

AWS Certified SysOps Administrator - Associate (CSOA) Complete Training Guide with Practice Labs

100+ Exam Practice Questions



- COVERS COMPLETE EXAM BLUEPRINT
- 100+ EXAM PRACTICE QUESTIONS
- COVERING 100% OF EXAM OBJECTIVES
- READY TO PRACTICE LABS
- EFFECTIVELY DEMONSTRATE AN OVERALL UNDERSTANDING OF THIS TRACK
- ENABLES YOU TO PASS THE EXAM IN YOUR VERY FIRST ATTEMPT



AWS Certified SysOps Administrator - Associate (CSOA) Complete Training Guide with Practice Labs

100+ Exam Practice Questions



- COVERS COMPLETE EXAM BLUEPRINT
- 100+ EXAM PRACTICE QUESTIONS
- COVERING 100% OF EXAM OBJECTIVES
- READY TO PRACTICE LABS

- EFFECTIVELY DEMONSTRATE AN OVERALL UNDERSTANDING OF THIS TRACK
- ENABLES YOU TO PASS THE EXAM IN YOUR VERY FIRST ATTEMPT

[in ip-specialist-global](#)

[f @IPSpecialist.net](#)

[t @IPspecialistnet](#)

[www.ipspecialist.net](#)

Document Control

Proposal Name : AWS Certified SysOps
Administrator - Associate Workbook

Document Version : 1.0

Document Release Date : 20 Jun 2018

Reference : WB_AWS_CSOA

Copyright © 2018 IPSpecialist LTD.

Registered in England and Wales

Company Registration No: 10883539

Registration Office at Office 32, 19-21 Crawford Street, London W1H 1PJ,
United Kingdom

www.ipspecialist.net

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from IPSpecialist LTD, except for the inclusion of brief quotations in a review.

Feedback:

If you have any comments regarding the quality of this book, or otherwise alter it to suit your needs better, you can contact us through email at info@ipspecialist.net

Please make sure to include the book title and ISBN in your message

About IPSpecialist

IPSpecialist LTD. IS COMMITTED TO EXCELLENCE AND DEDICATED TO YOUR SUCCESS.

Our philosophy is to treat our customers like family. We want you to succeed, and we are willing to do anything possible to help you make it happen. We have the proof to back up our claims. We strive to accelerate billions of careers with great courses, accessibility, and affordability. We believe that continuous learning and knowledge evolution are most important things to keep re-skilling and up-skilling the world.

Planning and creating a specific goal is where IPSpecialist helps. We can create a career track that suits your visions as well as develop the competencies you need to become a professional Network Engineer. We can also assist you with the execution and evaluation of proficiency level based on the career track you choose, as they are customized to fit your specific goals.

We help you STAND OUT from the crowd through our detailed IP training content packages.

Course Features:

Self-Paced learning

- Learn at your own pace and in your own time

Covers Complete Exam Blueprint

- Prep-up for the exam with confidence

Case Study Based Learning

- Relate the content to real-life scenarios

Subscriptions that suits you

- Get more pay less with IPS Subscriptions

Career Advisory Services

- Let industry experts plan your career journey

Virtual Labs to test your skills

- With IPS vRacks, you can testify your exam preparations

Practice Questions

- Practice Questions to measure your preparation standards

On Request Digital Certification

- On request, digital certification from IPSpecialist LTD.

About the Authors:

This book has been compiled with the help of multiple professional engineers. These engineers specialize in different fields, e.g., Networking, Security, Cloud, Big Data, & IoT. Each engineer develops content in its specialized field that is compiled to form a comprehensive certification guide.

About the Technical Reviewers:

Nouman Ahmed Khan

AWS-Architect, CCDE, CCIEX5 (R&S, SP, Security, DC, Wireless), CISSP, CISA, CISM is a Solution Architect working with a major telecommunication provider in Qatar. He works with enterprises, mega-projects, and service providers to help them select the best-fit technology solutions. He also works closely as a consultant to understand customer business processes and helps select an appropriate technology strategy to support business goals. He has more than 14 years of experience working in Pakistan/Middle-East & UK. He holds a Bachelor of Engineering Degree from NED University, Pakistan, and M.Sc. in Computer Networks from the UK.

Abubakar Saeed

Abubakar Saeed has more than twenty-five years of experience, Managing, Consulting, Designing, and implementing large-scale technology projects, extensive experience heading ISP operations, solutions integration, heading Product Development, Presales, and Solution Design. Emphasizing on adhering to Project timelines and delivering as per customer expectations, he always leads the project in the right direction with his innovative ideas and excellent management.

Syed Hanif Wasti

Syed Hanif Wasti is a Computer science graduate working professionally as a Technical Content Developer. He is a part of a team of professionals operating in the E-learning and digital education sector. He holds a bachelor's degree in Computer Sciences from PAF-KIET, Pakistan. He has completed training of MCP and CCNA. He has both technical knowledge and industry sounding information, which he uses perfectly in his career. He was working as a Database and Network administrator while having experience of software development

Muhammad Yousuf

Muhammad Yousuf is a professional technical content writer. He is Cisco Certified Network Associate in Routing and Switching, holding bachelor's degree in Telecommunication Engineering from Sir Syed University of Engineering and Technology. He has both technical knowledge and industry sounding information, which he uses perfectly in his career.

Table of Contents

About this Workbook

AWS Cloud Certifications

Role-Based Certifications

Specialty Certifications

AWS Certified SysOps Administrator – Associate

Pricing

Exam Length

Exam Content

Exam Results

Exam Validity

How to become an AWS Certified SysOps Administrator - Associate?

Chapter 1: Introduction to AWS

Amazon Web Services Cloud Platform

Introduction to Cloud Computing

Advantages of Cloud Computing

Types of Cloud Computing

Cloud Computing Deployments Models

The Cloud Computing Difference

IT Assets Become Programmable Resources

Global, Available, and Unlimited Capacity

Higher Level Managed Services

Security Built In

AWS Cloud Economics

AWS Virtuous Cycle

AWS Cloud Architecture Design Principles

Scalability

[Disposable Resources Instead of Fixed Servers](#)

[Automation](#)

[Loose Coupling](#)

[Services, Not Servers](#)

[Databases](#)

[Removing Single Points of Failure](#)

[Optimize for Cost](#)

[Caching](#)

[Security](#)

[AWS Global Infrastructure](#)

[What is a Region?](#)

[What is an Availability Zone?](#)

[What is an Edge Location?](#)

[Chapter 2: Monitoring, Metrics & Analysis](#)

[Introduction](#)

[Amazon CloudWatch](#)

[Introduction](#)

[CloudWatch Metrics](#)

[How Amazon CloudWatch Works?](#)

[How long are CloudWatch metrics Stored?](#)

[CloudWatch Alarms](#)

[Lab: 2.1 Create a Cloud Watch Role](#)

[Monitoring EC2](#)

[AWS EC2 Metrics](#)

[Custom Metrics](#)

[EC2 Status Check](#)

[Lab 2.2 Launch an Instance](#)

[Lab 2.3 Create CloudWatch Dashboard](#)

[Monitoring EBS](#)

[Amazon EBS Volume Types](#)

[General Purpose SSD – IOPS and Volumes](#)

[Pre-Warming EBS Volumes](#)

[EBS CloudWatch Metrics](#)

[Modifying EBS Volumes](#)

[Monitoring RDS](#)

[RDS Metrics](#)

[Monitoring ELB](#)

[ELB Metrics](#)

[Monitoring ElastiCache](#)

[ElastiCache Engines](#)

[AWS Organizations](#)

[Key Features of AWS Organizations](#)

[Consolidated Billing](#)

[Lab 2.4 Make an AWS Organization:](#)

[Monitor Charges Using Billing Alarms](#)

[Cost Optimization](#)

[On-Demand Instance](#)

[Reserved Instance](#)

[Spot Instance](#)

[Chapter 3: High Availability](#)

[Introduction](#)

[Fault Tolerance and High Availability](#)

[Elasticity and Scalability:](#)

[Introduction:](#)

[Elasticity:](#)

[Scalability:](#)

[Old Table Network Performance:](#)

[Instances Types:](#)

[EBS Optimized Network Performance:](#)

[Amazon Relational Database Service \(RDS\):](#)

[RDS Multi Failover:](#)

[Failover Conditions:](#)

[RDS Using Read Replicas:](#)

[Lab 3.1 RDS Multi-AZ Deployment Read Replicas](#)

[Bastion Host and High Availability:](#)

[Overview of the lab:](#)

[What is the Bastion Host:](#)

[Deployment Steps:](#)

[Troubleshooting and Potential Auto Scaling:](#)

[Mind Map:](#)

[Chapter 4: Deployment and Provisioning](#)

[Introduction](#)

[AWS Deployment Services](#)

[AWS Elastic Beanstalk](#)

[AWS CloudFormation](#)

[AWS OpsWorks](#)

[What is AWS OpsWorks?](#)

[Overview](#)

[What is Chef?](#)

[Root/Admin Access Services](#)

[Elastic Load Balancing](#)

[Introduction](#)

[ELB Configurations](#)

[Sticky Sessions](#)

[Lab 4.1 Elastic Load Balancer Configuration](#)

[Pre-Warming Elastic Load Balancer](#)

[Pre-Warming The Load Balancer:](#)

Chapter 5: Data Management

Disaster Recovery

Traditional Approaches to DR

Using AWS for DR

AWS Features and Services Essential for Disaster Recovery

Regions

Storage

Compute

Networking

Databases

Deployment Orchestration

RTO and RPO

Recovery Time Objective (RTO)

Recovery Point Objective (RPO)

Disaster Recovery Scenarios with AWS

Backup and Restore

Pilot Light for Quick Recovery

Warm Standby Solutions

Multi-Site Solution

Failing Back from a Disaster

Backup and restore

Pilot light, warm standby, and multi-site

Chapter 6: Security

Security Token Service (STS)

Common Scenarios for Temporary Credentials

Case Scenario

AWS Shared Responsibility Model

AWS Security Responsibilities

Customer Security Responsibilities

[AWS Global Infrastructure Security](#)

[AWS Compliance Program](#)

[Certifications / Attestations:](#)

[Laws, Regulations, and Privacy:](#)

[Alignments / Frameworks:](#)

[Physical and Environmental Security](#)

[Storage Device Decommissioning](#)

[Network Security](#)

[Transmission Protection](#)

[Amazon Corporate Segregation](#)

[Network Monitoring and Protection](#)

[AWS Account Security Features](#)

[AWS Credentials](#)

[AWS Trusted Advisor Security Checks](#)

[Amazon EC2 Security](#)

[Multiple Levels of Security](#)

[The Hypervisor](#)

[Instance Isolation](#)

[Amazon EBS Security](#)

[Amazon ELB Security](#)

[AWS Direct Connect Security](#)

[Auditing on AWS](#)

[Chapter 7: Networking](#)

[What is DNS?](#)

[Internet Protocol \(IP\)](#)

[Top Level Domain \(TLD\)](#)

[Domain Name Registration](#)

[DNS Records](#)

[Time to Live \(TTL\)](#)

[Alias Records](#)

[Introduction to Route 53](#)

[DNS Management](#)

[Traffic Management](#)

[Availability Monitoring](#)

[Domain Registration](#)

[Lab 7.1: Register a domain name – Route 53](#)

[Introduction to VPC](#)

[VPC Configuration Scenarios](#)

[Scenario 1: VPC with a Single Public Subnet](#)

[Scenario 2: VPC with Public and Private Subnets \(NAT\)](#)

[Scenario 3: VPC with Public and Private Subnets and Hardware VPN Access](#)

[Scenario 4: VPC with a Private Subnet and Hardware VPN Access](#)

[VPC Connectivity Options](#)

[Network-to-Amazon VPC Connectivity Options](#)

[Amazon VPC-to-Amazon VPC Connectivity Options](#)

[Internal User-to-Amazon VPC Connectivity Options](#)

[Hardware VPN](#)

[AWS Direct Connect](#)

[Software VPN](#)

[AWS Direct Connect+ VPN](#)

[AWS VPN Cloud Hub](#)

[Lab 7.2 Build A Custom VPC](#)

[Lab 7.3 Custom VPC with Private Subnet](#)

[Lab 7.4 Creating a NAT instance](#)

[Lab 7.5 Network ACLs Vs. Security groups](#)

About this Workbook

This Workbook provides an in-depth understanding and complete course material to pass the AWS Certified SysOps Administrator – Associate Exam. The workbook is designed to take a practical approach to learn real-life examples and case studies.

- Covers complete Exam Blueprint
- Summarized content
- Case Study based approach
- Ready to practice labs
- Exam tips
- Mind maps
- 100% pass guarantee

AWS Cloud Certifications

AWS Certifications are industry-recognized credentials that validate your technical cloud skills and expertise while assisting in your career growth. These are one of the most valuable IT certifications right now since AWS has established an overwhelming lead in the public cloud market. Even with the presence of several tough competitors such as Microsoft Azure, Google Cloud Engine, and Rackspace, AWS is by far the dominant public cloud platform today, with an astounding collection of proprietary services that continues to grow.

The two key reasons as to why AWS certifications are prevailing in the current cloud-oriented job market:

- There's a dire need for skilled cloud engineers, developers, and architects – and the current shortage of experts is expected to continue into the foreseeable future.
- AWS certifications stand out for their thoroughness, rigour, consistency, and appropriateness for critical cloud engineering positions.

Value of AWS Certifications

AWS places equal emphasis on sound conceptual knowledge of its entire platform, as well as on hands-on experience with the AWS infrastructure and its many unique and complex components and services.

Individuals

- Demonstrate your expertise to design, deploy, and operate highly available, cost-effective, and secure applications on AWS
- Gain recognition and visibility for your proven skills and proficiency with AWS

- Earn tangible benefits such as access to the AWS Certified LinkedIn Community, invite to AWS Certification Appreciation Receptions and Lounges, AWS Certification Practice Exam Voucher, Digital Badge for certification validation, AWS Certified Logo usage, access to AWS Certified Store
- Foster credibility with your employer and peers

Employers

- Identify skilled professionals to lead IT initiatives with AWS technologies
- Reduce risks and costs to implement your workloads and projects on the AWS platform
- Increase customer satisfaction

Types of Certification

Role-Based Certifications:

- ***Foundational*** - Validates the overall understanding of the AWS Cloud. Prerequisite to achieving Specialty certification or an optional start towards Associate certification.
- ***Associate*** - Technical role-based certifications. No prerequisite.
- ***Professional*** - Highest level technical role-based certification. Relevant Associate certification required.

Specialty Certifications:

- Validate advanced skills in specific technical areas
- Requires one active role-based certification

Certification Roadmap

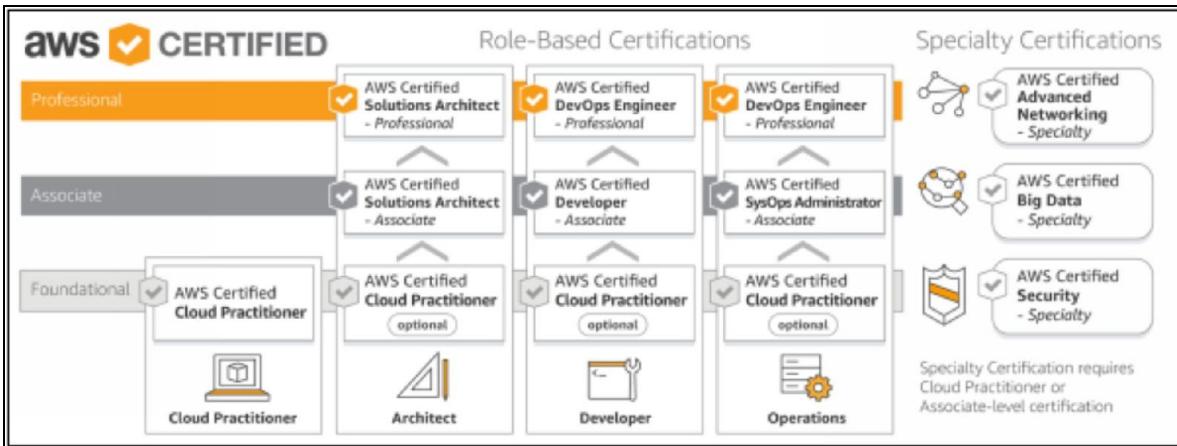


Figure 1. Certification Roadmap

AWS Certified SysOps Administrator – Associate

Operations exams validate technical knowledge for SysOps administrators, systems administrators, and those in a DevOps role who create automatable and repeatable deployments of applications, networks, and systems on the AWS platform.

Overview of AWS SysOps Administrator – Associate Certification

The AWS Certified SysOps Administrator – Associate exam validates technical expertise in deployment, management, and operations on the AWS platform. Exam concepts you should understand for this exam include:

- Deploying, managing, and operating scalable, highly available, and fault-tolerant systems on AWS
- Migrating an existing on-premises application to AWS
- Implementing and controlling the flow of data to and from AWS
- Selecting the appropriate AWS service based on compute, data, or security requirements
- Identifying appropriate use of AWS operational best practices
- Estimating AWS usage costs and identifying operational cost-control mechanisms

The basic knowledge and skills required at this level should include all of the following areas and objective components below:

AWS Knowledge

- Minimum of one year of hands-on experience with the AWS platform
- Professional experience managing/operating production systems on AWS

- A firm grasp of the seven AWS tenets – architecting for the cloud
- Hands-on experience with the AWS CLI and SDKs/API tools
- Understanding of network technologies as they relate to AWS
- Good grasp of fundamental Security concepts with hands-on inexperience in implementing Security controls and compliance requirements

General IT Knowledge

- 1-2 years' experience as a systems administrator in a systems operations role
- Experience understanding virtualization technology
- Monitoring and auditing systems experience
- Knowledge of networking concepts (DNS, TCP/IP, and Firewalls)
- Ability to collaborate with developers and the general business team/company wide

Intended Audience

Eligible candidates for this exam have:

- One or more years of hands-on experience operating AWS-based applications
- Experience provisioning, operating and maintaining systems running on AWS
- Ability to identify and gather requirements to define a solution to be built and operated on AWS
- Capabilities to provide AWS operations and deployment guidance and best practices throughout the life cycle of a project

Course Outline

The table below lists the main content domains and their weightings on the exam.

	Domain	% of Examination
Domain 1	Monitoring and Metrics	15%
Domain 2	High Availability	15%
Domain 3	Analysis	15%
Domain 4	Deployment and Provisioning	15%
Domain 5	Data Management	12%
Domain 6	Security	15%
Domain 7	Networking	13%
Total		100%

Following is the outline of the topic included in this examination; however, the list is not comprehensive.

Domain 1: Monitoring and Metrics

- 1.1 Demonstrate the ability to monitor availability and performance
- 1.2 Demonstrate the ability to monitor and manage billing and cost optimization process

Domain 2: High Availability

- 2.1 Implement scalability and elasticity based on the scenario
- 2.2 Ensure the level of fault tolerance based on business needs

Domain 3: Analysis

- 3.1 Optimize the environment to ensure maximum performance
- 3.2 Identify performance bottlenecks and implement remedies
- 3.3 Identify potential issues on a given application deployment

Domain 4: Deployment and Provisioning

- 4.1 Demonstrate the ability to build the environment to conform with the architected design

4.2 Demonstrate the ability to provision cloud resources and manage implementation automation

Domain 5: Data Management

5.1 Demonstrate the ability to create backups for different services

5.2 Demonstrate the ability to enforce compliance requirements

5.3 Manage backup and disaster recovery processes

Domain 6: Security

6.1 Implement and manage security policies

6.2 Ensure data integrity and access controls when using the AWS platform

6.3 Demonstrate an understanding of the shared responsibility model

6.4 Demonstrate the ability to prepare for security assessment use of AWS
Implement and manage security policies

Domain 7: Network Management

7.1 Demonstrate the ability to implement networking features of AWS

7.2 Demonstrate the ability to implement connectivity features of AWS

Exam Details

Pricing: USD 150

Exam Length: 80 minutes

Exam Content: Test item formats used in this examination are:

- Multiple-choice: examinee selects one option that best answers the question or completes a statement. The option can be embedded in a graphic where the examinee “points and clicks” on their selection choice to complete the test item.
- Multiple-response: examinee selects more than one option that best answers the question or completes a statement.

- Sample Directions: Read the statement or question and from the response options, select only the option(s) that represent the most correct or best answer(s) given the information.

The examinee selects from four (4) or more response options the option(s) that best completes the statement or answers the question. Distractors or wrong answers are response options that examinees with incomplete knowledge or skill would likely choose, but are generally plausible responses fitting into the content area defined by the test objective.

Exam Results: The AWS Certified SysOps Administrator - Associate examination is a pass or fail the exam. The examination is scored against a minimum standard established by AWS professionals who are guided by certification industry best practices and guidelines.

AWS Certification passing scores are set by using statistical analysis and are subject to change. AWS does not publish exam passing scores because exam questions and passing scores are updated to reflect changes in test forms as the content is updated.

Exam Validity: 2 years; Recertification is required every 2 years for all AWS Certifications.

How to become an AWS Certified SysOps Administrator - Associate?

Prerequisites

No prerequisite exam is required. It is recommended to have 1-2 years' experience as a systems administrator in a systems operations role along with the basic knowledge of networking protocols. Also, the candidates should have a basic understanding of IT services and their uses in the AWS Cloud platform.

Exam Preparation Guide

Exam preparation can be accomplished through self-study with textbooks, practice exams, and on-site classroom programs. This workbook provides you with all the information and knowledge to help you pass the AWS Certified Cloud Practitioner Exam. IPSpecialist offers full support for the candidates for them to pass the exam.

Step 1: Take an AWS Training Class

Training can help you advance your technical skills and learn best practices for working with AWS. This training course will assist in exam preparation:

- SysOps on AWS (aws.amazon.com/training/sysops)

Step 2: Review the Exam Guide and Sample Questions

Review the concepts covered in the Exam Blue Print and study the Sample Questions available at AWS website. Exam sample questions help you check your knowledge and pinpoint concepts and areas requiring more study.

Step 3: Read the Official Exam Study Guide

The AWS Certified SysOps Administrator - Associate Exam: Official Study Guide is written by AWS experts that prepare you to demonstrate your networking skills in an examination setting, covering exam objectives while

guiding you through hands-on exercises based on situations you'll likely encounter as an AWS Certified SysOps Administrator.

Step 4: Study AWS Whitepapers and FAQs

Broaden your technical understanding with whitepapers written by the AWS team, independent analysts, and AWS partners.

- AWS Cloud Computing Whitepapers (aws.amazon.com/whitepapers)
 - Overview of Security Processes
 - Storage Options in the Cloud
 - Defining Fault-Tolerant Applications in the AWS Cloud
 - Overview of Amazon Web Services
 - Compliance Whitepaper
 - Architecting for the AWS Cloud
- AWS Documentation (aws.amazon.com/documentation)
- Browse through these FAQs to find answers to commonly raised questions.
 - Amazon EC2
 - Amazon S3
 - Amazon VPC
 - Amazon S3
 - Amazon RDS
 - Amazon SQS

Step 5: Take a Practice Exam

Test your knowledge online in a timed environment by registering at ‘aws.training’.

Step 6: Schedule Your Exam and Get Certified

Schedule your exam at a testing center near you at ‘aws.training’.

Chapter 1: Introduction to AWS

Amazon Web Services Cloud Platform

Amazon Web Services (AWS) is a secure cloud services platform, offering computing power, database storage, content delivery and other functionality on-demand to help businesses scale and grow. AWS cloud products and solutions can be used to build sophisticated applications with increased flexibility, scalability and reliability.

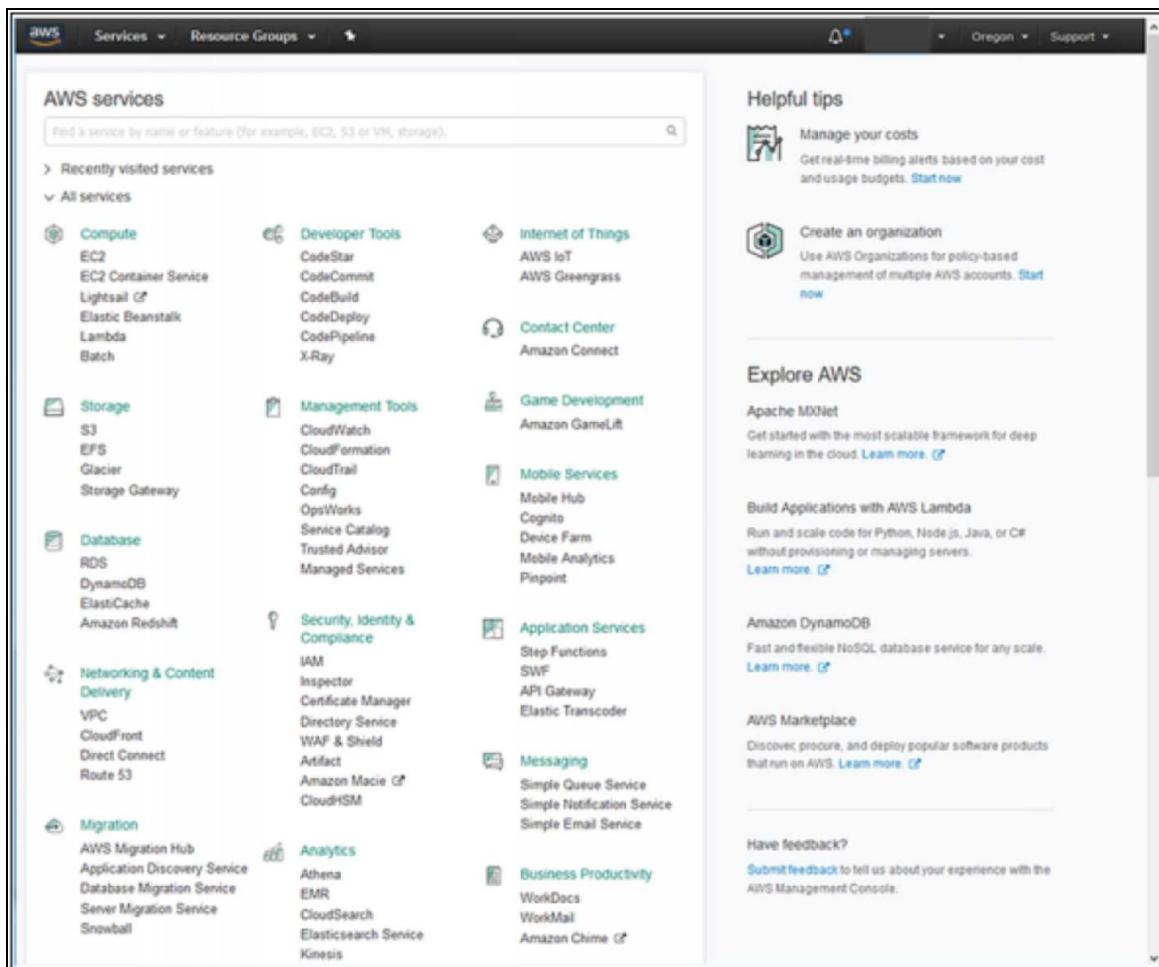


Figure 2. AWS Platform

Introduction to Cloud Computing

Cloud Computing is the practice of using a network of remote servers hosted on the internet to store, manage, and process data rather than using a local server or personal computer. It is the on-demand delivery of computing resources through a cloud services platform with pay-as-you-go pricing.

Advantages of Cloud Computing

1. Trade capital expense for variable expense:

Pay only for the resources consume instead of heavily investing in data centers and servers before knowing your requirements.

2. Benefit from massive economies of scale:

Achieve lower variable costs than you can get on your own. Cloud computing providers such as Amazon build their own data centers and achieve higher economies of scale which results in lower prices.

3. Stop guessing capacity:

Access as much or as little resources needed instead of buying too much or too few resources by guessing your needs. Scale up and down as required with no long-term contracts.

4. Increase speed and agility:

New IT resources are readily available so that you can scale up infinitely with demand. The result is a dramatic increase in agility for the organizations.

5. Stop spending money on running and maintaining data centers:

Eliminates the traditional need for spending money on running and maintaining data centers which are managed by the cloud provider.

6. Go global in minutes:

Provide lower latency at minimal cost by efficiently deploying your application in multiple regions around the world.

Types of Cloud Computing

Infrastructure as a Service (IaaS)	• Provides basic building blocks for cloud IT by offering access to networking features, computers, and data storage space
Platform as a Service (PaaS)	• Manages its own underlying infrastructure, usually hardware and operating systems, and provides application development platform
Software as a Service (SaaS)	• Offers a completed product as a web service that is run and maintained by the service provider along with the management of the underlying infrastructure

Figure 3. Types of Cloud Computing

Cloud Computing Deployments Models

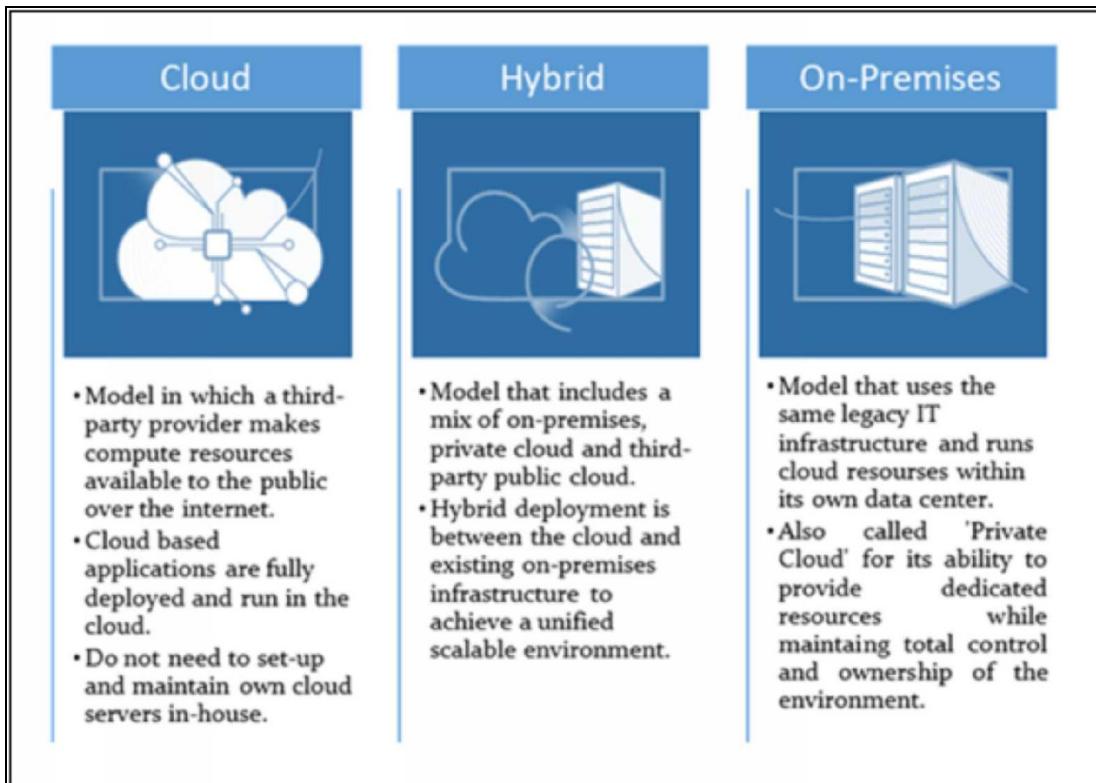


Figure 4. Cloud Deployment Model

The Cloud Computing Difference

This section compares cloud computing with the traditional environment and reviews why these new best practices have emerged.

IT Assets Become Programmable Resources

In a traditional environment, it would take days and weeks depending on the complexity of the environment to set up IT resources such as servers and networking hardware. On AWS, servers, databases, storage, and higher-level application components can be instantiated within seconds. These instances can be used as temporary and disposable resources to meet actual demand, while only paying for what you use.

Global, Available, and Unlimited Capacity

With AWS cloud platform you can deploy your infrastructure into different AWS regions around the world. Virtually unlimited on-demand capacity is available to enable future expansion of your IT architecture. The global support ensures high availability and fault tolerance.

Higher Level Managed Services

Apart from computing resources in the cloud, AWS also provides other higher level managed services such as storage, database, analytics, application, and deployment services. These services are instantly available to developers, consequently reducing dependency on in-house specialized skills.

Security Built In

In a non-cloud environment, security auditing would be a periodic and manual process. The AWS cloud provides plenty of security and encryption features with governance capabilities that enable continuous monitoring of your IT resources. Your security policy can be embedded in the design of your infrastructure.

AWS Cloud Economics

Weighing financial aspects of a traditional environment versus the cloud infrastructure is not as simple as comparing hardware, storage, and compute costs. You have to manage other investments, such as:

- Capital expenditures
- Operational expenditures
- Staffing
- Opportunity costs
- Licensing
- Facilities overhead

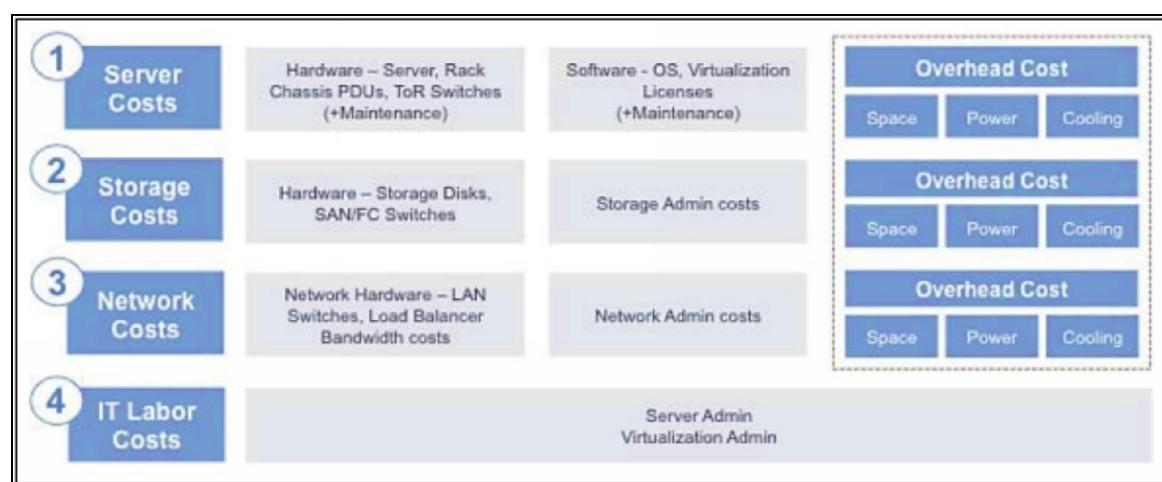


Figure 5. Typical Data Center Costs

On the other hand, a cloud environment provides scalable and powerful computing solutions, reliable storage, and database technologies at lower costs with reduced complexity, and increased flexibility. When you decouple from the data center, you can:

- **Decrease your TCO:** Eliminate the expenses related to building and maintaining data centers or colocation deployment. Pay for only the resources consumed.

- **Reduce complexity:** Reduce the need to manage infrastructure, investigate licensing issues, or divert resources.
- **Adjust capacity on the fly:** Scale up and down resources depending on the business needs using secure, reliable, and broadly accessible infrastructure.
- **Reduce time to market:** Design and develop new IT projects faster.
- **Deploy quickly, even worldwide:** Deploy applications across multiple geographic areas.
- **Increase efficiencies:** Use automation to reduce or eliminate IT management activities that waste time and resources.
- **Innovate more:** Try out new ideas as the cloud makes it faster and cheaper to deploy, test, and launch new products and services.
- **Spend your resources strategically:** Free your IT staff from handling operations and maintenance by switching to a DevOps model.
- **Enhance security:** Cloud providers have teams of people who focus on security, offering best practices to ensure you are compliant.

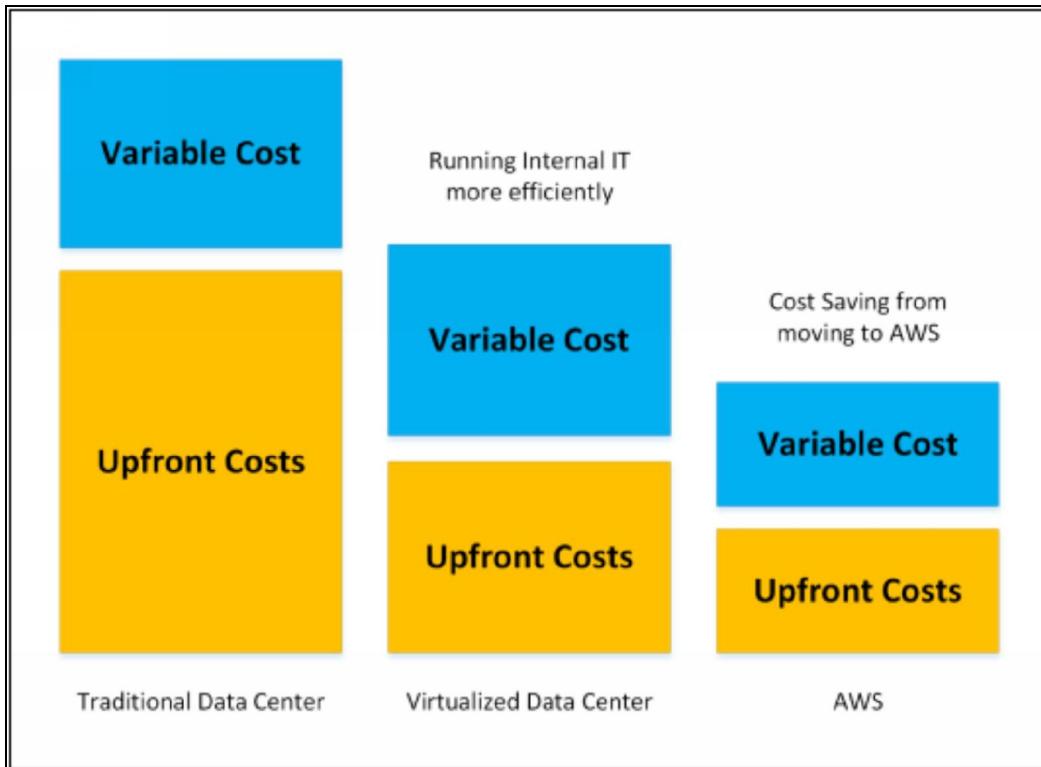


Figure 6. Cost Comparisons of Data Centers and AWS

AWS Virtuous Cycle

The AWS pricing philosophy is driven by a virtuous cycle. Lower prices mean more customers are taking advantage of the platform, which in turn results in further driving down costs.

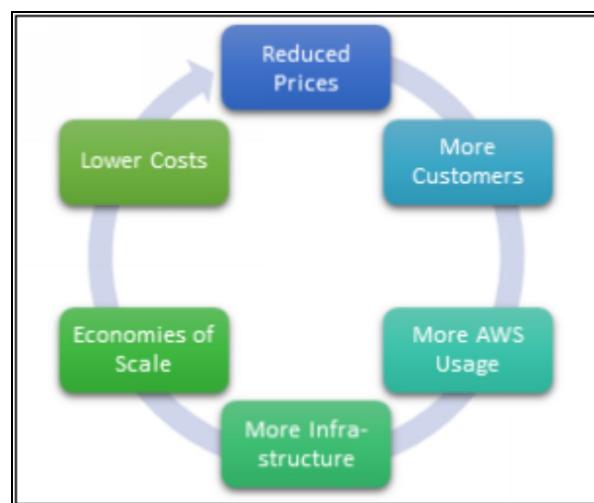


Figure 7. AWS Virtuous Cycle

AWS Cloud Architecture Design Principles

Excellent architectural design should take advantage of the inherent strengths of the AWS cloud computing platform. Below are the fundamental design principles that need to be taken into consideration while designing.

Scalability

Systems need to be designed in such a way that they are capable of growing and expanding over time with no drop in performance. The architecture needs to be able to take advantage of the virtually unlimited on-demand capacity of the cloud platform and scale in a manner where adding extra resources increases the ability to serve additional load. There are generally two ways to scale an IT architecture, vertically and horizontally.

- ***Scale Vertically*** - increase specifications such as RAM, CPU, IO, or networking capabilities of an individual resource.
- ***Scale Horizontally*** - increase the number of resources such as adding more hard drives to a storage array or adding more servers to support an application.
- **Stateless Applications** – An application that needs no knowledge of previous interactions and stores no session. It could be an application that when given the same input, provides the same response to an end user. A stateless application can scale horizontally since any of the available compute resources (e.g., Amazon EC2 instances, AWS Lambda functions) can service any request. With no session data to be shared, you can add more compute resources as needed and terminate them when the capacity is no longer required.
- **Stateless Components** - Most applications need to maintain some state information, for example, web

applications need to track previous activity such as whether a user is signed in, items already in the shopping cart, so that they might present personalized content based on past actions. A portion of these architectures can be made stateless by storing state in the client's browser using cookies. It makes servers relatively stateless because the user's browser stores the sessions.

- **Stateful Components** – Some layers of the architecture are stateful, such as a database. You need databases that can scale. Amazon RDS DB can scale up, and by adding read replicas, it can also scale out. Whereas, Amazon Dynamo DB scales automatically and is a better choice where the consistent addition of reading Replicas are required.
- **Distributed Processing** – Processing of extensive data requires a distributed processing approach where big data is broken down into pieces and have computing instances work on them separately in parallel. On AWS, the core service that handles this is Amazon Elastic MapReduce (EMR). It manages a fleet of EC2 instances that work on the fragments of data simultaneously.

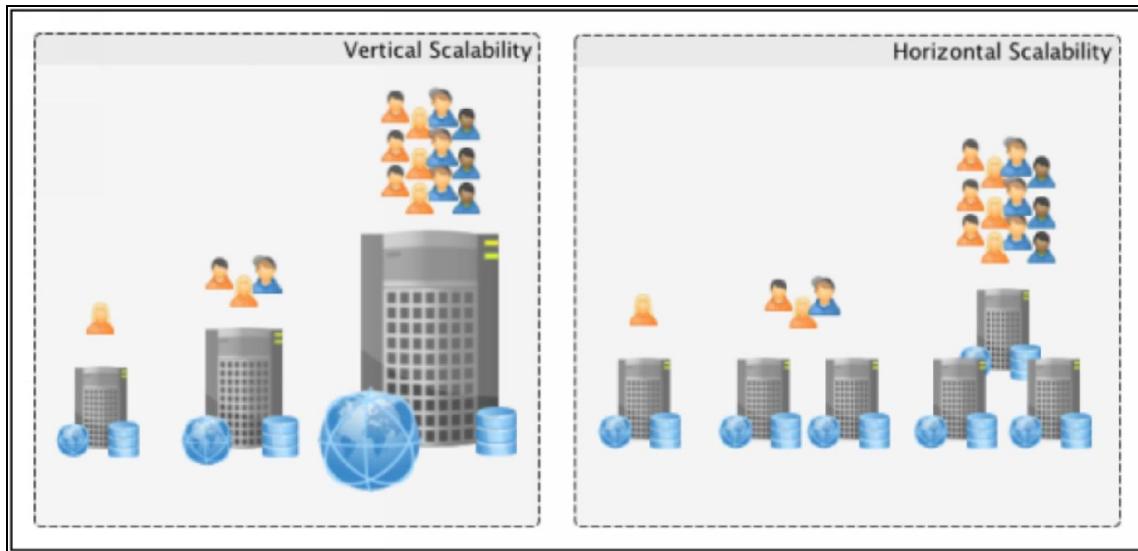


Figure 8. Vertical vs. Horizontal Scalability

Disposable Resources Instead of Fixed Servers

In a cloud computing environment, you can treat your servers and other components as temporary disposable resources instead of fixed elements. Launch as many as needed and use as long as you need them. If a server goes down or needs a configuration update, it can be replaced with the latest configuration server instead of updating the old one.

- ***Instantiating Compute Resources*** - When deploying resources for a new environment or increasing the capacity of the existing system, it is essential to keep the process of configuration and coding as an automated and repeatable process to avoid human errors and long lead times.
 - **Bootstrapping** – Executing bootstrapping after launching a resource with the default configuration, enables you to reuse the same scripts without modifications.
 - **Golden Image** – Certain resource types such as Amazon EC2 instances, Amazon RDS DB instances, Amazon Elastic Block Store (Amazon EBS) volumes, can be launched from a golden image, which is a snapshot of a particular state of that resource. It is used in auto-scaling, for example, by creating an Amazon Machine Image (AMI) of a customized EC2 instance; you can launch as many instances as needed with the same customized configurations.
 - **Hybrid** – Using a combination of both approaches, where some parts of the configuration are captured in a golden image, while others are configured dynamically through a bootstrapping action. AWS Elastic Beanstalk follows the hybrid model.
- ***Infrastructure as Code*** – AWS assets are programmable, allowing you to treat your infrastructure as code. It lets you repeatedly deploy the infrastructure across multiple regions without the need to go and provision everything manually. AWS

CloudFormation and AWS Elastic Beanstalk are the two such provisioning resources.

Automation

One of the design best practices is to automate wherever possible to improve the system's stability and efficiency of the organization using various AWS automation technologies. These include AWS Elastic Beanstalk, Amazon EC2 Auto recovery, Auto Scaling, Amazon CloudWatch Alarms, Amazon CloudWatch Events, AWS OpsWorks Lifecycle events and AWS Lambda Scheduled events.

Loose Coupling

IT systems should ideally be designed with reduced interdependency. Complex applications require breakdown into smaller loosely coupled components. The failure of any one component does not cascade down to other parts of the application. The more loosely coupled system is the more resilient it is.

- ***Well-Defined Interfaces*** – Using technology-specific interfaces such as RESTful APIs, components can interact with each other to reduce inter-dependability. It hides the technical implementation detail allowing teams to modify any underlying operations without affecting other components. Amazon API Gateway service makes it easier to create, publish, maintain and monitor thousands of concurrent API calls while handling all the tasks involved in accepting and processing including traffic management, authorization, and access control.
- ***Service Discovery*** – Applications deployed as a set of smaller services require the ability to interact with each other since the services may be running across multiple resources. Implementing Service Discovery allows smaller services to be used irrespective

of their network topology details through the loose coupling. In AWS platform service discovery can be achieved through Amazon's Elastic Load Balancer which uses DNS endpoints; so if your RDS instance goes down and you have Multi-AZ enabled on that RDS database, the Elastic Load Balancer redirects the request to the copy of the database in the other Availability Zone.

- **Asynchronous Integration** - Asynchronous Integration is a form of loose coupling where an immediate response between the services is not needed, and an acknowledgment of the request is sufficient. One component generates events while the other consumers. Both components interact through an intermediate durable storage layer, not through point-to-point interaction. An example is an Amazon SQS Queue. If a process fails while reading messages from the queue, messages can still be added to the queue for processing once the system recovers.

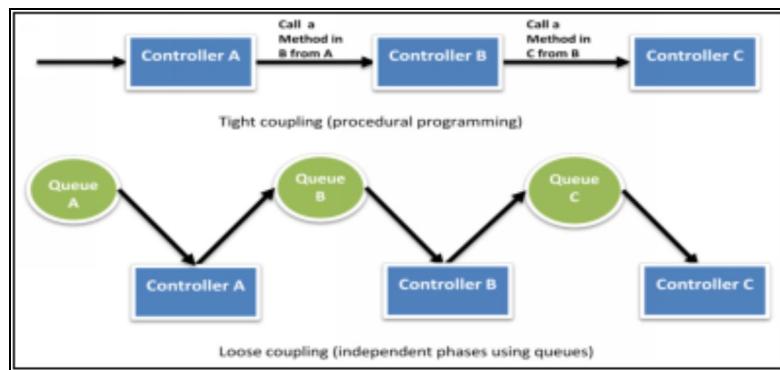


Figure 9. Tight and Loose Coupling

- **Graceful Failure** – Increase loose coupling by building applications that handle component failure in a graceful manner. In the event of component failure, this helps reduce the impact on the end users and increase the ability to progress on offline procedures.

Services, Not Servers

Developing large-scale applications require a variety of underlying technology components. Best design practice would be to leverage the broad set of computing, storage, database, analytics, application, and deployment services of AWS to increase developer productivity and operational efficiency.

- **Managed Services** - Always rely on services, not servers. Developers can power their applications by using AWS managed services that include databases, machine learning, analytics, queuing, search, email, notifications, and many more. For example, Amazon S3 can be used to store data without having to think about capacity, hard disk configurations, replication. Amazon S3 also provides a highly available static web hosting solution that can scale automatically to meet traffic demand.



