access credentials so you do not and should not share with others.
  + Don’t use an email address as that is only used for root account.
  + Once you create a new user, they will have a secret access key and password, the secret access key can only be seen once, so make sure you download the file and save it locally.
* Each newly created user is assigned a unique Amazon resource name (ARN) See below.
* You can also use an **inline policy** which is unique policy that can be created and is unique just for a single user. For example, you can give Matt all access policy to S3 but use the inline policy to deny him delete access to a particular bucket.

* 

* User credentials should **NEVER** be stored or passed to an EC2 instance
* Users can have group and regular user policies apply to them - meaning a user can have multiple IAM policies applied to them at any given time.
* By default,, an explicit deny always overrides an explicit allow from attached IAM policies.
* MFA can be configured on a per user basis for login and resource access/actions.



**IAM Groups:**

* Allows you to assign IAM permission policies to more than one user at a time
* This ability allows for easier access management to AWS resources

**CREATING  A GROUP**

* Once you create a group, you just need to assign the policy that everyone in that group should have
* Then add users to the group
* Once you create a group and add users to it, you can go back to the user and remove their individual permissions.

**IAM Roles:**

A role is something that another entity can assume, and in doing so, acquires temporary permissions defined by the role.

Entities in AWS that can assume a role include:

* AWS services such as EC2, Lambda
* IAM users, Groups, and Roles in same or different AWS Account.
* Federated users (active Directory, LDAP, aor Wen Identity)
* Roles must be used because policies cannot be directly attached to AWS services

**For Ex:**

If you are using an EC2 instance and it need to access an S3 bucket,

* The instance should assume a role from IAM with the proper required permissions (S3 read-only)
* The instance can then perform actions based on the roles it assumes.
* You should never pass or store permanent credentials on or to an EC2 instance
* You can associate a role with an instance before or after launch. Users must have the "PassRole" Permission
* **An EC2 instance can only have ONE role attached at a time**

**TRUST Policy**

Defines the entities that have permission to assume the role

* **Roles can also grant a different AWS account the permission to access services in your account.** This is mostly in cases where you need to grant access to an account but don’t want to create permanent permissions for that account.
* This is called **Cross Account Access.**This is mostly used in cases where a 3rd party that has a different account has to work in an organizations AWS account. Instead of creating an account for them, you can create a role and specify that their account ID can assume that role.
* This also applies in case where we have security auditors. An organization that has a team that audits the security in AWS accounts, you may need to grant them permission for read only access to see what you are doing in AWS. So you add their AWS account ID to the trust policy for the role that is created for them.

To create a role:

1. Go to IAM and under Role, click on create Role
2. You will be prompted to select a trusted entity which could either be an AWS service such as an EC2 instance, Another AWS account, Web Identity, or a SAM 2.0 federation.
3. Assuming the trust entity is another AWS account, you will enter the Account ID and click Next
4. You attach the permission policy to the account
5. Enter the name of the Role and description and create the role.

Once the role is created ( say for security auditors, ) you need to give your account ID and the name of the role to security auditors.

1. Go under your name and click on switch Role and the name will change to the name of the role that you just created.

**IAM API KEYS:**

* API access keys are just texts  used to sign a programatic request to AWS. They are required to make calls to AWS from the
* CLI
* Tools for windows power shell
* AWS SDKs
* Direct HTTP calls using the API for individual AWS services.
* **There are 02 parts to API access keys:** The access key ID and the secret Access key (only available when you create an account)
* The purpose of a key is to provide a request and prove that the request was made by someone who has access to make the request
* Access keys are permanent and don’t have an expiration.

For example: API credentials are used by developers configured in an IDE

**Important API key facts:**

1. API keys are only available ONE time, when a new user is created or when you reissue a new set of keys.
2. AWS will not regenerate the same set of access keys if you require new set of API credentials

**Security Token Service (STS)**

Roles also use access keys, but the access keys that they use comes from STS. STS allows you to create temporary security credentials that grant trusted users access to your AWS resources.