EXAM TIP: Amazon S3 is great for static website hosting.

- **Serverless Architectures** - Serverless architectures reduce the operational complexity of running applications. Event-driven and synchronous services can both be built without managing any server infrastructure. Example, your code can be uploaded to AWS lambda compute service that runs the code on your behalf. Develop scalable synchronous APIs powered by AWS Lambda using Amazon API Gateway. Lastly combining this with Amazon S3 for serving static content, a complete web application can be produced.



EXAM TIP: For event-driven managed service / serverless architecture, use AWS Lambda. If you want to customize for your own needs, then Amazon EC2 offers flexibility and full control.

Databases

AWS managed database services remove constraints that come with licensing costs and the ability to support diverse database engines. The

different categories of database technologies to keep in mind while designing system architecture:

Relational Databases

- Often called RDBS or SQL databases.
- Consists of normalized data in well-defined tabular structures known as tables, consisting of rows and columns.
- Provides powerful query language, flexible indexing capabilities, strong integrity controls, and ability to combine data from multiple tables fast and efficiently.
- Amazon Relational Database Service (Amazon RDS) and Amazon Aurora
- Scalability: Can scale vertically by upgrading to a larger Amazon RDS DB instance or adding more and faster storage. For read-heavy applications, use Amazon Aurora to scale by creating one or more read replicas horizontally.
- High Availability: using Amazon RDS Multi-AZ deployment feature creates synchronously replicated standby instance in a different Availability Zone (AZ). In case of failure of the primary node, Amazon RDS performs an automatic failover to the standby without manual administrative intervention.
- Anti-Patterns: If your application does not need joins or complex transactions, consider a NoSQL database instead. Store large binary files (audio, video, and image) in Amazon S3 and only hold the metadata for the files in the database.

Non-Relational Databases

- Often called NoSQL databases.
- The tradeoff query and transaction capabilities of relational databases for a more flexible data model.
- It utilizes a variety of data models including graphs, key-value pairs, and JSON documents.

- Amazon DynamoDB
- *Scalability*: Automatically scales horizontally by data partitioning and replication.
- *High Availability*: Synchronously replicates data across three facilities in an AWS region to provide fault tolerance in case of a server failure or Availability Zone disruption.
- *Anti-Patterns*: If your schema cannot be denormalized and requires joins or complex transactions, consider a relational database instead. Store large binary files (audio, video, and image) in Amazon S3 and only hold the metadata for the files in the database.



EXAM TIP: In any given scenario, if you are told to be working on complex transactions or using joins, then you would use Amazon Aurora, Amazon RDS, MySQL or any other relational database but if you are not then you want a non-relational database like Amazon DynamoDB.

Data Warehouse

- A particular type of relational database optimized for analysis and reporting of large amounts of data
- Used to combine transactional data from disparate sources making them available for analysis and decision-making
- Running complex transactions and queries on the production database create massive overhead and require immense processing power, hence the need for data warehousing
- Amazon Redshift
- *Scalability*: Amazon Redshift uses a combination of massively parallel processing (MPP), columnar data storage and targeted data compression encoding to achieve efficient storage and optimum query performance. It increases performance by increasing the number of nodes in the data warehouse cluster
- *High Availability*: By deploying production workloads in multi-node clusters enables the data written to a node to be automatically replicated to other nodes within the cluster. Data is

also continuously backed up to Amazon S3. Amazon Redshift automatically re-replicates data from failed drives and replaces nodes when necessary.

- *Anti-Patterns*: It is not meant to be used for online transaction processing (OLTP) functions as Amazon Redshift is a SQL-based relational database management system (RDBMS). For high concurrency workload or a production database, consider using Amazon RDS or Amazon DynamoDB instead.

Search

- Search service is used to index and search both structured and free text format
- Sophisticated search functionality typically outgrows the capabilities of relational or NO SQL databases. Therefore a search service is required.
- AWS provides two services, Amazon CloudSearch and Amazon Elasticsearch Service (Amazon ES)
- Amazon CloudSearch is a managed search service that requires little configuration and scales automatically; whereas Amazon ES offers an open source API offering more control over the configuration details
- *Scalability*: Both uses data partitioning and replication to scale horizontally
- *High-Availability*: Both services store data redundantly across Availability Zones

Removing Single Points of Failure

A system needs to be highly available to withstand any failure of the individual or multiple components (like hard disks, servers, network links). You should have resiliency built across various services as well as multiple availability zones to automate recovery and reduce disruption at every layer of your architecture.

- **Introducing Redundancy** - Have multiple resources for the same task. Redundancy can be implemented in either standby or active mode. In standby mode, functionality is recovered through secondary resource while the primary resource remains unavailable. In active mode, requests are distributed to multiple redundant compute resources when one of them fails.
- **Detect Failure** - Detection, and reaction to failure should both be automated as much as possible. Configure health checks and mask failure by routing traffic to healthy endpoints using services like ELB and Amazon Route53. Auto Scaling can be configured to replace unhealthy nodes using the Amazon EC2 auto recovery feature or services such as AWS OpsWorks and AWS Elastic Beanstalk.
- **Durable Data Storage** – Durable data storage is vital for data availability and integrity. Data replication can be achieved by introducing redundant copies of data. The three modes of replication that can be used are asynchronous replication, synchronous replication, and Quorum-based replication.
 - **Synchronous replication** only acknowledges a transaction after it has been durably stored in both the primary location and its replicas.
 - **Asynchronous replication** decouples the primary node from its replicas at the expense of introducing replication lag
 - **Quorum-based replication** combines synchronous and asynchronous replication to overcome the challenges of large-scale distributed database systems
- **Automated Multi-Data Center Resilience** – This is achieved by using the multiple availability zones offered by the AWS global

infrastructure. Availability zones are designed to be isolated from failures of the other availability zones. Example, a fleet of application servers distributed across multiple Availability Zones can be attached to the Elastic Load Balancing service (ELB). When health checks of the EC2 instances of a particular Availability Zone fail, ELB will stop sending traffic to those nodes. Amazon RDS provides automatic failover support for DB instances using Multi-AZ deployments, while Amazon S3 and Amazon DynamoDB stores data redundantly across multiple facilities.

- ***Fault Isolation and Traditional Horizontal Scaling*** – Fault isolation can be attained through sharding. Sharding is a method of grouping instances into groups called shards. Each customer is assigned to a specific shard instead of spreading traffic from all customers across every node. Shuffle sharding technique allows the client to try every endpoint in a set of shared resources until one succeeds.

Optimize for Cost

Reduce capital expenses by benefiting from the AWS economies of scale. Main principles of optimizing for cost include:

- ***Right-Sizing*** - AWS offers a broad set of options for instance types. Selecting the right configurations, resource types and storage solutions that suit your workload requirements can reduce cost.
- ***Elasticity*** - Implement Auto Scaling to horizontally scale up and down automatically depending upon your need to reduce cost. Automate turning off non-production workloads when not in use. Use AWS managed services wherever possible that helps in taking capacity decisions as and when needed.

- ***Take Advantage of the Variety of Purchasing Options*** – AWS provides flexible purchasing options with no long-term commitments. These purchasing options can reduce cost while paying for instances. Two ways to pay for Amazon EC2 instances are:
 - Reserved Capacity – Reserved instances enables you to get a significantly discounted hourly rate when reserving computing capacity as oppose to On-Demand instance pricing. It is ideal for applications with predictable capacity requirements.
 - Spot Instances - Available at discounted pricing compared to On-Demand pricing. Ideal for workloads that have flexible start and end times. Spot instances allow you to bid on spare computing capacity. When your bid exceeds the current Spot market price, your instance is launched. If the Spot market price increases above your bid price, your instance will be terminated automatically.

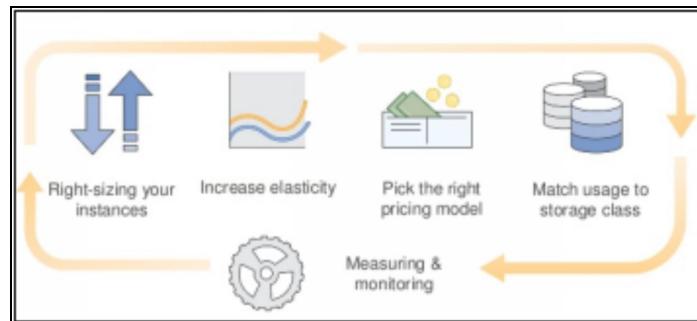


Figure 10. Cost Optimization Pillars

Caching

Caching is used to store previously calculated data for future use. This improves application performance and increases the cost efficiency of

implementation. A good practice is to implement caching in the IT architecture wherever possible.

- ***Application Data Caching*** – Application data can be stored in the cache for subsequent requests to improve latency for end users and reduce the load on back-end systems. Amazon ElastiCache makes it easy to deploy, operate, and scale an in-memory cache in the cloud.
- ***Edge Caching*** – Both static and dynamic content can be cached at multiple edge locations around the world using Amazon CloudFront. This allows content to be served by infrastructure that is closer to viewers, lowering latency and providing high, sustained data transfer rates to deliver large famous objects to end users at scale.

Security

AWS allows you to improve your safety in some ways, plus also letting the use of security tools and techniques that traditional IT infrastructures implement.

- ***Utilize AWS Features for Defense in Depth*** – Isolate parts of the infrastructure by building a VPC network topology using subnets, security groups, and routing controls. Setup web application firewall for protection using AWS WAF.
- ***Offload Security Responsibility to AWS*** - AWS manages the security of the underlying cloud infrastructure; you are only responsible for securing the workloads you deploy in AWS.
- ***Reduce Privileged Access*** –To avoid a breach of security reduce privileged access to the programmable resources and servers. For

Example, defining IAM roles to restrict root level access.

- **Security as Code** - AWS CloudFormation scripts can be used that incorporates your security policy and reliably deploys it. Security scripts can be reused among multiple projects as part of your continuous integration pipeline.
- **Real-Time Auditing** – AWS allows you to monitor and automate controls to minimize security risk exposures continuously. Services like AWS Config, Amazon Inspector, and AWS Trusted Advisor continually monitor IT resources for compliance and vulnerabilities. Testing and auditing in real-time are essential for keeping the environment fast and safe.

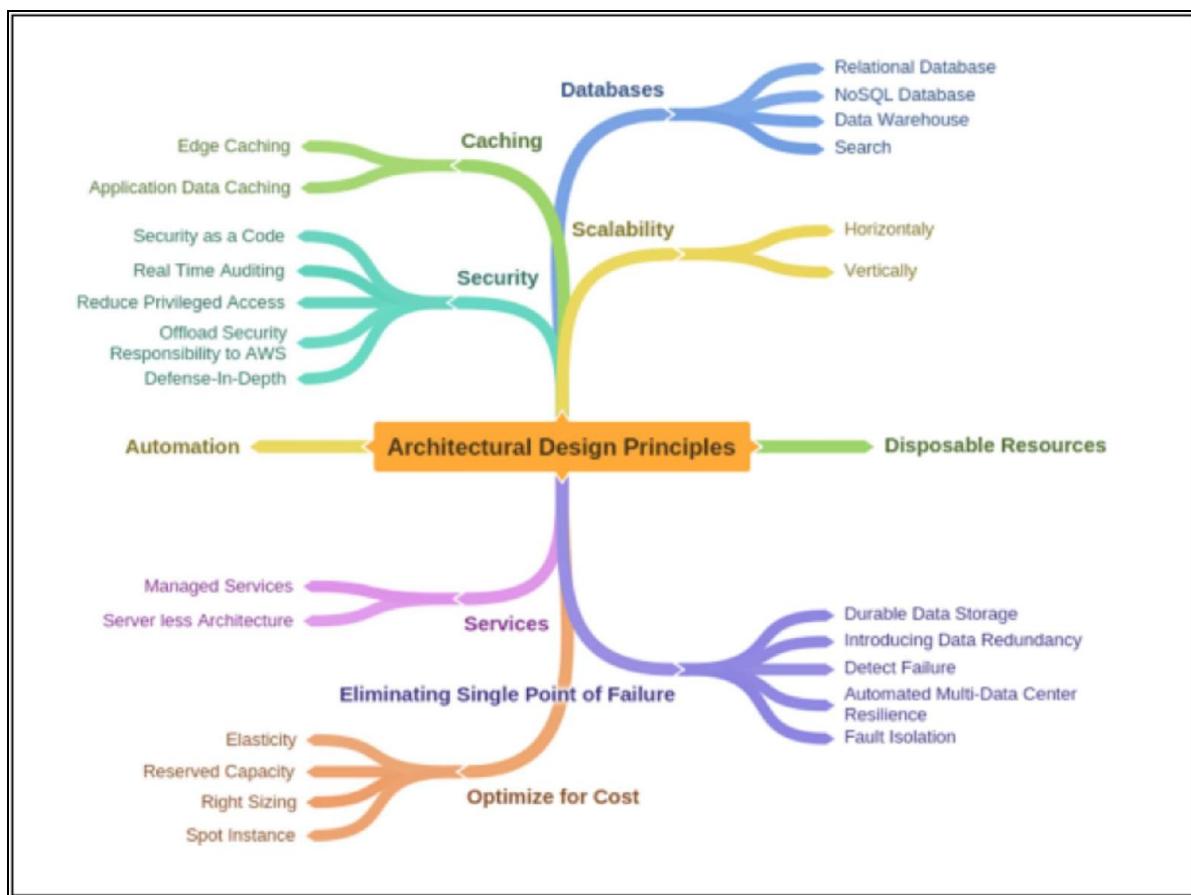


Figure 11. Mind Map of Architectural Design Principles

AWS Global Infrastructure

The AWS Cloud spans across 18 geographic Regions with 53 Availability Zones and 1 Local Region around the world, with further announced plans for 12 more Availability Zones and four more Regions in Bahrain, Hong Kong SAR, Sweden, and a second AWS GovCloud Region in the US.

What is a Region?

The region is an entirely independent and separate geographical area. Each region has multiple, physically separated and isolated locations known as Availability Zones. Examples of Region include London, Dublin, Sydney.

What is an Availability Zone?

Availability zone is simply a data center or a collection of data centers. Each Availability zone in a Region has separate power, networking and connectivity to reduce the chances of two zones failing simultaneously. No two Availability zones share a data center; however, the data centers within a particular Availability zone are connected to each other over redundant low-latency private network links. Likewise, all zones in a region are linked by highly resilient and very low latency private fiber optic connections for communication. The Availability zones would be at a certain length or distance apart from each other.

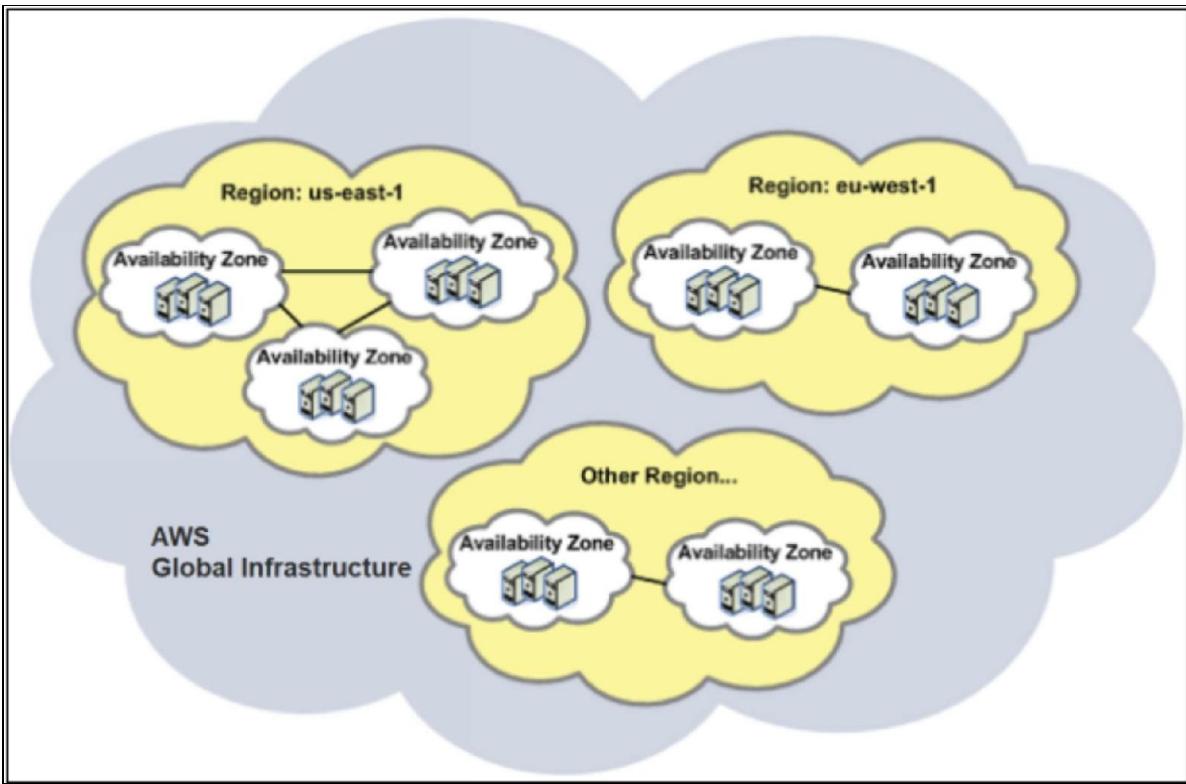


Figure 12. Regions and Availability Zones

What is an Edge Location?

Edge Locations are AWS sites deployed in major cities and highly populated areas across the globe. There are many more Edge locations than there are regions. Currently, there are over 102 edge locations. Edge Locations are used by AWS services such as AWS CloudFront to cache data and reduce latency for end-user access by using the Edge Locations as a global Content Delivery Network (CDN).

Therefore, End Users uses Edge Locations who are accessing and using your services. For example, you may have your website hosted within the Ohio region with a configured CloudFront distribution associated. When a user accesses your website from Europe, they will be re-directed to their closest Edge Location (in Europe) where cached data could be read on your website, significantly reducing latency.

Regional Edge Cache

In November 2016, AWS announced a new type of Edge Location, called a Regional Edge Cache. These sit between your CloudFront Origin servers and the Edge Locations. A Regional Edge Cache has a larger cache-width than each of the individual Edge Locations, and because data expires from the cache at the Edge Locations, the data is retained at the Regional Edge Caches.

Therefore, when data is requested at the Edge Location that is no longer available, the Edge Location can retrieve the cached data from the Regional Edge Cache instead of the Origin servers, which would have a higher latency.

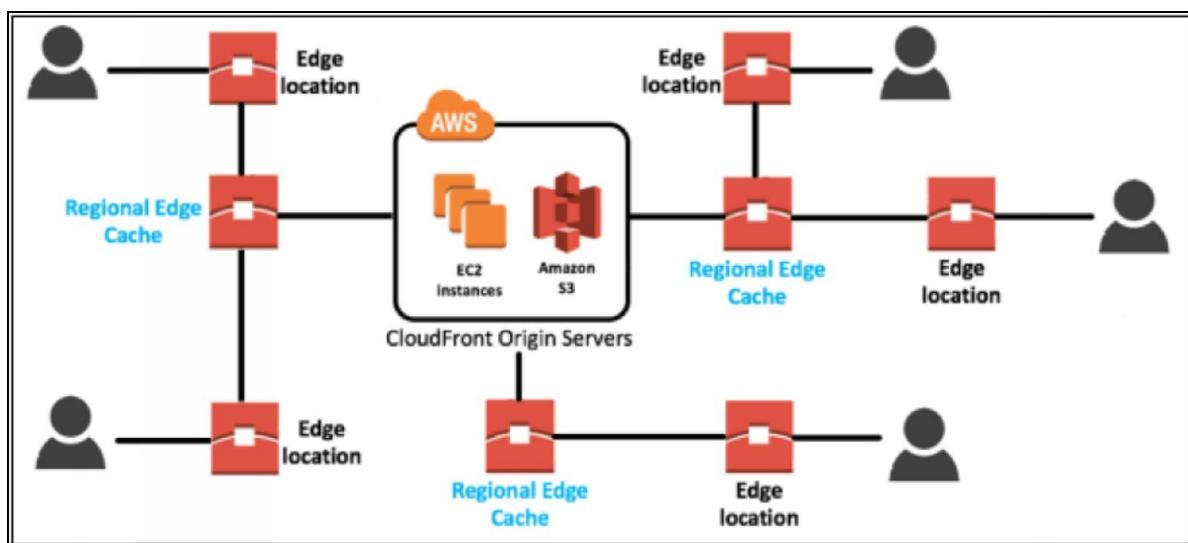


Figure 13. Edge Locations and Regional Edge Caches



EXAM TIP: Know the difference between the three: Region, Availability Zone, and Edge Location.

Chapter 2: Monitoring, Metrics & Analysis

Introduction

Amazon Web Services is the global leader in cloud computing that provides a wide variety of IT services. With an extraordinary breadth of available services to take advantage of, it is critical to monitor what you use and devise an alerting strategy that works for your organization. As you operate IT services in the cloud, you are financially responsible for the AWS costs you incur. Therefore it is essential to measure your use of AWS services. To help you identify changes in spending, you need to establish sound financial monitoring. In this chapter, we explore monitoring concepts, CloudWatch basics, ways to extend CloudWatch, create billing alerts, AWS config, how config rules can be used for troubleshooting and ways to automate actions based on changes within your AWS environment.

Amazon CloudWatch

Introduction

Amazon Cloud Watch is a service used for monitoring AWS cloud resources and the applications you run on AWS. Amazon CloudWatch is used to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. It can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. Below is a list of all supported AWS resources:



Figure 14. Usage of CloudWatch

You can access CloudWatch using any of the following methods:

- Amazon CloudWatch console
- AWS CLI
- CloudWatch API
- AWS SDKs

CloudWatch Metrics

Metrics are data about the performance of your systems. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications. CloudWatch alarms send notifications or automatically make changes to the resources you are monitoring based on rules that you define.

For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances and then use this data to determine whether you should launch additional instances to handle the increased load. You can also use this data to stop under-used instances to save money.

In addition to monitoring the built-in metrics that come with AWS, you can monitor your custom metrics. By default, several services provide free metrics for resources such as Amazon EC2 instances, Amazon EBS volumes, and Amazon RDS DB instances. You can also enable detailed monitoring of some resources, such as your Amazon EC2 instances, or publish your application metrics. Amazon CloudWatch can load all the metrics in your account (both AWS resource metrics and application metrics that you provide) for search, graphing, and alarms.

Metric data is kept for 15 months, enabling you to view both up-to-the-minute data and historical data.

How Amazon CloudWatch Works?

Amazon CloudWatch is a metrics repository. An AWS service—such as Amazon EC2—puts metrics into the repository, and you retrieve statistics based on those metrics. If you put your custom metrics into the repository, you can retrieve statistics on these metrics as well.

You can use metrics to calculate statistics and then present the data graphically in the CloudWatch console. You can configure alarm actions to stop, start, or terminate an Amazon EC2 instance when specific criteria are met. Also, you can create alarms that initiate Amazon EC2 Auto Scaling and Amazon Simple Notification Service (Amazon SNS) actions on your behalf.

How long are CloudWatch metrics Stored?

The AWS CloudWatch can store metrics for two weeks by default. However, you can also get the data longer than two weeks by using the “GetMetric Statistics API” or by using the third party resources, which are offered by AWS partners.

- One minute data points are available for 15 days
- Five minutes datapoints are available for 63 days
- One hour data points are available for 455 days

The metric granularity depends upon the AWS service for which CloudWatch is used. Many default metric for many default services are 1 minute, but it can be 3 or 5 minutes depending on the service. For custom metrics, the minimum granularity that you can have is 1 minute. You can also retrieve data from any terminated EC2 or ELB instance after its termination.

CloudWatch Alarms

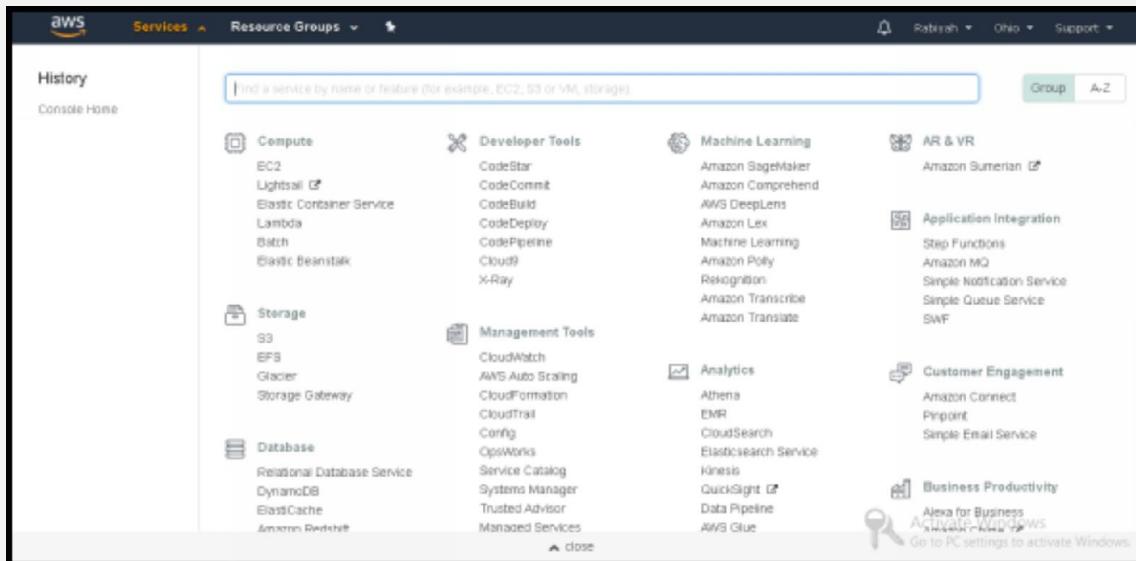
You can create an alarm to monitor any Amazon CloudWatch metric in your account. It includes EC2 CPU Utilization, Elastic Load Balancer Latency or even the charges on your AWS bill. The alarm performs one or more actions based on the value of the metric relative to a threshold over some time periods. The action can be an Amazon EC2 action, an Auto Scaling action, or a notification sent to an Amazon SNS topic. You can also add alarms to CloudWatch dashboards and monitor them visually. Alarms invoke actions for sustained state changes only. CloudWatch alarms do not invoke operations merely because they are in a particular state; the state must have changed and been maintained for a specified number of periods. After an alarm invokes an action due to a change in state, its subsequent behavior depends on the type of action that you have associated with the alarm.

Lab: 2.1 Create a Cloud Watch Role

To create a Cloud Watch role, you have to do the following steps in the AWS lab;

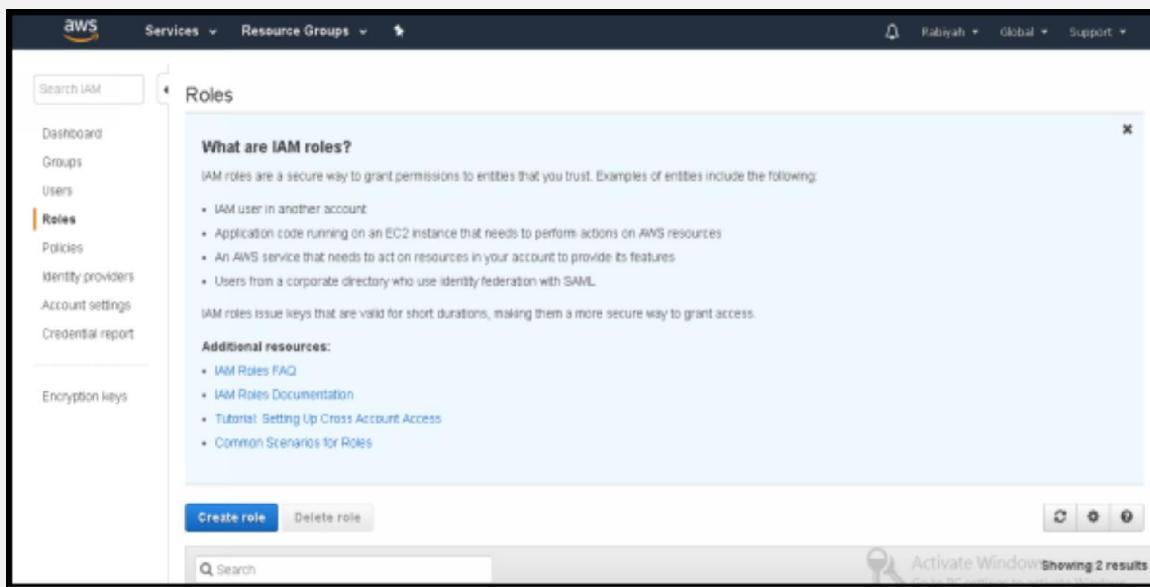
Step: 01

First of all login to your AWS account and go to services.



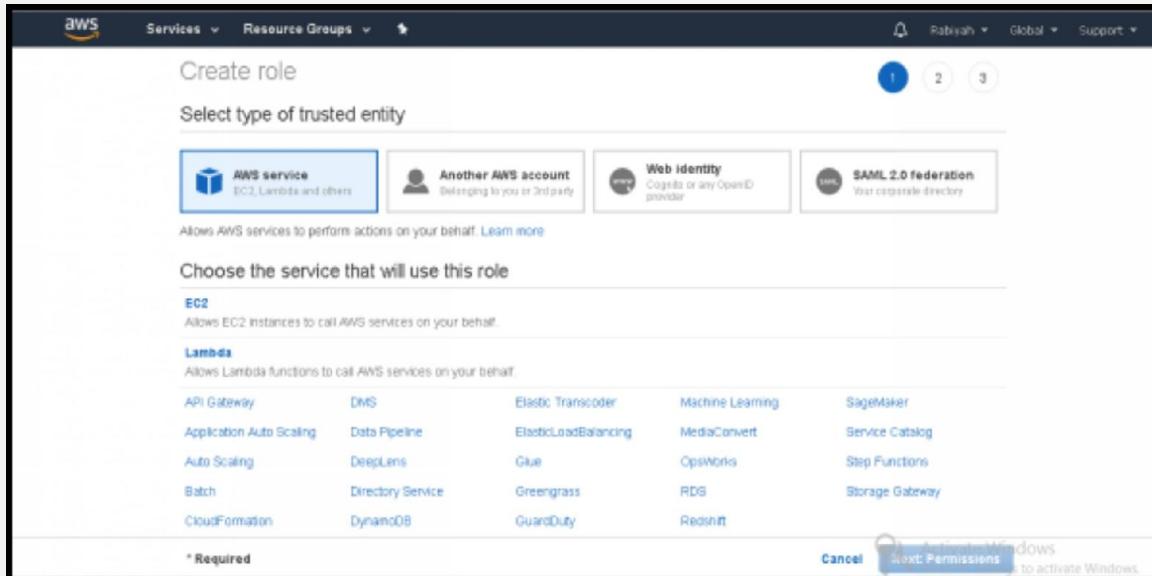
Step:02

Go to the “key” Security, Identity and Compliance and select IAM and then go to “roles.”

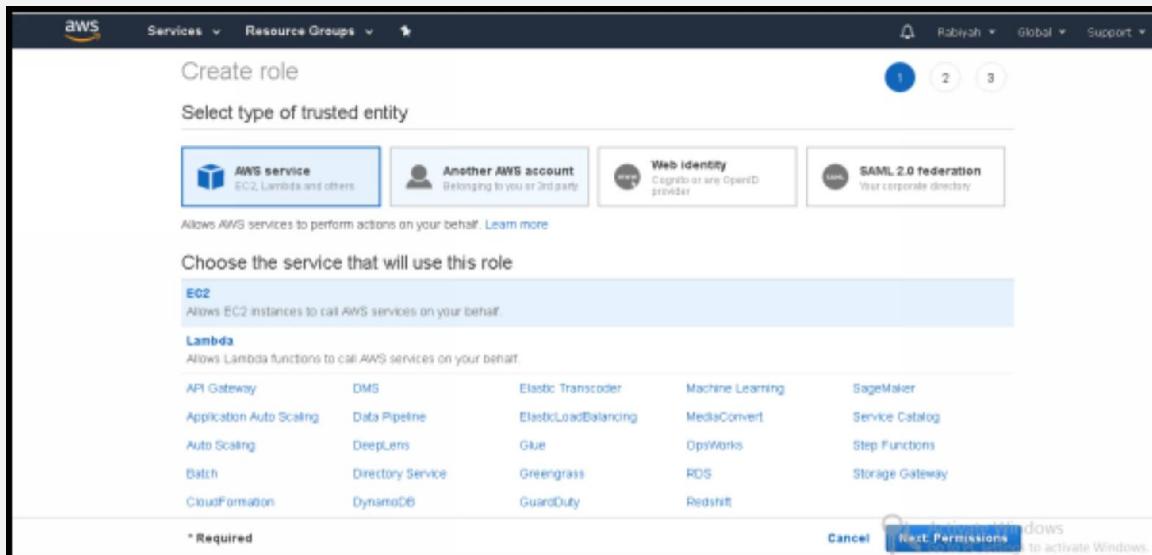


Click on Create role.

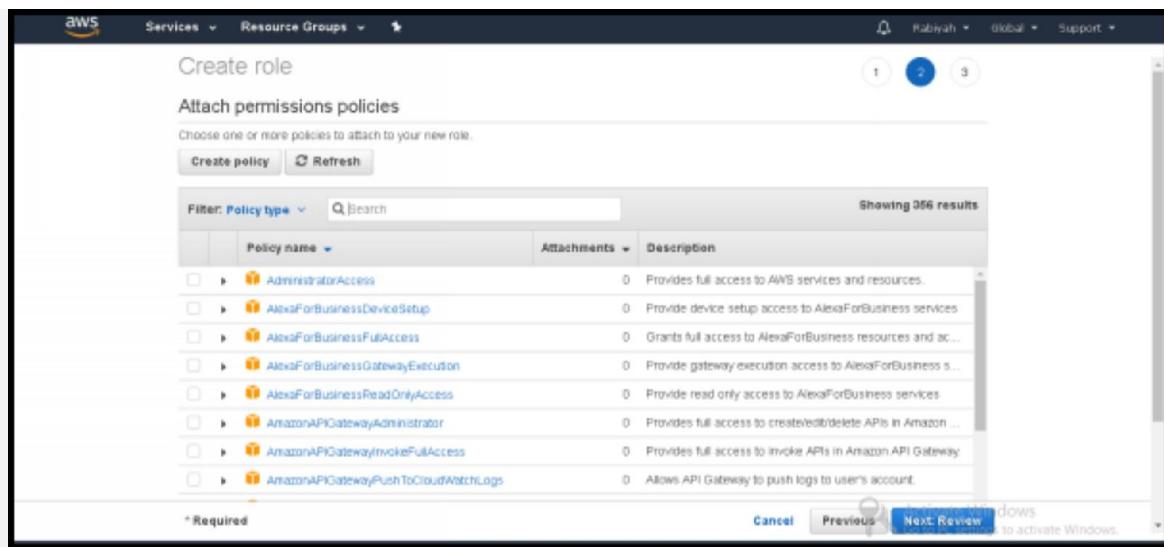
Choose any service from the options that will use that role.



Select type of any trusted entity to further continue. Click on next.

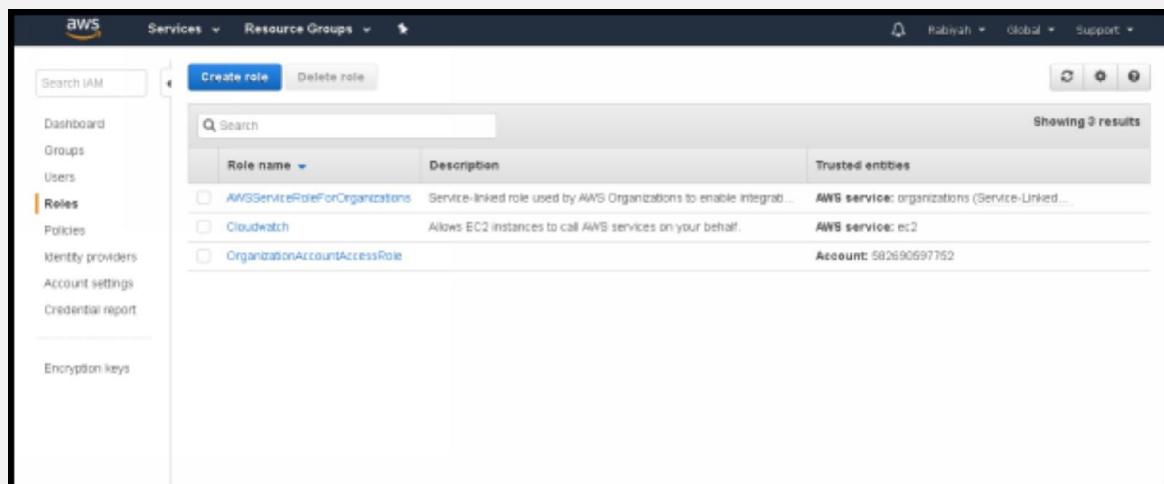


Now attach Permission Policies of your requirements.



After it, click on “Next Review.”

Below diagram is giving a review that you have successfully create a role.



After completing all of the steps, if you want to create a new role then again start the whole process from the step to “create a new role.”

Monitoring EC2

You can monitor EC2 instances automatically using CloudWatch without installing additional software. There are two types of monitoring available:

- Basic Monitoring: Seven pre-selected metrics at a five-minute frequency and three status check metrics at one-minute frequency, for no additional charge.
- Detailed Monitoring: All metrics available to Basic Monitoring at one-minute frequency, for an additional charge. Instances with Detailed Monitoring enabled allows data aggregation by Amazon EC2 AMI ID and instance type.

If you use Auto Scaling or Elastic Load Balancing, Amazon CloudWatch will also provide Amazon EC2 instance metrics aggregated by Auto Scaling group and by Elastic Load Balancer, regardless of whether you have chosen Basic or Detailed Monitoring.

Monitoring data is retained for fifteen months, even if your AWS resources have been terminated. However, it can be kept longer than two weeks by using the “GetMetric Statistics API” or by using third party resources that are offered by AWS partners.

This enables you to look back at the metrics preceding an event of interest to you. Basic Monitoring is enabled automatically for all Amazon EC2 instances, and you can access these metrics in either the Amazon EC2 tab or the Amazon CloudWatch tab of the AWS Management Console, or by using the Amazon CloudWatch API.

AWS EC2 Metrics

Amazon EC2 sends metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch. By default, each data point covers the

previous 5 minutes of activity for instance. If you've enabled detailed monitoring, each data point covers the previous 1 minute of activity.

Host-level metrics consists of the following:

- CPU – CPUUtilization
- Network – NetworkIn, NetworkOut, NetworkPacketsIn, NetworkPacketsOut
- Disk – DiskReadOps, DiskWriteOps, DiskReadBytes, DiskWriteBytes,
- Status Check – StatusCheckFailed, StatusCheckFailed_Instance, StatusCheckFailed_System

All other metrics apart from the above are custom metrics. RAM utilization is also a custom metric. By default, EC2 monitoring is 5-minute intervals unless you enable detailed monitoring which will then make it 1-minute intervals.

Custom Metrics

As mentioned above, Amazon Cloudwatch provides hypervisor-specific metrics. If you need OS-specific metrics, such as memory and disk-related metrics, you need to use CloudWatch custom metrics. You can publish your metrics to CloudWatch using the AWS CLI or an API. You can view statistical graphs of your published metrics with the AWS Management Console.

The Amazon CloudWatch Monitoring Scripts for Amazon Elastic Compute Cloud (Amazon EC2) Linux-based instances demonstrate how to produce and consume Amazon CloudWatch custom metrics. These sample Perl scripts comprise a fully functional example that reports memory, swap, and disk space utilization metrics for a Linux instance. Standard Amazon CloudWatch usage charges for custom metrics apply to your use of these scripts.

The package for the monitoring scripts contains the following files:

- **CloudWatchClient.pm** – Shared Perl module that simplifies calling Amazon CloudWatch from other scripts.
- **mon-put-instance-data.pl** – Collects system metrics on an Amazon EC2 instance (memory, swap, disk space utilization) and sends them to Amazon CloudWatch.
- **mon-get-instance-stats.pl** – Queries Amazon CloudWatch and displays the most recent utilization statistics for the EC2 instance on which this script is executed.
- **awscreds.template** – File template for AWS credentials that stores your access key ID and secret access key.
- **LICENSE.txt** – a Text file containing the Apache 2.0 license.
- **NOTICE.txt** – Copyright notice.

EC2 Status Check

You can monitor the status of your instances by viewing status checks and scheduled events for your instances. A status check gives you the information that results from automated checks performed by Amazon EC2. These automated checks detect whether specific issues are affecting your instances.

With status checks monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues. You can view the results of these status checks to identify specific and detectable problems.

Status checks are performed every minute, and each returns a pass or a fail status. If all checks pass, the overall status of the instance is OK. If one or more checks fail, the overall status is impaired. Status checks are built into Amazon EC2, so they cannot be disabled or deleted. You can, however, create or eliminate alarms that are triggered based on the result of the status

checks. For example, you can generate an alert to warn you if status checks fail on a specific instance. You can also create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers the instance if it becomes impaired due to an underlying issue.

Types Of Status Checks

There are two types of status checks;

1. System Status Checks

It checks the underlying physical host. It monitors the AWS systems on which your instance runs. These checks detect underlying problems with your instance that require AWS involvement to repair. When a system status check fails, you can choose to wait for AWS to fix the issue, or you can resolve it yourself.

Following are examples of problems that can cause system status checks to fail:

- Loss of network connectivity
- Loss of system power
- Software issues on the physical host
- Hardware issues on the physical host that impact network reachability

The best way to resolve these issues is to stop and then start the Virtual Machine again, which will allow the VM to start up on another physical host instead of the previous one with the above issues.

2. Instance Status Checks

It checks the Virtual Machine itself. It monitors the software and network configuration of your instance. These checks detect problems that require your involvement to repair. When an instance status check fails, typically you will need to address the issue yourself, for example, by rebooting the instance or by making instance configuration changes.

Following are examples of problems that can cause instance status checks to fail:

- Failed system status checks
- Incorrect networking or start-up configuration
- Exhausted memory
- Corrupted file system
- Incompatible kernel

The best way to troubleshoot is by rebooting the instance or making modifications to the instance's operating system.

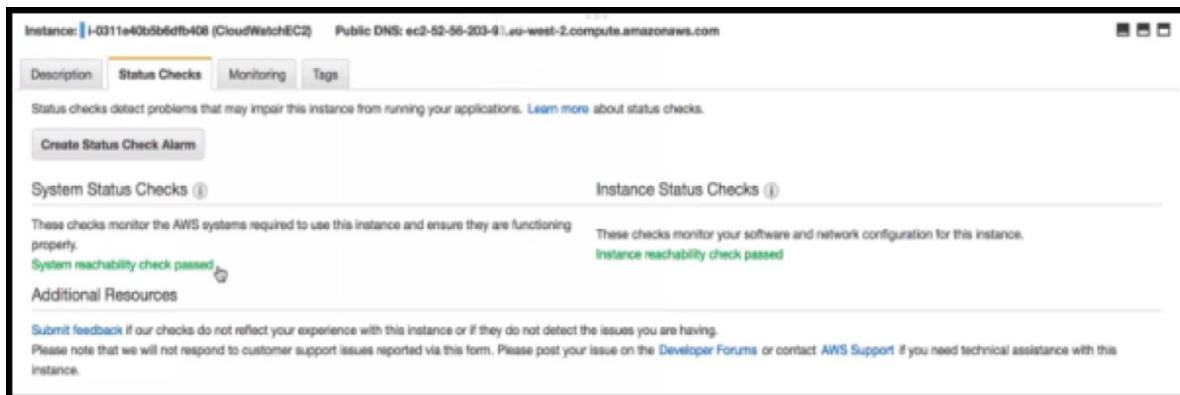
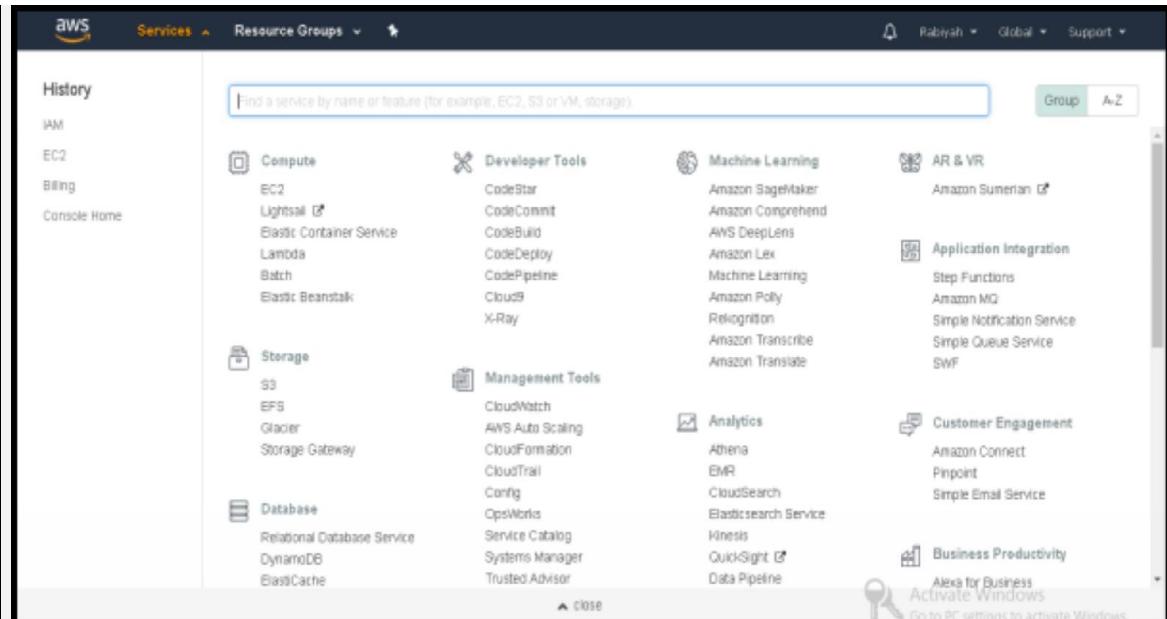


Figure 13. EC2 Instance Check

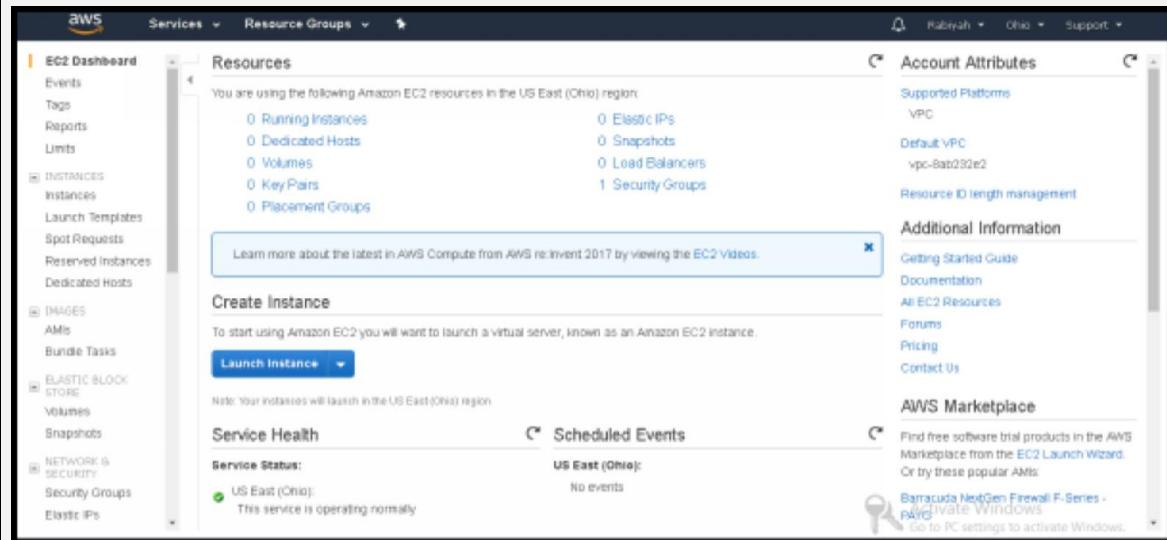
Lab 2.2 Launch an Instance

Step by Step Configuration:

1. First of all, login to your AWS account and click on services. From these services, select Computer and then click on EC2.



2. Now, form the EC2 dashboard click on the “Launch Instance.”



3. Choose an Amazon Machine Image (AMI) of your choice.

The screenshot shows the AWS CloudFormation console with the "Step 1: Choose an Amazon Machine Image (AMI)" step selected. The left sidebar has a "Quick Start" section with links for "My AMIs", "AWS Marketplace", "Community AMIs", and "Free tier only". The main area lists three AMI options:

- Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type** - ami-483b1193 (Selected, 64-bit): The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
- Amazon Linux 2 LTS Candidate AMI 2017.12.0 (HVM), SSD Volume Type** - ami-710e2414 (64-bit): Amazon Linux 2 is the next generation of Amazon Linux. It includes the latest LTS kernel (4.9) tuned for enhanced performance on Amazon EC2, systemd support, newer versions of glibc, glibc, and binutils, and an additional set of core packages for performance and security improvements.
- SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type** - ami-75143f10 (64-bit): SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

A "Select" button is available for each option. A watermark for "Activate Windows" is visible at the bottom right.

4. Choose an Instance type and click on Next Configure Instance details.

The screenshot shows the AWS CloudFormation console with the "Step 2: Choose an Instance Type" step selected. The left sidebar has a "Quick Start" section with links for "My AMIs", "AWS Marketplace", "Community AMIs", and "Free tier only". The main area lists instance types:

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

A "Review and Launch" button is visible at the bottom right. A watermark for "Activate Windows" is visible at the bottom right.

5. Now, click on “Next Add Storage.”

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot Instances

Network: vpc-8ab232e2 (default)

Subnet: No preference (default subnet in any Availability Zone)

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: Cloudwatch

Shutdown behavior: Stop

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply

6. Click on “Add Storage.”

Step 4: Add Storage

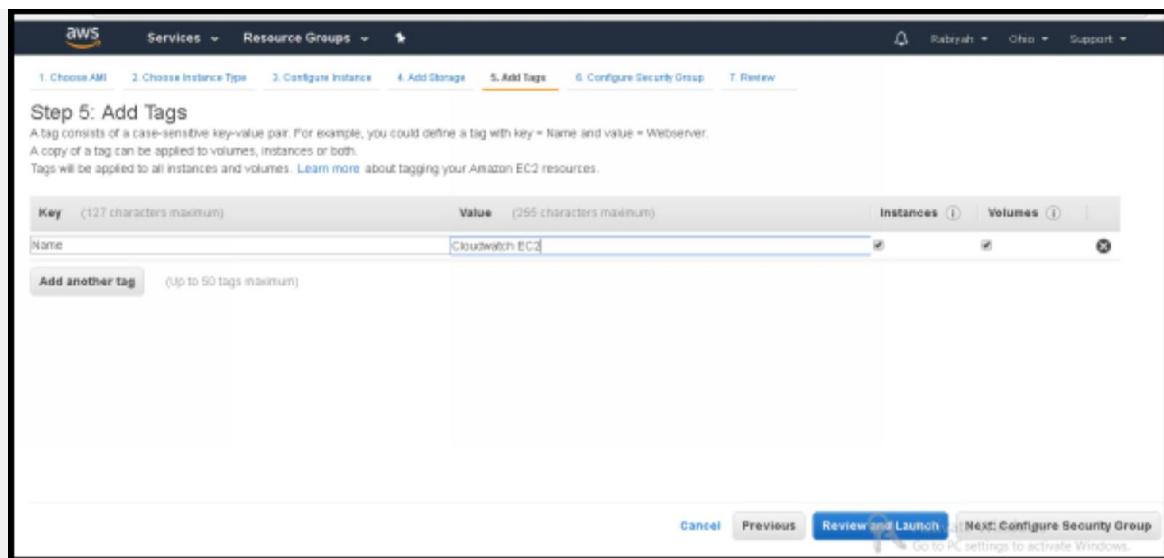
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0da722d3239fa8c7c	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

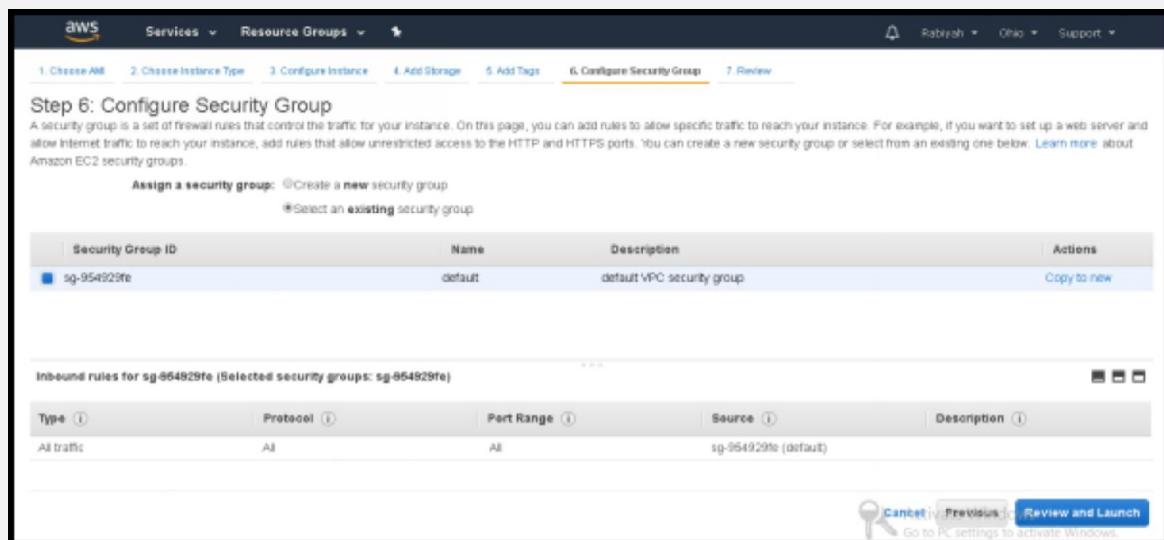
Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more about free usage tier eligibility and usage restrictions.](#)

7. To Add Tags, click on the “Next Configure Security Group” tab.



8. To configure security group, click on security group Id and click on Review and launch.



9. To review the instances and click on “Launch.”

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-053b1193

Free tier eligible

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Post Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security Group ID	Name	Description
sg-9549298	default	default VPC security group

Actions

Activate | Cancel | Previous | **Launch** | Go to EC2 settings to activate Windows

Before running this instance, you first have to download the private key file. For this purpose, create a new key or select an existing key pair and save it to a safe location.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

My EC2 key pair

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel | **Launch Instances**

10. Your instance is launched now successfully.

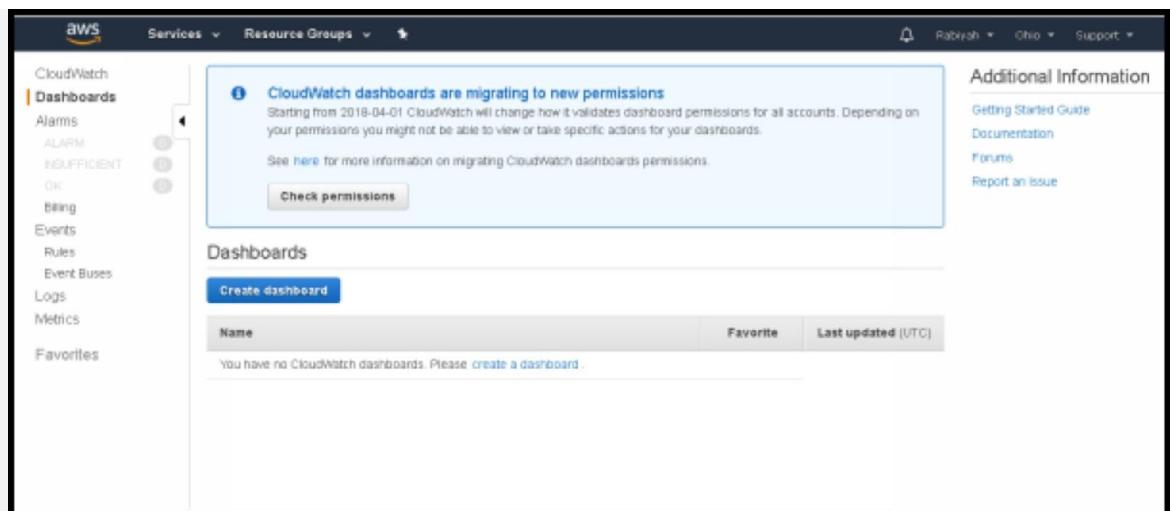
The screenshot shows the AWS Launch Status page. At the top, there's a green success message: "Your instances are now launching" with a link to "View launch log". Below it is a blue info message: "Get notified of estimated charges" with a link to "Create billing alerts". Under "How to connect to your instances", it says instances are launching and may take a few minutes to reach the running state. It also provides a link to "View Instances" and instructions on how to connect once instances are running. A sidebar on the left lists helpful resources like "How to connect to your Linux instance" and "Learn about AWS Free Usage Tier".

Lab 2.3 Create CloudWatch Dashboard

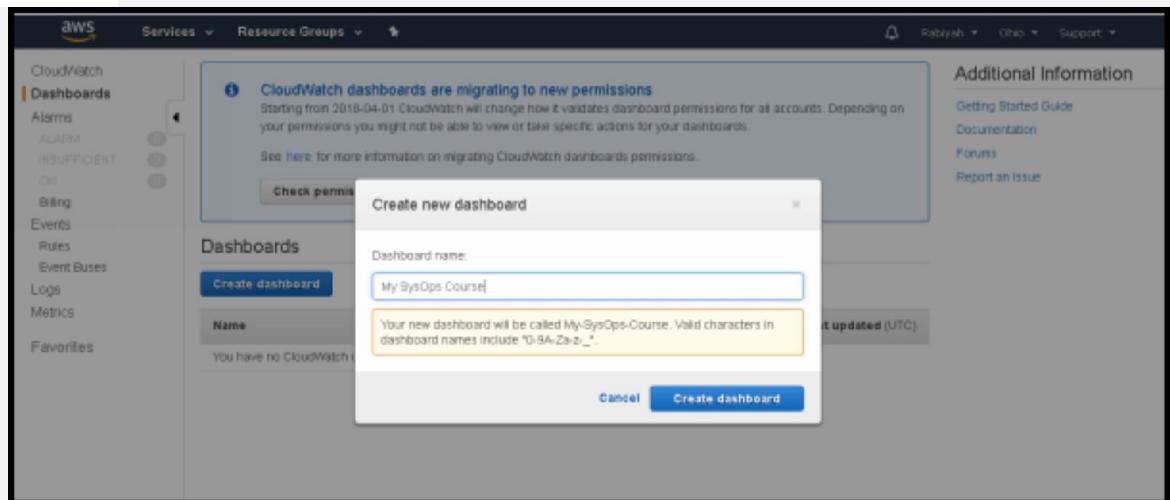
1. To create a CloudWatch Dashboard, first of all, click on the services and from the services select EC2 from “compute” option.

The screenshot shows the AWS Services dashboard. On the left, there's a sidebar with "History", "EC2", "IAM", "Billing", and "Console Home". The main area has a search bar at the top. Below it, the "Compute" section is expanded, showing services like EC2, Lightsail, Elastic Container Service, Lambda, Batch, and Elastic Beanstalk. Other sections like "Storage", "Database", "Developer Tools", "Management Tools", "Analytics", and "Machine Learning" are also visible, each with their respective service icons and names.

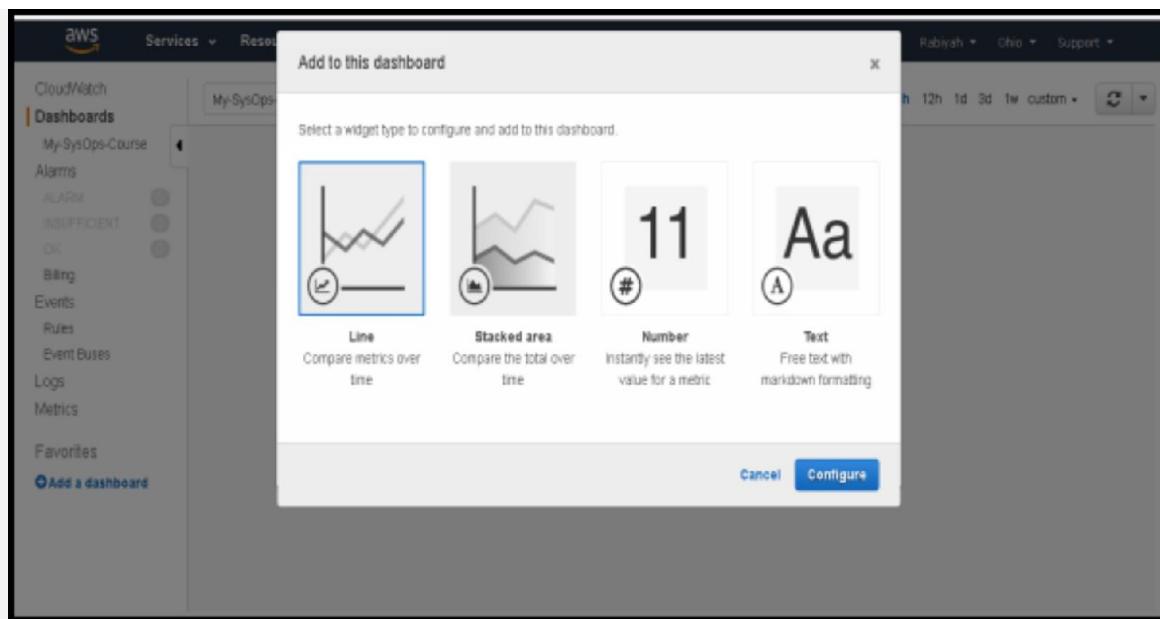
2. Select “Dashboard” from the list and then click on “Create dashboard.”



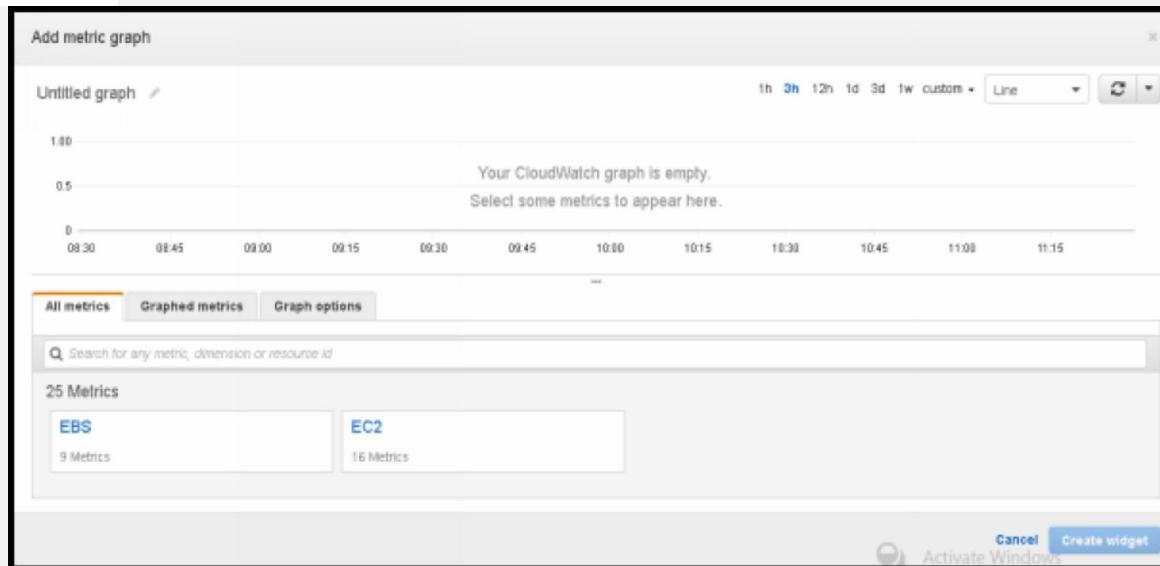
3. Name the dashboard of your own Choice in the empty box.



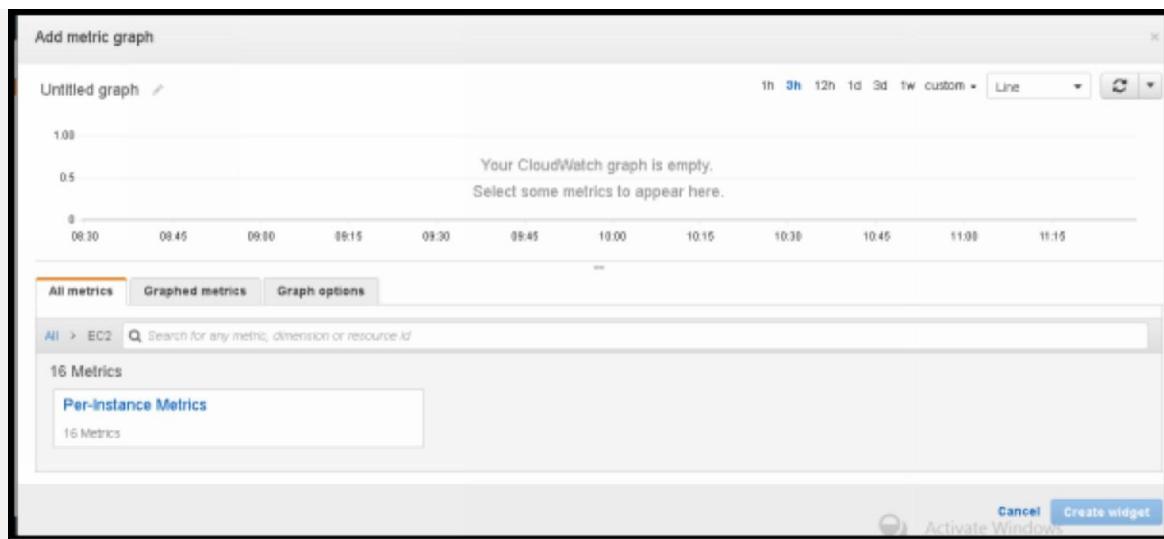
4. Select the widget type to configure the dashboard.



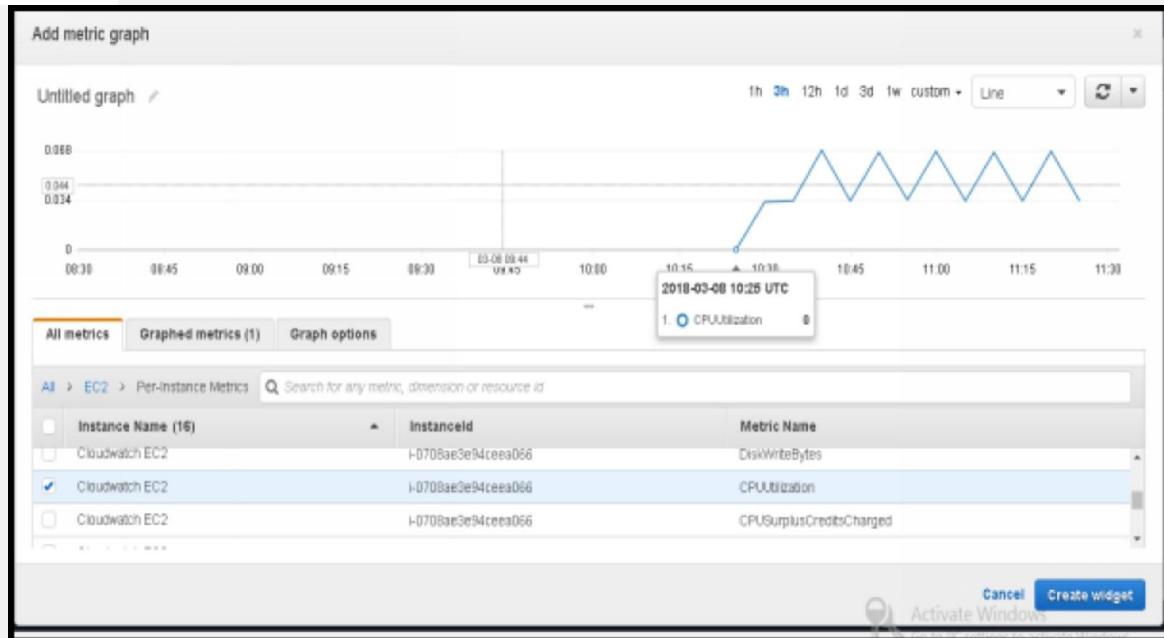
5. Now add the metric graph as shown in below diagram.



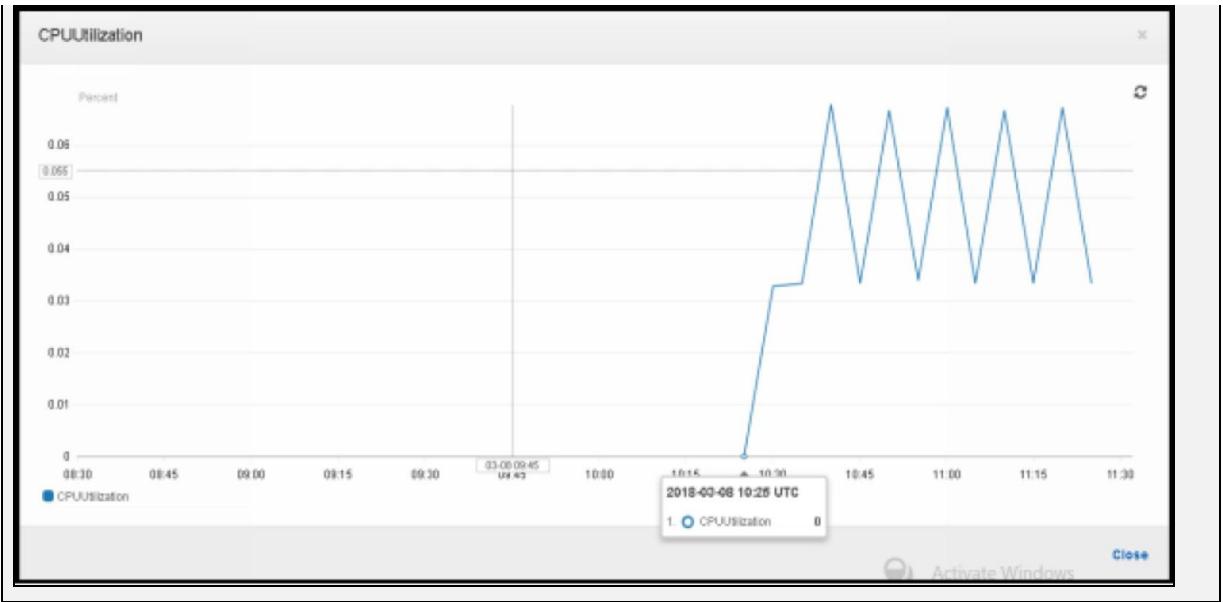
6. Now click on pre-Interface Metrics.



7. Now to calculate the maximum threshold level of energy, select the “CPU Utilization” from the list of metrics.



8. Now you can see that there is a graph which is showing the maximum threshold level of CPU utilization.



Monitoring EBS

Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. EBS allows you to create storage volumes and attach them to Amazon EC2 instances in the same Availability Zone. Once attached, it appears as a mounted device similar to any hard drive or other block device and the instance can interact with the volume just as it would with a local drive, format it with a file system, run a database, install applications on it directly or use them in any other way you would use a block device.

Amazon EBS volumes can be used as boot partitions for Amazon EC2 instances, which lets you preserve your boot partition data irrespective of the life of your instance, and bundle your AMI in one-click. You can also stop and restart instances that boot from Amazon EBS volumes while preserving state, with very fast start-up times. However, the HDD volumes cannot be a boot volume.

Amazon EBS Volume Types

- General Purpose SSD(gp2)
 - General purpose SSD balances price and performance for a variety of transactional workloads.
 - Use Cases: Boot-volumes, low-latency interactive apps, dev & test
 - Volume Size: 1 GB - 16 TB
 - Max IOPS: 10,000
 - Max throughput/volume: 160 MB/s
- Provisioned IOPS SSD (io1)
 - Highest performance SSD, designed for latency-sensitive transactional workloads.
 - Use Cases: I/O-intensive applications, NoSQL & relational databases.

- Volume Size: 4 GB - 16 TB
 - Max IOPS: 32,000
 - Max throughput/volume: 500 MB/s
- Throughput Optimized HDD (st1)
 - Low-cost HDD, designed for frequently accessed, throughput-intensive workloads.
 - Use Cases: Big data, data warehouses, log processing
 - Volume Size: 500 GB - 16 TB
 - Max Volume: 500
 - Max throughput/volume: 500 MB/s
- Cold HDD (sc1)
 - Lowest cost HDD, designed for less frequently accessed workloads.
 - Use Cases: Colder data requiring fewer scans per day such as File Servers
 - Volume Size: 500 GB - 16 TB
 - Max Volume: 250
 - Max throughput/volume: 250 MB/s

General Purpose SSD – IOPS and Volumes

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 10,000 IOPS (at 3,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver the provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

I/O Credits and Burst Performance

When your volume requires more than the baseline performance I/O level, it simply uses I/O credits in the credit balance to burst to the level of performance needed, up to a maximum of 3,000 IOPS.

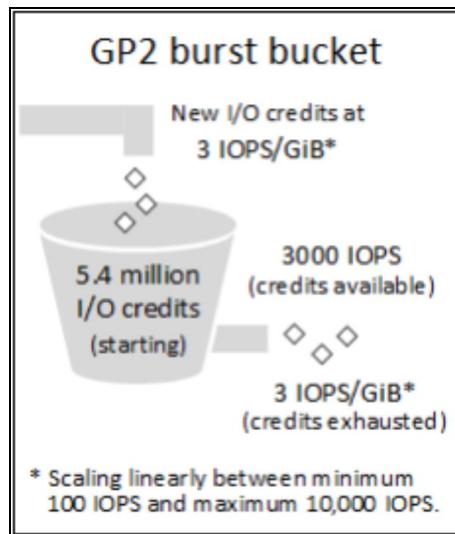


Figure 16. GP2 Burst Bucket

Each volume receives an initial I/O credit balance of 5,400,000 I/O credits. This is enough to sustain the maximum burst performance of 3,000 IOPS for 30 minutes. When you are not going over your provisioned IO level, (bursting), you will be earning credits.

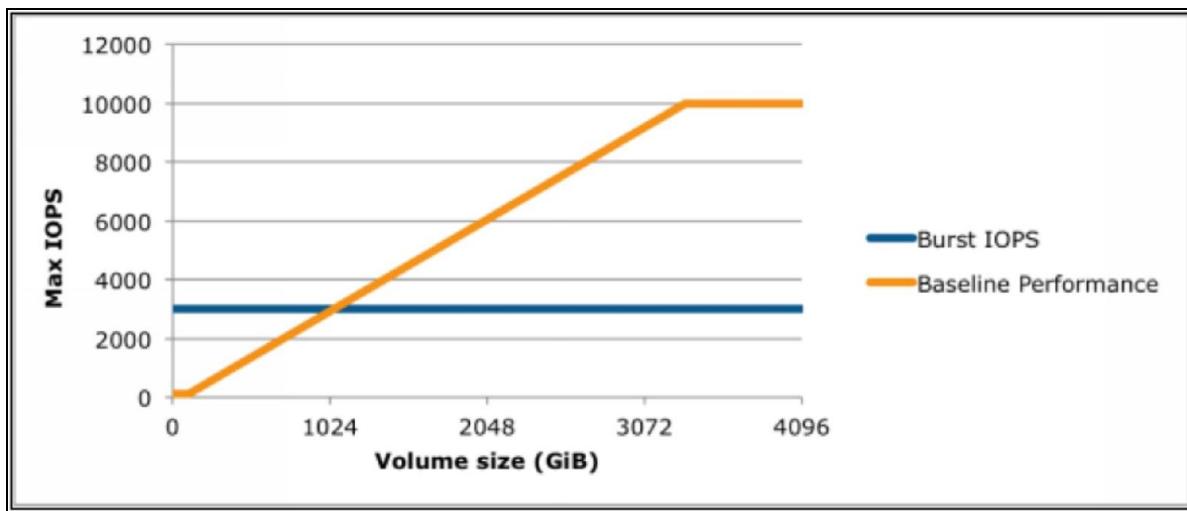


Figure 15. Burst Performance Graph

The following table lists several volume sizes and the associated baseline performance of the volume (which is also the rate at which it accumulates I/O credits), the burst duration at the 3,000 IOPS maximum (when starting with a full credit balance), and the time in seconds that the volume would take to refill an empty credit balance.

Volume size (GiB)	Baseline performance (IOPS)	Maximum burst duration @ 3,000 IOPS (seconds)	Seconds to fill empty credit balance
1	100	1862	54,000
100	300	2,000	18,000
214 (Min. size for max. throughput)	642	2,290	8,412
250	750	2,400	7,200
500	1,500	3,600	3,600
750	2,250	7,200	2,400
1,000	3,000	N/A	N/A
3,334 (Min. size for max. IOPS)	10,000	N/A	N/A
16,384 (16 TiB, max. volume size)	10,000	N/A	N/A

Table 1. Volume sizes and baseline performances

Pre-Warming EBS Volumes

New EBS volumes receive their maximum performance the moment that they are available and do not require initialization (formerly known as pre-warming). However, storage blocks on volumes that were restored from snapshots must be initialized (pulled down from Amazon S3 and written to the volume) before you can access the block. This preliminary action takes time and can cause a significant increase in the latency of an I/O operation the first time each block is accessed. For most applications, amortizing this

cost over the lifetime of the volume is acceptable. Performance is restored after the data is accessed once.

You can avoid this performance hit in a production environment by reading from all of the blocks on your volume before you use it. This process is called initialization. For a new volume created from a snapshot, you should read all the blocks that have data before using the volume.

EBS CloudWatch Metrics

Amazon Elastic Block Store (Amazon EBS) sends data points to CloudWatch for several metrics. Amazon EBS General Purpose SSD (gp2), Throughput Optimized HDD (st1), Cold HDD (sc1), and Magnetic (standard) volumes automatically send five-minute metrics to CloudWatch. Provisioned IOPS SSD (io1) volumes automatically send one-minute metrics to CloudWatch. Data is only reported to CloudWatch when the volume is attached to an instance. The AWS/EBS namespace includes the following metrics:

Metric	Description
VolumeReadBytes	
VolumeWriteBytes	It provides information on the I/O operations in a specified period. The Sum statistic reports the total number of bytes transferred during the period. The Average statistic reports the average size of each I/O operation during the period, except on volumes attached to a C5, C5d, i3.metal, M5, or M5d instance, where the average represents the average over the specified period. The SampleCount statistic reports the total number of I/O operations during the period, except on volumes attached to a C5, C5d, M5, or M5d instance, where the sample count

	<p>represents the number of data points used in the statistical calculation. Data is reported to CloudWatch only when the volume is active.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>Units: Bytes</p>
VolumeReadOps VolumeWriteOps	<p>The total number of I/O operations in a specified period.</p> <p>To calculate the average I/O operations per second (IOPS) for the period, divide the total operations in the period by the number of seconds in that period.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>Units: Count</p>
VolumeTotalReadTime VolumeTotalWriteTime	<p>The total number of seconds spent by all operations that completed in a specified period. If multiple requests are submitted at the same time, this total could be greater than the length of the period. For example, for 5 minutes (300 seconds): if 700 operations completed during that period, and each</p>

	<p>operation took 1 second, the value would be 700 seconds.</p> <p>The Average statistic on this metric is not relevant for volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>Units: Seconds</p>
VolumeIdleTime	<p>The total number of seconds in a specified period when no read or write operations were submitted.</p> <p>The Average statistic on this metric is not relevant for volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>Units: Seconds</p>
VolumeQueueLength	<p>The number of reads and write operation requests waiting to be completed in a specified period.</p>

	<p>The Sum statistic on this metric is not relevant for volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>The Minimum and Maximum statistics on this metric are supported only by volumes attached to a C5, C5d, i3.metal, M5, or M5d instance.</p> <p>Units: Count</p>
VolumeThroughputPercentage	<p>Used with Provisioned IOPS SSD volumes only. The percentage of I/O operations per second (IOPS) delivered of the total IOPS provisioned for an Amazon EBS volume. Provisioned IOPS SSD volumes deliver within 10% of the provisioned IOPS performance 99.9% of the time over a given year.</p> <p>During a write, if there are no other pending I/O requests in a minute, the metric value will be 100 percent. Also, a volume's I/O performance may become degraded temporarily due to an action you have taken (for example, creating a snapshot of a volume during peak usage, running the volume on a non-EBS-optimized instance, or accessing data on the volume for the first time).</p> <p>Units: Percent</p>

VolumeConsumedReadWriteOps	<p>Used with Provisioned IOPS SSD volumes only. The total amount of reading and write operations (normalized to 256K capacity units) consumed in a specified period.</p> <p>I/O operations that are smaller than 256K each count as one consumed IOPS. I/O operations that are larger than 256K are counted in 256K capacity units. For example, a 1024K I/O would count as four consumed IOPS.</p> <p>Units: Count</p>
-----------------------------------	---

Table 2. AWS/EBS namespaces and metrics

EBS Volume Status Check

Volume status checks enable you to better understand, track, and manage potential inconsistencies in the data on an Amazon EBS volume. They are designed to provide you with the information that you need to determine whether your Amazon EBS volumes are impaired, and to help you control how a potentially inconsistent volume is handled.

Volume status checks are automated tests that run every 5 minutes and return a pass or fail status. If all checks pass, the status of the volume is ok. If a check fails, the status of the volume is impaired. If the status is insufficient data, the checks may still be in progress on the volume. You can view the results of volume status checks to identify any impaired volumes and take any necessary actions.

Volume Status	I/O enabled Status	I/O performance status (only available for Provisioned IOPS volumes)
ok	Enabled (I/O Enabled or I/O Auto-Enabled)	Normal (Volume performance is as expected)

warning	Enabled (I/O Enabled or I/O Auto-Enabled)	Degraded (Volume performance is below expectations) Severely Degraded (Volume performance is well below expectations)
impaired	Enabled (I/O Enabled or I/O Auto-Enabled) Disabled (Volume is offline and pending recovery, or is waiting for the user to enable I/O)	Stalled (Volume performance is severely impacted) Not Available (Unable to determine I/O performance because I/O is disabled)
insufficient-data	Enabled (I/O Enabled or I/O Auto-Enabled) Insufficient Data	Insufficient Data

Table 3. Volume Status Checks

Know that Degraded or Severely Degraded is “warning” and Stalled or Not Available is “impaired.”

Modifying EBS Volumes

If your Amazon EBS volume is attached to a current generation EC2 instance type, you can increase its size, change its volume type, or (for an io1 volume) adjust its IOPS performance, all without detaching it. You can apply these changes to detached volumes as well.

- Issue the modification command (console or command line)
- Monitor the progress of the modification
- If the size of the volume was modified, extend the volume’s file system to take advantage of the increased storage capacity.

Monitoring RDS

With Amazon RDS, you can monitor network throughput, I/O for reading, write, and metadata operations, client connections, and burst credit balances for your DB instances. AWS provides various tools that you can use to monitor Amazon RDS. You can configure some of these tools to do the monitoring for you, while some of the tools require manual intervention.

There are two types of monitoring available for RDS:

- In CloudWatch you can monitor RDS by metrics
- In RDS itself, you can monitor RDS by events

Amazon CloudWatch Metrics – Amazon RDS automatically sends metrics to CloudWatch every minute for each active database instance and cluster. You are not charged additionally for Amazon RDS metrics in CloudWatch.

Amazon RDS Events – Subscribe to Amazon RDS events to be notified when changes occur with a DB instance, DB cluster, DB snapshot, DB cluster snapshot, DB parameter group, or DB security group.

RDS Metrics

The AWS/RDS namespace includes the following metrics:

Metric	Description
BinLogDiskUsage	The amount of disk space occupied by binary logs on the master. It applies to MySQL read replicas. Units: Bytes
CPU utilization	It shows the percentage of CPU utilization. Units: Percent
DatabaseConnections	The number of database connections in use.

	Units: Count
DiskQueueDepth	The number of outstanding IOs (read/write requests) waiting to access the disk.
	Units: Count
FreeableMemory	The amount of available random access memory.
	Units: Bytes
FreeStorageSpace	The amount of available storage space.
	Units: Bytes
NetworkReceiveThroughput	The incoming (Receive) network traffic on the DB instance, including both customer database traffic and Amazon RDS traffic used for monitoring and replication.
	Units: Bytes/second
NetworkTransmitThroughput	The outgoing (Transmit) network traffic on the DB instance, including both customer database traffic and Amazon RDS traffic used for monitoring and replication.
	Units: Bytes/second
read IOPS	The average number of disk read I/O operations per second.
	Units: Count/Second

read latency	The average amount of time taken per-disk I/O operation. Units: Seconds
ReadThroughput	The average number of bytes read from disk per second. Units: Bytes/Second
ReplicaLag	The amount of time a Read Replica DB instance lags behind the source DB instance. It applies to MySQL, MariaDB, and PostgreSQL Read Replicas. Units: Seconds
swap usage	The amount of swap space used on the DB instance. Units: Bytes
WriteIOPS	The average number of disk write I/O operations per second. Units: Count/Second
WriteLatency	The average amount of time taken per-disk I/O operation. Units: Seconds
write throughput	The average number of bytes written to disk per second. Units: Bytes/Second

Table 4. Metrics & Descriptions

Monitoring ELB

You can use CloudWatch to monitor your load balancers, analyze traffic patterns, and troubleshoot issues with your load balancers and back-end instances. Elastic Load Balancing publishes data points to Amazon CloudWatch for your load balancers and your back-end instances.

Elastic Load Balancing reports metrics to CloudWatch only when requests are flowing through the load balancer. If requests are flowing through the load balancer, Elastic Load Balancing measures and send its metrics in 60-second intervals. If no requests are flowing through the load balancer or no data for a metric, the metric is not reported.

ELB Metrics

The AWS/ELB namespace includes the following metrics:

Metric	Description
HealthyHostCount	<p>The number of healthy instances registered with your load balancer. A newly registered instance is considered healthy after it passes the first health check. If cross-zone load balancing is enabled, the number of healthy instances for the LoadBalancerName dimension is calculated across all Availability Zones. Otherwise, it is calculated per Availability Zone.</p> <p>Reporting criteria: There are registered instances</p> <p>Statistics: The most useful statistics are Average and Maximum. The load balancer nodes determine these statistics. Note that</p>

	<p>some load balancer nodes might conclude that an instance is unhealthy for a brief period while other nodes determine that it is healthy.</p>
UnHealthyHostCount	<p>The number of unhealthy instances registered with your load balancer. An instance is considered unhealthy after it exceeds the unhealthy threshold configured for health checks. An unhealthy instance is considered healthy again after it meets the healthy threshold configured for health checks.</p> <p>Reporting criteria: There are registered instances</p> <p>Statistics: The most useful statistics are Average and Minimum. The load balancer nodes determine these statistics. Note that some load balancer nodes might decide that an instance is unhealthy for a brief period while other nodes determine that it is healthy.</p>
RequestCount	<p>The number of requests completed or connections made during the specified interval (1 or 5 minutes).</p> <p>[HTTP listener] The number of requests received and routed, including HTTP error responses from the registered instances.</p> <p>[TCP listener] The number of connections made to the registered instances.</p>

	<p>Reporting criteria: There is a nonzero value</p> <p>Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average all return 1.</p>
Latency	<p>[HTTP listener] The total time elapsed, in seconds, from the time the load balancer sent the request to a registered instance until the instance started to address the response headers.</p> <p>[TCP listener] The total time elapsed, in seconds, for the load balancer to successfully establish a connection to a registered instance.</p> <p>Reporting criteria: There is a nonzero value</p> <p>Statistics: The most useful statistic is Average. Use Maximum to determine whether some requests are taking substantially longer than the average. Note that Minimum is typically not helpful.</p>
HTTPCode_ELB_4XX	<p>[HTTP listener] The number of HTTP 4XX client error codes generated by the load balancer. Client errors are generated when a request is malformed or incomplete.</p> <p>Reporting criteria: There is a nonzero value</p>

	<p>Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average are all 1.</p>
HTTPCode_ELB_5XX	<p>[HTTP listener] The number of HTTP 5XX server error codes generated by the load balancer. This count does not include any response codes generated by the registered instances. The metric is reported if there are no healthy instances registered to the load balancer, or if the request rate exceeds the capacity of the instances (spillover) or the load balancer.</p> <p>Reporting criteria: There is a nonzero value</p> <p>Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average are all 1.</p>
HTTPCode_Backend_2XX , HTTPCode_Backend_3XX , HTTPCode_Backend_4XX , HTTPCode_Backend_5XX	<p>[HTTP listener] The number of HTTP response codes generated by registered instances. This count does not include any response codes generated by the load balancer.</p> <p>Reporting criteria: There is a nonzero value</p> <p>Statistics: The most useful statistic is Sum. Note that Minimum, Maximum, and Average are all 1.</p>
BackendConnectionErrors	The number of connections that were not successfully established between the load balancer and the registered instances.

	<p>Because the load balancer retries the connection when there are errors, this count can exceed the request rate. Note that this count also includes any connection errors related to health checks.</p> <p>Reporting criteria: There is a nonzero value</p> <p>Statistics: The most useful statistic is Sum. Note that Average, Minimum, and Maximum are reported per load balancer node and are not typically used. However, the difference between the minimum and maximum (or peak to average or average to trough) might be helpful to determine whether a load balancer node is an outlier.</p>
SurgeQueueLength	<p>The total number of requests (HTTP listener) or connections (TCP listener) that are pending routing to a healthy instance. The maximum size of the queue is 1,024. Additional requests or connections are rejected when the queue is full. For more information, see SpilloverCount.</p> <p>Reporting criteria: There is a nonzero value.</p> <p>Statistics: The most useful statistic is Maximum because it represents the peak of queued requests. The Average statistic can be helpful in combination with Minimum and Maximum to determine the</p>

	range of queued requests. Note that Sum is not helpful.
SpilloverCount	<p>The total number of requests that were rejected because the surge queue is full.</p> <p>[HTTP listener] The load balancer returns an HTTP 503 error code.</p> <p>[TCP listener] The load balancer closes the connection.</p> <p>Reporting criteria: There is a nonzero value</p> <p>Statistics: The most useful statistic is Sum. Note that Average, Minimum, and Maximum are reported per load balancer node and are not typically used.</p>

Table 5. ELB Metrics & Descriptions

Monitoring ElastiCache

ElastiCache Engines

Amazon ElastiCache is a managed, in-memory data store service. It simplifies and offloads the management, monitoring, and operation of in-memory cache environments, enabling you to focus on the differentiating parts of your applications. Amazon ElastiCache provides support for two engines, Memcached and Redis.

- Amazon ElastiCache for Redis – Manage and analyze fast moving data with a versatile in-memory data store.
- Amazon ElastiCache for Memcached – Build a scalable Caching Tier for data-intensive apps.

Amazon ElastiCache for Redis is a blazing fast in-memory data store that provides sub-millisecond latency to power internet-scale real-time applications. Built on open-source Redis and compatible with the Redis APIs, ElastiCache for Redis works with your Redis clients and uses the open Redis data format to store your data. Amazon ElastiCache for Redis is a great choice for real-time transactional and analytical processing use cases such as caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media streaming, queues, real-time analytics, and session store.

Amazon ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. It delivers the performance, ease-of-use, and simplicity of Memcached. ElastiCache for Memcached is fully managed, scalable, and secure - making it an ideal candidate for use cases where frequently accessed data must be in-memory. It is a popular choice for use cases such as Web, Mobile Apps, Gaming, Ad-Tech, and E-Commerce.



AWS Organizations

AWS Organizations is an account management service that allows you to consolidate multiple AWS accounts into an organization, enabling you to create a hierarchical structure that can be managed centrally.

With AWS Organizations, you can create multiple groups of AWS accounts known as the Organizational Units and then apply policies to those Organizational Units, commonly referred to as Service Control Policies (SCPs). These policies centrally control the use of AWS services across multiple AWS accounts, without the need for custom scripts and manual processes. Entities in the AWS accounts can only use the AWS services allowed by both the SCP and the AWS IAM policy for the account.

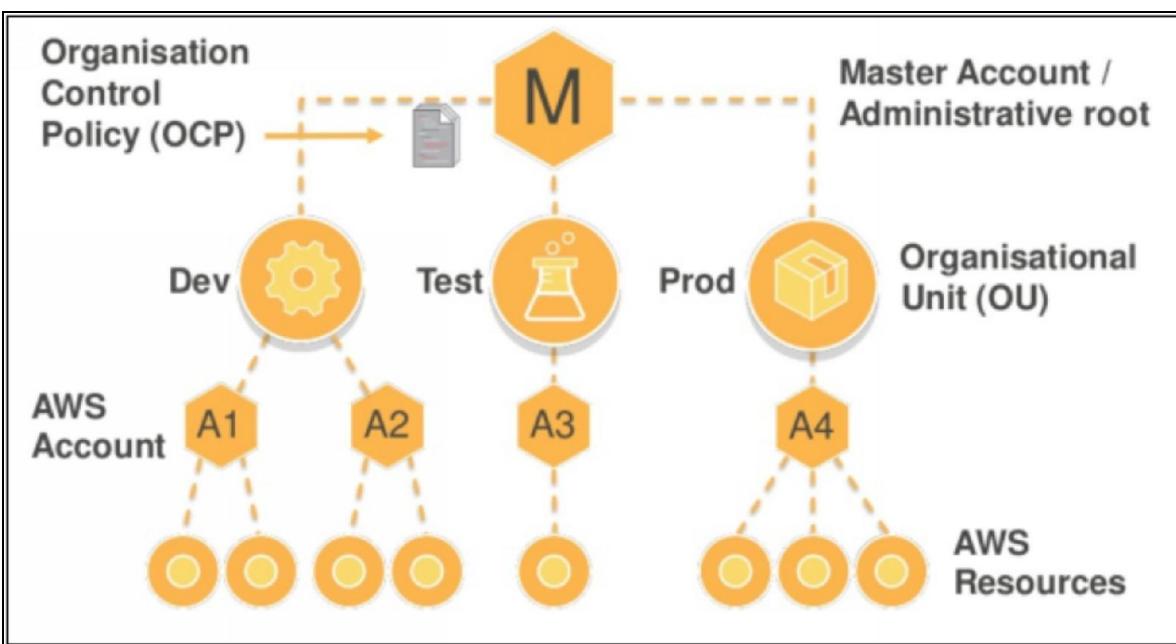


Figure 18. AWS Organization

AWS Organizations is available to all AWS customers at no additional charge in two feature sets:

- Only Consolidated billing features: This mode only provides the consolidated billing features and does not include the other advanced features of AWS Organizations, such as the use of policies to restrict what users and roles in different accounts can access.
- All features: This mode is the complete feature set that includes all the functionality of consolidated billing in addition to the advanced features that provides more control over the accounts in your organization.



Key Features of AWS Organizations

- Group-based account management:
Create separate groups of AWS accounts to use with development and production resources, and then apply different policies to each group.



- Policy framework for multiple AWS accounts:
AWS Organizations provides a policy framework for multiple AWS accounts. Apply policies to a group of accounts or all the accounts in your organization.



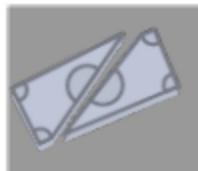
- API level control of AWS services:

Use service control policies (SCPs) to manage and centrally control access to AWS services at an API level across multiple AWS accounts.



- Account creation and management APIs

Automate the creation and management of new AWS accounts through APIs. APIs create new accounts programmatically.



- Consolidated billing

Set up a single payment method for all the AWS accounts in your organization through consolidated billing. It provides a combined view of charges incurred by all your accounts.



- Enable only consolidated billing features

Create new organizations with only the consolidated billing features enabled. Advanced policy controls such as Service Control Policies (SCPs) are not enabled.

Consolidated Billing

One of the critical features of AWS Organizations is the consolidation of the billing of all the AWS accounts in your organization, where you have a single AWS account as the paying master account linked with a set of all

other AWS accounts to form a simple one-level hierarchy. At the end of the month, you obtain a combined view of charges incurred by all of your AWS accounts. It also provides a cost report for each member account that is associated with the master paying account. Consolidated billing is available at no additional cost.

Consolidated billing has the following key benefits:

- One Bill – Get one bill for multiple accounts.
- Easy Tracking – Easily track each account's charges.
- Combined Usage – Combine usage from all accounts in the organization results in volume discounts.