* These are temporary credentials for short term use only with a configurable session duration between 15 minutes and 12 or 36 hours. ]
* Once expired, they can no longer be used to access your AWS resources
* When requested through an STS API call, a credential object is returned containing
* Security token
* Access key ID
* Secret access key
* Expiration time stamp
* **STS Benefits**:
* Avoid distributing or embedding long term AWS security credentials in an application.
* Grant access to AWS without having to create an IAM user
* Since the credentials are temporary, you don’t have to manually rotate or revoke them.
* **When to use STS**

* **Identity Federation:**
  + Enterprise identity federation (authenticate through your companies network)
    - STS supports security assertion markup language **(SAML),** which allows for use of Microsoft active directory (of your own solutions)
  + Web Identity federation (3rd party identity providers such as Facebook, google, amazon)
* **Roles for cross account access:**
  + Used for organizations that have more than one AWS account.
* **Roles for Amazon EC2 (and other AWS services)**
  + Grant access to an application running on an EC2 instance to access other AWS services without having to embed credentials.

There are 5 different API calls we can make with STS:

[ADD SCREEN SHOT HERE]

**IDENTITY FEDERATION**

Federation enables you to manage access to your AWS cloud resources centrally. This is a way of authenticating users or applications outside the normal IAM user process. With federation, you can use single sign on (SSO) to access your AWS account using credentials from your corporate directory. Federation uses open standards such as Security assertion markup language 2.0 (SAML) to exchange identity and security information between an identity provider and an application.

* What if you have a large organization  with a lot of people that need to log into the console or need programmatic access to AWS? The task of creating and managing a lot of IAM users and groups could seem daunting particularly when you already have in place some type of identity broker in-house such as an **active directory** with all the company's employees associated by group or departments already in the directory.  It will be a lot of effort to maintain 02 separate directories. AWS allows you to do an identity federation.
* There are 03 different types of identity federation offered by AWS:
  + **Custom Identity Provider**:  This is when you have an already on prem identity provider that is used to authenticate users to other on prem application.  When you want to use the  custom identity provider you obtain credentials using STS.
    - For example, assume a user wants to access an internal application that needs access to the AWS console, the user will log into the identity broker application which validates the identity. Once authenticated, the identity makes a call to STS to get a call to get a set of credentials. Once the credentials are received and forwarded to the client who can start accessing AWS services with these credentials.
    - This means a code will need to be written in the identity broker to make the call to STS.

* + **LDAP/ Active Directory**:  This uses SAML which is an open standard for changing authentication and authorization data between 02 different parties such as your active directory and AWS.
  + **Web Identity**:  Here you typically use in web based applications or mobile app. Instead of having every user of your application needing a set of application credentials they can log in using an open ID connect provider such as Facebook, google etc.

Amazon Cognito lets you add user sign-up , sign-in and access control to your web and mobile applications quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers such as facebook, google and amazon and enterprise Identity providers via SAML 2.0

**Delegate Access Across AWS Accounts Using IAM Roles**

This section teaches you how to use a role to delegate access to resources that are in different AWS accounts that you own (Production and Development). You share resources in one account with users in a different account. By setting up cross-account access in this way, you don't need to create individual IAM users in each account. In addition, users don't have to sign out of one account and sign into another in order to access resources in different AWS accounts.

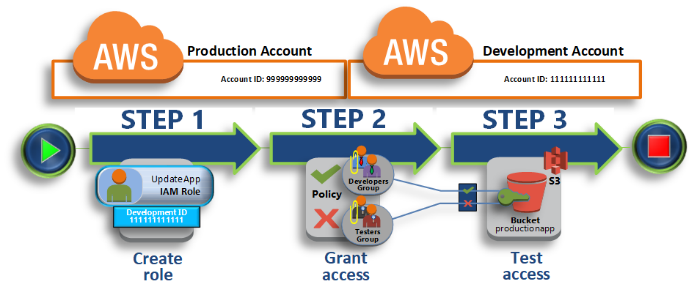
In this tutorial, imagine that the Production account is where live applications are managed. The Development account is a sandbox where developers and testers can freely test applications. In each account, application information is stored in Amazon S3 buckets. You manage IAM users in the Development account, where you have two IAM groups: Developers and Testers. Users in both groups have permissions to work in the Development account and access resources there. From time to time, a developer must update the live applications in the Production account. These applications are stored in an Amazon S3 bucket called production app.

At the end of this tutorial, you have the following:

* Users in the Development account (the trusted account) that are allowed to assume a specific role in the Production account.
* A role in the Production account (the trusting account) that is allowed to access a specific Amazon S3 bucket.
* The production app bucket in the Production account.