 **EXAM TIP:** Paying Account should be used for billing purposes only. Do not deploy resources to the Paying Account. When monitoring is enabled on the Paying Account, billing data for all linked accounts are included. You can also create billing alerts for individual accounts separately.

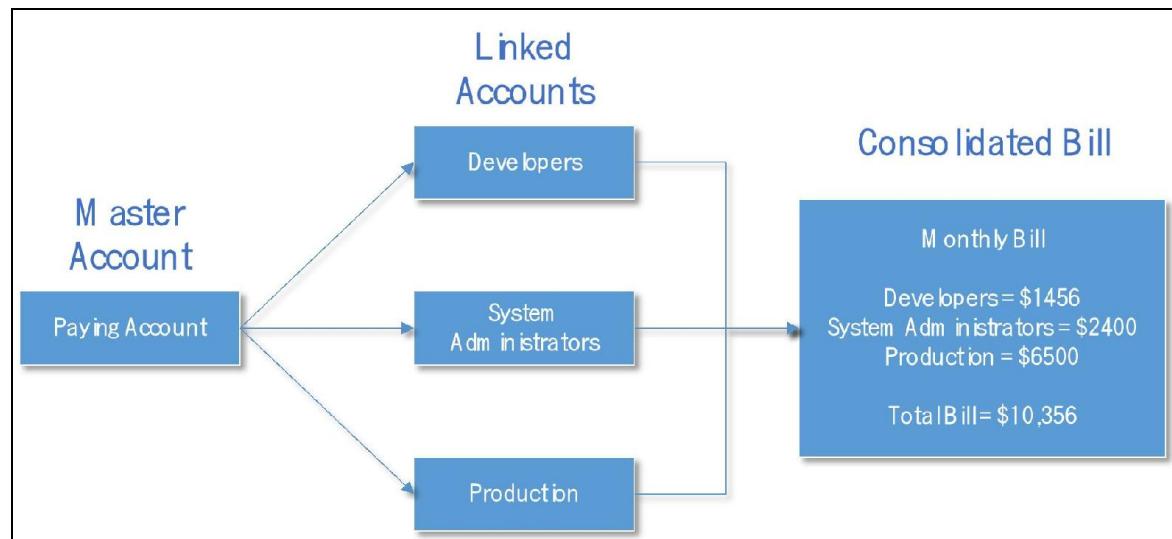


Figure 19. Consolidated Billing

With only the consolidated billing feature enabled, each member account is independent of the other member accounts. Unless the master account explicitly restricts linked accounts using policies, the owner of each member account can independently access resources, sign up for AWS

services and use AWS Premium Support. Account owners use their own IAM username and password with individually assigned account permissions in the organization.

Currently, there is a soft limit of 20 accounts per organization and a hard limit of one level of billing hierarchy; i.e., a master (paying) account cannot be in the same organization as another master (paying) account.



EXAM TIP: AWS CloudTrail is a service used to monitor account activity and deliver generated event logs to the associated account S3 Bucket. You can aggregate Log Files from multiple regions to a single S3 bucket of the Paying Account.

Consolidated Billing Examples

1. Volume Discounts

Services such as Amazon EC2 and Amazon S3 have tiered volume pricing that offers lower prices, the more you use the service. With consolidated billing, AWS determine which volume pricing tiers to apply by combining the usage from all accounts. Consider the following scenario:

Account Name	Data Transfer OUT
Developers	8 TB
System Administrators	5 TB
Production	3 TB

Table 6. Account's Data Out Stats

The Data Transfer OUT rates from Amazon S3 to the internet for US East (N. Virginia) Region are as follows:

Data Transfer Volume	Pricing	
Up to 1 GB / Month	\$0.00 per GB	
Next 10 TB / Month	\$0.09 per GB	

Next 40 TB / Month	\$0.085 per GB	
---------------------------	----------------	--

Table 7. Data Out Pricing

Without Consolidated billing, the cost will be calculated as:

- 8 TB will be charged as $(8 * 1024) * \$0.09 = \$ 737.28$
- 5 TB will be charged as $(5 * 1024) * \$0.09 = \$ 460.80$
- 3 TB will be charged as $(3 * 1024) * \$0.09 = \$ 276.48$
- Total Bill = \$ 1474.56 for 16 TB of data transfer

With Consolidated billing, data transfer charges for a total of 16 TB will be:

- Tier 1: First 10 TB will be charged as $(10 * 1024) * \$0.09 = \$ 921.60$
- Tier 2: Next 6 TB will be charged as $(6 * 1024) * \$0.085 = \$ 522.24$
- Total Consolidated Bill = \$ 1443.84 for 16 TB of data transfer

2. Reserved Instances:

As AWS Organizations deals with all the linked accounts in the organization as a single account, every member account can, therefore, get the hourly cost-benefit of Reserved Instances purchased by any other member account within the organization. Consider the following scenario of two linked accounts:

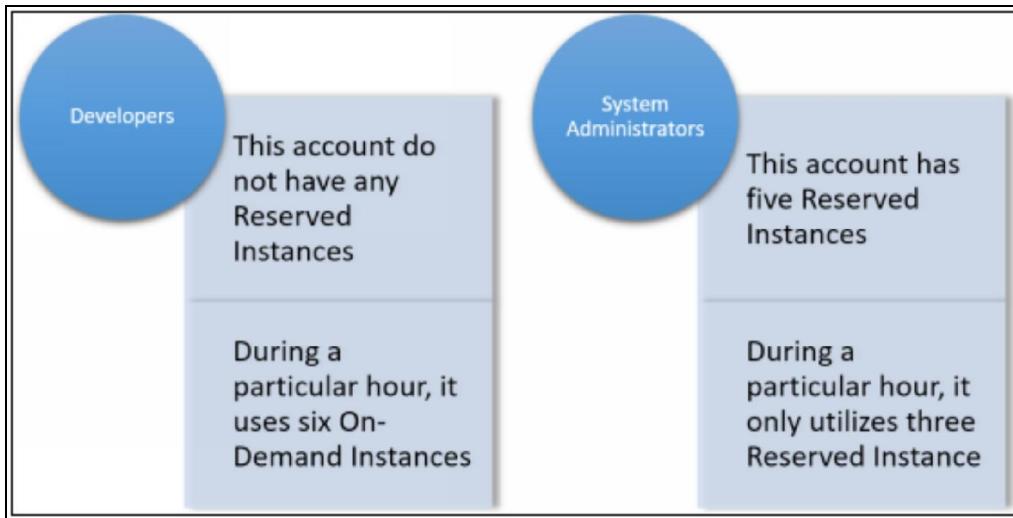


Figure 20. Linked accounts in an AWS organization

On the organization's consolidated bill, only nine instances will be charged from which five of them will be charged as Reserved Instances and the remaining four as regular On-Demand Instances. If the accounts were not linked to a single consolidated bill, six On-Demand Instances and five Reserved Instances would have been charged.

The linked accounts receive the cost-benefit from each other's Reserved Instances only if the launched instances are in the same Availability Zone having the same instance size and belonging to the same family of instance types.



EXAM TIP: Consolidated Billing allows you to get volume discounts on all your accounts. When consolidated billing is enabled, unused reserved instances for EC2 are applied across the group.

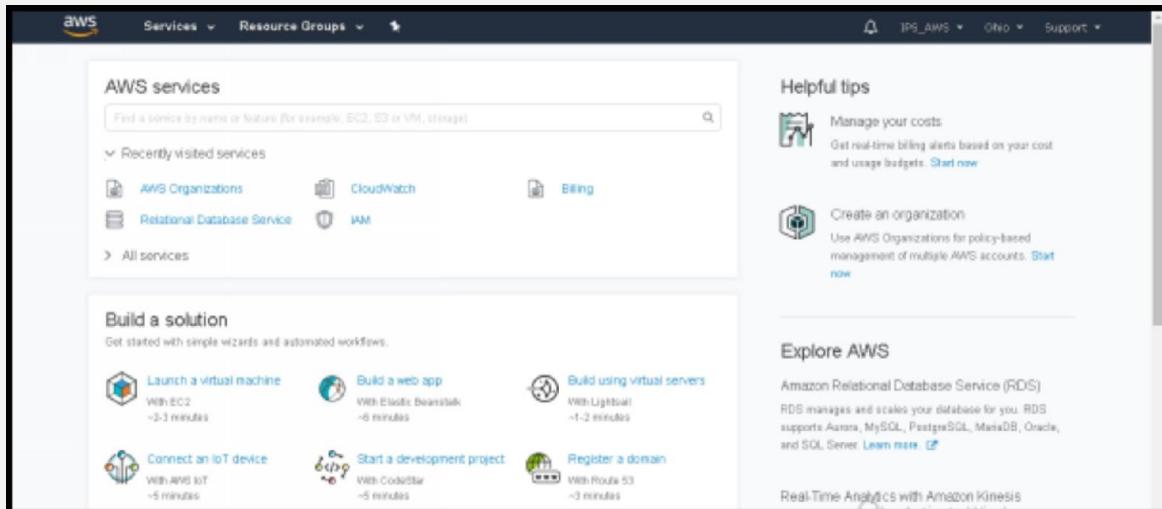


EXAM TIP: Going into the exam you are going to get scenario-based questions asking about how you can save cost, the answer to it is 'Consolidated Billing.'

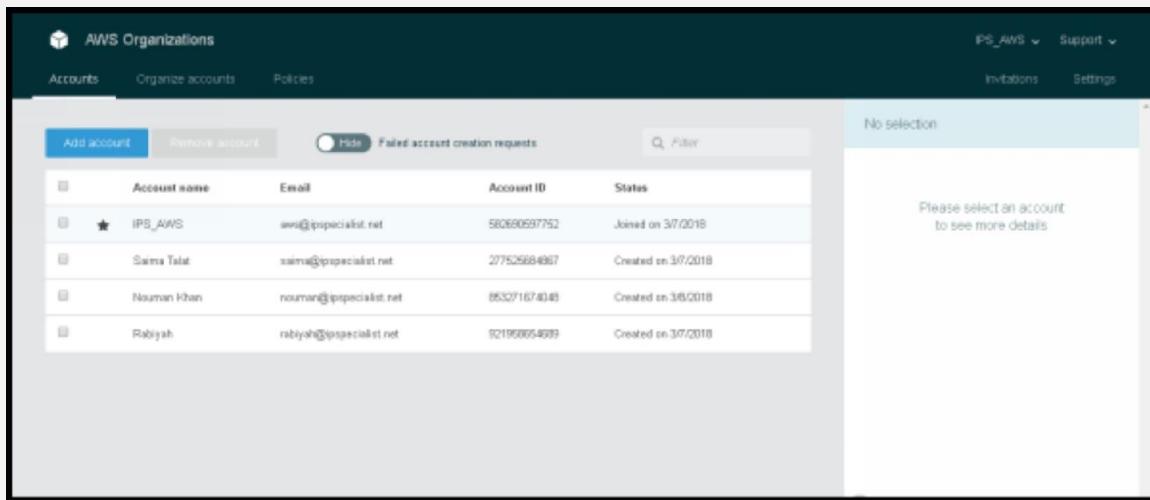
Lab 2.4 Make an AWS Organization:

Here are the steps to make an organization in your AWS management console.

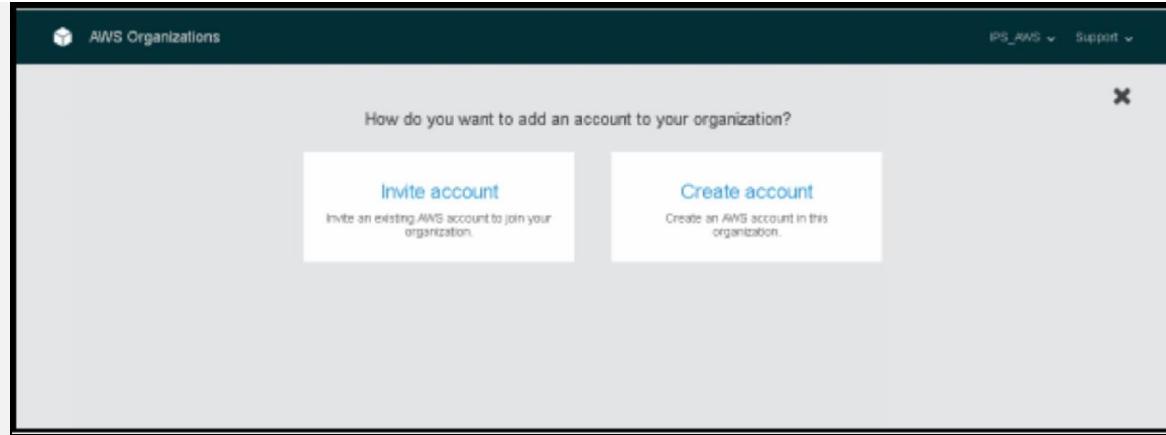
Step:01) Go to console and click on “Create an Organization.”



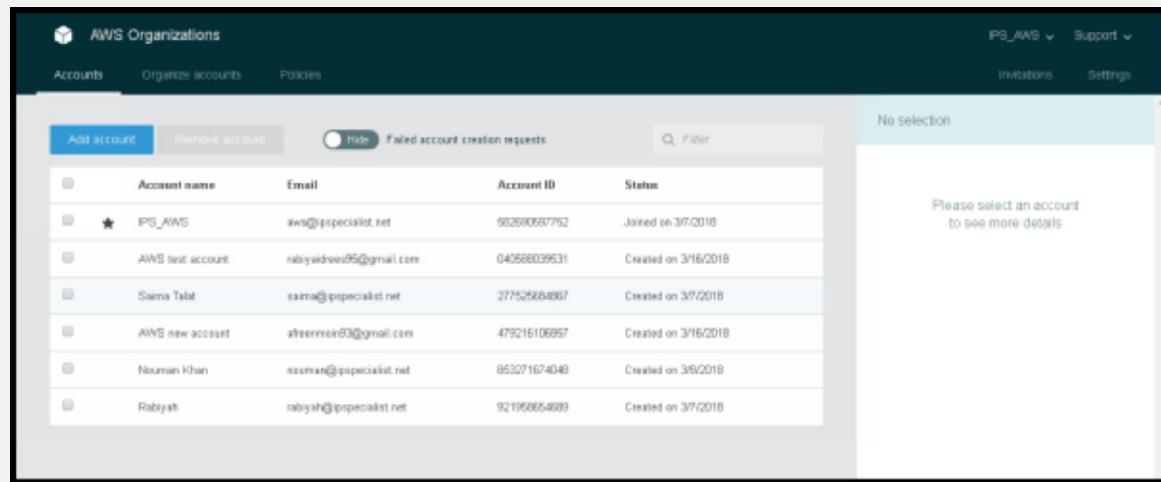
Step:02) If you have wanted to work on existing account, select from the list of existing accounts. If you are a new member, create a new account.



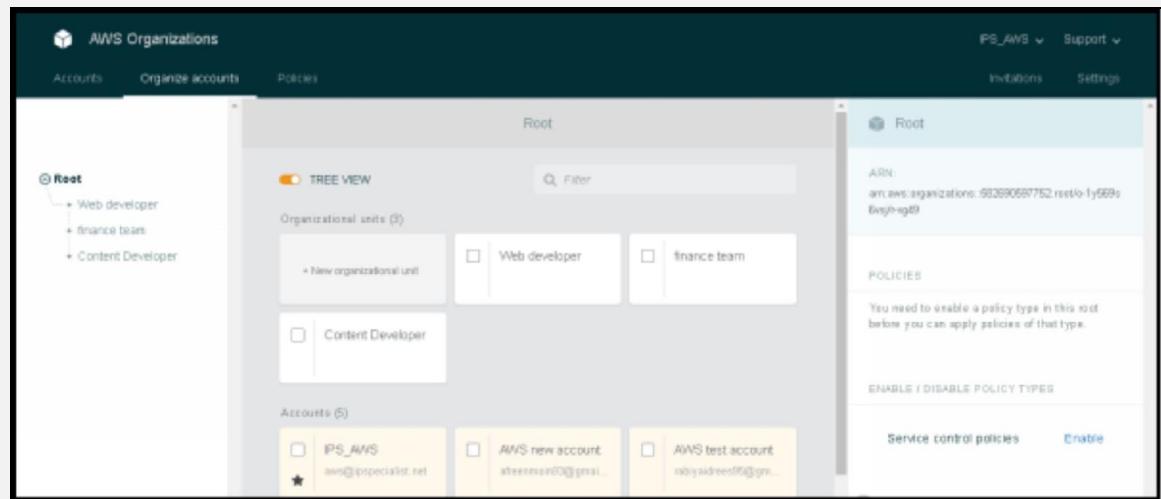
Step:03) Click on “Create account.”



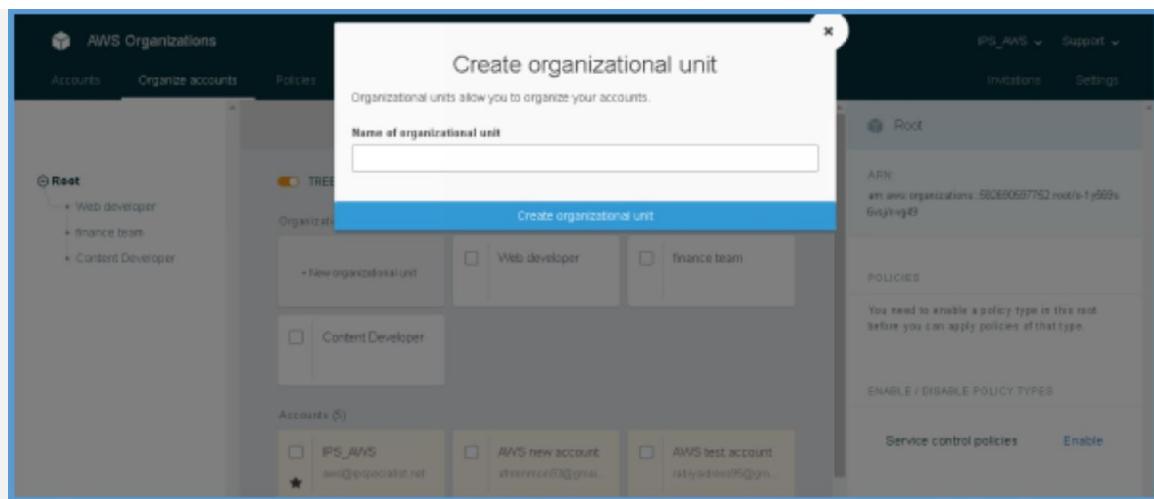
Step:04) In this step, the new accounts are created.



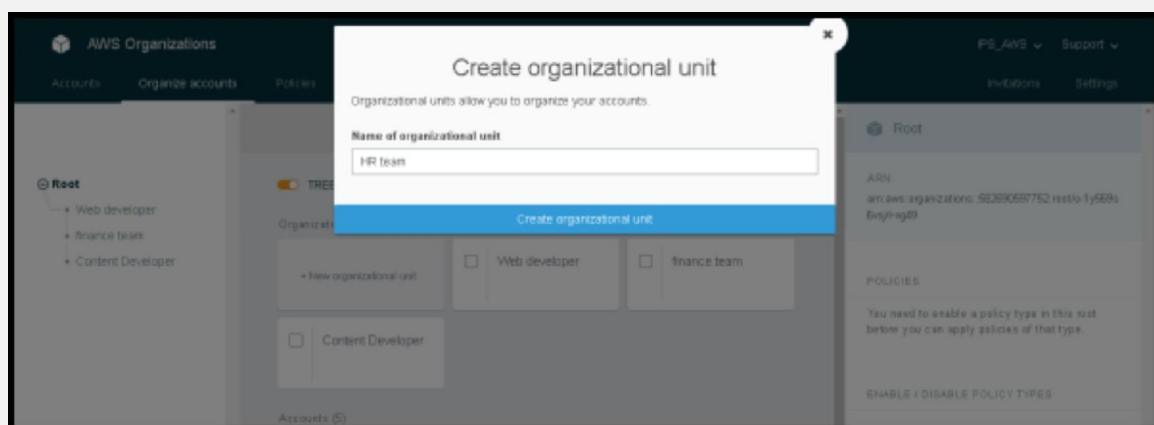
Step:05) Now click on “organize account.”



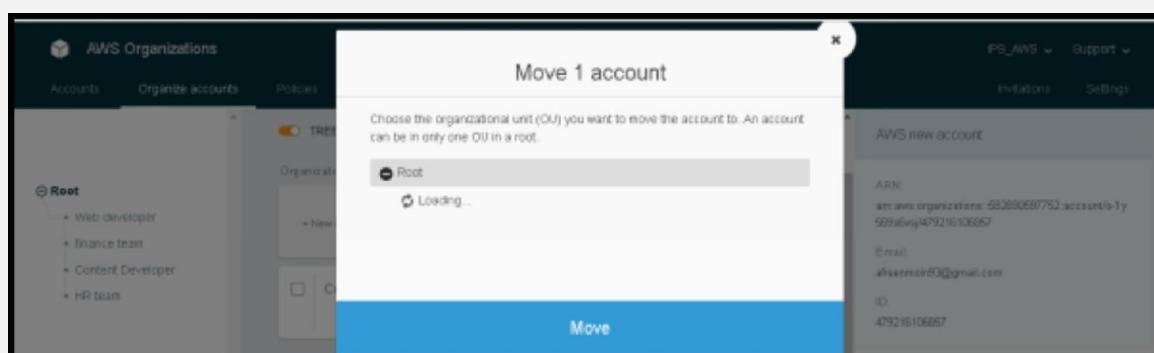
Step:06) Make different Organizational units for maintaining accounts.

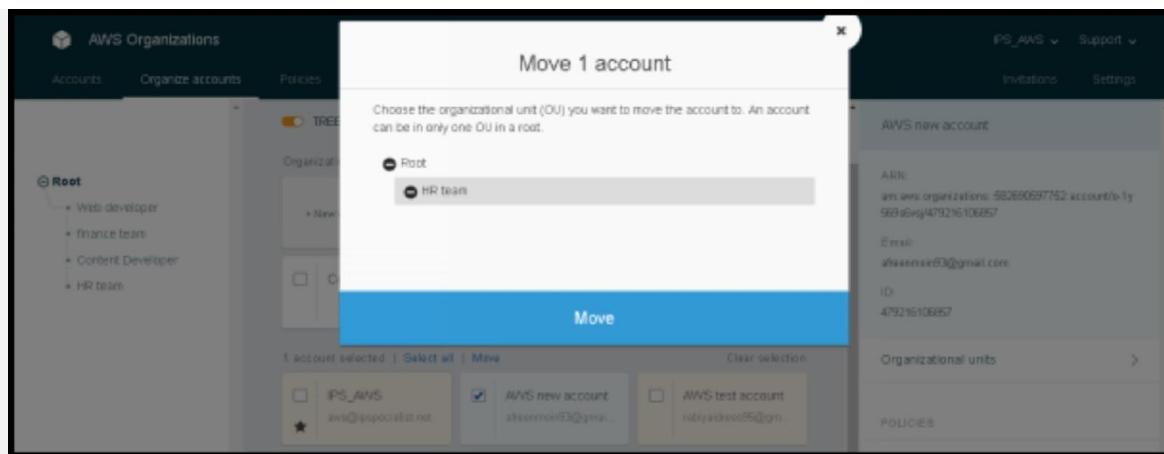


Step:07) Name the Organizational unit according to your requirement.

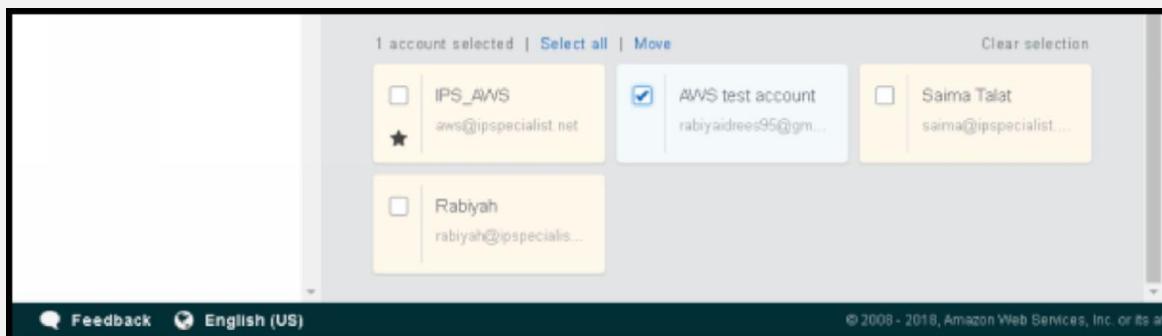


Step:08) Move the accounts to the “OU” of your own choice.

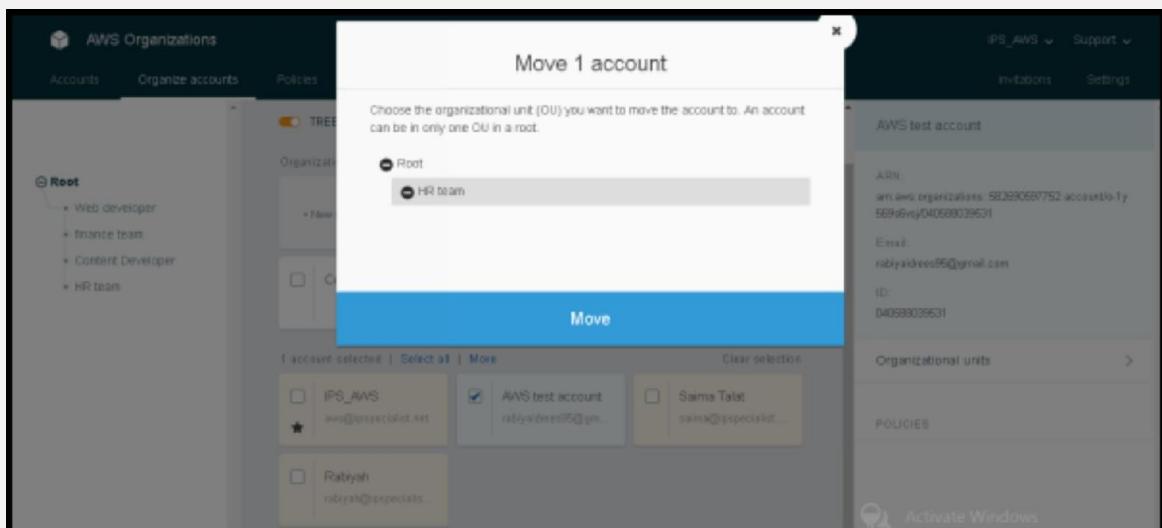




Step:10) Here is shown the selected account to move to “AU.”



Step:12) The below diagram shows the process of moving an account in an “AU.”



Step:13) To add different policies to these account click on Add Policy”.

The screenshot shows the AWS Organizations Policies page. At the top, there are tabs for 'Accounts', 'Organize accounts', and 'Policies'. Below the tabs, there are buttons for 'Create policy' and 'Delete policy'. A table lists one policy: 'FullAWSAccess' (Service control) which 'Allows access to every operation'. To the right of the table, a message says 'No selection' and 'Please select a policy to see more details'. There is also a small icon of a document.

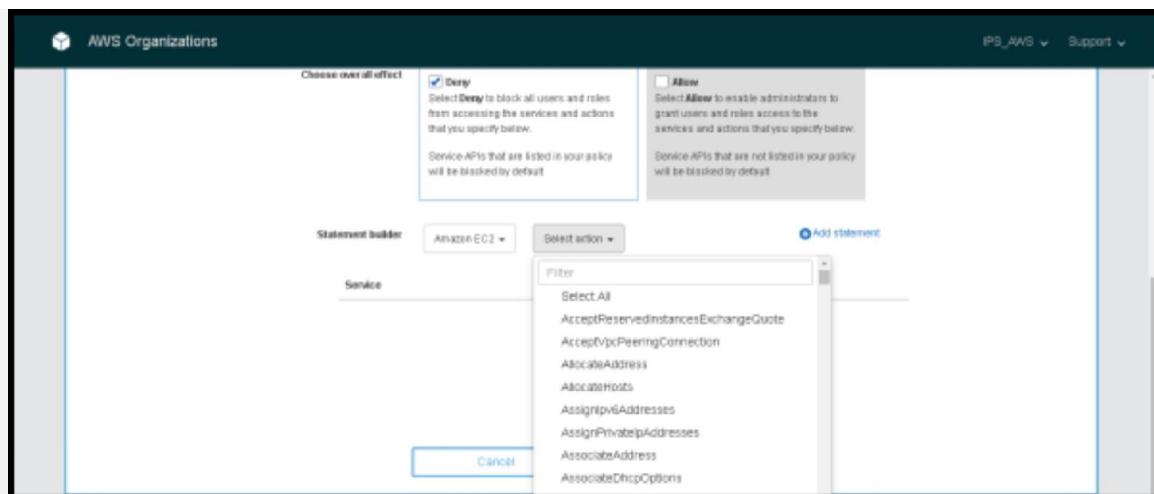
Step:14) Give the name of the policy of your own choice.

The screenshot shows the 'Create policy' wizard. It has a 'Policy name' field with 'IPS AWS Policy' entered, and a 'Description' field which is empty. Under 'Choose effect', there are two options: 'Deny' (selected) and 'Allow'. Both options have explanatory text. Below this is a 'Statement builder' section with dropdown menus for 'Service' (set to 'Amazon EC2'), 'Actions', and 'Effect'. There is also a 'Select service' button and a 'Select actions' button. An 'Add statement' button is located at the bottom right of the builder area.

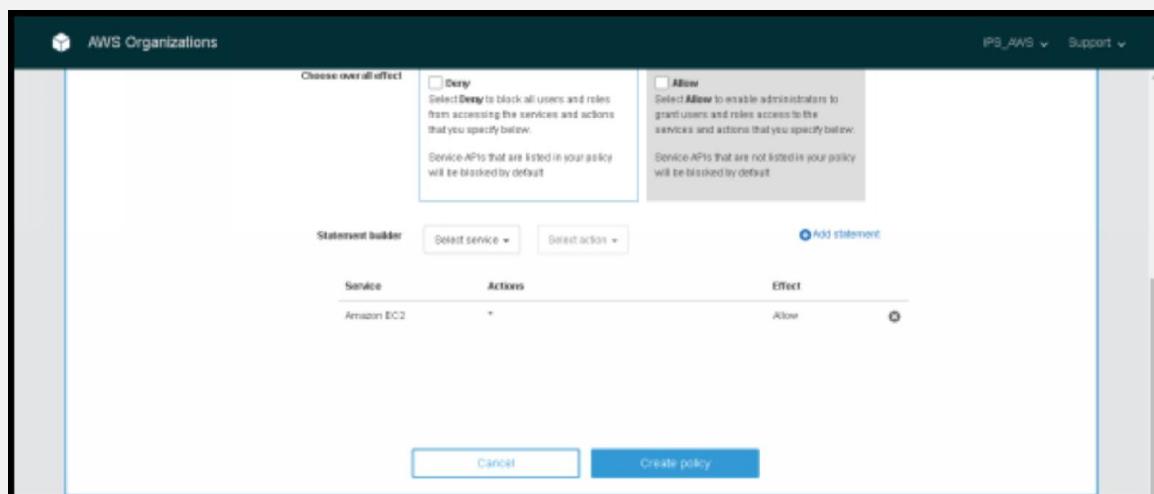
Step:15) Click on “Statement Builder” and select the option of your own choice.

The screenshot shows the 'Statement builder' interface. It has a 'Choose overall effect' section with 'Deny' selected. Below it is a 'Statement builder' area with a 'Service' dropdown set to 'Amazon EC2', a 'Selection' dropdown, and an 'Add statement' button. A scrollable list of services is shown under 'Service': AWS Config, AWS Cost and Usage Report, AWS Database Migration Service, AWS Device Farm, AWS Direct Connect, AWS Directory Service, Amazon DynamoDB, Amazon DynamoDB Accelerator, Amazon EC2 (which is checked), and Amazon EC2 Container Registry.

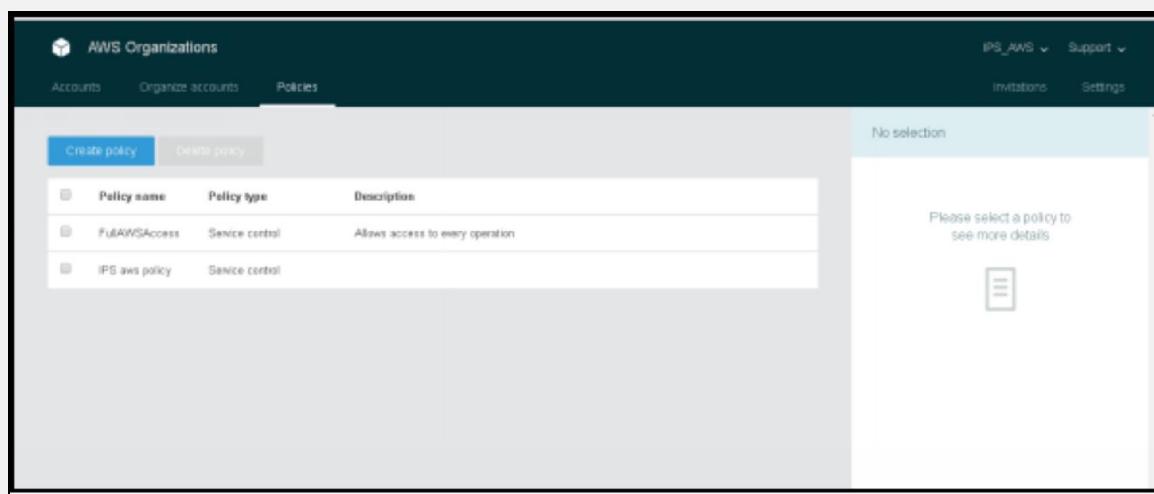
Step:16) To add some policies on the accounts, click on “allow” and to deny some policies, click on “Deny.”



Step: 17) Click on “Create Policy.”



Step:18) Select the policy you want to add on the account.



The screenshot shows the AWS Organizations console with the 'Policies' tab selected. On the left, there's a list of policies: 'FwIAWSAccess' (Service control, description: 'Allows access to every operation') and 'IPS aws policy' (Service control, checked). A modal window is open for the 'IPS aws policy', displaying its ARN (arn:aws:organizations::58269397752:policy/o-1y689a6nj/service_control_policy/p-y1p7mg) and a 'Policy editor' button.

Step:20) Select an account and attach policy on it.

The screenshot shows the same AWS Organizations console as before, but now the 'Attach' button is visible next to each account name in the list on the right. The accounts listed are: 'AWS test account' (40958839531), 'Salma Talat' (277525884867), 'AWS new account' (479218106857), 'IPS_AWS' (58269397752), and 'Nouman Khan' (05121874848).

Monitor Charges Using Billing Alarms

You can monitor your estimated AWS charges using Amazon CloudWatch. When you enable the monitoring of estimated charges for your AWS account, the estimated charges are calculated and sent several times daily to CloudWatch as metric data. This data includes the estimated charges for every service in AWS that you use, in addition to the estimated overall total of your AWS charges. The alarm triggers when your account billing exceeds the threshold you specify. It triggers only when actual billing exceeds the threshold.

Enable Billing Alerts

Before you can create an alarm for your estimated charges, you must enable billing alerts, so that you can monitor your estimated AWS charges and create an alarm using billing metric data.

Create a Billing Alarm

After you've enabled billing alerts, you can create a billing alarm. In this procedure, you create an alarm that sends an email message when your estimated charges for AWS exceed a specified threshold.

Check the Alarm Status

You can check the status of your billing alarm.

Delete a Billing Alarm

You can delete your billing alarm when you no longer need it.

Cost Optimization

By following a few simple steps, you can effectively control your AWS costs:

1. Right-size your services to meet capacity needs at the lowest cost
2. Save money when you reserve
3. Use the spot market
4. Monitor and track service usage
5. Use Cost Explorer to optimize savings.

Many large enterprise organizations customize their contracts with AWS to further optimize their costs and meet their needs. Different pricing models are available for some of the AWS products, offering you the flexibility to access services according to your requirements.

On-Demand Instance

With on-demand instances, you pay for computing capacity by the hour, with no minimum commitments required.

Reserved Instance

Reserved Instances allow you to reserve computing capacity in advance for long-term savings. It provides significant discounts (up to 60%) compared to On-Demand Instance pricing.

The following table compares one-year and three-year savings from the use of reserved instances versus on-demand instances. The figures are based on pricing as of January 2015 on an m3. Large Linux instance type in the US East (N. Virginia) region.

	No Upfront	Partial Upfront	All Upfront	On-Demand
1 Year	\$876.00	\$767.12	\$751.00	\$1226.40

3 Years		\$1461.40	\$1373.00	\$3679.20
Savings Year	1	29%	37%	39%
Savings Years	3		60%	63%

Table 8. 1 year and 3 years' price comparison

Spot Instance

You can bid for unused Amazon Elastic Compute Cloud (Amazon EC2) capacity. Instances are charged at Spot Price, which is set by Amazon EC2 and fluctuates, depending on supply and demand. If your bid exceeds the current Spot Price, your requested instances will run until either you terminate them or the Spot Price increases above your bid.

Pricing is tiered for storage and data transfer. The more you use, the less you pay per gigabyte (GB). Volume discounts are also available.

Chapter 3: High Availability

Introduction

Amazon Web Services provides services and infrastructure to build reliable, fault-tolerant, and highly available systems in the cloud. Most of the higher-level services, such as Amazon Simple Storage Service (S3), Amazon SimpleDB, Amazon Simple Queue Service (SQS), and Amazon Elastic Load Balancing (ELB), have been built with fault tolerance and high availability in mind. Services that provide basic infrastructures, such as Amazon Elastic Compute Cloud (EC2) and Amazon Elastic Block Store (EBS), provide specific features, such as availability zones, elastic IP addresses, and snapshots, that a fault-tolerant and the highly available system must take advantage of and use correctly.

Fault Tolerance and High Availability

Load balancing is an effective way to increase the availability of a system. Instances that fail can be replaced seamlessly behind the load balancer while other instances continue to operate. Elastic Load Balancing can be used to balance across instances in multiple availability zones of a region.



Figure 21. AWS services to provide Fault-Tolerance and High availability

Elasticity and Scalability:

Introduction:



Note: Remember that in the world of cloud computing these two words are closely related but they are not really the same thing.

When we talk about the implementation of [cloud computing](#) in the enterprise networks, CIOs and other scholars must estimate possible cloud solutions on different criteria. Most IT departments are focusing on cost, performance, security & reliability. However, IT managers are also starting to add two more criteria to the list of cloud computing. These two criteria are scalability and elasticity.

Elasticity:

Elasticity is primarily the creation of virtual machines to meet the real-time requirements of resources in Cloud Computation.

By using Elasticity in a programmed manner, the total amount of infrastructure of the system is saved. For this purpose, the foundation of the system is compressed or expanded according to its desires and needs.

Scalability:

Scalability is mainly used to handle the increasing work on the application layer and also to make a place for that growing data in the system .

Scalability increases the ability of infrastructure in the system (hardware and software) without affecting their performance.

Systems that are expected to grow over time need to be built on top of a scalable architecture. Scalable architecture can support the growth of users, traffic, or data size with no drop in performance. This system should provide that scale in a right manner where adding extra resources results in at least a proportional increase in ability to serve additional load. Growth should introduce economies of scale, and cost should follow the same measurement that creates business value out of that system. Whereas cloud computing

provides virtually infinite on-demand capacity, but your design must be able to take advantage of those resources smoothly.

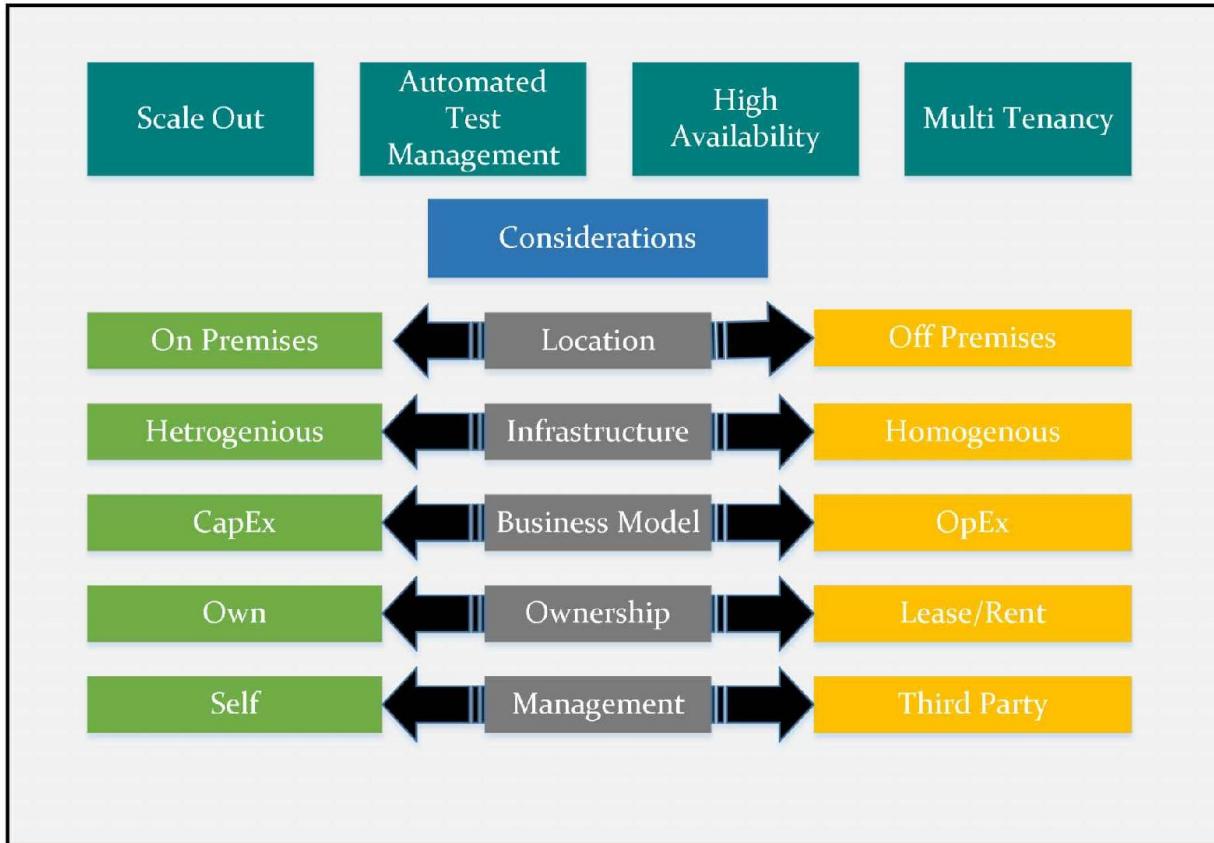


Figure 22. Business Module of Scale up and Scale out

There are two ways to scale an IT architecture:

- Vertically
- Horizontally.

Scaling Vertically

Vertical Scaling takes place through an increase in the requirements of an individual resource (e.g., upgrading a server with a larger hard drive or a faster CPU). If you are using an Amazon EC2, this can simply be achieved by stopping an instance and resizing it to an instance type that contains more RAM, CPU, IO, or networking capabilities.

Scaling Horizontally

Horizontal Scaling takes place through an increase in the number of resources (e.g., adding more hard drives to a storage group or adding more servers to support an application). This is a great way to architect Internet-scale applications that have control over the elasticity of cloud computing. Remember that not all architectures are designed to distribute their workload to several resources.

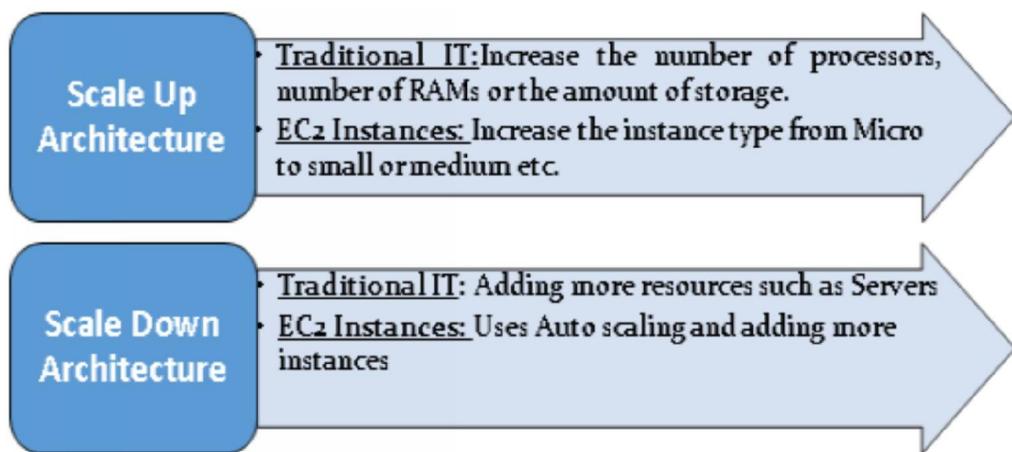


Figure 23. Vertical and Horizontal Scaling

Scale Out:

The following diagram gives the idea behind the Scale-out infrastructure.

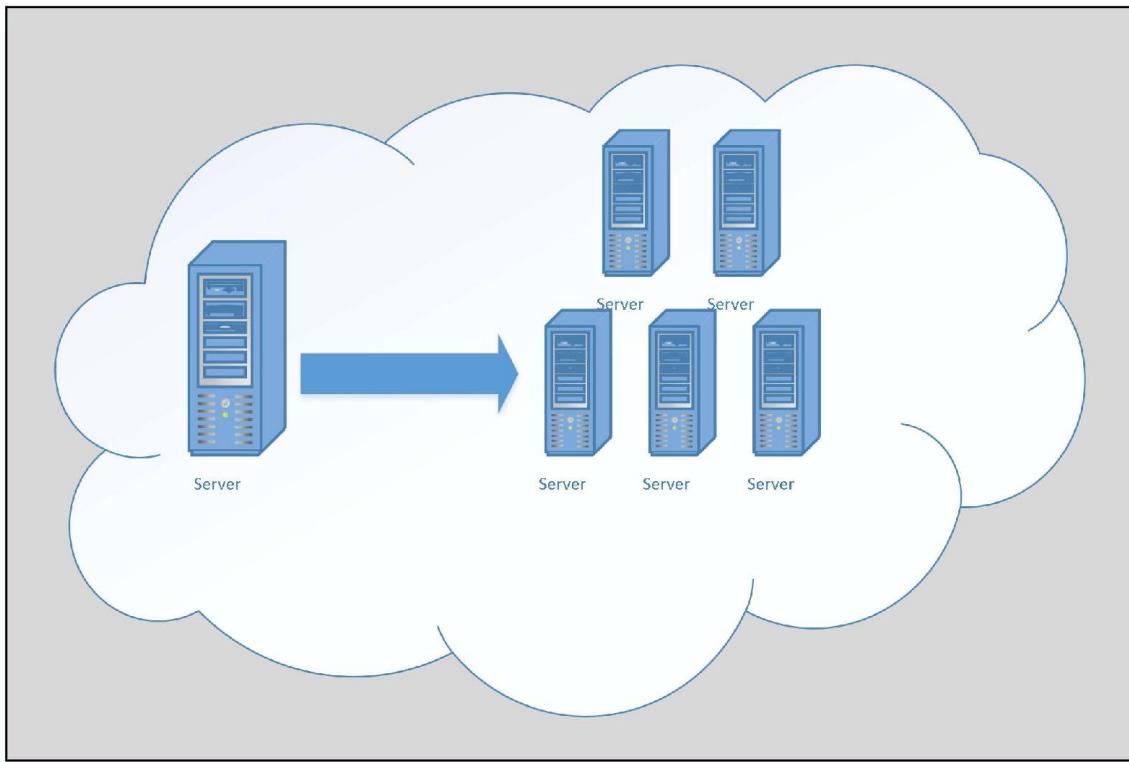


Figure 24. Scale-Out Infrastructure

Scale Up:

The following diagram gives the idea behind the scale up infrastructure.

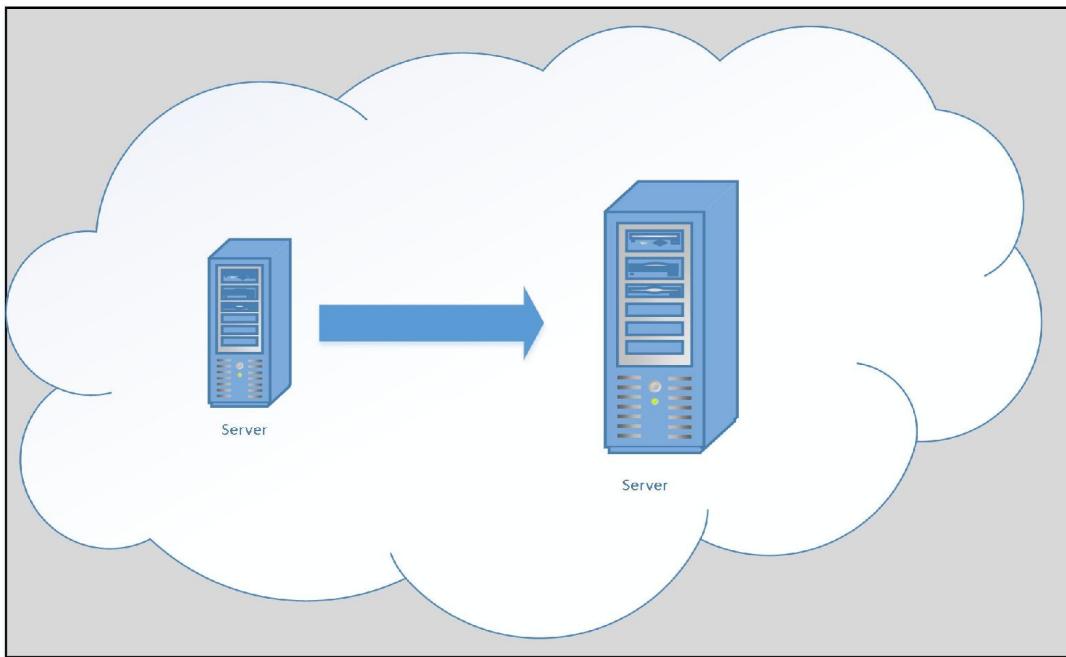


Figure 25. Scale Up Infrastructure

Below are the tables to show the difference between old network performance and EBS Optimized Network Performance.

Old Table Network Performance:

Instances Types:

The following table lists the applications for each of Amazon's EC2 instance type;

Instance Family	Instance Type	Architecture	vCPU	ECU	Memory (GIB)	Instance Storage	Ebs Optimized bandwidth	IP	Network performance
General Purpose	M1.small	32 bit or 64 bit	1	1	1.7	1*160	-	8	Low
General Purpose	m.medium	32 bit or 64 bit	1	2	3.75	1*410	-	12	Moderate
General Purpose	M1.large	64 bit	2	4	7.5	2*420	1000 Mbps	30	Modertae
General Purpose	M1.xlarge	64 bit	4	8	15	2*840	5000 Mbps	60	High
General Purpose	M3.xlarge	64 bit	4	13	15	0-EBS only	500 Mbps	60	Moderate
General Purpose	M3.2xlarge	64 bit	8	26	30	0-EBS only	1000 Mbps	120	Moderate
Compute Optimized	C1.medium	32 bit or 64 bit	2	5	1.7	1*350	1000mbps	120	High
Compute Optimized	C1.xlarge	64 bit	8	20	7	4*420	-1000 Mbps	12	Moderate high
Compute Optimized	Cc1.4xlarge	64 bit	16	33.5	22.5	2*840	-*	1	Moderate
Compute Optimized	Cc2.8xlarge	64 bit	32	68	60.5	4*840	-	240	High
Memory Optimzed	M2.xlarge	64 bit	2	6.5	17.1	1*420	-	60	10 Gigabit
Memory Optimiozed	M2.2xlarge	64 bit	4	13	34.2	1*850	500 Mbps	120	Moderate

Table 9. Old instance's network performances

EBS Optimized Network Performance:

Instance Type	Dedicated EBS Throughput (Mbps)*	Max IOPS**	16K Bandwidth (Mb/s)**
c1.xlarge	1,000	8,000	125
c3.xlarge	500	4,000	62.5
c3.2xlarge	1,000	8,000	125
c3.4xlarge	2,000	16,000	250
c4.large	500	4,000	62.5
c4.xlarge	750	6,000	93.75
c4.2xlarge	1,000	8,000	125
c4.4xlarge	2,000	16,000	250
c4.8xlarge	4,000	32,000	500
d2.xlarge	750	6,000	93.75
d2.2xlarge	1,000	8,000	125
d2.4xlarge	2,000	16,000	250
d2.8xlarge	4,000	32,000	500
g2.2xlarge	1,000	8,000	125
i2.xlarge	500	4,000	62.5
i2.2xlarge	1,000	8,000	125

Table 10. EBS Optimized network Performance

Amazon Relational Database Service (RDS):

Amazon RDS makes it easy to set up, operate and scale the relational database in the cloud. When you do time to consume administrative tasks on the cloud such as hardware establishment, database setup, recovery, and backups. It offers cost-efficient and resizable capacity. By using Amazon RDS, you are free to focus on your applications so that you can give them the fast performance, high availability, security and compatibility they required.

There are several database engines on which Amazon RDS can be used which include the followings;

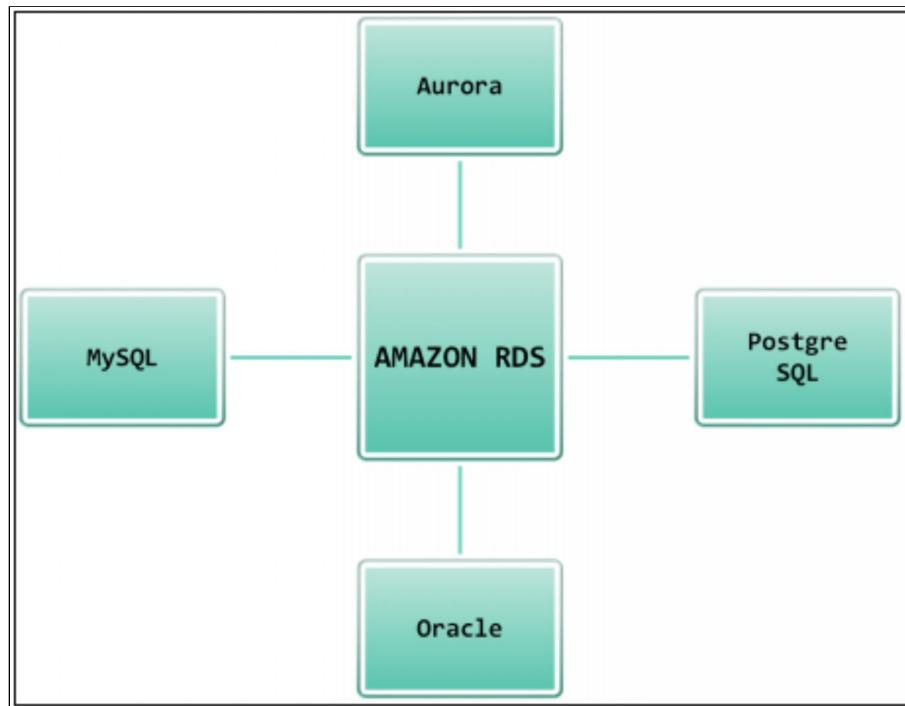


Figure 26. Amazon RDS Database Engines

To migrate your services to Amazon RDS, you can efficiently use the [AWS Database Migration Service](#). Also, you can replicate your existing databases to Amazon RDS.

RDS Multi Failover:

- Amazon RDS Multi-AZ deployments provide improved availability and durability for Database (DB) Instances.
- When you establish a Multi-AZ DB Instance, Amazon RDS creates a primary DB Instance by itself and replicates the data synchronously to a

backup instance in a different Availability Zone (AZ).

- Each AZ runs on its own physically distinct, independent infrastructure, and is designed to be highly reliable.
- In the case of infrastructure failure, Amazon RDS executes an automatic failover to the backup so that you can continue database operations as soon as the failover is complete.
- Since the endpoint for your DB Instance remains the same after a failover, your application can continue to run database operation without the need for manual administrative actions.

Advantages:

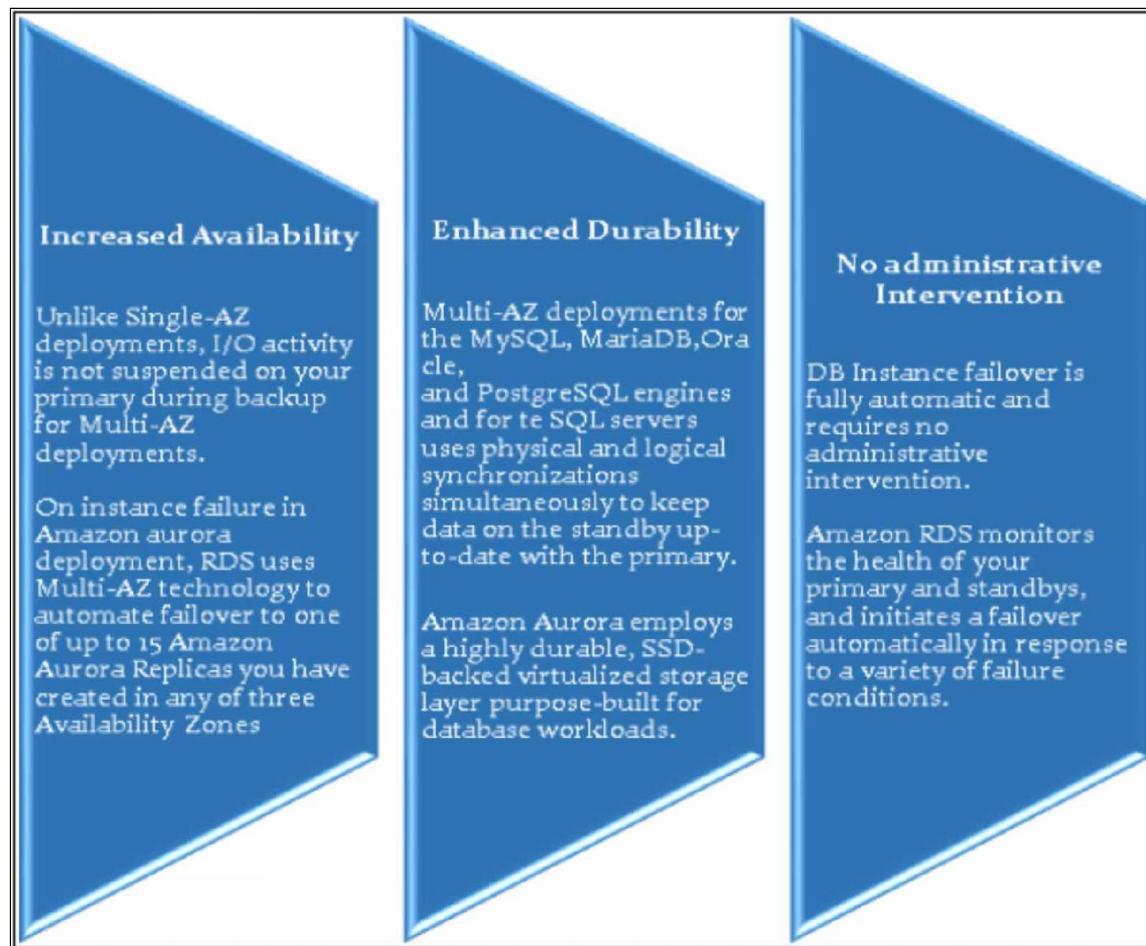


Figure 27. Advantages of Amazon RDS



Exam Tip:

You can force a failover from one AZ to another by restarting your instance. For this purpose, you have to restart your DB Instance API call in “AWS

Failover Conditions:

Amazon RDS identifies and automatically recovers from the most common failure scenarios for Multi-AZ deployments so that you can continue database operations as quickly as possible without administrative intervention. Amazon RDS automatically performs a failover in the event of any of the following:

- Loss of availability in primary Availability Zone
- Damage of network connectivity to the primary
- Compute unit failure on the primary



Note:

When operations such as DB Instance scaling or system upgrades like OS fixing are initiated for Multi-AZ deployments, for enhanced availability, they are applied first on the standby prior to an automatic failover. As a result, your availability impact is limited only to the time required for automatic failover to complete. Note that Amazon RDS Multi-AZ deployments do not failover automatically in response to

- Storage failure on the primary

RDS Using Read Replicas:

Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. This feature makes it easy to elastically scale out beyond the capacity limitations of a single DB Instance for read-heavy database workloads. You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from several copies of your data, thereby increasing total read throughput. Read replicas can also be promoted when

needed to become standalone DB instances. Read replicas are available in Amazon RDS for [MySQL](#), [MariaDB](#), and [PostgreSQL](#) as well as [Amazon Aurora](#).

- For the MySQL, MariaDB and PostgreSQL database engines, Amazon RDS creates a second DB instance using a snapshot of the source DB instance.
- It then uses the engines' built-in asynchronous replication to update the read replica whenever there is a change to the source DB instance.
- The read replica operates as a DB instance that allows only read-only connections; applications can connect to a read replica just as they would to any DB instance. Amazon RDS replicates all databases in the source DB instance.
- [Amazon Aurora](#) employs an SSD-backed virtualized storage layer purpose-built for database workloads.
- Amazon Aurora replicas share the same underlying storage as the source instance, lowering costs and avoiding the need to copy data to the replica nodes.

Advantages:

- Enhanced Performance:

Reduce the performance of load on your source DB Instance by sending read requests from your applications to the read replica.

To elastically scale out beyond the capacity limitations of a single DB Instance for read-heavy database workloads by using Read Replicas.

To further maximize read performance, add table indexes directly to Read Replicas, by using Amazon RDS for MySQL.

Because read replicas can be promoted to master status, they are valuable as part of a sharding implementation. To shard your database, add a read replica and push it to master status, then, from each of the resulting DB Instances, delete the data that belongs to the other shard.

- Increased Availability:

To provide complete availability mechanism to [Amazon RDS Multi-AZ Deployments](#) Read replicas in Amazon RDS for MySQL, MariaDB, and PostgreSQL are used.

If the source DB instance fails, then you can easily support a read replica.

You also have a facility to replicate DB instances across [AWS Regions](#) as a measure of your disaster rescue approach.

This functionality complements the synchronous replication, automatic failure detection, and failover provided with Multi-AZ deployments.

- Designed for Security:

When you create a read replica for Amazon RDS for MySQL, MariaDB, and PostgreSQL, Amazon RDS sets up a secure communications channel using public key encryption between the source DB instance and the read replica, even when replicating across regions.

Amazon RDS has a capability of establishing any AWS security configurations, such as adding security group entries, which are required to facilitate the secure channel.

Read replicas are also be created in any database engine such as MySQL, MariaDB, and PostgreSQL of your Amazon RDS by using [AWS Key Management Service \(KMS\)](#).

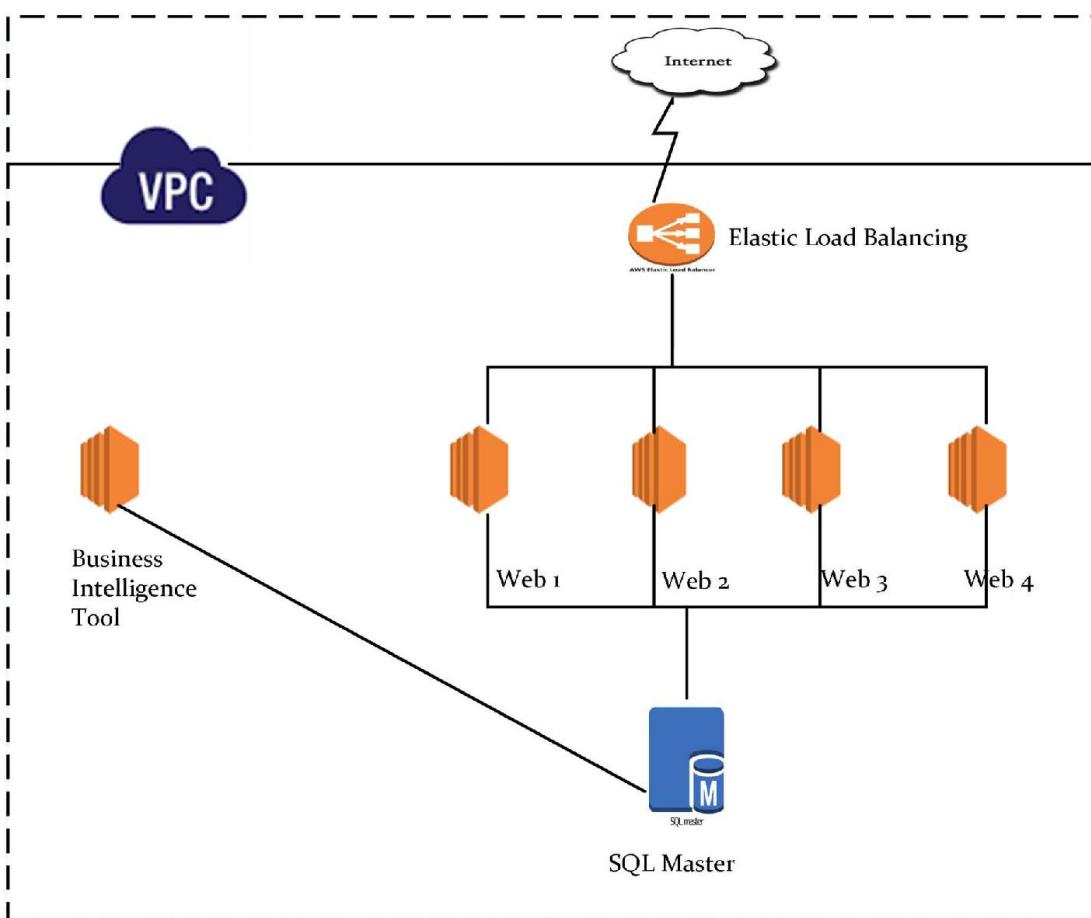


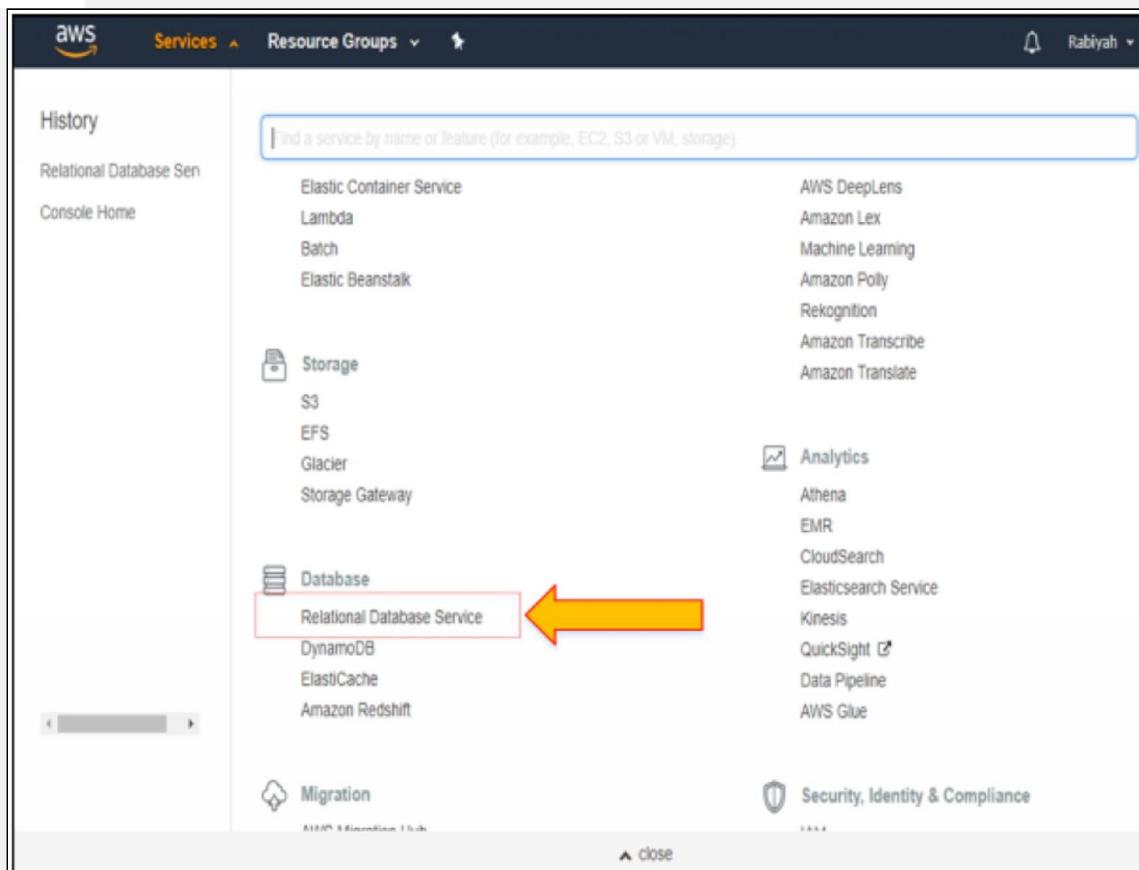
Figure 28. Amazon RDS using read Replicas

Lab 3.1 RDS Multi-AZ Deployment Read Replicas

This lab comes in the exam with multiple scenarios. You first have to clear the concept of the difference between Multi-AZ and Read Replicas. Then you will be able to do the lab of the

This lab includes the following steps.

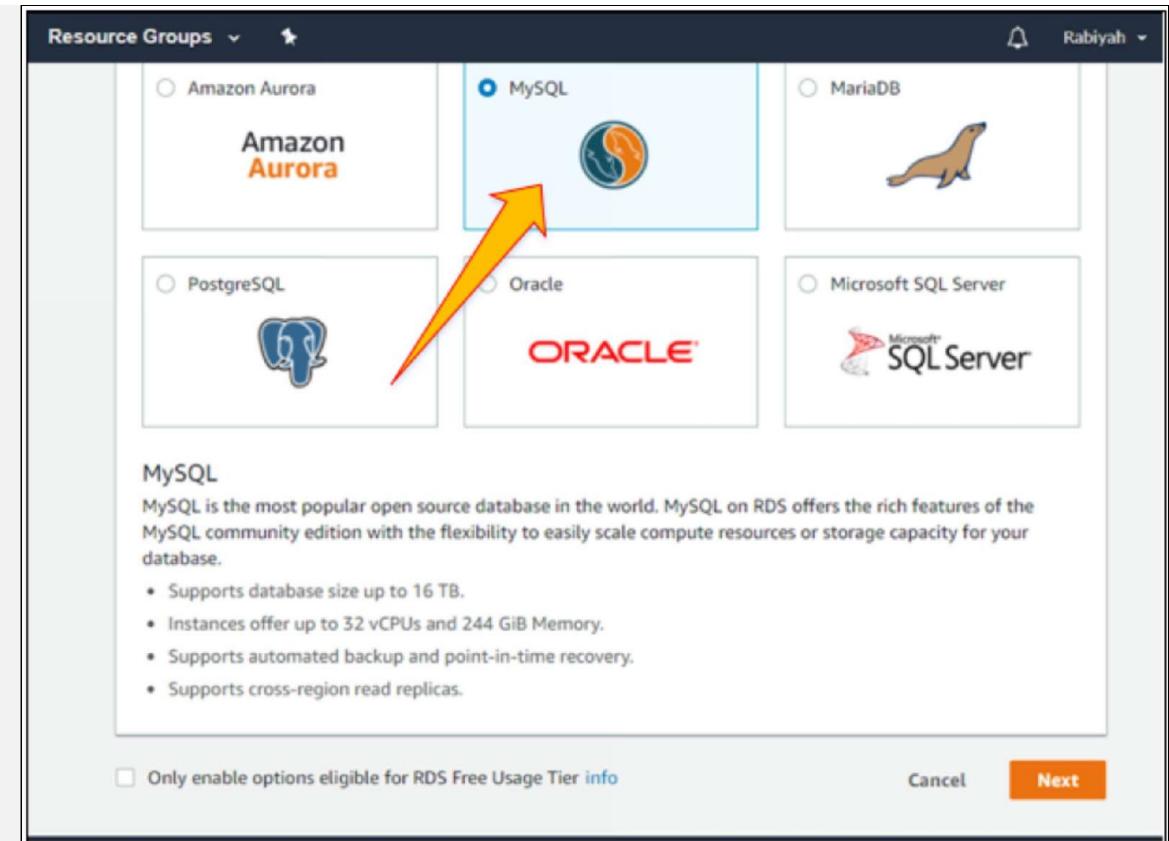
1. Login to your AWS Management Console and click on “Services. Then go to the “database” and click on “RDS.”



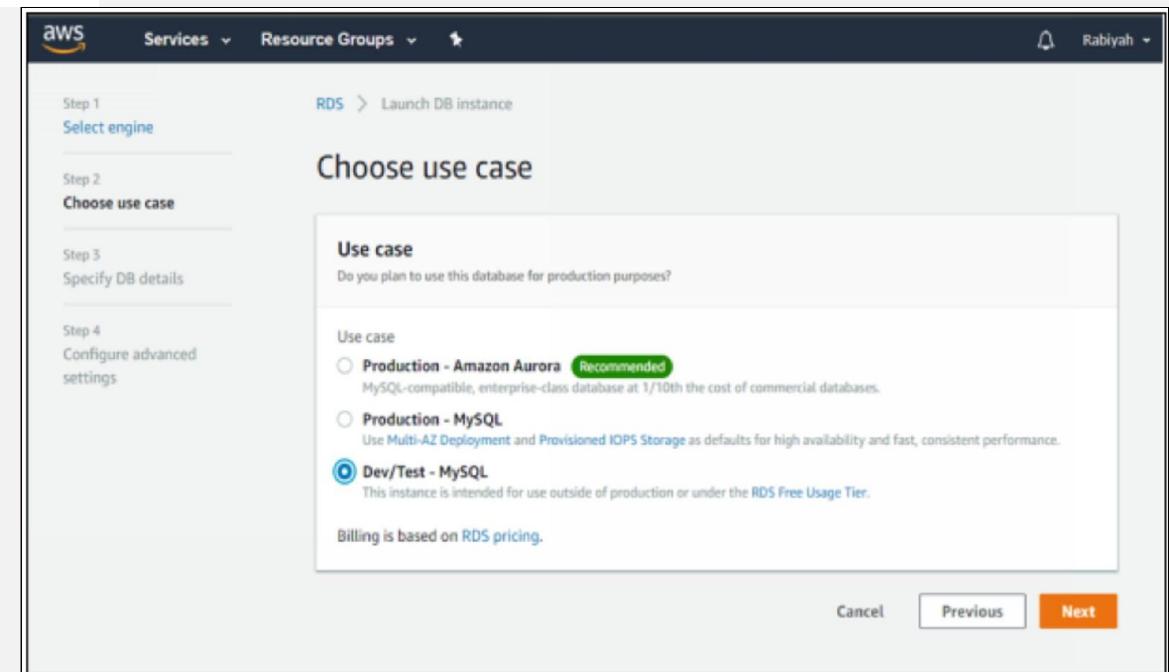
If you are creating an instance for the very first time, click on “Launch a DB instance.” If you have already existing instances, then first delete them and then take a final snapshot of them and then launch a DB instance.

The screenshot shows the AWS RDS Dashboard. On the left, there's a sidebar with various navigation options like Dashboard, Instances, Clusters, etc. The main area has sections for Reserved instances, Option groups, Snapshots, Subnet groups, Parameter groups, and so on. Below that is a 'Create instance' section with a note about launching in the US East (N. Virginia) region. At the bottom is a 'Service health' section showing the service is operating normally. A prominent yellow arrow points to the 'Launch a DB instance' button.

Amazon will show different engines to launch an instance. To select any one of them, click on “MySQL” and then click on “Next.”



1. Choose the “Use case” you want for this database and click on Next.



2. Apply DB instance identifier info, Master username info, and password for that username and click on Next.

AWS Services Resource Groups

Settings

DB instance identifier [info](#)
Specify a name that is unique for all DB instances owned by your AWS account in the current region.

AWS098IPS

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance".
Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server).
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Master username [info](#)
Specify an alphanumeric string that defines the login ID for the master user.

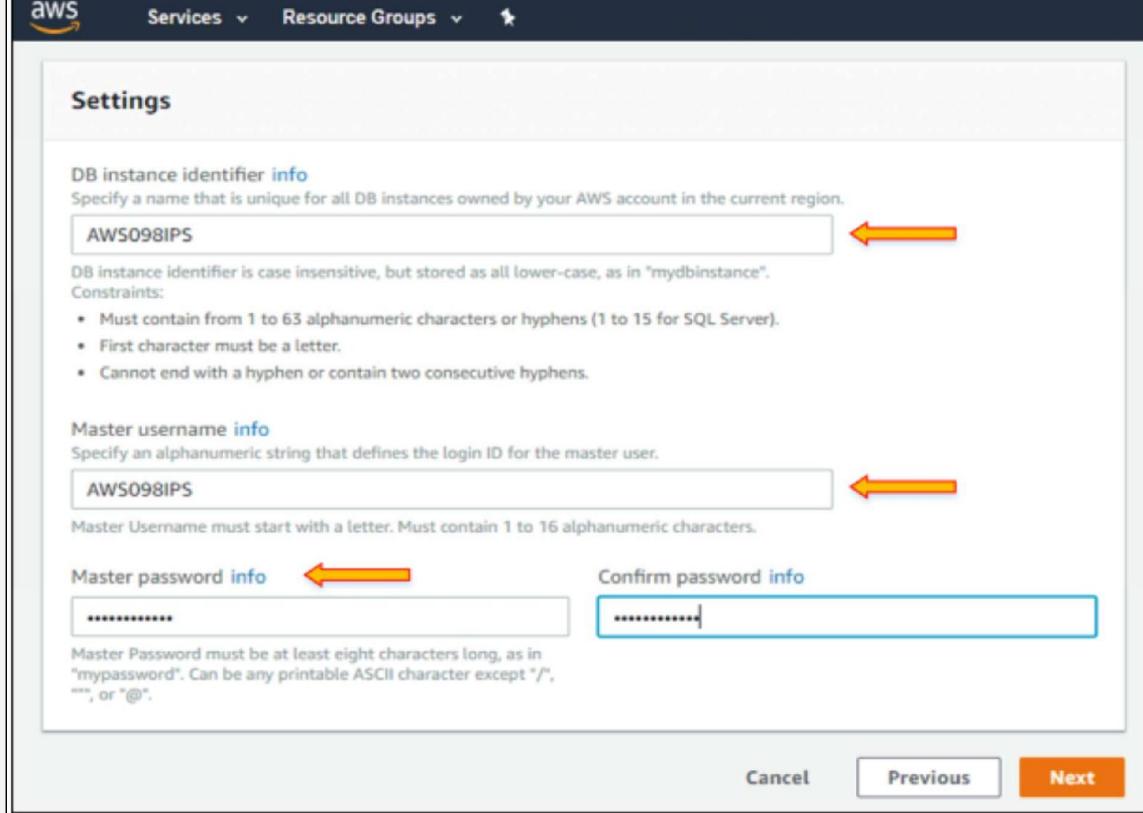
AWS098IPS

Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

Master password [info](#) Confirm password [info](#)

Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", "\n", or "@".

Cancel Previous Next



3. Now specify the DB details as shown the figure.

RDS - AWS Console

Secure | https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:ct=dashboard:

AWS Services Resource Groups

RDS > Launch DB instance

Configure advanced settings

Step 4 of 4

Amazon RDS

- Dashboard
- Instances
- Clusters
- Performance Insights [info](#)
- Snapshots
- Reserved instances
- Subnet groups
- Parameter groups
- Option groups
- Events
- Event subscriptions

Network & Security

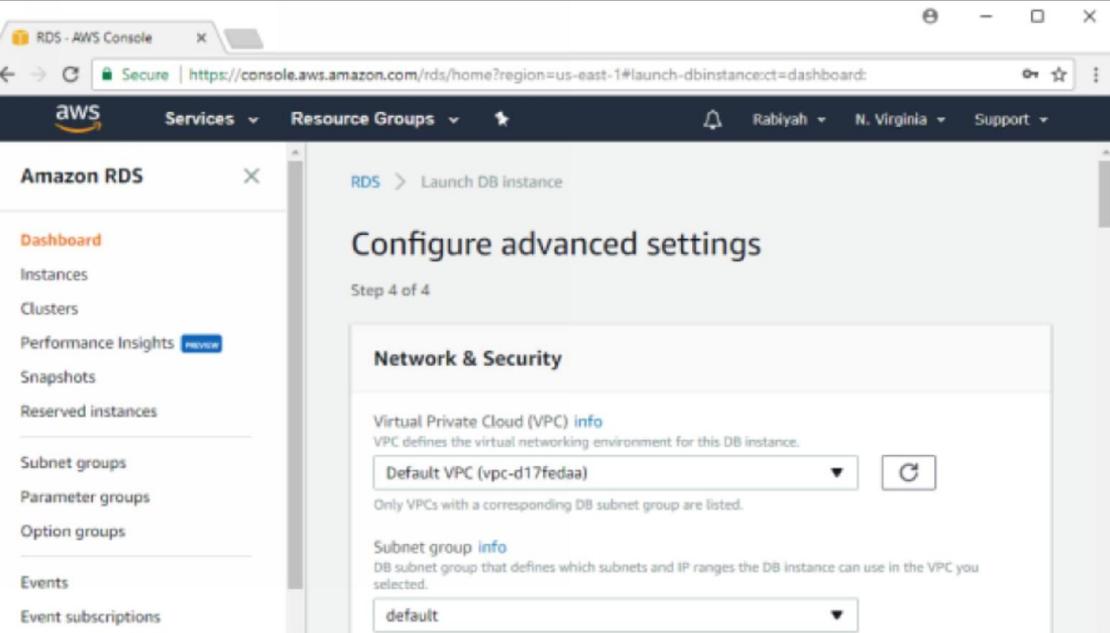
Virtual Private Cloud (VPC) [info](#)
VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-d17fedaa)

Only VPCs with a corresponding DB subnet group are listed.

Subnet group [info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default



4. Click on the options as shown in the picture.

RDS - AWS Console

Secure | https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:ct=dashboard:

aws Services Resource Groups default

Amazon RDS

Dashboard Instances Clusters Performance Insights review Snapshots Reserved instances Subnet groups Parameter groups Option groups Events Event subscriptions

Public accessibility info

Yes EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Availability zone info

No preference

VPC security groups Security groups have rules authorizing connections from all the EC2 instances and devices that need to access the DB instance.

Create new VPC security group

Choose existing VPC security groups

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

RDS - AWS Console

Secure | https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:ct=dashboard:

aws Services Resource Groups Database options

Amazon RDS

Dashboard Instances Clusters Performance Insights review Snapshots Reserved instances Subnet groups Parameter groups Option groups Events Event subscriptions

Database name

AWS098IPS

Note: if no database name is specified then no initial MySQL database will be created on the DB instance.

Database port

TCP/IP port the DB instance will use for application connections.

3306

DB parameter group info

default.mysql5.6

Option group info

default:mysql-5-6

Copy tags to snapshots

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

RDS - AWS Console

Secure | https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:ct=dashboard:

aws Services Resource Groups

Amazon RDS

Dashboard Instances Clusters Performance Insights Snapshots Reserved instances Subnet groups Parameter groups Option groups Events Event subscriptions

Encryption

Encryption

Enable Encryption Select to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console. [Learn More](#).

Disable Encryption

ⓘ The selected engine or DB instance class does not support storage encryption.

Backup

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to detail [here](#).

Feedback English (US)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the 'Encryption' section of the AWS RDS console. It features two radio buttons: 'Enable Encryption' and 'Disable Encryption'. A note below the buttons states: 'The selected engine or DB instance class does not support storage encryption.' The left sidebar lists various RDS management options like Dashboard, Instances, Clusters, and Performance Insights.

RDS - AWS Console

Secure | https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:ct=dashboard:

aws Services Resource Groups

Amazon RDS

Dashboard Instances Clusters Performance Insights Snapshots Reserved instances Subnet groups Parameter groups Option groups Events Event subscriptions

Backup

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to detail [here](#).

Backup retention period [info](#)
Select the number of days that Amazon RDS should retain automatic backups of this DB instance.
0 days

ⓘ A backup retention period of zero days will disable automated backups for this DB instance.

Backup window [info](#)
 Select window

Feedback English (US)

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

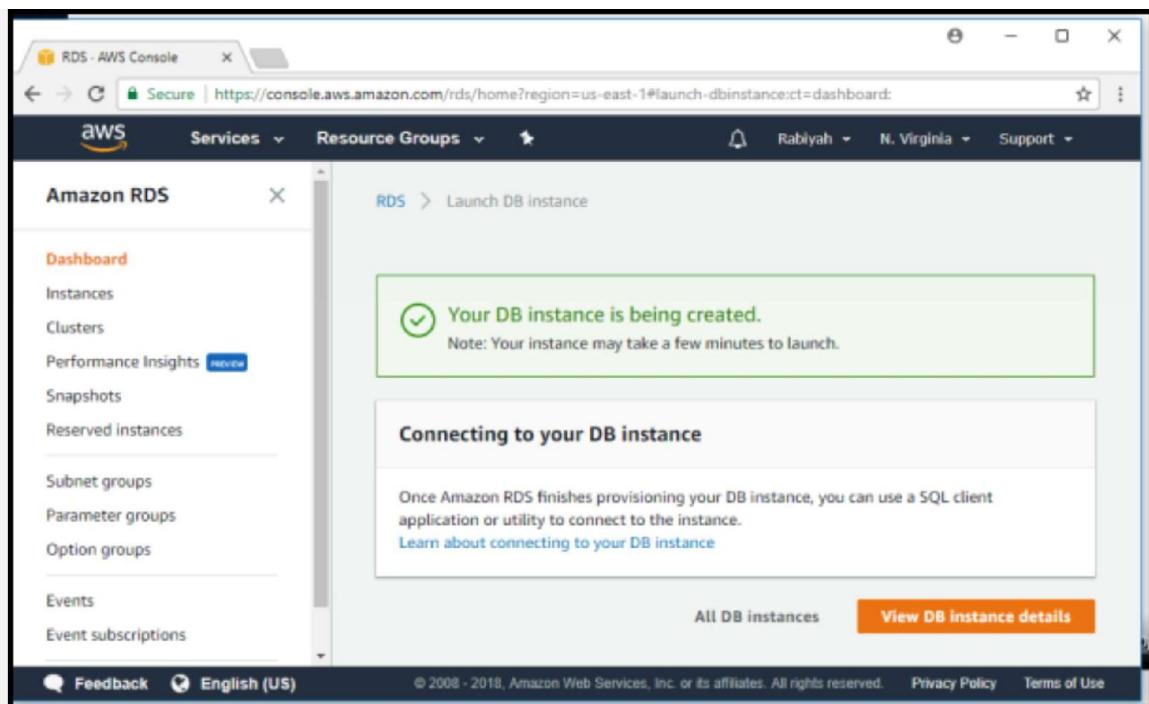
The screenshot shows the 'Backup' section of the AWS RDS console. It includes a note about backup support for InnoDB only, a dropdown for backup retention period (set to 0 days), and a note about the backup window. The left sidebar lists various RDS management options like Dashboard, Instances, Clusters, and Performance Insights.

The screenshot shows the AWS RDS Dashboard. On the left, there's a sidebar with options like Dashboard, Instances, Clusters, Performance Insights, Snapshots, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, and Event subscriptions. The main panel is titled 'Monitoring' and contains two sections: 'Enhanced monitoring' (with 'Disable enhanced monitoring' selected) and 'Log exports' (with 'Error log' checked). At the bottom right of the main panel are 'Feedback', 'English (US)', and links to 'Privacy Policy' and 'Terms of Use'.

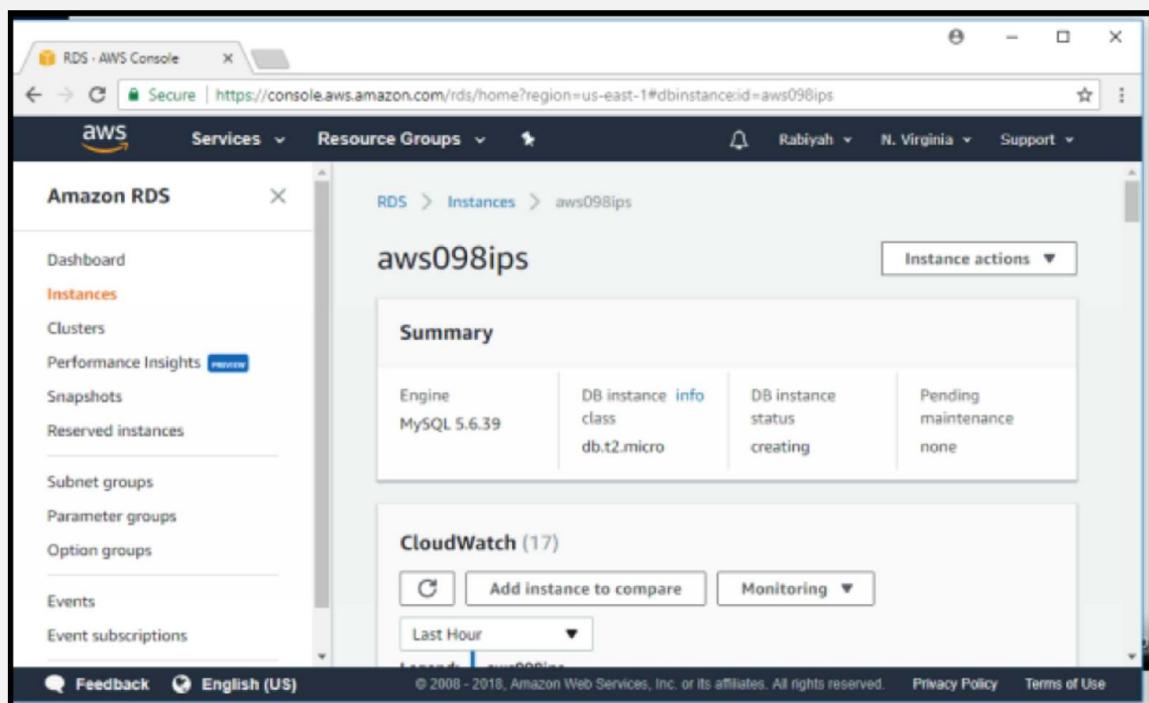
5. Now click on “Launch DB instance.”

The screenshot shows the AWS RDS Dashboard. The sidebar is identical to the previous one. The main panel is titled 'Maintenance' and contains two sections: 'Auto minor version upgrade' (with 'Disable auto minor version upgrade' selected) and 'Maintenance window' (with 'No preference' selected). At the bottom right of the main panel are 'Cancel', 'Previous', and a large orange 'Launch DB instance' button.

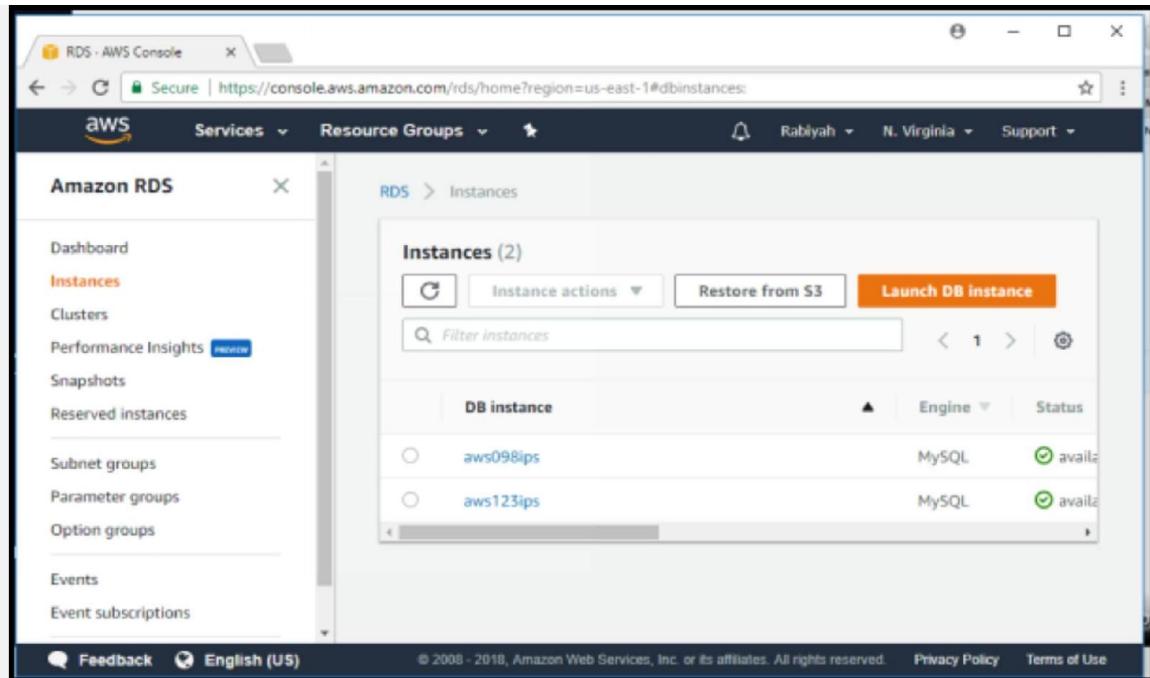
6. After launching the instance click on “View DB instance details.”



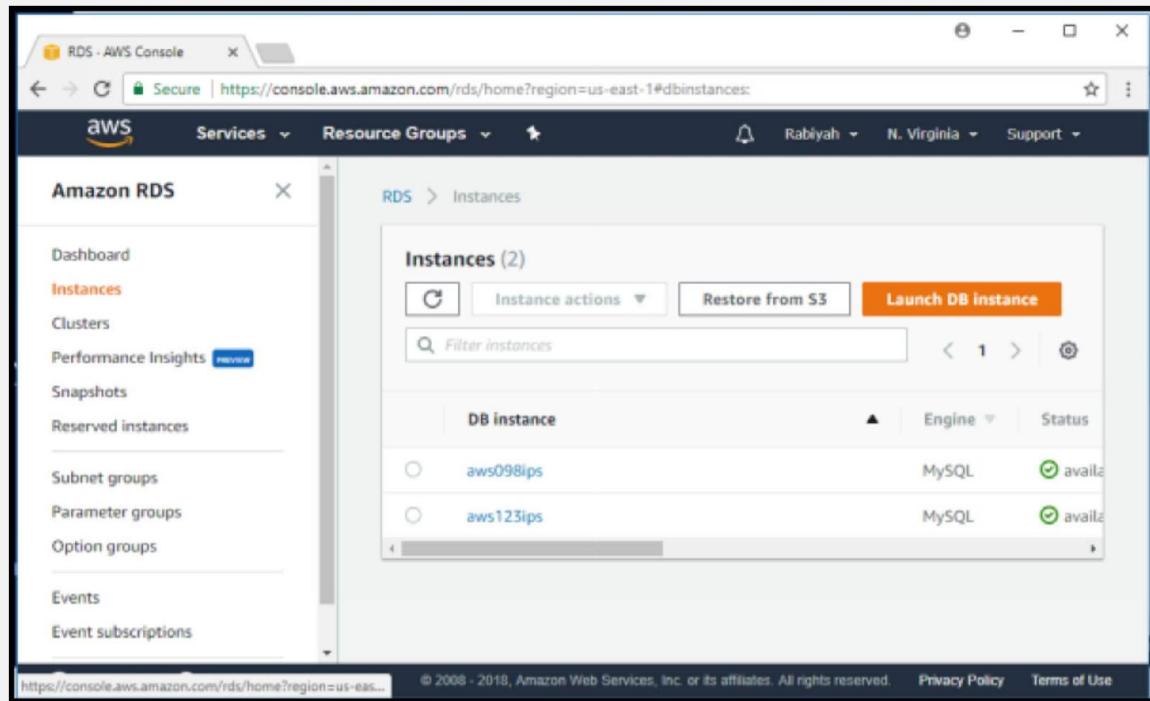
These are the details of your instance that you have launched.



In this step, the status of pending the instance is now changed into available.