Developers can use the role in the AWS Management Console to access the production app bucket in the Production account. They can also access the bucket by using API calls that are authenticated by temporary credentials provided by the role. Similar attempts by a Tester to use the role fail.

This workflow has three basic steps.



[Step 1: Create a Role](https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html#tutorial_cross-account-with-roles-1)

First, you use the AWS Management Console to establish trust between the Production account (ID number 999999999999) and the Development account (ID number 111111111111). You start by creating an IAM role named UpdateApp. When you create the role, you define the Development account as a trusted entity and specify a permissions policy that allows trusted users to update the production app bucket.

[Step 2: Grant Access to the Role](https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html#tutorial_cross-account-with-roles-2)

In this step of the tutorial, you modify the IAM group policy so that Testers are denied access to the UpdateApp role. Because Testers have PowerUser access in this scenario, we must explicitly deny the ability to use the role.

Add the following policy statement to allow the AssumeRole action on the UpdateApp role in the Production account. Be sure that you change *PRODUCTION-ACCOUNT-ID* in the Resource element to the actual AWS account ID of the Production account.

{   
  "Version": "2012-10-17",   
  "Statement": {   
    "Effect": "Allow",   
    "Action": "sts:AssumeRole",   
    "Resource": "arn:aws:iam::*PRODUCTION-ACCOUNT-ID*:role/UpdateApp"   
  }   
}

[Step 3: Test Access by Switching Roles](https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html#tutorial_cross-account-with-roles-3)

Finally, as a Developer, you use the UpdateApp role to update the production app bucket in the Production account. You see how to access the role through the AWS console, the AWS CLI, and the API.

**AWS ORGANIZATIONS:**

Increasingly, companies that use AWS find that they end up with multiple AWS accounts to manage. Large organizations usually have the desire to enforce some time of control on which services each account should be implemented and how much money they should be spending on their account. Large organizations need this visibility for security and cost controls.  So, AWS has come up with an easy way to help control sub accounts within an organization using AWS organizations:

* AWS organizations allows you to create a root or master AWS account and invite other accounts within the organization to join that organization beneath that root account. You can group accounts into organizational units and apply a service control policy to accounts or organizational units that will restrict what these accounts can do.
* These policies are very similar to IAM policies and will prevent a unit from doing something that it shouldn’t do.
* These units can follow the org chart of the company, or departments such as finance, HR, etc
* The root account should be able to apply service control policies at the organizational level.

**CONSOLIDATED BILLING:**

* Every account created in AWS has to have a payment method associated with it. AWS organizations allow you to have all your account bills roll up to the root account and the bill pays the bill for the rest of the account.

**Boot Camp Objectives**

* Understand and use essential tools for handling files, directories, command-line environments, and documentation
* Operate running systems, including booting into different run levels, identifying processes, starting and stopping virtual machines, and controlling services
* Configure local storage using partitions and logical volumes
* Create and configure file systems and file system attributes, such as permissions, encryption, access control lists, and network file systems
* Deploy, configure, and maintain systems, including software installation, update, and core services
* Manage users and groups, including use of a centralized directory for authentication
* Manage security, including basic firewall and SELinux configuration

**January 25, 2020: 1:00pm to 9:30pm- Linux Session 1**

**1. Using Essential Tools**

1. **Introduction**

An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn't function.

What is LINUX?

Linux is a family of free and open-source software operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991 by Linus Torvalds. Linux is typically packaged in a Linux distribution.

What is Linux Distribution?

A **Linux distribution** (often abbreviated as **distro**) is an operating system made from a software collection, which is based upon the **Linux** kernel and, often, a package management system. ... The software is usually adapted to the **distribution** and then packaged into software packages by the **distribution's**maintainers. Common Linux Distributions are:

* Slackware
* Debian
* Red hat Enterprise Linux
* SUSE
* Ubuntu
* Fedora
* OpenSUSE
* CentOS

Why use Linux?

This is the one question that most people ask. Why bother learning a completely different computing environment, when the operating system that ships with most desktops, laptops, and servers works just fine?

To answer that question, I would pose another question. Does that operating system you're currently using really work just fine?? Or, do you find yourself battling obstacles like viruses, malware, slow downs, crashes, costly repairs, and licensing fees?

If you struggle with the above, Linux might be the perfect platform for you. Linux has evolved into one of the most reliable computer ecosystems on the planet. Combine that reliability with zero cost of entry and you have the perfect solution for a desktop platform.