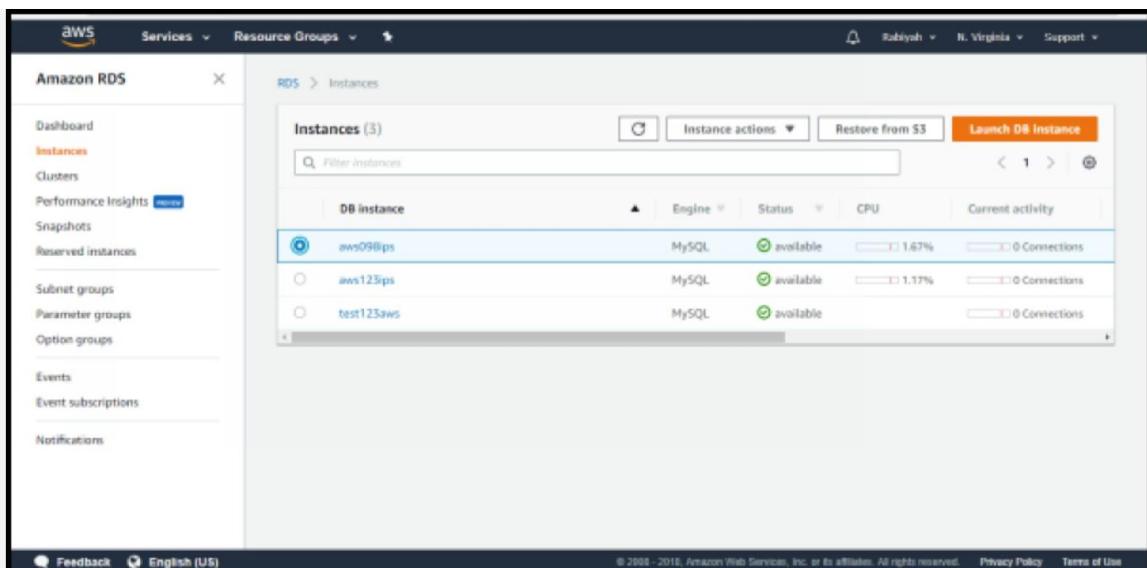


The screenshot shows the AWS RDS Instances page. On the left, there's a navigation sidebar with options like Dashboard, Instances (which is selected), Clusters, Performance Insights, Snapshots, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, and Event subscriptions. The main content area is titled 'Instances (2)'. It features a search bar with 'Filter instances' and a button 'Launch DB instance'. Below is a table with columns 'DB instance', 'Engine', and 'Status'. Two entries are listed: 'aws098ips' and 'aws123ips', both running MySQL and marked as 'available'.

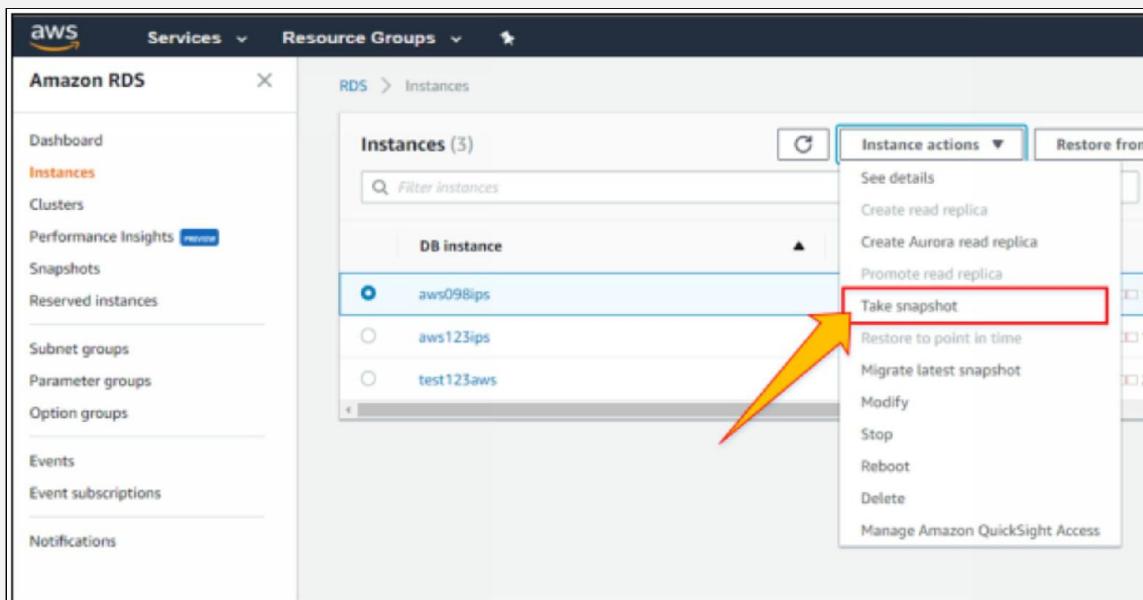


This screenshot is identical to the one above, showing the AWS RDS Instances page with two MySQL DB instances listed: 'aws098ips' and 'aws123ips', both in the 'available' status.

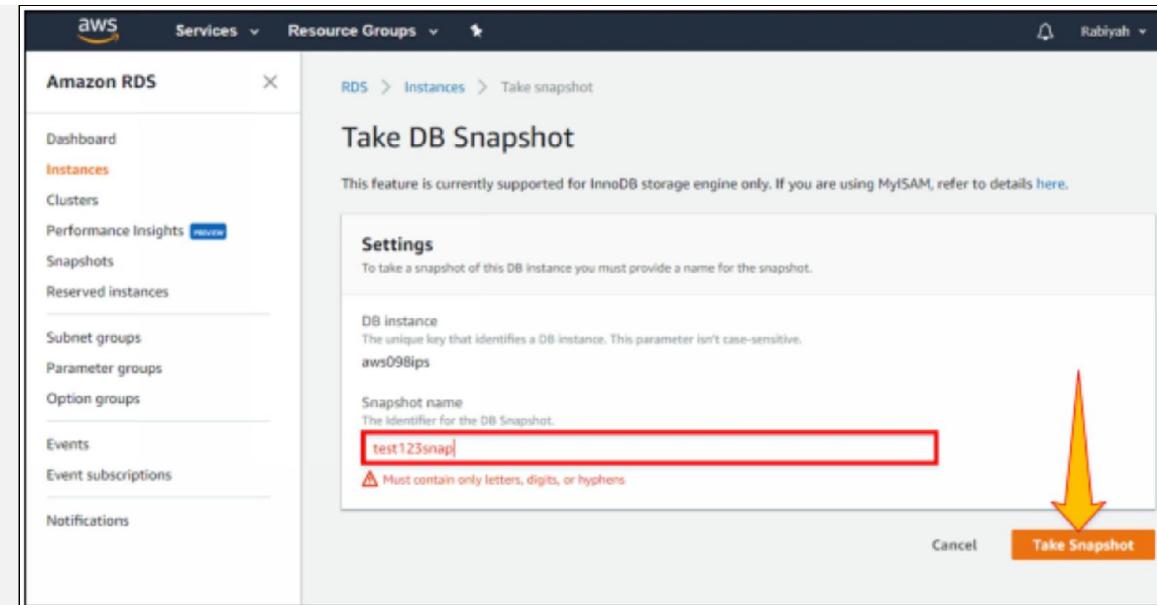
7. Now click on the instance that you have launched, and click on “Instance actions.”



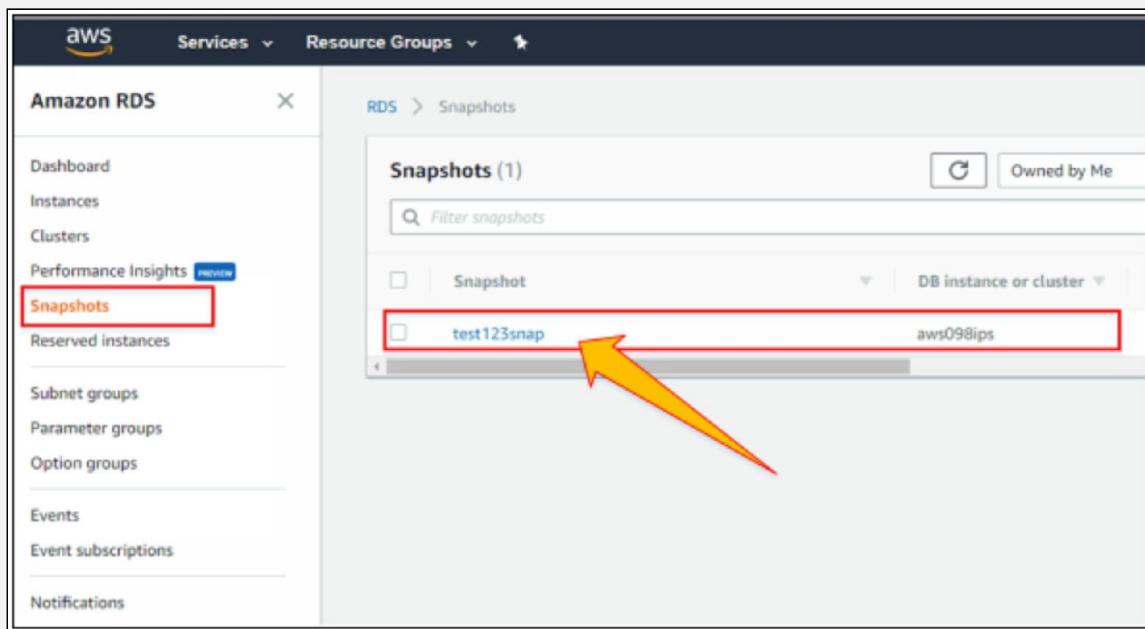
8. To make replicas of the instance, you have to take the snapshots of the instance. For this purpose, click on “Take Snapshot” after selecting an instance.



9. To take a snapshot of the instance, you have to give the name of the snapshot. Moreover, then click on “take Snapshot” box.



This is the required snapshot of your instance.



10. To know the details of that snapshot, click on it.

Screenshot of the AWS RDS console showing the details of a database snapshot named "test123snap".

The left sidebar shows the following navigation:

- Dashboard
- Instances
- Clusters
- Performance Insights
- Snapshots** (highlighted)
- Reserved instances

The main content area displays the following details for the snapshot:

Details	
ARN	arn:aws:rds:us-east-1:921958654689:snapshot:test123snap
Instance/Cluster Name	aws098ips
Snapshot type	manual
DB engine version	5.6.39
Master username	AWS098IPS
Zone	us-east-1f
DB snapshot name	test123snap
VPC	vpc-d17fedaa
DB engine	mysql
License model	general-public-license
Status	available
DB storage	20GB

11. Now, delete the snapshot as it is required only for making a replica of the instance.

Screenshot of the AWS RDS console showing the "Delete DB snapshots" confirmation dialog.

The left sidebar shows the following navigation:

- Dashboard
- Instances
- Clusters
- Performance Insights
- Snapshots** (highlighted)
- Reserved instances

The main content area displays the following confirmation message:

Are you sure you want to Delete these DB snapshots?
• test123snap

Buttons available:

- Cancel
- Delete (highlighted with a red arrow)

Page footer:

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

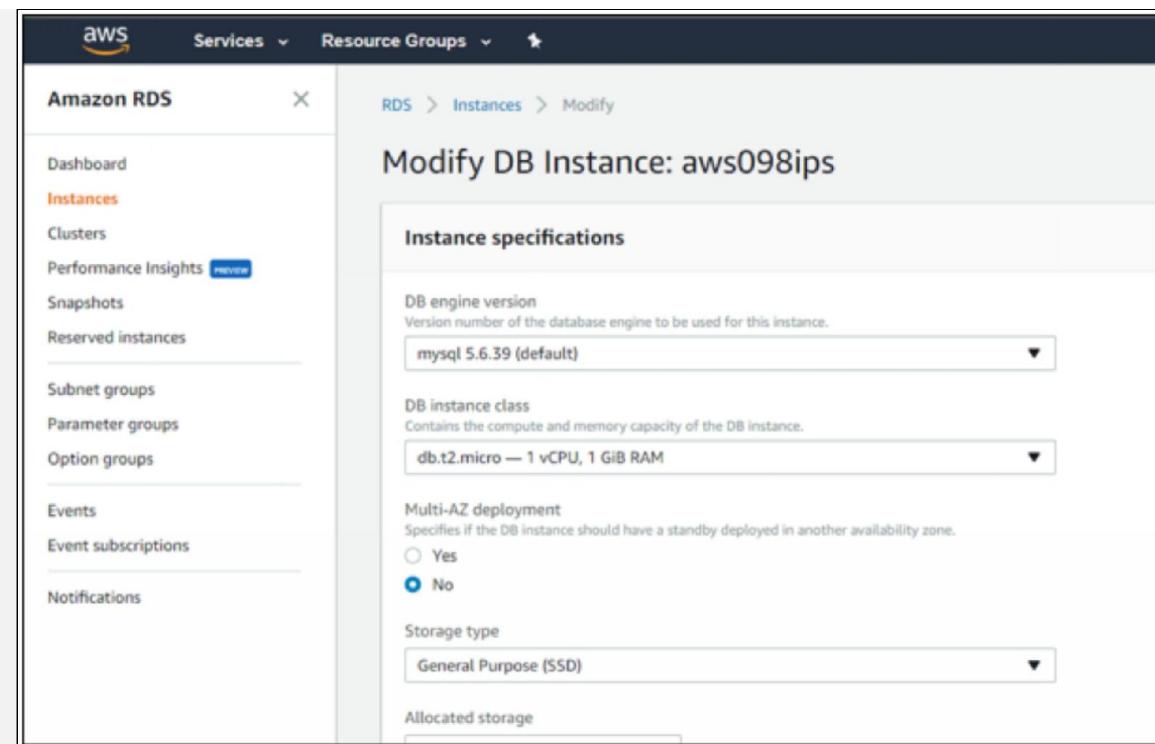
12. To modify the instance again select the instance and click on the “instance actions.”

The screenshot shows the AWS RDS Instances page. On the left, there's a sidebar with options like Dashboard, Instances (which is selected), Clusters, Performance Insights, Snapshots, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Notifications. The main area displays 'Instances (3)'. A search bar at the top says 'Filter instances'. Below it is a table with columns: DB instance, Engine, Status, CPU, and Current activity. The first row, 'aws098ips', is highlighted with a blue circle icon. The other two rows show 'aws123ips' and 'test123aws'. Each row has a small circular icon next to it.

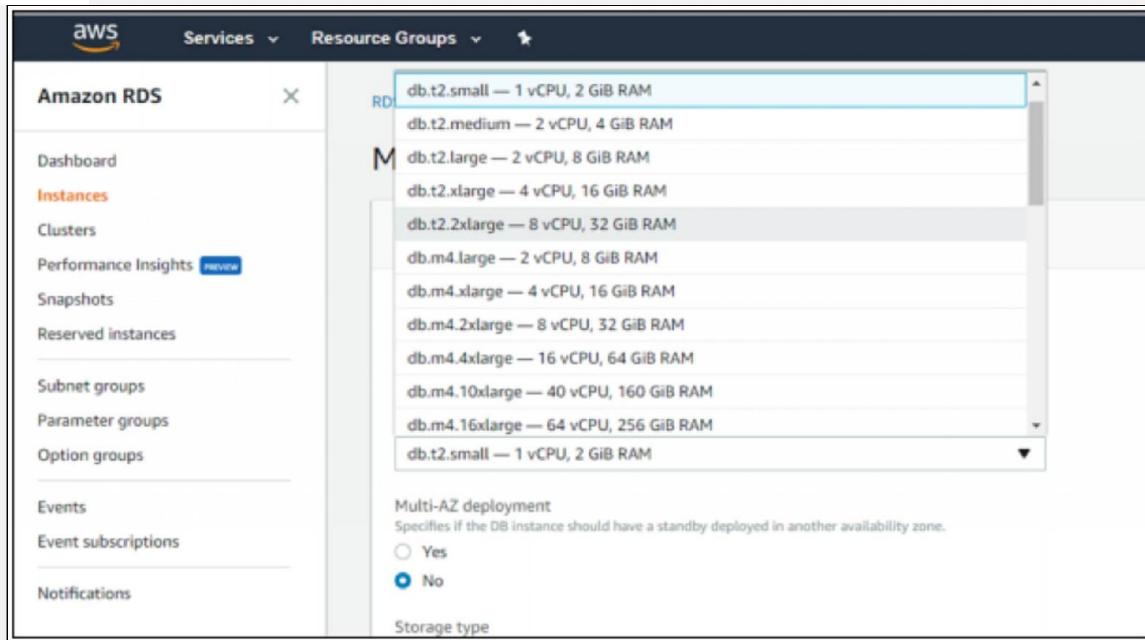
13. Click on “modify.”

This screenshot is similar to the previous one, showing the AWS RDS Instances page. The 'Instances' section is selected in the sidebar. In the main area, the 'aws098ips' instance is selected, indicated by a blue circle icon. A context menu is open over the 'aws098ips' row, with the 'Instance actions' button highlighted with a red box. The menu itself is also highlighted with a red box and contains the following options: See details, Create read replica, Create Aurora read replica, Promote read replica, Take snapshot, Restore to point in time, Migrate latest snapshot, Modify (which is highlighted with a red box), Stop, Reboot, Delete, and Manage Amazon QuickSight Access.

You will get the following window.



14. Modify the settings as shown in below diagrams.



AWS Services Resource Groups

Amazon RDS

Instances

Dashboard Clusters Performance Insights Preview Snapshots Reserved instances Subnet groups Parameter groups Option groups Events Event subscriptions Notifications

Instance specifications

DB engine version Version number of the database engine to be used for this instance.
mysql 5.6.39 (default)

DB instance class Contains the compute and memory capacity of the DB instance.
db.t2.small — 1 vCPU, 2 GiB RAM

Multi-AZ deployment Specifies if the DB instance should have a standby deployed in another availability zone.
 Yes
 No

Storage type General Purpose (SSD)

Allocated storage 20 GB

This instance supports multiple storage ranges between 20 and 16384 GB. See all

AWS Services Resource Groups Rubiyah

Amazon RDS

Instances

Dashboard Clusters Performance Insights Preview Snapshots Reserved instances Subnet groups Parameter groups Option groups Events Event subscriptions Notifications

In higher latencies upon exhaustion of the initial General Purpose (SSD) IO Credit balance. [Learn more](#) for more details.

Settings

DB instance identifier DB instance identifier. This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (e.g. mydbinstance).
aws09Bips

New master password Set a new password for the master DB instance user.
.....

Network & Security

Subnet group Use this field to move the DB Instance to a new subnet group in another VPC. [Learn more](#).
default

AWS Services Resource Groups Rabiyah

Amazon RDS X

Instances

Clusters

Performance Insights Review

Snapshots

Reserved instances

Subnet groups

Parameter groups

Option groups

Events

Event subscriptions

Notifications

Subnet group
Use this field to move the DB instance to a new subnet group in another VPC. Learn more.

default

Security group
List of DB security groups to associate with this DB instance.

Choose security groups

default (sg-ac9aa9da) (vpc-d17fedaa) X

rds-launch-wizard-1 (sg-e920129f) (vpc-d17fedaa) X

Certificate authority
Certificate authority for this DB instance

rds-ca-2015

Public accessibility info

Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Backup

Backup retention period

The number of days for which automated backups are retained. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

0 days

0 days

1 day

2 days

3 days

4 days

5 days

6 days

7 days

8 days

9 days

This will disable automated backups and **delete all** existing automated snapshots. A service outage will occur if you change the backup retention period from 0 to a zero value to 0.

Automated backups are created if automated backups are enabled.

Duration

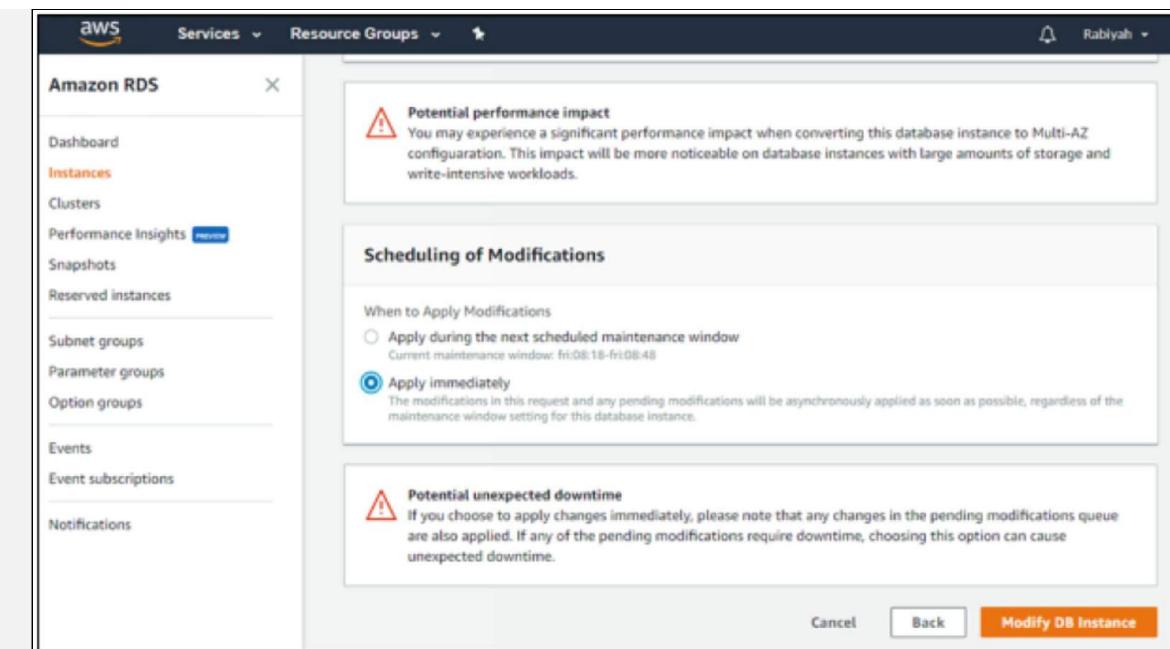
0.5 hours

The screenshot shows the AWS RDS service dashboard. On the left, there's a sidebar with options like Dashboard, Instances (which is selected), Clusters, Performance Insights, Snapshots, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Notifications. The main panel is titled 'Monitoring' and contains two sections: 'Enhanced monitoring' and 'Log exports'. Under 'Enhanced monitoring', there are two radio buttons: 'Enable enhanced monitoring' (with a note that it's useful for CPU usage) and 'Disable enhanced monitoring' (which is selected). Under 'Log exports', there are four checkboxes: 'Audit log' (unchecked), 'Error log' (checked), 'General log' (unchecked), and 'Slow query log' (unchecked).

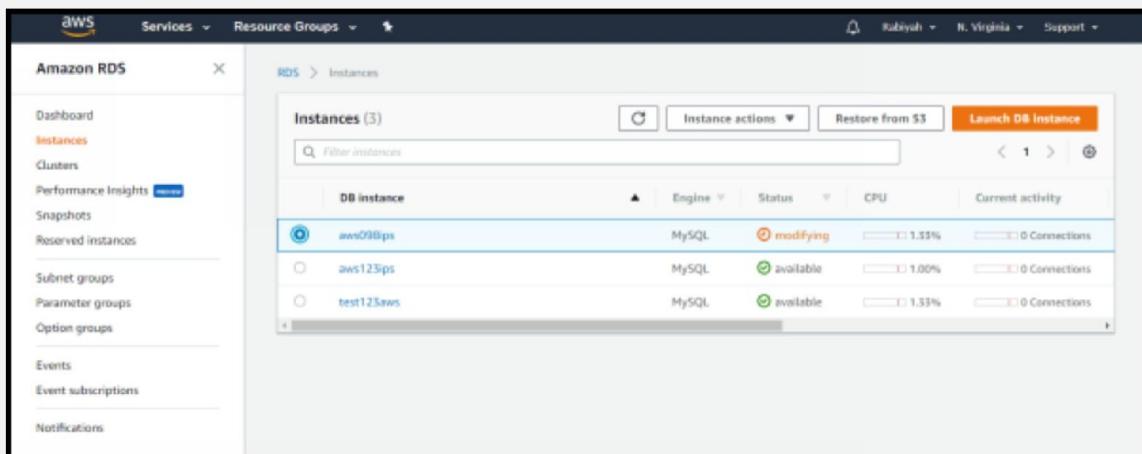
15. Click on Continue.

The screenshot shows the AWS RDS service dashboard. The sidebar includes the same set of options as the previous screenshot. The main panel is titled 'Maintenance' and contains two sections: 'IAM role' and 'Maintenance window'. Under 'IAM role', it says 'The following service-linked role is used for publishing logs to CloudWatch Logs.' followed by 'RDS Service Linked Role'. Under 'Maintenance window', it says 'The weekly time range (in UTC) during which system maintenance can occur.' with fields for 'Start Day' (set to Friday), 'Start Time' (set to 08:18 UTC), and 'Duration' (set to 0.5 hours). At the bottom right, there are 'Cancel' and 'Continue' buttons, with 'Continue' being highlighted.

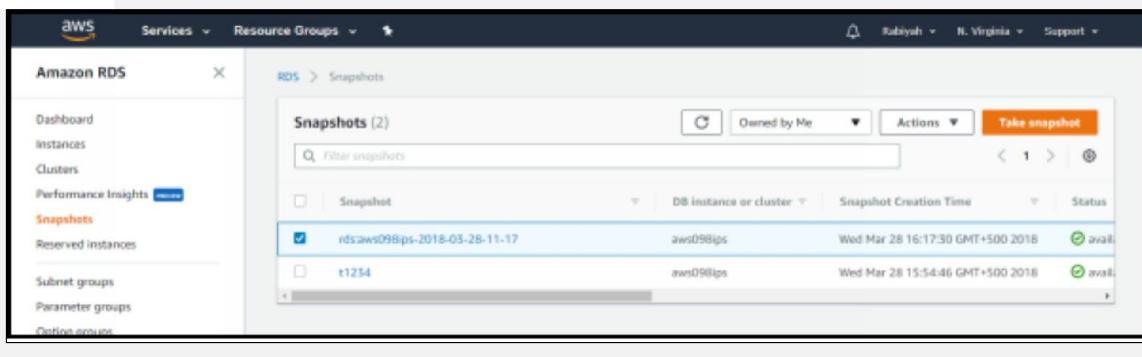
16. Now click on “modify DB instance.”



The instance modification is in the process as shown in the below diagram.



17. Now again go to snapshot option and to see what happens in the instance click on the existing snapshot.



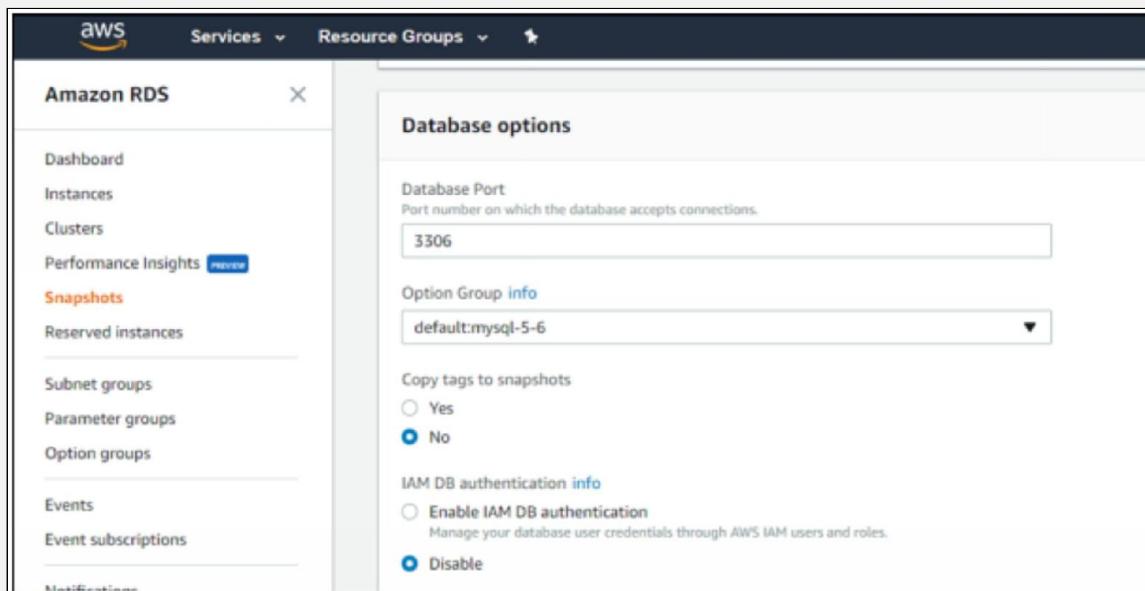
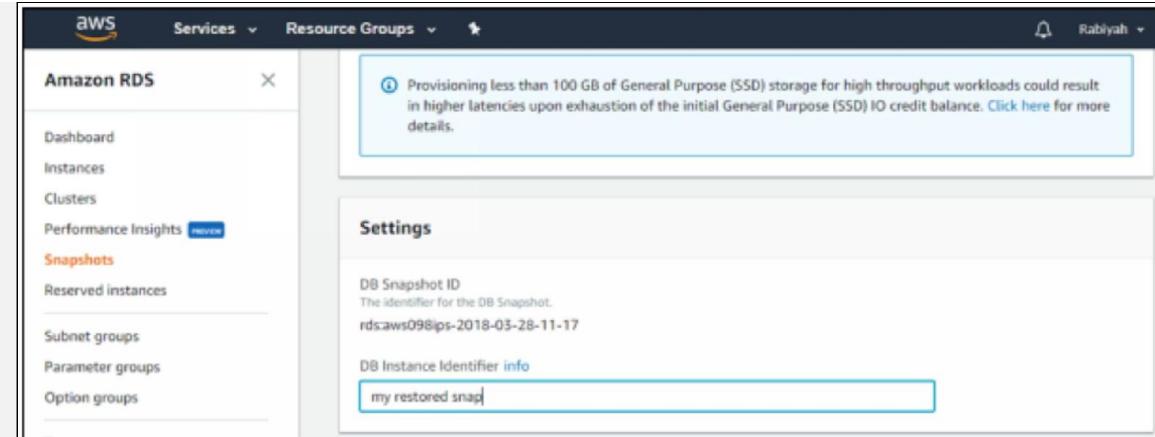
18. Click on “Actions” and then click on “Restore Snapshot.”

The screenshot shows the AWS RDS console with the 'Schemas' tab selected. In the main area, there are two snapshots listed: 'rds.aws098ips-2018-03-28-11-17' and 't1234'. To the right of the snapshots, there is an 'Actions' dropdown menu with several options: 'Restore Snapshot', 'Modify Snapshot', 'Copy Snapshot', 'Share Snapshot', 'Migrate Snapshot', and 'Delete Snapshot'. The 'Restore Snapshot' option is highlighted with a blue border.

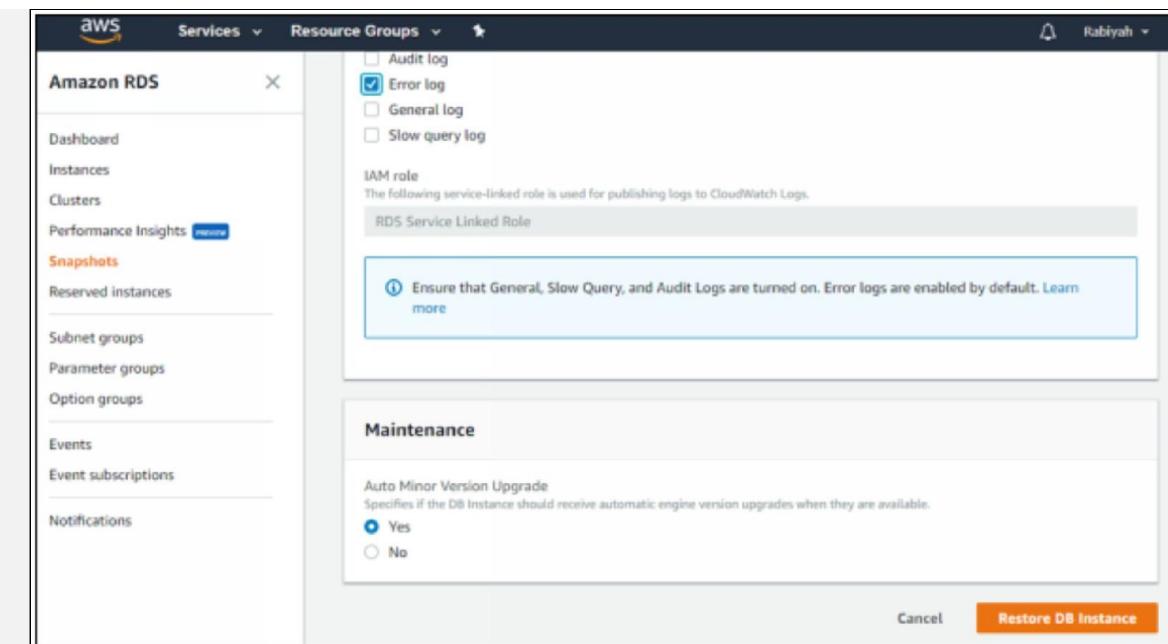
19. You will get the following window. Do the edition as shown below?

The screenshot shows the 'Create New DB Instance' wizard. On the left, the 'Schemas' tab is selected in the navigation bar. The main area is titled 'Instance specifications' and contains the following fields:

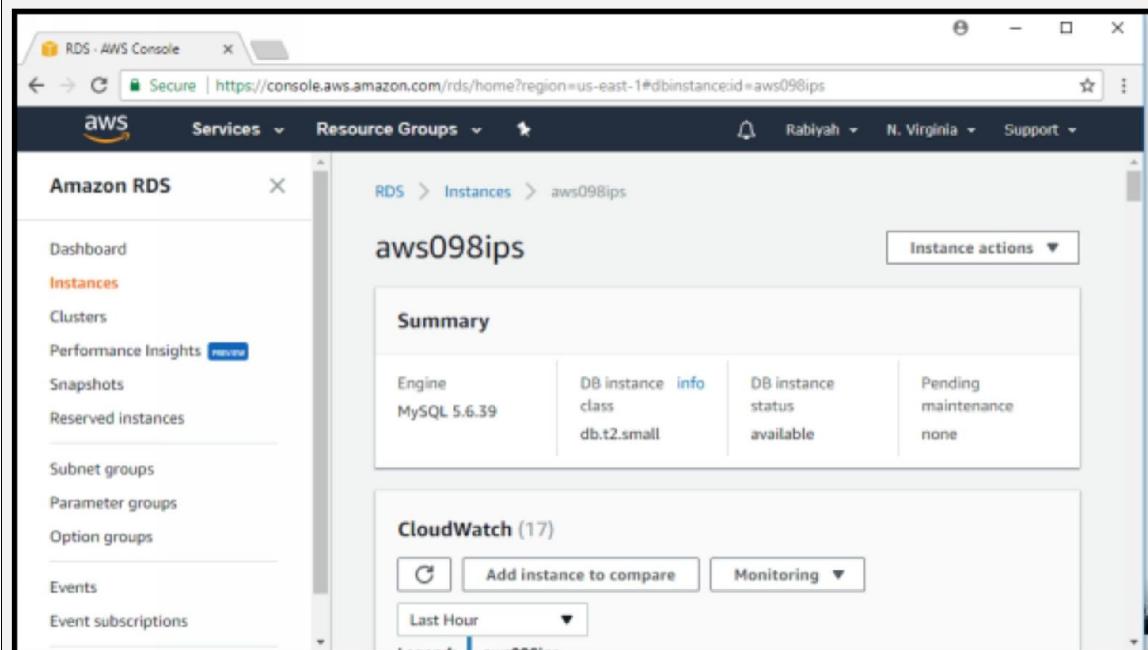
- DB Engine: mysql
- License Model: general-public-license
- DB Instance Class: db.m4.xlarge — 4 vCPU, 16 GiB RAM
- Multi-AZ Deployment: No (radio button selected)
- Storage type: General Purpose (SSD)

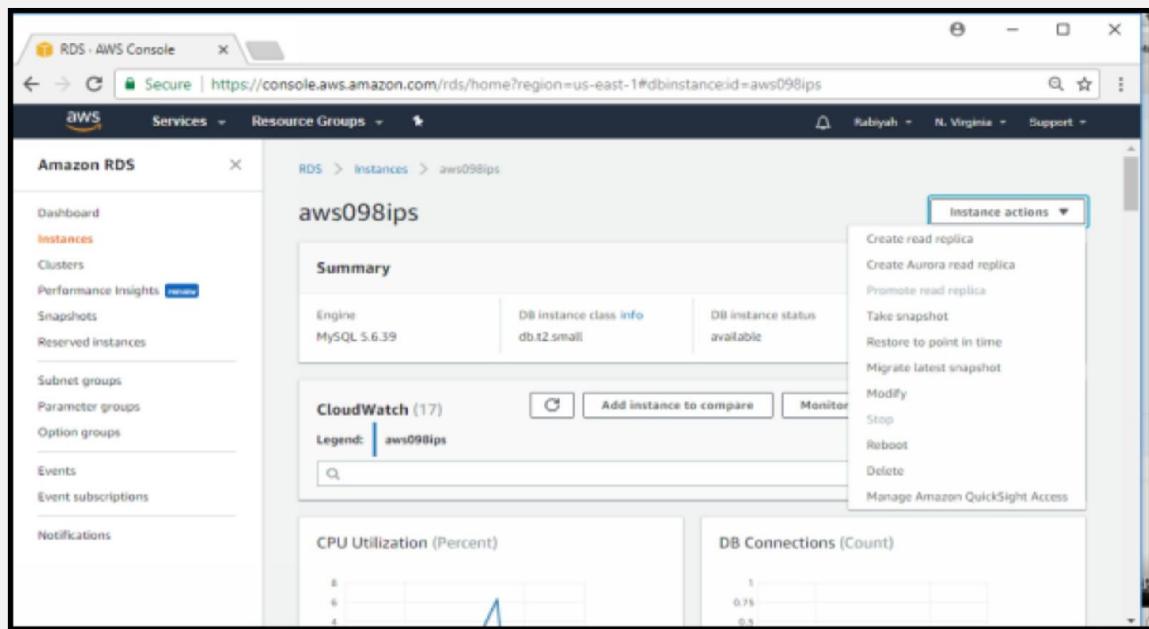
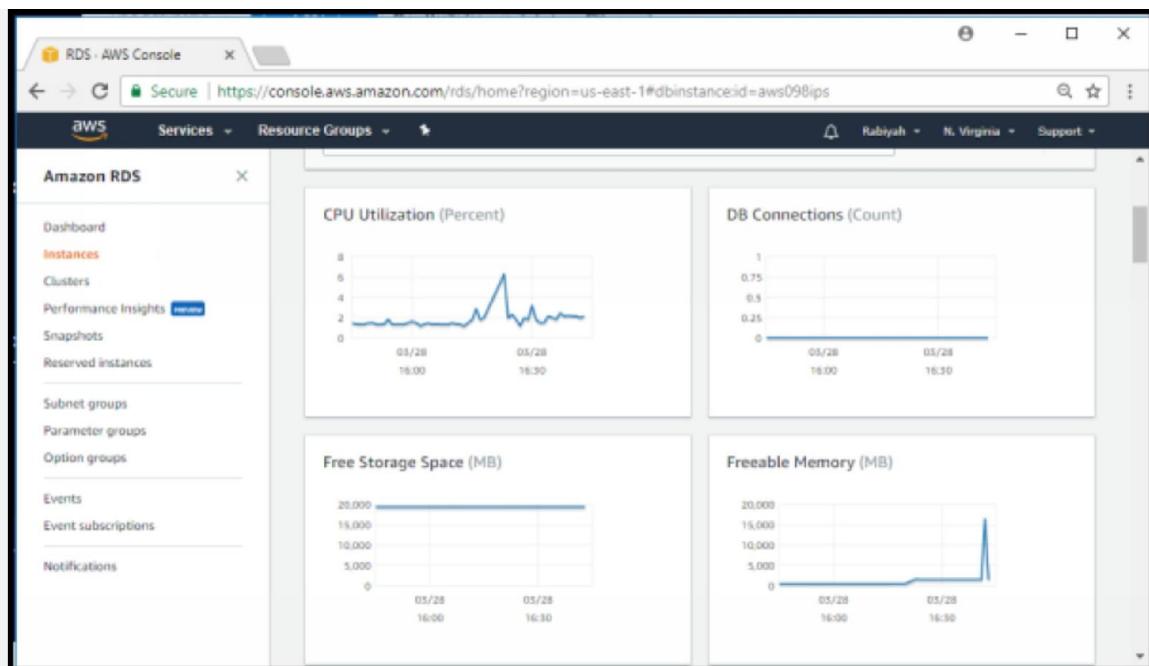


20. Click on “Restore DB instance.”



Here are the details of your restored DB instance.





21. To create read replica, select the instance, then “instance actions” and then on “Promote read replica.”

RDS - AWS Console

Secure | https://console.aws.amazon.com/rds/home?region=us-east-1#dbinstanceid=aws098ips

AWS Services Resource Groups

RDS Instances aws098ips Create read replica

Rubysh N. Virginia Support

Create read replica DB instance

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.

Network & Security

Destination region
The region in which the replica will be launched.

US East (N. Virginia)

Asia Pacific (Osaka-Local)

US East (N. Virginia)

EU (Frankfurt)

Asia Pacific (Singapore)

Asia Pacific (Tokyo)

Asia Pacific (Mumbai)

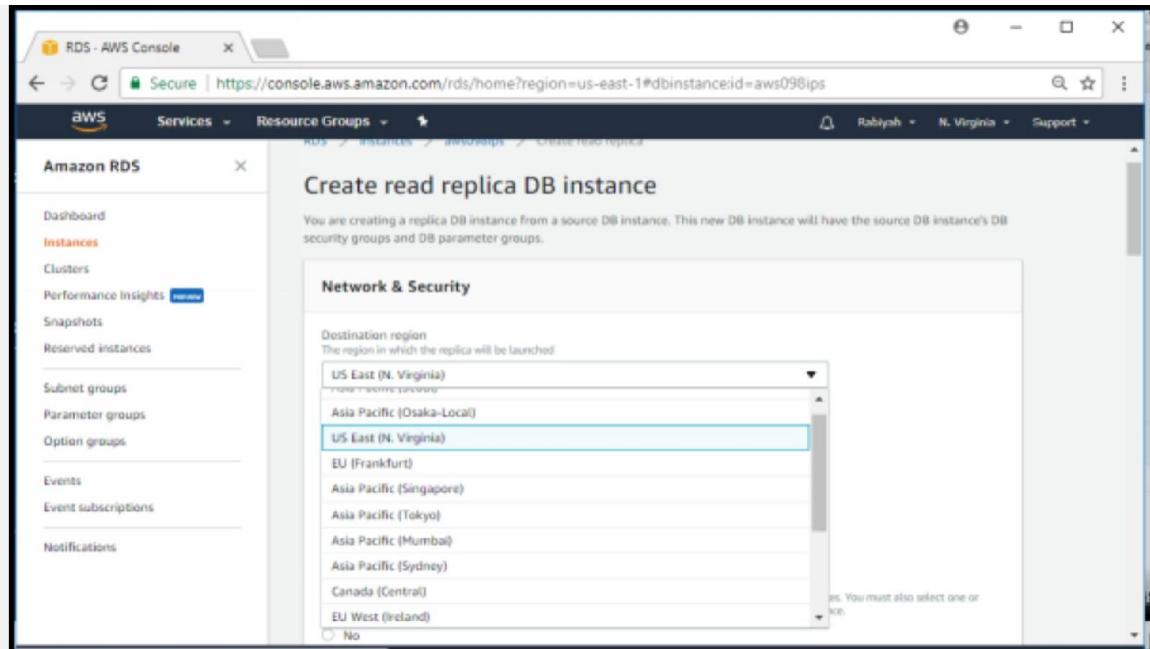
Asia Pacific (Sydney)

Canada (Central)

EU West (Ireland)

No

You must also select one or more security groups.



AWS Services Resource Groups

Amazon RDS

Instances

Dashboard Clusters Performance Insights Snapshots Reserved instances Subnet groups Parameter groups Option groups Events Event subscriptions Notifications

Settings

Read replica source
Source DB instance Identifier: aws098ips

DB instance identifier
DB instance identifier: aws instance identifier

Database options

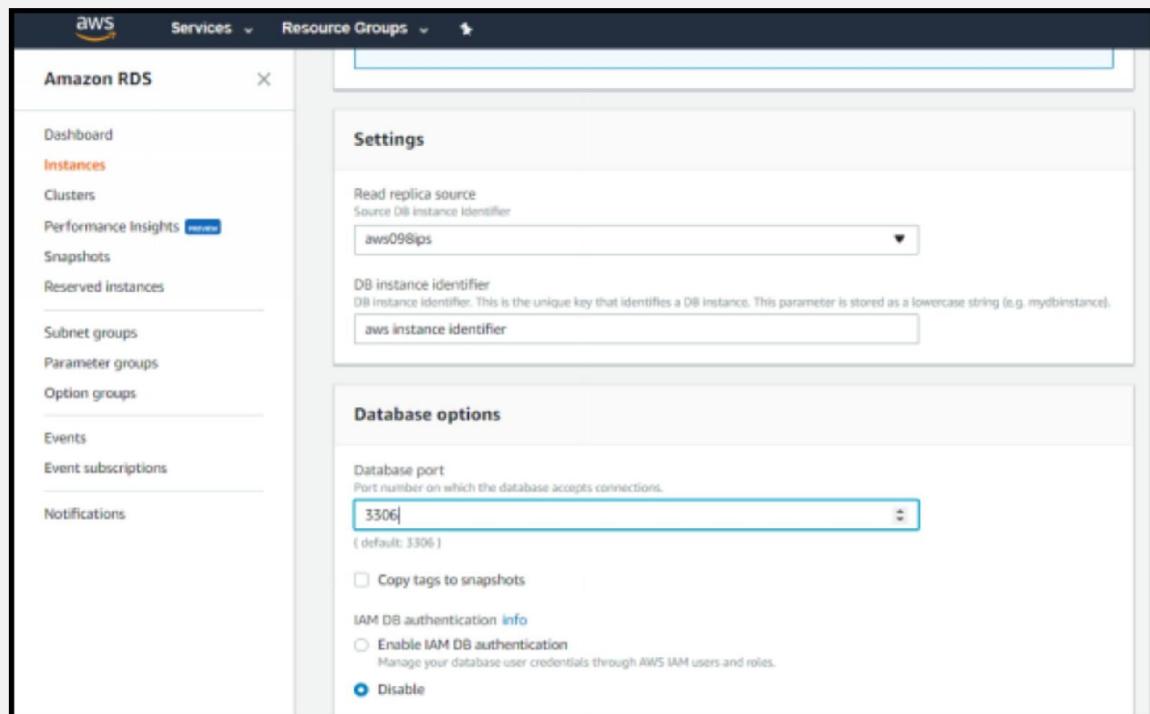
Database port
Port number on which the database accepts connections: 3306 (default: 3306)

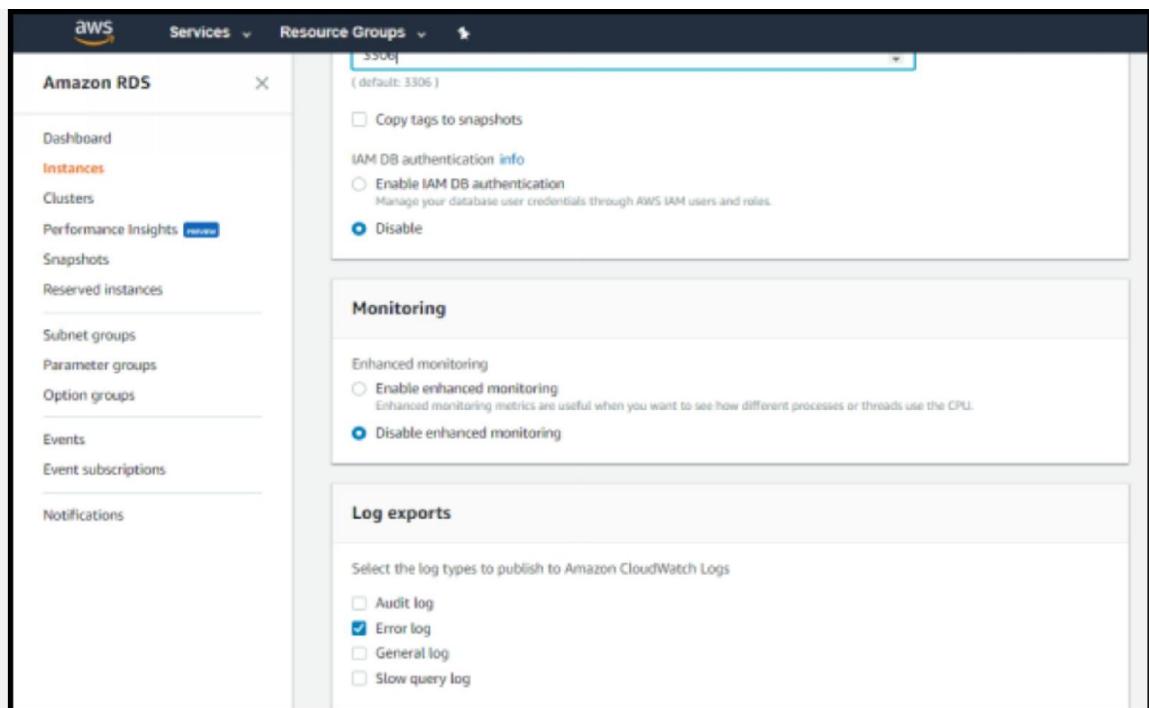
Copy tags to snapshots

IAM DB authentication [info](#)

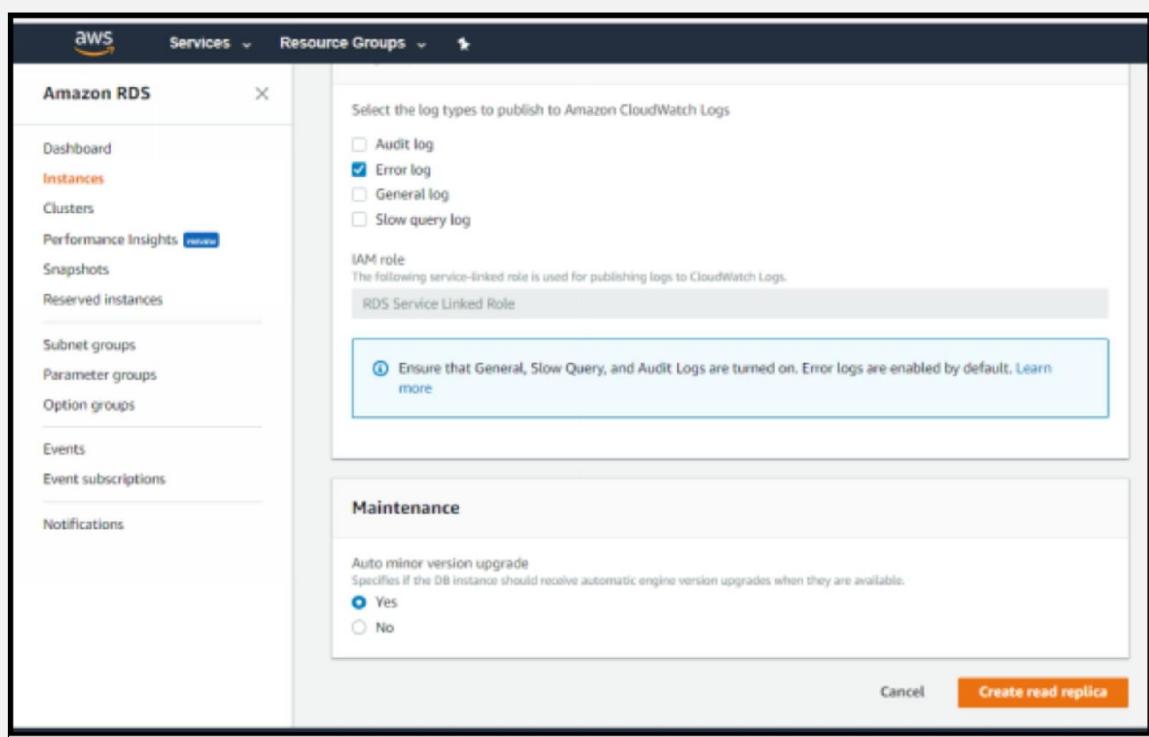
Enable IAM DB authentication Manage your database user credentials through AWS IAM users and roles.

Disable

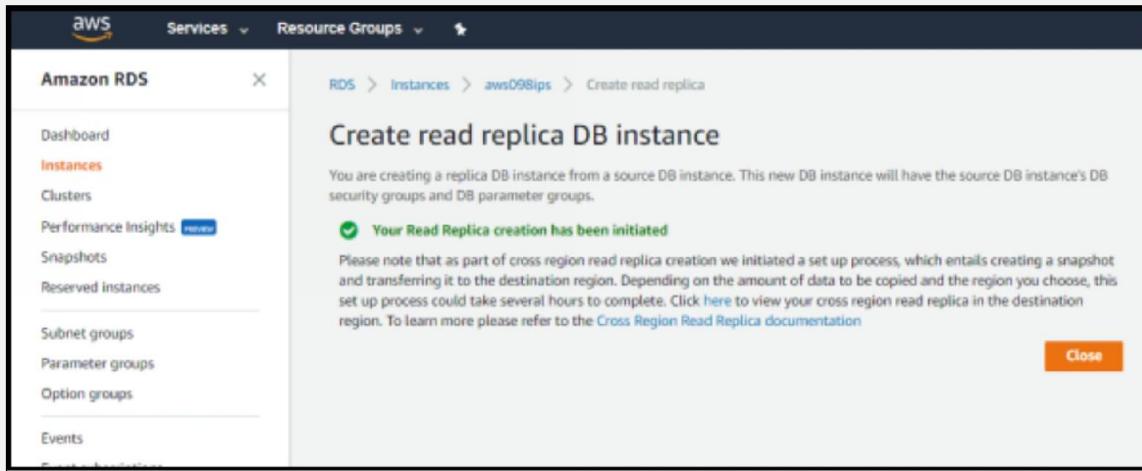




22. After doing all changing as shown in the diagrams, click on “Create Read Replica.”



It is the required Read Replica, click on “Close.”



If you want to know the details of your instances that you have created, click on an instance and then check.

Instances (4)							
	DB instance	Engine	Status	CPU	Current activity	Maintenance	
●	aws098ips	MySQL	available	1.83%	0 Connections	none	
○	aws123ips	MySQL	available	1.17%	0 Connections	none	
○	s0987	MySQL	available	0.21%	0 Connections	none	
○	test123aws	MySQL	available	1.19%	0 Connections	none	

Instances (4)							
	Status	CPU	Current activity	Maintenance	Class	VPC	Multi-AZ
●	available	1.83%	0 Connections	none	db.t2.small	vpc-d17fedaa	Yes
●	available	1.17%	0 Connections	none	db.t2.micro	vpc-d17fedaa	No
●	available	0.21%	0 Connections	none	db.m4.2xlarge	vpc-d17fedaa	No
●	available	1.19%	0 Connections	none	db.t2.micro	vpc-d17fedaa	No

Screenshot of the AWS RDS Instances page showing event logs and replication details.

Amazon RDS

Instances

Events

Replication (2)

DB instance	Role	Zone	Replication source	Lag
aws098ips	master	eu-east-1f	-	-
mydbinstance (London)	replica	eu-west-2	aws098ips	-

Event Log Details:

Time	User	Action
March 28, 2018 at 4:16:48 PM UTC+5	aws098ips	DB instance restarted
March 28, 2018 at 4:16:57 PM UTC+5	aws098ips	DB instance shutdown
March 28, 2018 at 4:16:58 PM UTC+5	aws098ips	Enabled automated backups
March 28, 2018 at 5:55:38 PM UTC+5	aws098ips	Finished DB instance backup
March 28, 2018 at 5:54:35 PM UTC+5	aws098ips	Backing up DB instance
March 28, 2018 at 5:29:06 PM UTC+5	aws098ips	Finished DB instance backup
March 28, 2018 at 5:28:00 PM UTC+5	aws098ips	Backing up DB instance
March 28, 2018 at 2:08:47 PM UTC+5	aws098ips	CloudWatch Logs Export enabled for logs [error]
March 28, 2018 at 2:08:43 PM UTC+5	aws098ips	DB instance created
March 28, 2018 at 2:08:10 PM UTC+5	aws098ips	DB instance restarted

Bastion Host and High Availability:

Overview of the lab:

This lab provides the deployment of Linux bastion for AWS Cloud infrastructures. For this purpose, we deploy a virtual private cloud (VPC) by using the Amazon VPC Quick Start reference deployment, sets up private and public subnets, and deploys Linux bastion instances into that VPC.

You also have a facility to deploy Linux bastion hosts into your existing AWS infrastructure.

What is the Bastion Host:

To provide secure access to Linux instances that are located in the private and public subnets “Bastion Host” is used. The architecture that is used for this purpose is Quick Start architecture which deploys Linux bastion into every public subnet to give the environment the readily accessible administrative access. The Quick Start sets up a multi-az climate which consists of two Availability Zones. If highly available bastion access is not necessary, you can stop the instance in the second Availability Zone and start it up when needed.

Architecture Diagram :

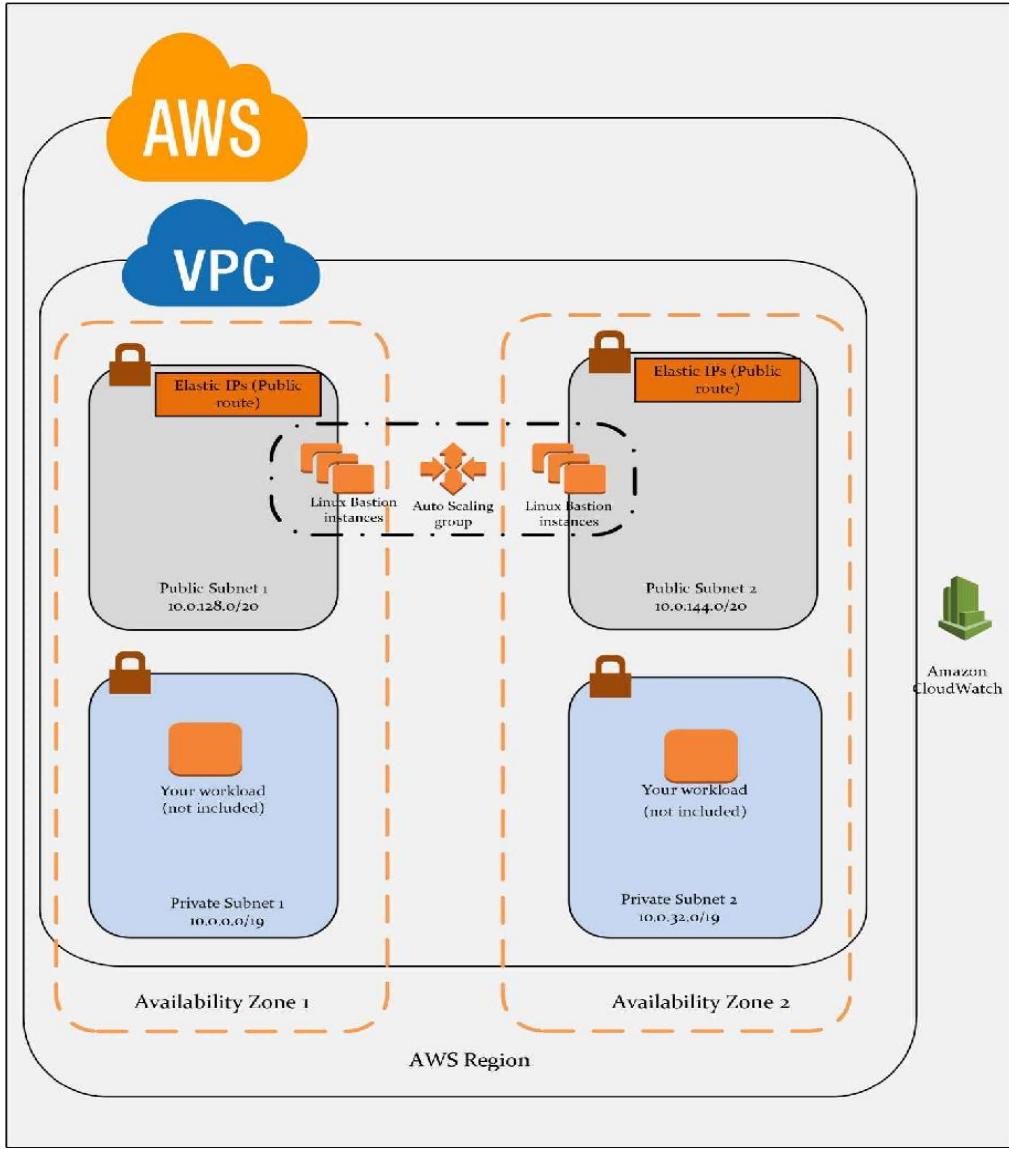


Figure 29. Bastion Host Network Diagram

The Quick Start forms a networking environment that comprises the following components.

- A high-available architecture is possessing two Availability Zones.
- A VPC which is configured with private and public subnets to provide you with your virtual environment on AWS.
- An Internet gateway to let access to the Internet. Bastion host uses this internet gateway to send and receive the traffic.
- To allow the outbound internet access for that resources which are in the private subnets by managing NAT gateways.

- To allow incoming Secure Shell (SSH) access to the instances which are in the private and public subnets by establishing a Linux Bastion host with an Elastic IP address.
- A security group for fine-arrangement of incoming access control.
- An Amazon EC2 Auto Scaling group with a configurable number of instances.
- A group of Elastic IP addresses that resemble the number of bastion host instances. If the Auto Scaling group relaunches any instances, these addresses are reassigned with the new instances.
- An Amazon CloudWatch Logs record groups to hold the Linux bastion host shell history logs.

Deployment Steps:

Follow the step-by-step instructions in this section to make the virtual network environment.

Step:01 Prepare an AWS Account

1. If you don't already have an AWS account, create one at <https://aws.amazon.com> by following the on-screen instructions.
2. Use the region selector in the navigation bar to choose the AWS Region where you want to deploy the Quick Start on AWS as shown below;

US East (N. Virginia)

US East (Ohio)

US West (N. California)

US West (Oregon)

Asia Pacific (Mumbai)

Asia Pacific (Seoul)

Asia Pacific (Singapore)

Asia Pacific (Sydney)

Asia Pacific (Tokyo)

Canada (Central)

EU (Frankfurt)

EU (Ireland)

EU (London)

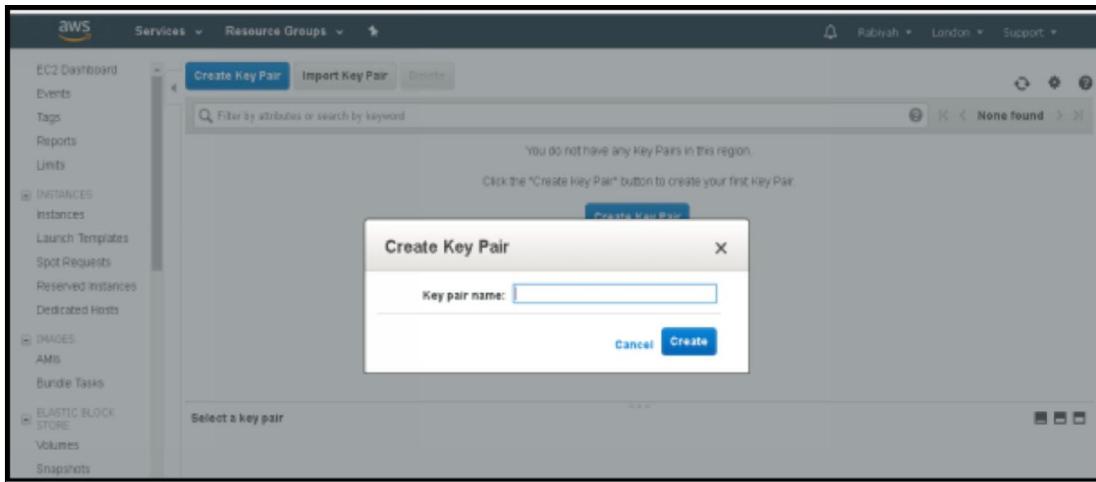
EU (Paris)

South America (São Paulo)



ExamTip: Consider choosing a region closest to your data centre or corporate network to reduce network latency between systems running on AWS and the systems and users on your corporate network

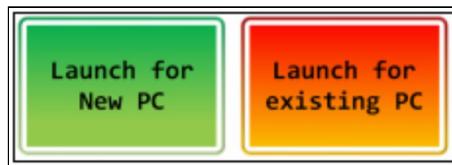
3. Create a key pair in your preferred region. To do this, in the navigation pane of the Amazon EC2 console, choose Key Pairs, Create Key Pair, type a name, and then choose to Create.



Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To be able to log in to your instances, you must create a key pair. If you are doing this on Linux, the key pair is used to verify SSH login.

Step:02 Launch The Stack

1. Use one of the following options to launch the Quick Start into your AWS account.



2. Each stack takes around 5 minutes to generate.



Note:

You are responsible for the cost of the AWS services used while running this

3.

On the Select Template page, keep the default setting for the Amazon S3 template URL, and then choose Next.

4. On the Specify Details page, analyze the parameters for the template, provide values for parameters that need your input, and modify the default settings as necessary.

In the following tables, parameters are listed and described separately for deploying the bastion host into a new VPC or an existing VPC.

Parameters For Deploying Linux Bastion Host For Existing VPC:

These parameters include the following processes;

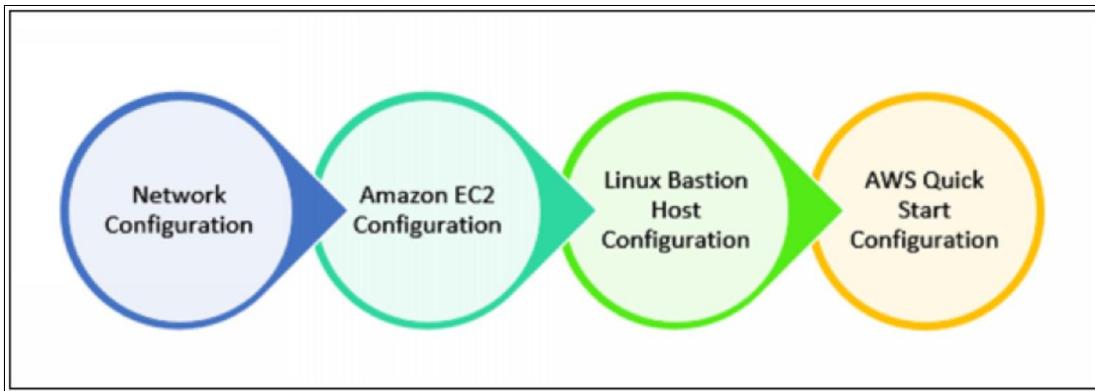


Figure 30. Bastion host deployment parameters

Network Configuration:

Parameter label (Name)	Default	Description
Availability Zones	Requires input that you have given	The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and reserves the logical order you specify
VPC CIDR	10.0.0.0/16	CIDR block for the VPC
Private Subnet 1 CIDR	10.0.0.0/19	CIDR block for the private subnet located in Availability Zone 1.
Private Subnet 2 CIDR	10.0.32.0/19	CIDR block for the private subnet located in Availability Zone 2.
Private Subnet CIDR	10.0.128.0/20	CIDR block for the public subnet located in Availability Zone 1.
Public Subnet 2 CIDR	10.0.144.0/20	CIDR block for the public subnet located in Availability Zone 2.
Allowed Bastion Subnet CIDR	Requires the input that you have given in Availability Zones.	CIDR block that allows SSH external access to the bastion hosts. We recommend that you set this value to a trusted CIDR block. For example, you might want to limit access to your commercial network.

Table 11. Network Configuration in Bastion Host Environment

Amazon EC2 Configuration:

Parameter Label (Name)	Default	Description
Key Pair Name	Requires the input	Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred region.
Bastion AMI Operating System	Amazon-Linux VM	Distributes the Linux for the AMI to use by the bastion host instances. If you select CentOS, firstly make sure that you have a subscription to the CentOS AMI in AWS Marketplace.
Bastion Instance Type	t2.micro	EC2 instance type for the bastion host instances

Table 12. Amazon EC2 Configuration in Bastion Host Network

Linux Bastion Configuration:

Parameter label (Name)	Default	Description
No of bastion hosts	1	The number of Linux bastion hosts that are running. The maximum range of bastion hosts is four.
Enable Banner	false	Includes or compress the banner when you make a link to the bastion host with SSH. To display the banner, set this parameter to true.
Bastion banner	Default URL	URL for the ASCII text file that contains the banner text to display upon login.
Enable TCP forwarding	false	Setting this value to true enables TCP forwarding (SSH tunneling). We recommend keeping the default (disabled) setting unless required.
Enable X11 forwarding	false	Set this value to true to enable X Windows over SSH. We recommend you to keep the default (disabled) setting except when necessary.

Table 13. Linux Configuration in Bastion Host Network

AWS Quick Start Configuration:

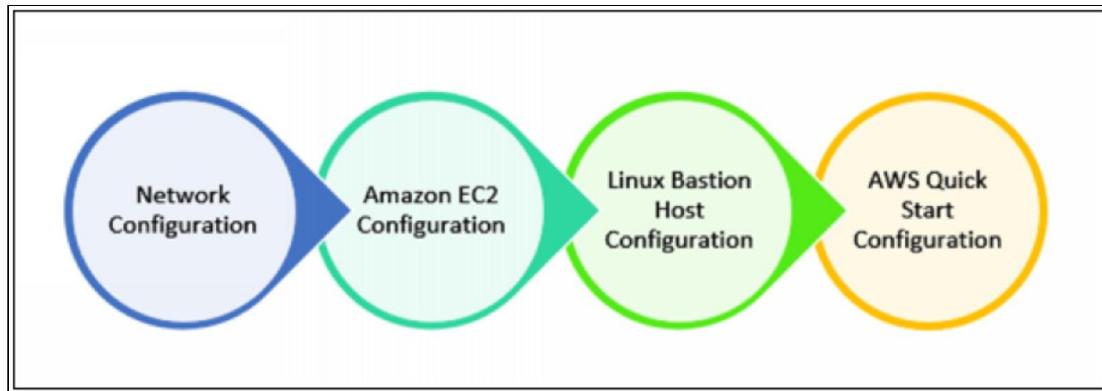
Parameter label (name)	Default	Description
Quick Start S3 Bucket Name	aws-quickstart	You have to create an S3 bucket for making a copy of Quick Start assets if you decide to modify or extend the Quick Start for your use. The bucket name includes numbers, lowercase letters, uppercase letters, and hyphens, but it should not start or end with a hyphen.
Quick Start S3 Key Prefix	quickstart-Linux bastion/	The S3 key name prefix is used to simulate a folder for your copy of Quick Start assets if you decide to modify or extend the Quick Start for your use. This prefix might include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.

*Table 14.
AWS
Quick
Start
Configurat
ion*

**Param
eters
for**

Deploying Linux Bastion Host for Existing VPC:

These parameters also include the following steps;



Network Configuration:

Parameter label (name)	Default	Description
VPC ID	Requires input	The ID of your existing VPC (e.g., vpc-0343606e).
Public subnet 1 ID	Requires input	The ID of the public subnet you want to provide to the first bastion host (e.g., subnet-a0246dcf)
Public Subnet 2 ID	Requires input	The ID of the public subnet you want to provide the second bastion host (e.g., subnet-e3246d8e).
Allowed bastion External Access CIDR	Requires input	CIDR block that permits SSH external access to the bastion hosts. You have to set this value to a trusted CIDR block. For example, you want to limit access to your corporate network.

Table 15. Network Configuration in Bastion Host Network

Amazon EC2 Configuration:

Parameter label (name)	Default	Description
Key pair name	Requires input	Public/private key pair, which allows you to connect securely to your instance after it launches. When you create an AWS account, a key pair will be created in your preferred region.
Bastion AMI Operating System	Amazon-Linux-HVM	The Linux distribution for the AMI to be used for the bastion host instances. If you select CentOS, firstly make sure that you have a subscription to the CentOS AMI in AWS Marketplace.
Bastion Instance Type	T2.micro	EC2 instance type for the bastion host instances.

Table 16. Amazon EC2 Configuration in Bastion Host network

Linux Bastion Configuration:

Parameter label (name)	Default	Description
Number of Bastion Hosts	1	The number of Linux bastion hosts to run. The maximum range of bastion hosts is 4.
Enable Banner	false	Includes or compress the banner when you connect to the bastion host via SSH. To show the banner, set this parameter to true.
Bastion banner	Default URL	URL for the ASCII text file that contains the banner text for presenting on login.

Table 17.
Linux
Bastion
Configurati
on

Enable TCP forwarding	false	By setting this value to true enables TCP forwarding (SSH tunneling). We recommend keeping the default (disabled) setting except where necessary.
Enable X11 Forwarding	false	Setting this value to true enables X Windows over SSH. X11 forwarding can be a useful tool, but it is a security risk. It is recommended to keep the default (disabled) setting except where necessary.
Parameter label (name)	Default	Description
Quick Start S3 Bucket Name	aws-quickstart	The S3 bucket you have created for your copy of Quick Start assets if you decide to customize or extend the Quick Start for your use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.
Quick Start S3 Key Prefix	quickstart-Linux bastion/	The crucial S3 name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to modify or extend the Quick Start for your use. This prefix can contain numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.

Table 18. AWS Quick Start Configuration

When you have done reviewing and modifying the parameters, choose Next.

5. On the Options page, you can specify tags (key-value pairs) for resources in your stack and set advanced options. When you're done, choose Next.
6. On the Review page, review and confirm the template settings. Under Capabilities, select the checkbox to acknowledge that the template will create IAM resources. Now to deploy the stack, click on Create.
7. Monitor the status of the stack. When the status is CREATE_COMPLETE, the stack is ready.

Step 3: Add AWS Services

After you use this Quick Start to build your VPC environment with Linux bastion hosts, you can deploy additional Quick Starts or deploy your applications on top of this AWS infrastructure. If you decide to extend your AWS environment with other Quick Starts for trial or productional use; we recommend that you choose the option to deploy the Quick Start into an existing VPC, where that option is available.

Troubleshooting and Potential Auto Scaling:

Following are the reasons if your instances are not launching into auto-scaling groups;

- Autoscaling Configuration is not working properly.
- Security group does not exist.
- Autoscaling group does not found.
- AZ is no longer supported.
- Invalid Device EBS Mapping.
- Autoscaling service is not enabled on your computer.
- Associated Key Pair does not exist.
- Attempting to attach and EBS block device to an instance-store AMI.
- Instance type specified is not supported in the AZ.

Mind Map:

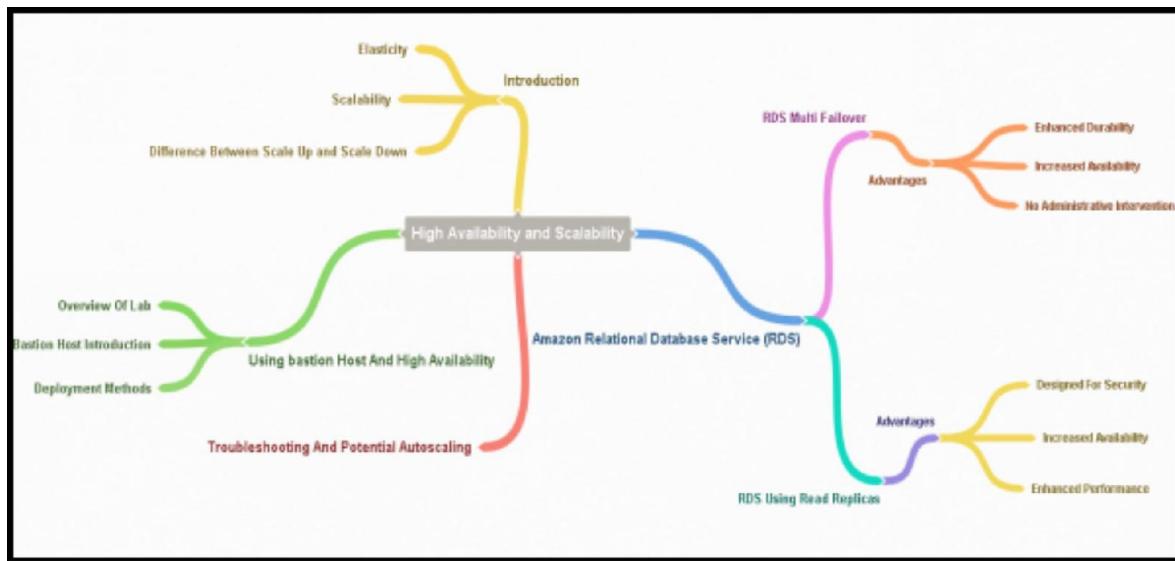


Figure 31. High Availability Mind Map

Chapter 4: Deployment and Provisioning

Introduction

Amazon Web Services offers multiple options for provisioning your IT infrastructure and the deployment of your applications. Whether it is a simple three-tier application or a complex set of workloads, the deployment model varies from customer to customer.

The AWS platform is designed to address scalability, performance, security, ease of deployment, tools to help migrate applications and an ecosystem of developers and architects who are deeply involved in the growth of its products and services.

AWS Deployment Services

When it comes to deployment services, AWS has multiple options too. The following diagram summarizes different deployment services in AWS.

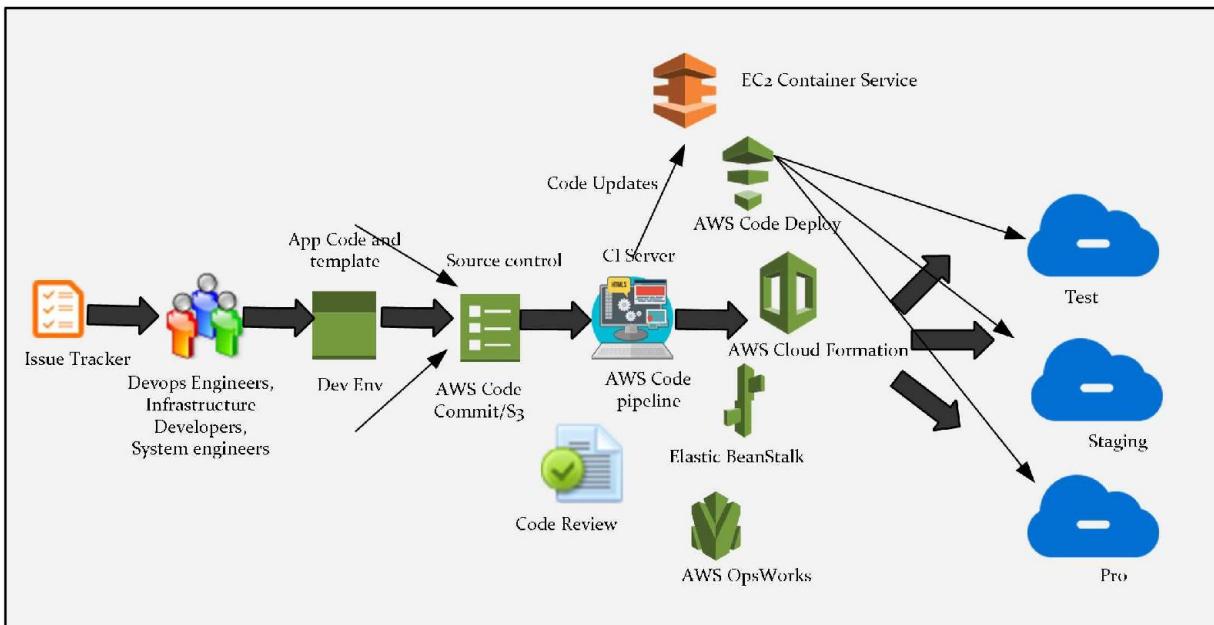


Figure 32. Overview of AWS Deployment Services

AWS Elastic BeanStalk

AWS Elastic Beanstalk is the fastest and most straightforward way to get an application up and running on AWS. It is perfect for developers who want to deploy code and not worry about managing the underlying infrastructure. Elastic Beanstalk is ideal if you have a standard three-tier PHP, Java, Python, Ruby, Node.js, .NET, Go or Docker application that can run on an app server with a database. Elastic Beanstalk uses Auto Scaling and Elastic Load Balancing to support highly variable amounts of traffic easily and works for you if you want to start small and scale up. Common use cases include web apps, content management systems (CMS), and API backends.

AWS CloudFormation

AWS CloudFormation provides the sysadmin, network architect, and other IT personnel the ability to provision and manage stacks of AWS resources based on templates you create to model your infrastructure architecture. You can handle anything from a single Amazon EC2 instance to a complex multilayer, multiregional application. Using templates means you can impose version control on your infrastructure and easily replicate your infrastructure stack quickly and with repeatability. AWS CloudFormation is recommended if you want a tool for granular control over the provisioning and management of your infrastructure.

AWS OpsWorks

AWS OpsWorks is an application management service that makes it easy for both developers and operations personnel to deploy and operate applications of all shapes and sizes. AWS OpsWorks works best if you want to use your code, have some abstraction from the underlying infrastructure, and have an application more complicated than a three-tier architecture. AWS OpsWorks is also recommended if you want to manage your infrastructure with a configuration management system such as Chef.

What is AWS OpsWorks?

Overview

Cloud-based applications usually require a group of related resources such as application servers, database servers, etc. that must be created and managed collectively. This collection of instances is called a stack. AWS OpsWorks provides a simple and straightforward way to create and manage stacks and their associated applications and resources.

OpsWorks consists of two elements, Stacks, and Layers. A stack is a container (or group) of resources such as ELBs, EC2 instances, RDS instances, etc. A layer exists within a stack and consists of things like a web application layer, an application-processing layer or a database layer. When you create a layer, rather than going and configuring everything manually (like installing Apache, PHP, etc.) OpsWorks takes care of this for you. A simple application stack might look something like the following:

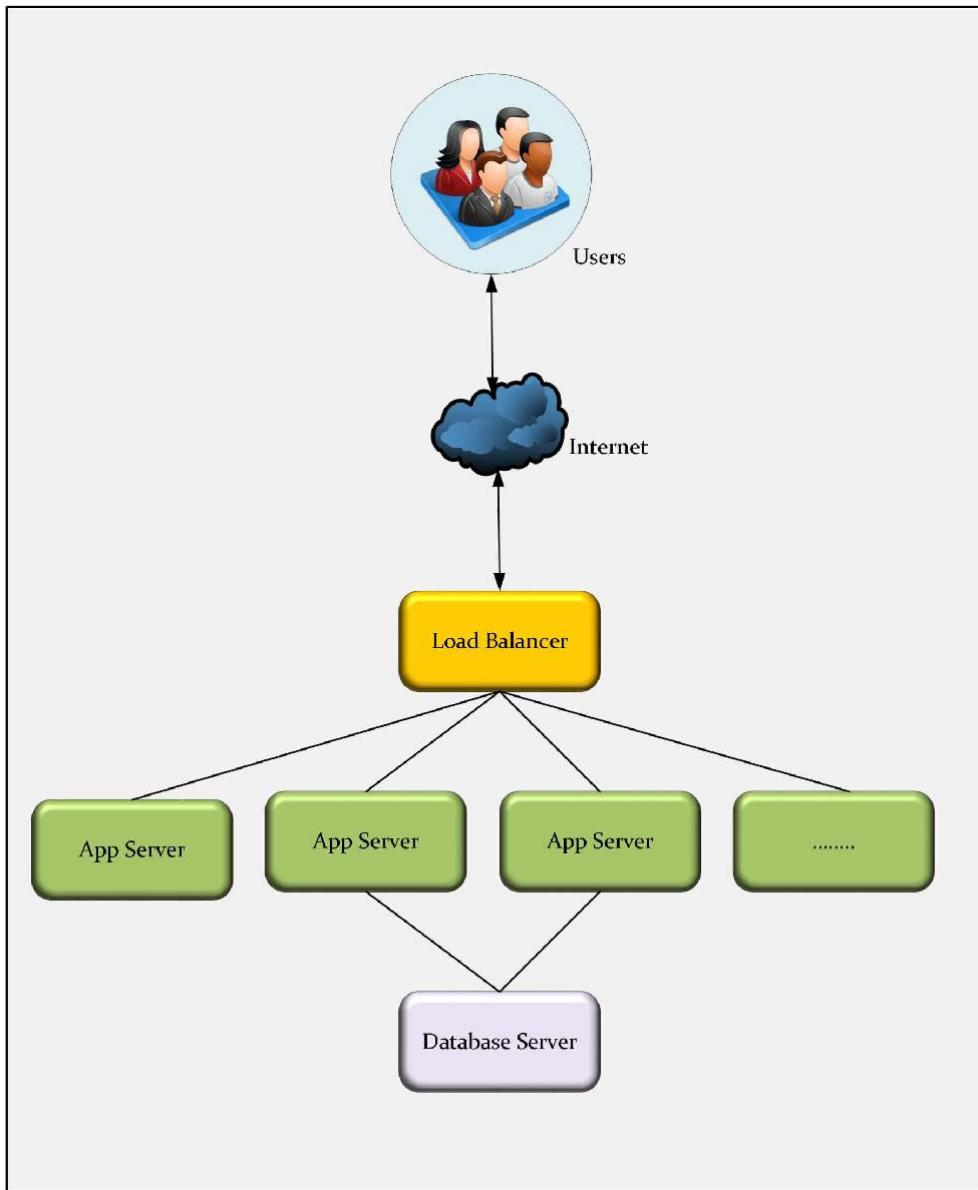


Figure 33. *OpsWorks Stack*

You need one or more layer in a stack. An instance must be assigned to at least one layer. So if you got an EC2 instance, it has to be in the web server layer, application layer or the database layer. Which Chef layers run, are determined by the layer to which the instance belongs. OpsWorks have a list of preconfigured layers that include applications, databases, load balancers and caching layers.

OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed and managed across your Amazon EC2 instances or on-premises compute environments. OpsWorks has three offerings, AWS Opsworks for Chef Automate, AWS OpsWorks for Puppet Enterprise, and AWS OpsWorks Stacks.

What is Chef?

Chef turns infrastructure into code. With Chef, you can automate how you build, deploy, and manage your system's infrastructure. Your system becomes as versionable, testable, and repeatable as application code.

Chef server stores your recipes as well as other configuration data. The Chef client is installed on each server, virtual machine, container, or networking device you manage called as nodes. The client periodically polls Chef server latest policy and state of your network. If anything on the node is out of date, the client brings it up to date.

For configuration management, you can use Chef cookbooks and Chef Automate by using AWS OpsWorks Stacks while AWS OpsWorks for Puppet Enterprise lets you configure a [Puppet Enterprise](#) master server in AWS. Puppet offers a set of tools for imposing the preferred state of your organization and automating on-demand tasks.

Root/Admin Access Services

Services with root/admin access to operating system include the following four:

1. Elastic Beanstalk
2. Elastic MapReduce
3. OpsWorks
4. EC2

Remember that you do not have root/admin access to RDS, DynamoDB, S3 or Glacier.

Elastic Load Balancing

Introduction

Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple EC2 instances. It seamlessly provides necessary load balancing capacity required for application traffic distribution so that you can achieve higher levels of fault tolerance in your applications.

Elastic Load Balancing supports three types of load balancers, Application Load Balancers, Network Load Balancers, and Classic Load Balancers. You can select a load balancer based on your application needs. These load balancers feature high availability, automatic scaling, and robust security.

Application Load Balancer	Network Load Balancer	Classic Load Balancer
<ul style="list-style-type: none">Makes routing decisions at the application layer (layer 7) and is best suited for load balancing of HTTP and HTTPS traffic.Application Load Balancer routes traffic to targets - EC2 instances, containers and IP addresses within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.Ideal for applications requiring advanced routing capabilities, micro-services, and container-based architectures.	<ul style="list-style-type: none">Makes routing decisions at the transport layer (Layer 4) and is best suited for load balancing of TCP traffic where extreme performance is required.Network Load Balancer routes connections to targets - Amazon EC2 instances, containers and IP addresses based on IP protocol data.Optimized to handle sudden and volatile traffic patterns and is capable of handling millions of requests per second while maintaining ultra-low latencies.	<ul style="list-style-type: none">Makes routing decisions at the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS) and supports either EC2 Classic or a VPC. However, it is recommended to use Application Load Balancer for Layer 7 and Network Load Balancer for Layer 4 when using Virtual Private Cloud (VPC).Classic Load Balancer routes traffic based on either application or network level information.Ideal for simple load balancing of traffic across multiple EC2 instances.

Figure 34. Elastic Load Balancer

Use the Application Load Balancer for flexible application management and TLS termination. If extreme performance and static IP is needed for your application, then use Network Load Balancer. Use Classic Load Balancer if your application is built within the EC2 Classic network.

It helps the system to be fault tolerance in the following ways;

- Serves as a single point of contact for all the customers, so that the availability of applications increases.
- When your requirements and needs change, you can add or remove the instances from your load balancer.
- ELB also provides you with the facility to configure the health checks, which are used to monitor the health of the registered instances.

ELB Configurations

A load balancer accepts incoming traffic from clients and routes requests to its registered targets (such as EC2 instances) in one or more Availability Zones. The load balancer also monitors the health of its designated targets and ensures that it routes traffic only to healthy targets. When the load balancer detects an unhealthy target, it stops routing traffic to that target and then resumes routing traffic to that target when it discovers that the objective is healthy again.

You configure your load balancer to accept incoming traffic by specifying one or more listeners. A listener is a process that checks for connection requests. It is configured with a protocol and port number for connections from clients to the load balancer and a protocol and port number for connections from the load balancer to the targets.

The critical difference between the three load balancers is the way you configure these load balancers. With Application Load Balancers and Network Load Balancers, you register targets in target groups and route traffic to the target groups. With Classic Load Balancers, you register instances with the load balancer.

You can use Elastic Load Balancers to load balance across different availability zones within the same region, but not to the different areas (or different VPC's) themselves. An ELB and a NAT are two entirely different things.

Configuration Types

There are two types of ELB configuration:

1. External Elastic Load Balancer

An External Load balancer, also known as Internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the Internet to the EC2 instances that are registered with the load balancer.

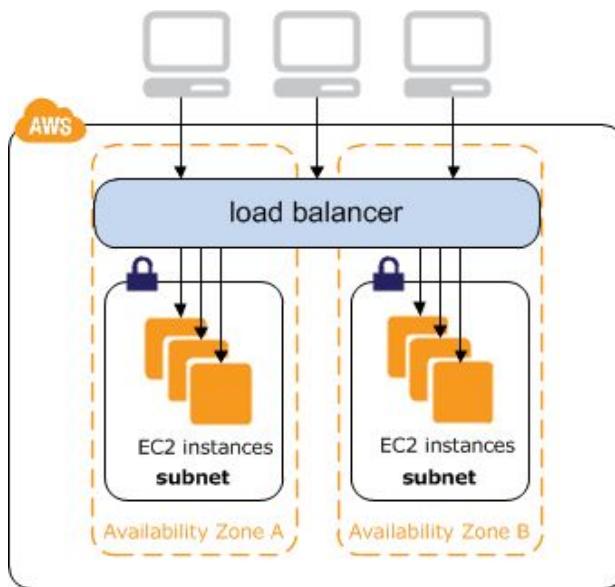


Figure 35. Internet Facing Load Balancer

An Internet-facing load balancer routes traffic from the Internet to your EC2 instances. If a load balancer is in a VPC with ClassicLink enabled, its instances can be linked EC2-Classic instances. If a load balancer is in EC2-Classic, its instances must be in EC2-Classic.

2. Internal Elastic Load Balancer

The nodes of an internal load balancer have only private IP addresses. The DNS name of an internal load balancer is publicly resolvable to the private IP addresses of the nodes. Therefore, internal load balancers can only route requests from clients with access to the VPC for the load balancer.

If your application has multiple tiers, for example, web servers that must be connected to the Internet and database servers that are only connected to the web servers, you can design an architecture that uses both internal and Internet-facing load balancers. Create an Internet-facing load balancer and register the web servers with it. Create an internal load balancer and register the database servers with it. The web servers receive requests from the Internet-facing load balancer and send requests for the database servers to the internal load balancer. The database servers receive requests from the internal load balancer.

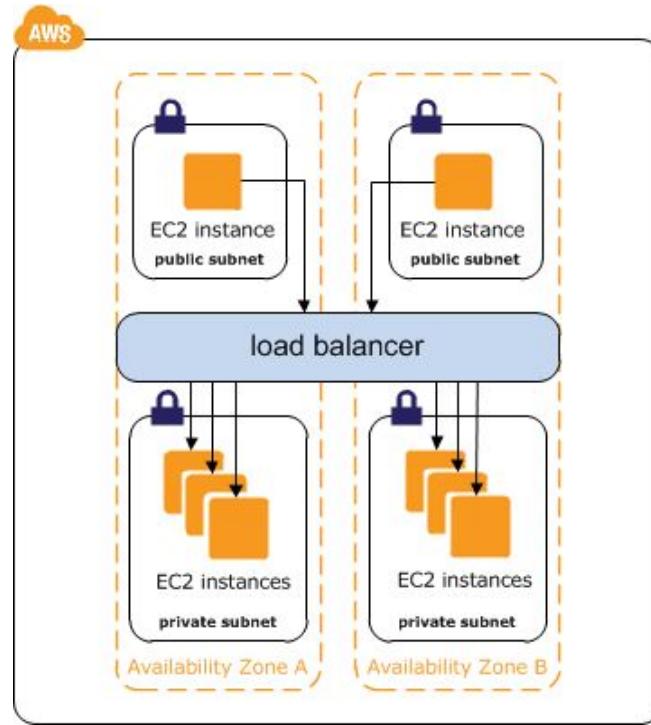


Figure 36. Internal Load Balancer

Health Checks Configuration

Load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances to discover the availability of your EC2 instances. These tests are called health checks. The status of the instances that are healthy at the time of the health check is InService. The status of any instances that are unhealthy at the time of the health check is OutOfService. The load balancer performs health checks on all registered instances, whether the instance is in a healthy state or an unhealthy state.

The load balancer routes request only to the healthy instances. When the load balancer determines that an instance be unhealthy, it stops routing requests to that instance. The load balancer resumes routing requests to the instance when it has been restored to a healthy state.

The load balancer checks the health of the registered instances using either the default health check configuration provided by Elastic Load Balancing or a health check configuration that you configure.

A health configuration contains the information that a load balancer uses to determine the health state of the registered instances. The following table describes the health check configuration fields.

Field	Description
Ping Protocol	The protocol to use to connect with the instance. Valid values: TCP, HTTP, HTTPS, and SSL Console default: HTTP CLI/API default: TCP
Ping Port	The port to use to connect with the instance, as a protocol:port pair. If the load balancer fails to communicate with the instance at the specified port within the configured response timeout period, the instance is considered unhealthy. Ping protocols: TCP, HTTP, HTTPS, and SSL Ping port range: 1 to 65535 Console default: HTTP:80

	CLI/API default: TCP:80
Ping Path	<p>The destination for the HTTP or HTTPS request.</p> <p>An HTTP or HTTPS GET request is issued to the instance on the ping port and the ping path. If the load balancer receives any response other than "200 OK" within the response timeout period, the instance is considered unhealthy. If the response includes a body, your application must either set the Content-Length header to a value greater than or equal to zero or specify Transfer-Encoding with a value set to 'chunked.'</p> <p>Default: /index.html</p>
Response Timeout	<p>The amount of time to wait when receiving a response from the health check, in seconds.</p> <p>Valid values: 2 to 60</p> <p>Default: 5</p>
HealthCheck Interval	<p>The amount of time between health checks of an individual instance, in seconds.</p> <p>Valid values: 5 to 300</p> <p>Default: 30</p>
Unhealthy Threshold	<p>The number of consecutive failed health checks that must occur before declaring an EC2 instance unhealthy.</p> <p>Valid values: 2 to 10</p> <p>Default: 2</p>
Healthy Threshold	<p>The number of consecutive successful health checks that must occur before declaring an EC2 instance healthy.</p> <p>Valid values: 2 to 10</p> <p>Default: 10</p>

Table 17. Health check configuration

The load balancer sends a health check request to each registered instance every Interval seconds, using the specified port, protocol, and ping path. Each health check request is independent and lasts the entire interval. The time it takes for the instance to respond does not affect the delay for the

next health check. If the health checks exceed `UnhealthyThresholdCount` consecutive failures, the load balancer takes the instance out of service. When the health checks exceed `HealthyThresholdCount` successive successes, the load balancer puts the instance back in service.

An HTTP/HTTPS health check succeeds if the instance returns a 200 response code within the health check interval. A TCP health check succeeds if the TCP connection succeeds. An SSL health check succeeds if the SSL handshake succeeds.

Sticky Sessions

By default, a Load Balancer routes each request independently to the registered instance with the smallest load. However, you can use the sticky session feature (also known as session affinity), which enables the load balancer to bind a user's session to a specific instance. This ensures that all requests from the user during the session are sent to the same instance.