That’s right, zero cost of entry… as it's free. You can install Linux on as many computers as you like without paying a cent for software or server licensing.

1. **Basic Shell Skills**

**What is "the Shell"?**

Simply put, the shell is a program that takes commands from the keyboard and gives them to the operating system to perform. In the old days, it was the only user interface available on a Unix-like system such as Linux. Nowadays, we have graphical user interfaces (GUIs) in addition to command line interfaces (CLIs) such as the shell.

On most Linux systems a program called bash (which stands for Bourne Again Shell, an enhanced version of the original Unix shell program, sh, written by Steve Bourne) acts as the shell program. Besides bash, there are other shell programs that can be installed in a Linux system. These include: ksh, tcsh and zsh.

**What's a "Terminal?"**

It's a program called a terminal emulator. This is a program that opens a window and lets you interact with the shell. There are a bunch of different terminal emulators you can use. Most Linux distributions supply several, such as: gnome-terminal, konsole, xterm, rxvt, kvt, nxterm, and eterm.

Basic Linux Commands

ls : lists directory contents

Cat : concatenates files

The cat command is important as a basic command because it serves two very important functions: concatenating (merging) files (as the name suggests) and printing the contents of a file to the screen. Printing the contents of files is by far the more frequent use of this command. If you want to see a file's contents, use the following format:

$ cat <filename>

For example, you might type the following to display the contents of the system's passwd file on the screen:

$ cat /etc/passwd

To use cat for its file concatenation powers, the general form of the command is:

$ cat file1 file2 > file1file2

For example, to redirect the contents of grocerylist.txt and todo\_list.txt into the Saturday.txt file:

$ cat grocerylist.txt todo\_list.txt > Saturday.txt

Touch:  The common use for touch is to create an empty file as a placeholder. Some programs require that a file exists to operate correctly, and this is one method of kickstarting such a process. Otherwise, this use offers a quick way to create a file without opening a text editor and then saving an empty file:

$ touch today.txt

$ ls -l

pwd : prints the working directory

cd : changes directory

Very closely related to the pwd command is the cd command. Changing directories is a frequent activity on a Linux system. As stated before, when you first log in, you're placed into your home directory. Every user on a Linux system has a home directory. Regular user accounts have personal directories under the /home directory. Your home directory is under /home/<username>. To view all user's home directories, cd to the /home directory.

$ cd /home

$ ls

rm : removes files and directories

cp : copies files and directories

$ cp file.txt /opt/files

To copy an entire directory and all its contents, including subdirectories, use the -R (Recurse) option. Copy the data directory in your home directory to /opt/files. You can use either the -r or -R to recurse copy files:

$ cp -R data /opt/files

mkdir : makes directories

If you're an organized person, you'll want to create directories to satisfy your need to properly arrange your files and data into separate compartments (directories). It's easy to create directories. Issue the mkdir command followed by the directory name you wish to create:

$ mkdir data

$ mkdir -p data/documents/forms/tests/outgoing

The -p option tells the system that you are creating a parent directory and subdirectories.

1. **Editing Files with vi/vim**

Vim is a text editor to create or edit a text file, config file and programs on a Linux, macOS, and Unix-like system. There are two modes in vim:

1. **Command mode** : In this mode you can move around the file, delete text, copy/paste/undo/redo and more.
2. **Insert mode** : In this mode you can insert text or edit text.

**How do I change mode from one to another when using vim?**

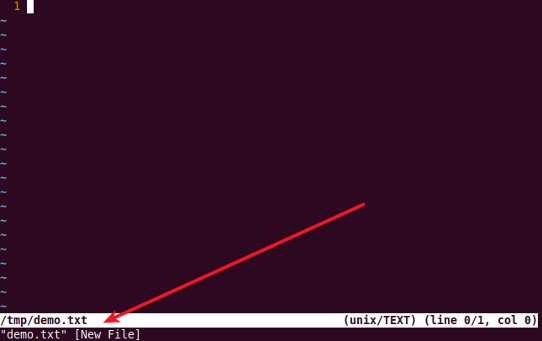
To switch from insert mode to command mode press or type **escape** key (Esc). If you are ever unsure about something you typed, just press ESC to place you in Normal mode. Then retype the command you wanted.

**TO SWITCH FROM COMMAND MODE TO INSERT MODE TYPE ANY ONE OF THE FOLLOWING CHARACTERS:**

* **a** : Append text following the current cursor position
* **A** : Append text to the end of the current line
* **i** : Insert text at the current cursor position
* **I** : Insert text at the beginning of the cursor line
* **o** : Begin a new line below the cursor to insert text
* **O** : Begin a new line above the cursor to add text

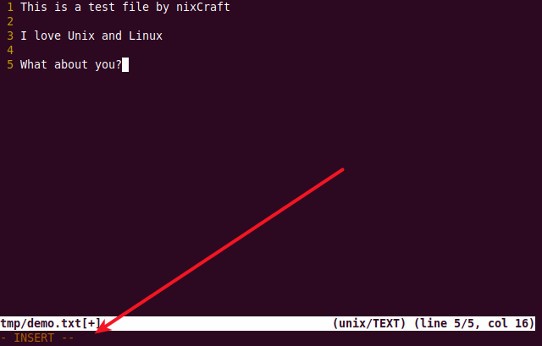
**Step to create and edit a file using vim**

1. Log into your server using SSH command: ssh user@cloud-vm-ip
2. Type vim command to create a file named demo.txt: vim demo.txt



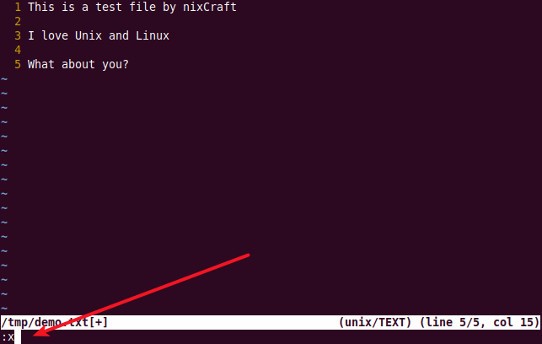
Vim started with a new blank file

1. To enter insert mode and to append text type the letter i (usually you do this by press Esc followed by i)



Enter INSERT mode in vim

1. Start entering text
2. When done editing press the Esc key to go back to command mode
3. Once in command mode to save and exit the file type a colon (:) followed by x. Press the Enter key.



To save the file and exit the vim

**How to quit the vi/vim editor without saving your changes**

You can quit the vim or vi text editor without saving any changes you may have made to the file as follows:

1. First, you need to press Esc key to get out of insert or append mode.
2. Next type colon (: symbol)
3. You see the cursor at the lower left corner of the screen.
4. Finally, type the following command to quit without saving anything to a file:   
   q!

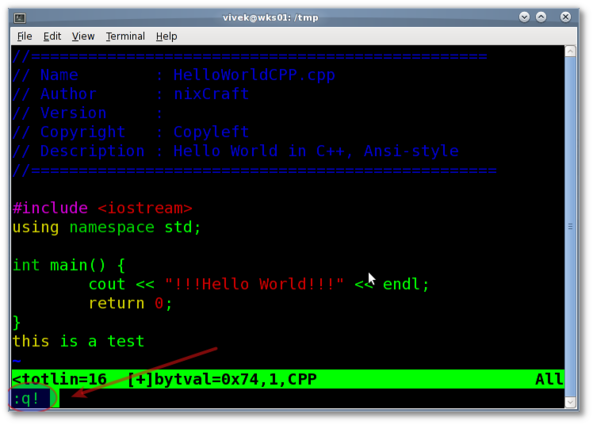


Fig.01: Vim / vi text editor quit without saving changes command demo

1. **Understanding the Shell Environment**
2. **Finding Help**

**2. Essential File Management Tools**

* Working with the File System Hierarchy
* Managing Files
* Using Links
* Working with Archives and Compressed Files

**3. User and Group Management**

* **Managing Users and Groups**

There are four main user administration files −

* /etc/passwd − Keeps the user account and password information. This file holds the majority of information about accounts on the Unix system.
* /etc/shadow − Holds the encrypted password of the corresponding account. Not all the systems support this file.
* /etc/group − This file contains the group information for each account.
* /etc/gshadow − This file contains secure group account information.

Check all the above files using the cat command.

The following table lists out commands that are available on majority of Linux systems to create and manage accounts and groups

|  |  |
| --- | --- |
| **Sr.No.** | **Command & Description** |
| 1 | **useradd**  Adds accounts to the system |
| 2 | **usermod**  Modifies account attributes |
| 3 | **userdel**  Deletes accounts from the system |
| 4 | **groupadd**  Adds groups to the system |
| 5 | **groupmod**  Modifies group attributes |
| 6 | **groupdel**  Removes groups from the system |

* **Create an Account**

Let us see how to create a new account on your Unix system. Following is the syntax to create a user's account −

useradd -d homedir -g groupname -m -s shell -u userid accountname 

The following table lists out the parameters −

|  |  |
| --- | --- |
| **Sr.No.** | **Option & Description** |
| 1 | -d homedir  Specifies home directory for the account |
| 2 | -g groupname  Specifies a group account for this account |
| 3 | -m  Creates the home directory if it doesn't exist |
| 4 | -s shell  Specifies the default shell for this account |
| 5 | -u userid  You can specify a user id for this account |
| 6 | accountname  Actual account name to be created |

If you do not specify any parameter, then the system makes use of the default values. The useradd command modifies the /etc/passwd, /etc/shadow, and /etc/group files and creates a home directory.

* **Create a Group**

We will now understand how to create a group. For this, we need to create groups before creating any account otherwise, we can make use of the existing groups in our system. We have all the groups listed in */etc/groups* file.

All the default groups are system account specific groups and it is not recommended to use them for ordinary accounts. So, following is the syntax to create a new group account −

 groupadd [-g gid [-o]] [-r] [-f] groupname 

The following table lists out the parameters −

|  |  |
| --- | --- |
| **Sr.No.** | **Option & Description** |
| 1 | -g GID  The numerical value of the group's ID |
| 2 | -o  This option permits to add group with non-unique GID |
| 3 | -r  This flag instructs groupadd to add a system account |
| 4 | -f  This option causes to just exit with success status, if the specified group already exists. With -g, if the specified GID already exists, other (unique) GID is chosen |
| 5 | groupname  Actual group name to be created |

If you do not specify any parameter, then the system makes use of the default values.

Following example creates a *developers* group with default values, which is very much acceptable for most of the administrators.

$ groupadd developers 

* **Modify a Group**

To modify a group, use the groupmod syntax −

$ groupmod -n new\_modified\_group\_name old\_group\_name 

To change the developers\_2 group name to developer, type −

$ groupmod -n developer developer\_2 

Here is how you will change the financial GID to 545 −

$ groupmod -g 545 developer 

**Exam Question**: Create User Account

Create the following users, group and group membership:

adminuser group

#groupadd adminuser

User natasha, using adminuser as a sub group

**Option1**

#useradd natasha

#usermod -aG adminuser natasha

#id natasha

Option2:

#useradd -G adminuser natasha

#id natasha

User Harry, also using adminuser as a sub group

User sarah, cannot access the SHELL which is interactive in the system, and is not a member of adminuser,

natasha, Harry, sarah password is redhat.

**Solutions**:

#groupadd adminuser

#useradd natasha -G adminuser

#useradd Harry -G adminuser

#useradd sarah -s /sbin/nologin

#Passwd user name // to modify password or echo redhat | passwd --stdin user name id natasha

// to view user group.

**4. Permissions Management**

* Managing File Ownership

Just as every user has an ID and is a member of one primary group, so every file on a Linux system has one owner and one group associated with it.

Use the ls -l command to display the owner and group.

$ chmod u+x file\_name

Or, to add read and write permissions for the group that owns the file, you would run:

$ chmod g+rw file\_name

* Managing Basic Permissions

The commands for modifying file permissions and ownership are:

chmod – change permissions

chown – change ownership.

# chown user:user file1

# chown user:group filename

Instead of adding permissions, the symbolic syntax of chmod can also be used to subtract or set to some absolute value as shown in these examples:

$ chmod o-w file\_name

$ chmod u=rwx,g=rx,o= file\_name

The chmod command can also explicitly set permissions using a numerical representation. For example, to set permissions on a file to rwxrwxr–, you would run:

$ chmod 774 file\_name

The breakdown of permissions looks like this:

* u – user
* g – group
* o – other

The ‘other’ entry is the dangerous one, as it effectively gives everyone permission for the folder/file. The permissions you can give to a file or folder are:

* r – read
* w – write
* x – execute

Table 2. Numeric permissions

|  |  |
| --- | --- |
| **Symbolic** | **Octal** |
| rwx | 7 |
| rw- | 6 |
| r-x | 5 |
| r-- | 4 |
| -wx | 3 |
| -w- | 2 |
| --x | 1 |
| — | 0 |

#chmod 755 file1

#chmod 66

* Managing Advanced Permissions

* Setting Default Permissions with umask

You can simply subtract the umask from the base permissions to determine the final permission for file as follows:

666 – 022 = 644

File base permissions : 666

umask value : 022

subtract to get permissions of new file (666-022) : 644 (rw-r–r–)

**5. Managing Software**

* Managing Software Packages with yum
* Using yum

**CHECKING FOR AND UPDATING PACKAGES**

To see which installed packages on your system have updates available, use the following command:

yum check-update

To update a single package, run the following command as root:

yum update package\_name

To update all packages and their dependencies, enter yum update (without any arguments):

yum update

**UPDATING PACKAGES AUTOMATICALLY**

It is also possible to set up periodical automatic updates for your packages.

For this purpose, Red Hat Enterprise Linux uses the yum-cron package. It provides a Yum interface for

the cron daemon and downloads metadata from your package repositories.

Metadata is a set of data that describes and gives information about other data.

With the yum-cron service enabled, the user can schedule an automated daily Yum update as a cron

job.

To install yum-cron issue the following command:

yum install yum-cron

By default, the yum-cron service is disabled and needs to be activated and started manually:

chkconfig yum-cron on

service yum-cron start

To verify the status of the service, run the following command:

service yum-cron status

**INSTALLING PACKAGES**

Yum allows you to install both a single package and multiple packages, as well as a package group of

your choice.

To install a single package and all of its non-installed dependencies, enter a command in the following

form:

yum install package\_name

You can also install multiple packages simultaneously by appending their names as arguments:

yum install package\_name package\_name…

**REMOVING PACKAGES**

Similarly to package installation, Yum allows you to uninstall (remove in RPM and Yum terminology)

both individual packages and a package group.

To uninstall a particular package, as well as any packages that depend on it, run the following command

as root:

yum remove package\_name…

* Managing Software Packages with rpm

**February 15, 2020: 9:00am to 8:00pm- Linux session 2**

**6. Managing Processes**

* Introduction to Process Management
* Using Common Command-Line Tools for Process Management
* Using top to Manage Processes
* Using tuned to Optimize Performance

**7. Scheduling Tasks**

* Configuring Cron to Automate Recurring Tasks
* Configuring at to Schedule Future Tasks

**8. Managing Storage**

* Understanding MBR and GPT Partitions
* Managing Partitions and File Systems
* Mounting File Systems

**9. Managing Advanced Storage**

* Understanding LVM
* Creating LVM Logical Volumesv
* Resizing LVM Logical Volumes

**February 29 2020 8:00am to 8:0pm- Linux session 3**

**10. Configuring Networking**

* Networking Fundamentals
* Managing Network Addresses and Interfaces
* Validating Network Configuration
* Configuring Network Configuration with nmtui and nmcli
* Setting Up Hostname and Name Resolution

**11. Essential Troubleshooting Skills**

TBD

**12. An Introduction to Bash Shell Scripting**

* Understanding Shell Scripting Core Elements
* Using Variables and Input
* Using Conditional Loops

**13. Configuring SSH (Read)**

* Hardening the SSH Server
* Using Other Useful sshd Options
* Configuring Key-Based Authentication with Passphrases

**14. Managing Apache HTTP Services**

* Configuring a Basic Apache Server
* Understanding Apache Configuration Files
* Creating Apache Virtual Hosts

**15. Managing SELinux**

* Understanding SELinux Working and Modes
* Understanding Context Settings and the Policy
* Restoring Default File Contexts
* Using Boolean Settings to Modify SELinux Settings
* Diagnosing and Addressing SELinux Policy Violations

**16. Configuring a Firewall**

* Understanding Linux Firewalling
* Working with Firewalld

**17. Accessing Network Storage**

* Using NFS Services
* Using CIFS Services
* Mounting Remote File Systems Through fstab
* Using Automount to Mount Remote File Systems

**18. Configuring Time Services**

* Understanding Local Time
* Using Network Time Protocol
* Managing Time on Red Hat Enterprise Linux