The key to managing sticky sessions is to determine how long your load balancer should consistently route the user's request to the same instance. If your application has its session cookie, then you can configure Elastic Load Balancing so that the session cookie follows the duration specified by the application's session cookie. If your application does not have its session cookie, then you can configure Elastic Load Balancing to create a session cookie by specifying your stickiness duration.

Elastic Load Balancing creates a cookie, named `AWSELB`, which is used to map the session to the instance. Requirements are:

- An HTTP/HTTPS load balancer.
- At least one healthy instance in each Availability Zone.

Sticky Session Types

1. Duration-Based Session Stickiness
2. Application-Controlled Session Stickiness

Duration-Based Session Stickiness

The load balancer uses a particular cookie to track the instance for each request to each listener. When the load balancer receives a request, it first checks to see if this cookie is present in the request. If so, the request is sent to the instance specified in the cookie. If there is no cookie, the load balancer chooses an instance based on the existing load balancing algorithm. A cookie is inserted into the response for subsequent binding requests from the same user to that instance.

The stickiness policy configuration defines a cookie expiration, which establishes the duration of validity for each cookie. The load balancer does not refresh the expiry time of the cookie and does not check whether the cookie is expired before using it. After a cookie expires, the session is no longer sticky. The client should remove the cookie from its cookie store upon expiry.

If an instance fails or becomes unhealthy, the load balancer stops routing requests to that instance and chooses a new healthy instance based on the existing load balancing algorithm. The request is routed to the new instance as if there is no cookie and the session is no longer sticky. If a client switches to a listener with a different backend port, stickiness is lost.

Application-Controlled Session Stickiness

The load balancer uses a particular cookie to associate the session with the instance that handled the initial request but follows the lifetime of the application cookie specified in the policy configuration. The load balancer only inserts a new stickiness cookie if the application response includes a new application cookie. The load balancer stickiness cookie does not update with each request. If the application cookie is explicitly removed or expires, the session stops being sticky until a new application cookie is issued.

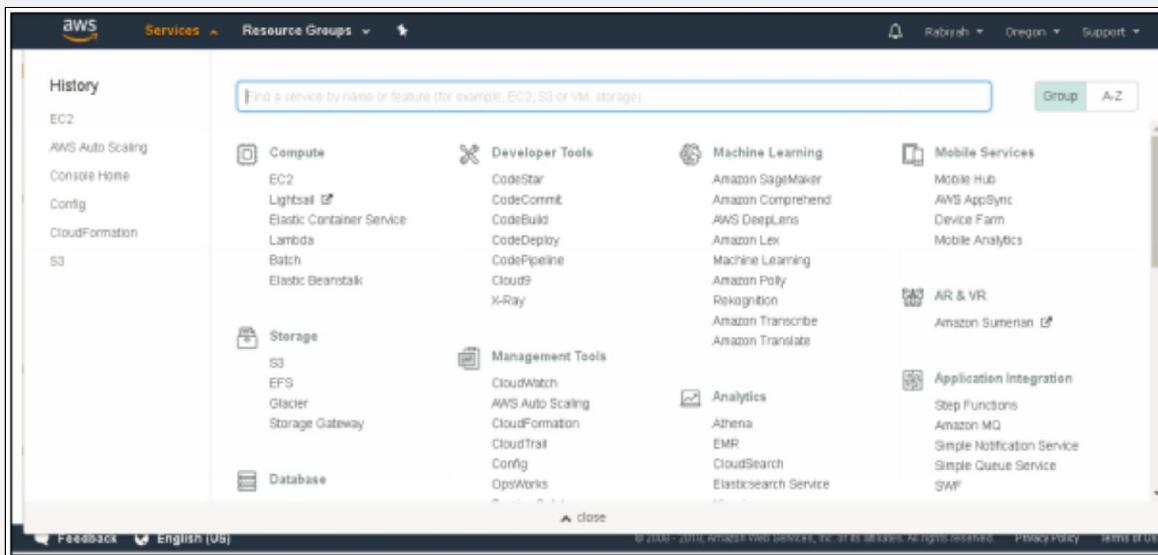
If an instance fails or becomes unhealthy, the load balancer stops routing requests to that instance and chooses a new healthy instance based on the existing load balancing algorithm. The load balancer treats the session as now "stuck" to the new healthy instance, and continues routing requests to that instance even if the failed instance comes back. However, it is up to the

new application instance whether and how to respond to a session which it has not previously seen.

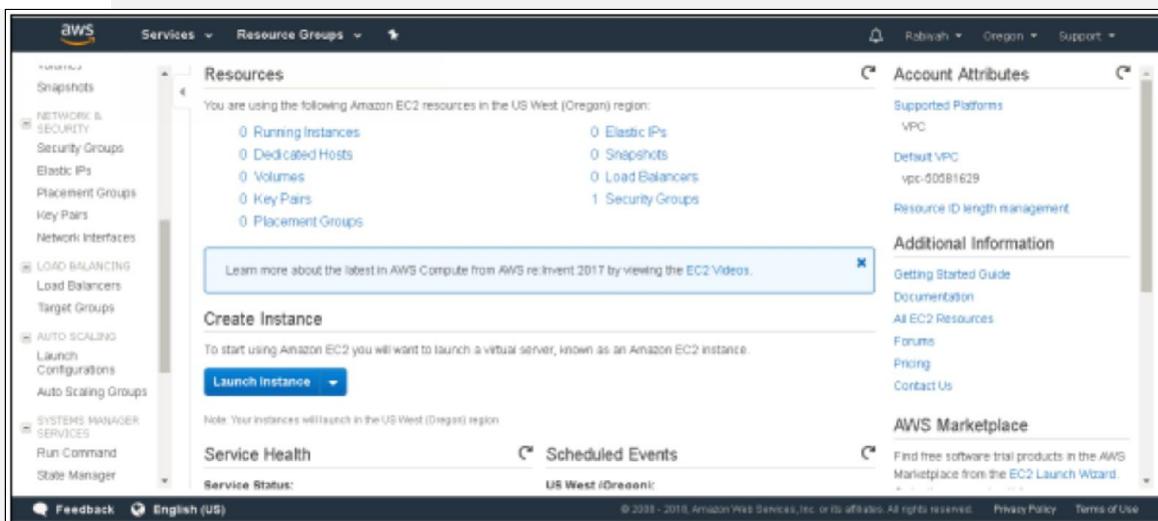
Lab 4.1 Elastic Load Balancer Configuration

Here are the steps to deploy an Elastic Load Balancer for your instance;

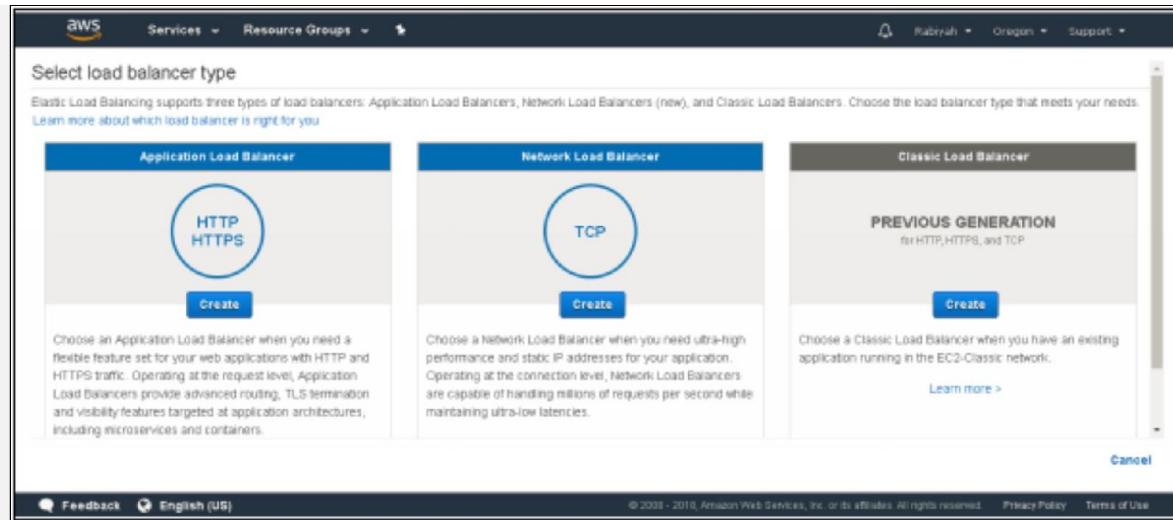
1. Go to the AWS Management Console and login to your account.
2. Go to the option “compute” and select the “EC2”.



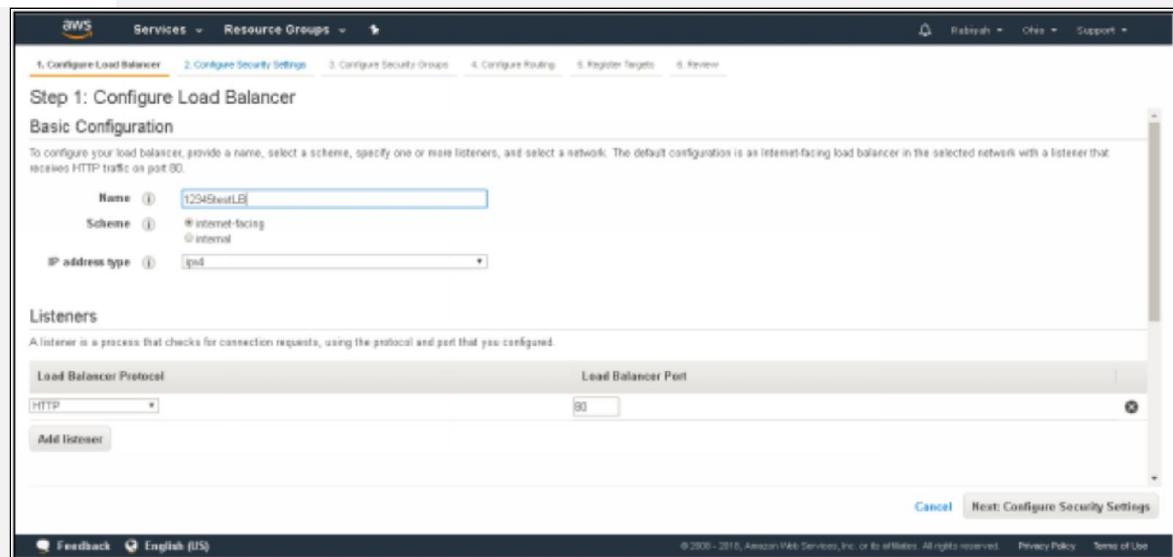
3. Go to the Load Balancer option and click on the “Create Load Balancer.”



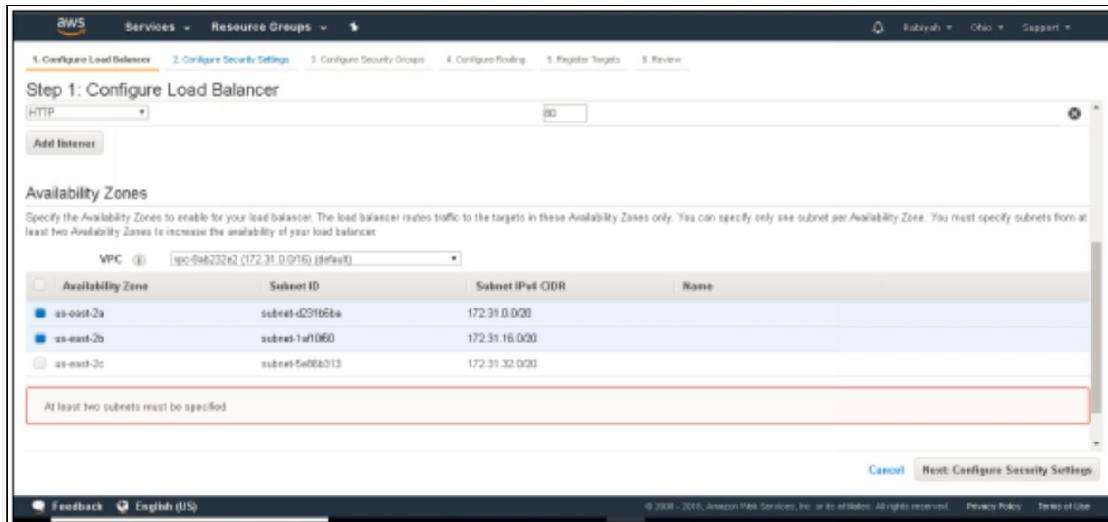
4. Choose the type of Load Balancer you want, and click on “Create.”



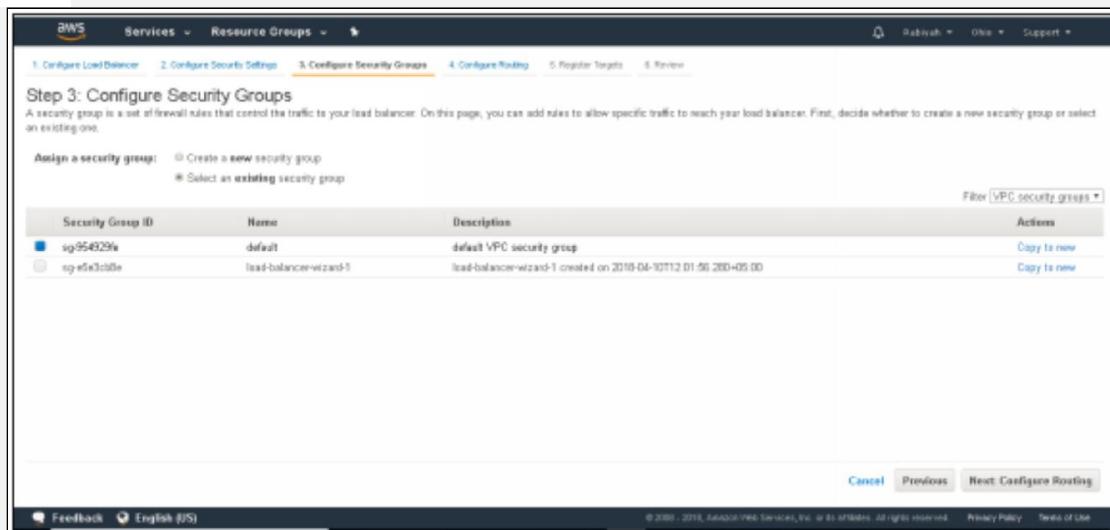
5. Configure Load Balancer. To do this, give a name to your load balancer.



6. Select the “Availability Zones” for your Load Balancer, as shown in the figure.



7. Now configure the security groups for your instances. For this purpose, select the load balancer which you have made.



8. To route the request to the registered instances, you have to configure the Load Balancer. Name the “Target Group” and click on “Next.”

Step 4: Configure Routing
Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

Target group

Target group	New target group
Name	8289
Protocol	HTTP
Port	80
Target type	instance

Health checks

Protocol	HTTP
Path	/

Advanced health check settings

Buttons: Cancel, Previous, Next: Register Targets

9. Click on “Add to registered” and then click on “Next.”

Step 5: Register Targets
Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets
To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	State	Security groups	Zone
i-0706ae3e9400ea886	Cloudwatch EC2	80	running	default	us-east-2b

Instances
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered on port 80

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CDR
i-0706ae3e9400ea886	Cloudwatch EC2	running	default	us-east-2b	subnet-0aff0988	172.31.16.8/20

Buttons: Cancel, Previous, Next: Review

10. You will get the complete review of your load balancer which has been created.

Step 6: Review
Please review the load balancer details before continuing.

Load balancer

Name	12345testLB
Scheme	internet-facing
Listeners	Port 80 - Protocol HTTP
IP address type	IPv4
VPC	vpc-0ab232e2
Subnets	subnet-0231668a, subnet-1aff0980
Tags	

Security groups

Security group	sg-954509fe
----------------	-------------

Routing

Buttons: Cancel, Previous, Review

The screenshot shows the 'Targets' section of the CloudWatch Metrics console. A new target group named '6789' is being created. The configuration includes:

- Target type: instance
- Protocol: HTTP
- Health check protocol: HTTP
- Path: /
- Health check port: traffic port
- Healthy threshold: 5
- Unhealthy threshold: 2
- Timeout: 5
- Interval: 30
- Success codes: 200

The 'Targets' section lists one instance: i-070bae3e94ceea066 (Cloudwatch EC2) 80.

Buttons at the bottom include 'Cancel', 'Previous', and 'Create'.

11. The following figure shows that Load Balancer has been successfully created.

The screenshot shows the 'Load Balancer Creation Status' page. It displays a green success message: "Successfully created load balancer 1234testLB was successfully created. Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks." A 'Close' button is at the bottom right.

12. To edit the configuration, attributes or any other setting in your load balancer, select the specific Load Balancer and click on “Description.” Here you will get the option of edit.

The screenshot shows the 'Load Balancers' section of the AWS Services page. The 'Description' tab is selected for the load balancer '1234testLB'. The basic configuration details are displayed:

Name	ARN	Creation time	Hosted zone
1234testLB	arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/1234testLB	April 10, 2018 at 12:11:06 PM UTC+5	Z3AAUDQH6KTL2

Other tabs available are 'Listeners' and 'Monitoring'.

AWS Services Resource Groups

Bundle Tasks

ELASTIC BLOCK STORE

- Volumes
- Snapshots

NETWORK & SECURITY

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

LOAD BALANCING

- Load Balances**
- Target Groups

AUTO SCALING

- Launch Configurations
- Auto Scaling Groups

SYSTEMS MANAGER SERVICES

- Run Command
- State Manager

Create Load Balance Actions

Filter by tags and attributes or search by keyword

1 to 2 of 2

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
12345testLB	12345testLB-963T07017.u...	provisioning	vpc-fab232e2	us-east-2a, us-east-2a	application	April 10, 21
09876vTestLB	09876vTestLB-1341369163...	active	vpc-fab232e2	us-east-2b, us-east-2a	application	April 10, 21

Security

Security groups: sg-9549295, default

- default VPC security group

Edit security groups

Attributes

Deletion protection	Disabled
Idle timeout	60 seconds
HTTP2	Enabled
Access logs	Disabled

Edit attributes

The screenshot shows the AWS Load Balancing console. On the left, there's a navigation sidebar with various services like Elastic Block Store, Network & Security, Auto Scaling, and Systems Manager. The 'Load Balances' section is selected. The main area displays two load balancers: '12345testLB' and '09876vTestLB'. The '12345testLB' is currently in a 'provisioning' state, while '09876vTestLB' is 'active'. Both are application load balancers within the same VPC and across two availability zones. The security group assigned to both is the 'default' VPC security group. There are also settings for deletion protection, idle timeout, HTTP2 support, and access logs.

Pre-Warming Elastic Load Balancer

If you are expecting massive traffic at the same time in any event in Amazon, your system will be at high risk of responding poorly. ELBs scale can work much better if there is a gradual increase in the traffic load instead of a massive spike in traffic.

Therefore, the AWS staff has made a solution to handle this problem in which they pre-configure the Load Balancer so that the Load Balancers has the appropriate level of traffic according to the incoming traffic. This method of pre-configuration is known as “Pre-Warming” a Load Balancer.

Pre-Warming The Load Balancer:

AWS staff can pre-configure the load balancer to have the appropriate level of capacity based on expected traffic in specific scenarios, such as when flash traffic is expected, or in the case where a load test cannot be configured to increase traffic gradually.

Before requesting load balancing, you need to know that:

- The start and end dates of your expected flash traffic or test
- The expected request rate per second
- The total size of the typical request/response that you will be experiencing.

Chapter 5: Data Management

Disaster Recovery

Disaster recovery (DR) is about preparing for and recovering from a disaster. A disaster is an event that has a negative impact on a company's business continuity or finances. DR includes hardware or software failure, a network outage, a power outage, physical damage to a building like fire or flooding, human error, or some other significant event.

Companies invest time and resources to plan and prepare for minimizing the impact of a disaster, to train employees, and to document and update processes. AWS also gives you the flexibility to quickly change and optimize resources during a DR event, which can result in significant cost savings.

Traditional Approaches to DR

A traditional approach to DR involves different levels of off-site duplication of data and infrastructure. Critical business services are set up and maintained on this infrastructure and tested at regular intervals. The disaster recovery environment's location and the source infrastructure should be a significant physical distance apart to ensure that the disaster recovery environment is isolated from faults that could impact the source site.

At a minimum, the infrastructure that is required to support the duplicate environment should include the following:

- Facilities to house the infrastructure, including power and cooling.
- Security to ensure the physical protection of assets.
- Suitable capacity to scale the environment.
- Support for repairing, replacing, and refreshing the infrastructure.
- Contractual agreements with an Internet service provider (ISP) to provide Internet connectivity that can sustain bandwidth utilization for the environment under a full load.
- Network infrastructure such as firewalls, routers, switches, and load balancers.
- Enough server capacity to run all mission-critical services, including storage appliances for the supporting data, and servers to run applications and backend services such as user authentication, Domain Name System (DNS),
- Dynamic Host Configuration Protocol (DHCP), monitoring, and alerting.

Using AWS for DR

AWS services and features can leverage for your disaster recovery (DR) processes to significantly minimize the impact on your data, your system, and your overall business operations. AWS also gives you the flexibility to quickly change and optimize resources during a DR event, which can result in significant cost savings. Applications deployed on AWS have the multi-site capability using multiple Availability Zones. Availability Zones are different locations that are designed to be isolated from each other. They provide low-cost, low-latency network connectivity within the same region.

- Only minimal hardware is required for ‘data replication.’
- Allows you to be flexible depending on what the disaster is and how to recover from it.
- Opex cost model (pay as you use) rather than heavy investment upfront. Scaling is quick and easy.
- Automate disaster recovery deployment.

AWS Features and Services Essential for Disaster Recovery

Before we discuss the various approaches to DR, it is essential to review the AWS services and features that are the most relevant to disaster recovery.

In the preparation phase of DR, it is essential to consider the use of services and features that support data migration and durable storage, because they enable you to restore backed-up, critical data to AWS when disaster strikes. For some of the scenarios that involve either a scaled-down or a fully scaled deployment of your system in AWS, compute resources will be required as well.

When reacting to a disaster, it is essential to either quick commission compute resources to run your system in AWS or to orchestrate the failover to already running resources in AWS. The critical infrastructure pieces include DNS, networking features, and various Amazon Elastic Compute Cloud (Amazon EC2) features described later in this section.

Regions

Amazon Web Services are available in multiple regions around the globe, so you can choose the most appropriate location for your DR site, in addition to the place where your system is fully deployed. AWS has multiple general-purpose regions in the Americas, EMEA, and the Asia Pacific that anyone with an AWS account can access. Special-use regions are also available for government agencies and China.

Storage

Amazon Simple Storage Service (S3) - Amazon S3 provides a highly durable (provide durability of 99.99999999%) storage infrastructure designed for mission-critical and primary data storage. AWS offers further protection for data retention and archiving through:

- Versioning in Amazon S3
- AWS multi-factor authentication (AWS MFA)
- Bucket policies

- AWS Identity and Access Management (IAM)

Amazon Glacier - provides exceptionally low-cost storage for data archiving and backup. Objects are optimized for infrequent access, for which retrieval times of several hours are adequate and is designed for the same durability as Amazon S3.

Amazon Elastic Block Store (EBS) - provides the ability to create point-in-time snapshots of data volumes. You can use the snapshots as the starting point for new Amazon EBS volumes, and you can protect your data for long-term durability because snapshots are stored within Amazon S3.

Amazon Import/Export - accelerates moving large amounts of data into and out of AWS by using portable storage devices for transport. AWS Import/Export bypasses the Internet and transfers your data directly onto and off of storage devices using the high-speed internal network of Amazon You can use AWS Import/Export to migrate data into and out of Amazon S3 buckets and Amazon Glacier vaults or into Amazon EBS snapshots.

Amazon Storage Gateway - is a service that connects an on-premises software appliance with cloud-based storage to provide seamless and highly secure integration between your on-premises IT environment and the storage infrastructure of AWS. It supports three different configurations:

- **Gateway-cached volumes** — You can store your primary data in Amazon S3 and retain your frequently accessed data locally.
- **Gateway-stored volumes** — If you need low-latency access to your entire data set, you can configure your gateway to store your primary data locally, and asynchronously back up point-in-time snapshots of this data to Amazon S3.
- **Gateway-virtual tape library (gateway-VTL)** — With gateway-VTL, you can have an almost limitless collection of virtual tapes. You can store each virtual tape in a virtual tape library (VTL) backed by Amazon S3 or a virtual tape shelf (VTS) supported by Amazon Glacier.

Compute

Amazon EC2 - provides resizable compute capacity in the cloud. Within minutes, you can create Amazon EC2 instances, which are virtual machines over which you have complete control.

Amazon EC2 VM Import Connector - virtual appliance enables you to import virtual machine images from your existing environment to Amazon EC2 instances.

Networking

Amazon Route 53 - is a highly available and scalable Domain Name System (DNS) web service. It includes some global load-balancing capabilities (which can be useful when you are dealing with DR scenarios such as DNS endpoint health checks) and the ability to failover between multiple endpoints and even static websites hosted in Amazon S3.

Elastic Load Balancing - automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve even greater fault tolerance in your applications by seamlessly providing the load-balancing capacity that is needed in response to incoming application traffic.

Amazon Virtual Private - lets you provision a private, isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. In the context of DR, you can use Amazon VPC to extend your existing network topology to the cloud; this can be especially appropriate when recovering enterprise applications that are typically on the internal network.

Amazon Direct Connect - makes it easy to set up a dedicated network connection from your premises to AWS. In many cases, this can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

Databases

Amazon Relational Database - Amazon RDS makes it easy to set up, operate, and scale a relational database in the cloud. You can use Amazon RDS either in the preparation phase for DR to hold your critical data in a database that is already running, or in the recovery phase to run your production database. When you want to look at multiple regions, Amazon RDS gives you the ability to snapshot data from one region to another, and also to have a read replica running in another region.

Amazon Dynamo DB - is a fast, fully managed NoSQL database service that makes it simple and cost-effective to store and retrieve any amount of data and serve any level of request traffic. It has reliable throughput and single-digit, millisecond latency. You can also use it in the preparation phase to copy data to DynamoDB in another region or to Amazon S3. During the recovery phase of DR, you can scale up seamlessly in a matter of minutes with a single click or API call.

Amazon Redshift - is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools. You can use Amazon Redshift in the preparation phase to snapshot your data warehouse to be durably stored in Amazon S3 within the same region or copied to another region. During the recovery phase of DR, you can quickly restore your data warehouse into the same region or within another AWS region.

Deployment Orchestration

Deployment automation and post-startup software installation/configuration processes and tools can be used in Amazon EC2. This can be very helpful in the recovery phase, enabling you to create the required set of resources in an automated way.

AWS Cloud Formation - gives developers and systems administrators an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion. You can create templates for your environments and deploy associated collections of resources (called a stack) as needed.

AWS Bean Stalk - is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, and Docker. You can deploy your application code, and AWS Elastic Beanstalk will provision the operating environment for your applications.

AWS OpsWorks - is an application management service that makes it easy to deploy and operate applications of all types and sizes. You can define your environment as a series of layers, and configure each layer as a tier of your application. AWS OpsWorks has automatic host replacement, so in the event of an instance failure, it will be automatically replaced. You can use AWS OpsWorks in the preparation phase to template your environment, and you can combine it with AWS CloudFormation in the recovery phase. You can quickly provision a new stack from the stored configuration that supports the defined RTO.

RTO and RPO

The two common industry terms for disaster planning includes:

- Recovery time objective (RTO) — is the length of time from which you can recover from a disaster. It is measured from when the crash first occurred as to when you have fully recovered from it.
- Recovery point objective (RPO) — is the amount of data your organization is prepared to lose in the event of a disaster.

Typically the lower the RTO & RPO threshold, the more costly its solution will be.

Recovery Time Objective (RTO)

The time it takes after a disruption to restore a business process to its service level, as defined by the operational level agreement (OLA). For example, if a disaster occurs at 12:00 PM (noon) and the RTO is eight hours, the DR process should restore the business process to the acceptable service level by 8:00 PM.

Recovery Point Objective (RPO)

The acceptable amount of data loss measured in time. For example, if a disaster occurs at 12:00 PM (noon) and the RPO is one hour, the system should recover all data that was in the system before 11:00 AM. Data loss will span only one hour, between 11:00 AM and 12:00 PM (noon).

Disaster Recovery Scenarios with AWS

This section outlines four DR scenarios that highlight the use of AWS and compare AWS with traditional DR methods. The following figure shows a spectrum for the four scenes, arranged by how quickly a system can be available to users after a DR event.

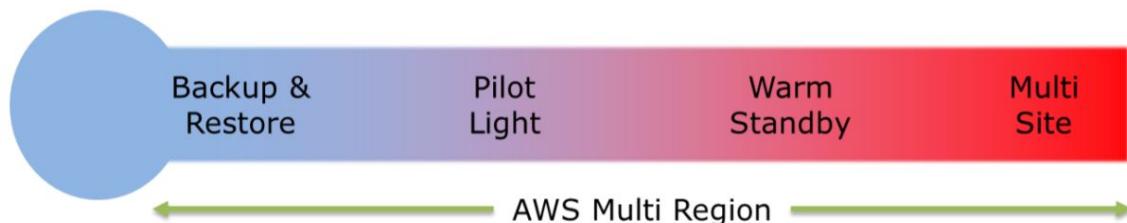


Figure 37. The spectrum of Disaster Recovery Options

Backup and Restore

In most traditional environments, data is backed up to tape and sent off-site regularly. If you use this method, it can take a long time to restore your system in the event of a disruption or disaster. Amazon S3 is an ideal destination for backup data that might be needed quickly to perform a restore. Transferring data to and from Amazon S3 is typically done through the network and is therefore accessible from any location.

You can use AWS Import/Export to transfer extensive data sets by shipping storage devices directly to AWS. For longer-term data storage where retrieval times of several hours are adequate, there is Amazon Glacier, which has the same durability model as Amazon S3. Amazon Glacier and Amazon S3 can be used in conjunction to produce a tiered backup solution.

The following figure shows data backup options to Amazon S3, from either on-site infrastructure or AWS.

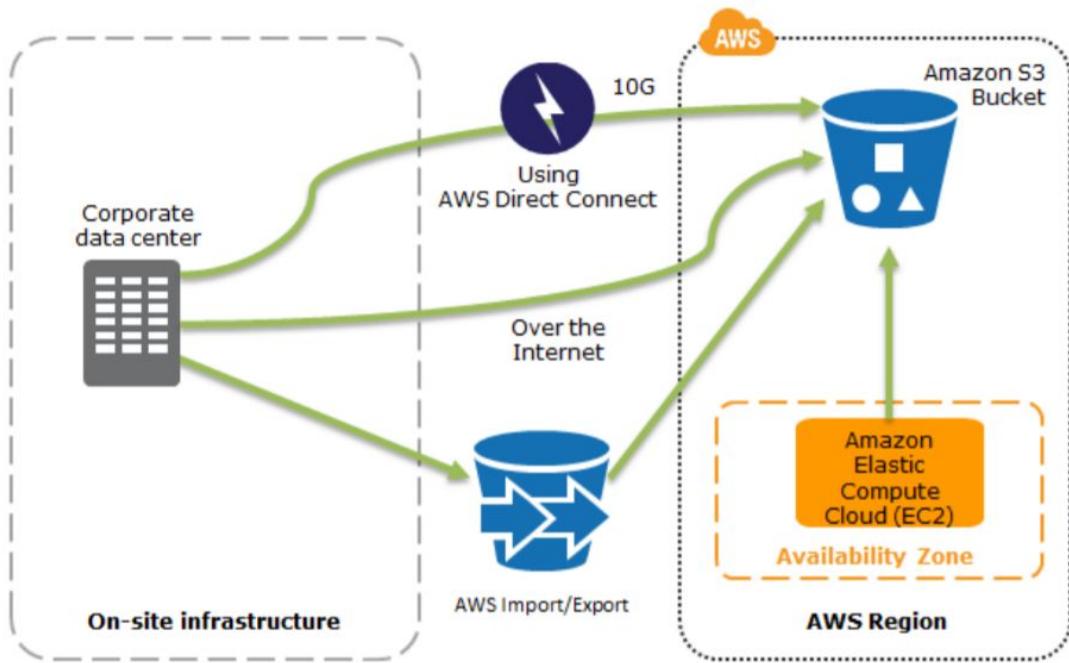


Figure 38. Data Backup Options to Amazon S3 from On-Site Infrastructure or AWS

The following diagram shows how you can quickly restore a system from Amazon S3 backups to Amazon EC2.

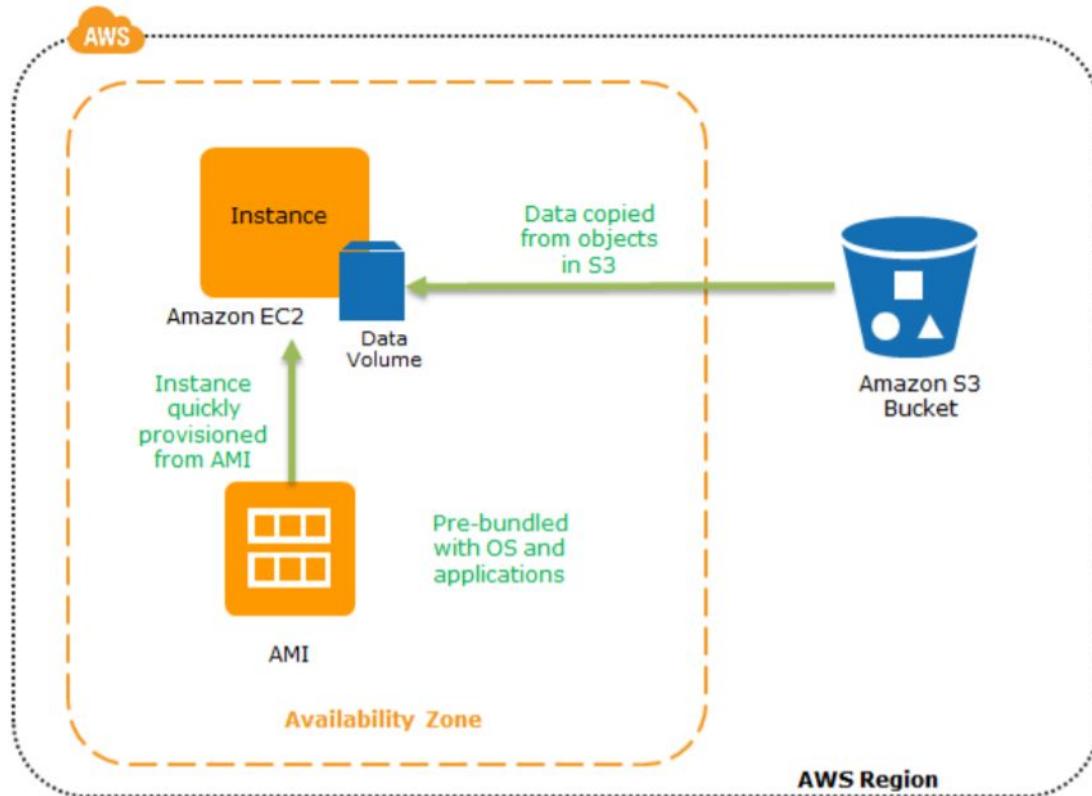


Figure 39. Restoring a System from Amazon S3 Backups to Amazon EC2

Critical steps for backup and restore:

1. Select an appropriate tool or method to back up your data into AWS.
2. Ensure that you have an appropriate retention policy for this data.
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 the restoration of your system.

Pilot Light for Quick Recovery

The term pilot light is often used to describe a DR scenario in which a minimal version of an environment is always running in the cloud. The idea of the pilot light is an analogy that comes from the gas heater. In a gas heater, a small flame that's always on can quickly ignite the entire furnace to heat up a house.

This scenario is similar to a backup-and-restore situation. For example, with AWS you can maintain a pilot light by configuring and running the most critical core elements of your system in AWS. When the time comes for recovery, you can rapidly provision a full-scale production environment around the vital core.

Infrastructure elements for the pilot light itself typically include your database servers, which would replicate data to Amazon EC2 or Amazon RDS. Depending on the system, there might be other critical data outside of the database that needs to be replicated to AWS. This is the crucial core of the system (the pilot light) around which all other infrastructure pieces in AWS (the rest of the furnace) can quickly be provisioned to restore the complete system.

To provision the remainder of the infrastructure to restore business-critical services, you would typically have some preconfigured servers bundled as Amazon Machine Images (AMIs), which are ready to be started up at a moment's notice. When starting recovery, instances from these AMIs come up quickly with their pre-defined role (for example, Web or App Server) within the deployment around the pilot light. From a networking point of view, you have two main options

for provisioning:

- Use Elastic IP addresses, which can be pre-allocated and identified in the preparation phase for DR, and associate them with your instances. Note that for MAC address-based software licensing, you can use elastic network interfaces (ENIs), which have a MAC address that can also be pre-allocated to provision licenses. You can associate these with your instances, just as you would with Elastic IP addresses.
- Use Elastic Load Balancing (ELB) to distribute traffic to multiple instances. You would then update your DNS records to point at your Amazon EC2 instance or point to your load balancer using a CNAME. We recommend this option for traditional web-based applications.

Preparation Phase

The following figure shows the preparation phase, in which you need to have your regularly changing data replicated to the pilot light, the small core around which the full environment will be started in the recovery phase. You're less frequently updated data, such as operating systems and applications, can be periodically updated and stored as AMIs.

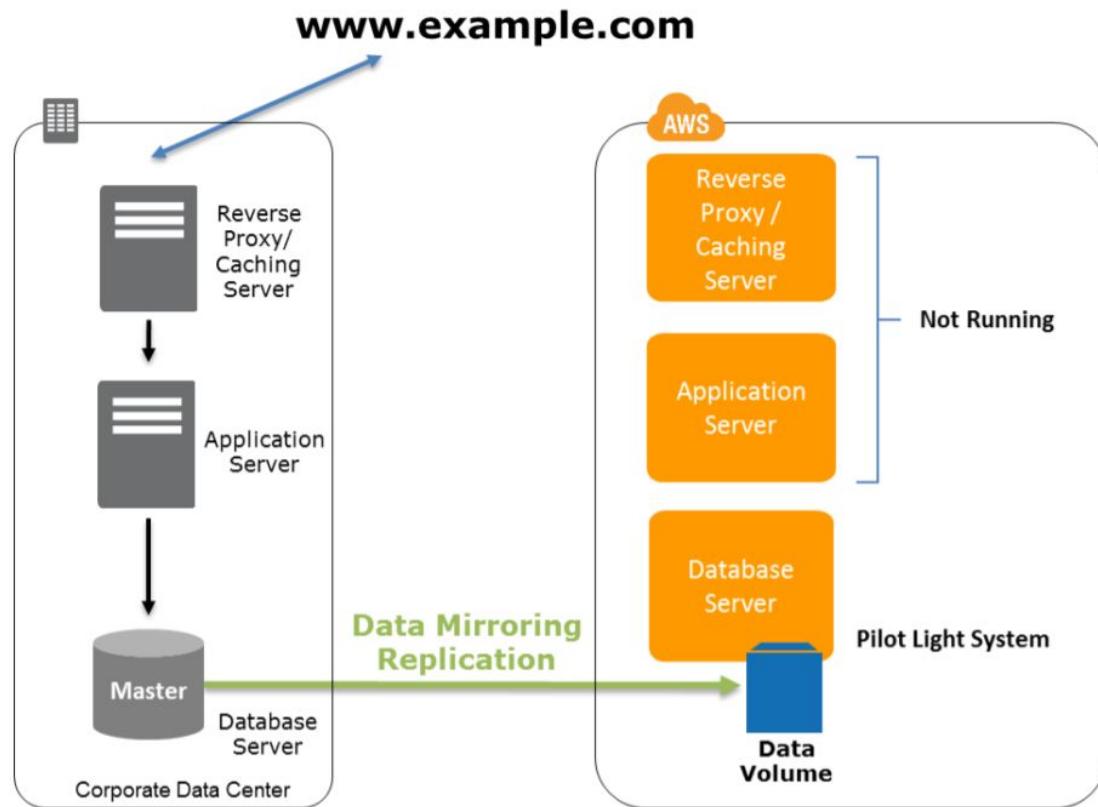


Figure 40. Preparation Phase of the Pilot Light Scenario

Key steps for preparation:

1. Set up Amazon EC2 instances to replicate or mirror data.
2. Ensure that you have all supporting custom software packages available in AWS.
3. Create and maintain AMIs of critical servers where fast recovery 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.*

Recovery Phase

To recover the remainder of the environment around the pilot light, you can start your systems from the AMIs within minutes on the appropriate instance types. For your dynamic data servers, you can resize them to handle production volumes as needed or add capacity accordingly. Horizontal scaling often is the most cost-effective and scalable approach to add capacity to a system. For example, you can add more web servers at peak times. However, you can also choose larger Amazon EC2 instance types, and thus scale vertically for applications such as relational databases. From a networking perspective, any required DNS updates can be done in parallel.

The following figure shows the recovery phase of the pilot light scenario.

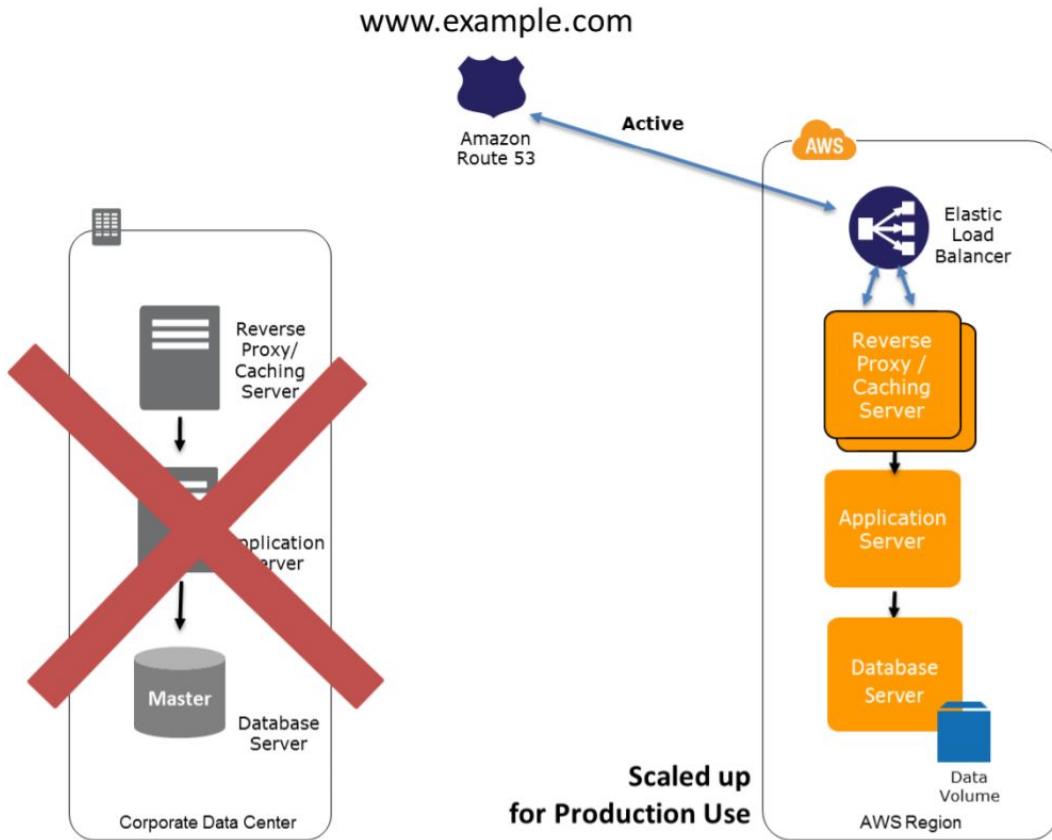


Figure 41. Recovery Phase of Pilot Light Scenario

Key steps for recovery:

1. Start your application Amazon EC2 instances 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.

Warm Standby Solutions

The term warm standby is used to describe a DR scenario in which a scaled-down version of a fully functional environment is always running in the cloud. A warm standby solution extends the pilot light elements and preparation. It further decreases the recovery time because some services are always running. By identifying your business-critical systems, you can fully duplicate these systems on AWS and have them always on.

These servers can be running on a minimum-sized fleet of Amazon EC2 instances on the smallest sizes possible. This solution is not scaled to take a full-production load, but it is fully functional. It can be used for non-production work, such as testing, quality assurance, and internal use.

In a disaster, the system is scaled up quickly to handle the production load. In AWS, this can be done by adding more instances to the load balancer and by resizing the small capacity servers to run on larger Amazon EC2 instance types. As stated in the preceding section, horizontal scaling is preferred over vertical scaling.

Preparation Phase

The following figure shows the preparation phase for a warm standby solution, in which an on-site solution and an AWS solution run side-by-side.

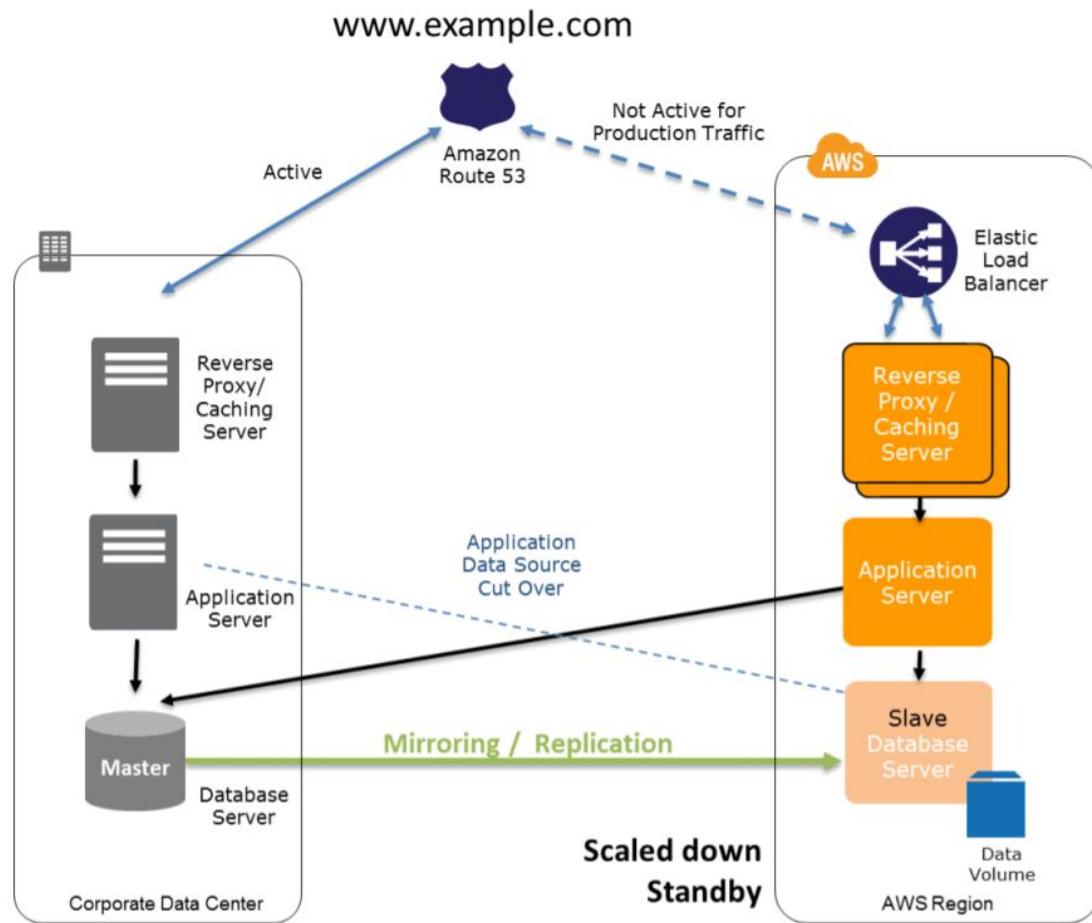


Figure 42. Preparation Phase of the Warm Standby Scenario

Key steps for preparation:

1. Set up Amazon EC2 instances to replicate or mirror data.
2. Create and maintain AMIs.
3. Run your application using a minimal footprint of Amazon EC2 instances or AWS infrastructure.
4. Patch and update software and configuration files in line with your live environment.

Recovery Phase

In the case of failure of the production system, the standby environment will be scaled up for production load, and DNS records will be changed to route all traffic to AWS.

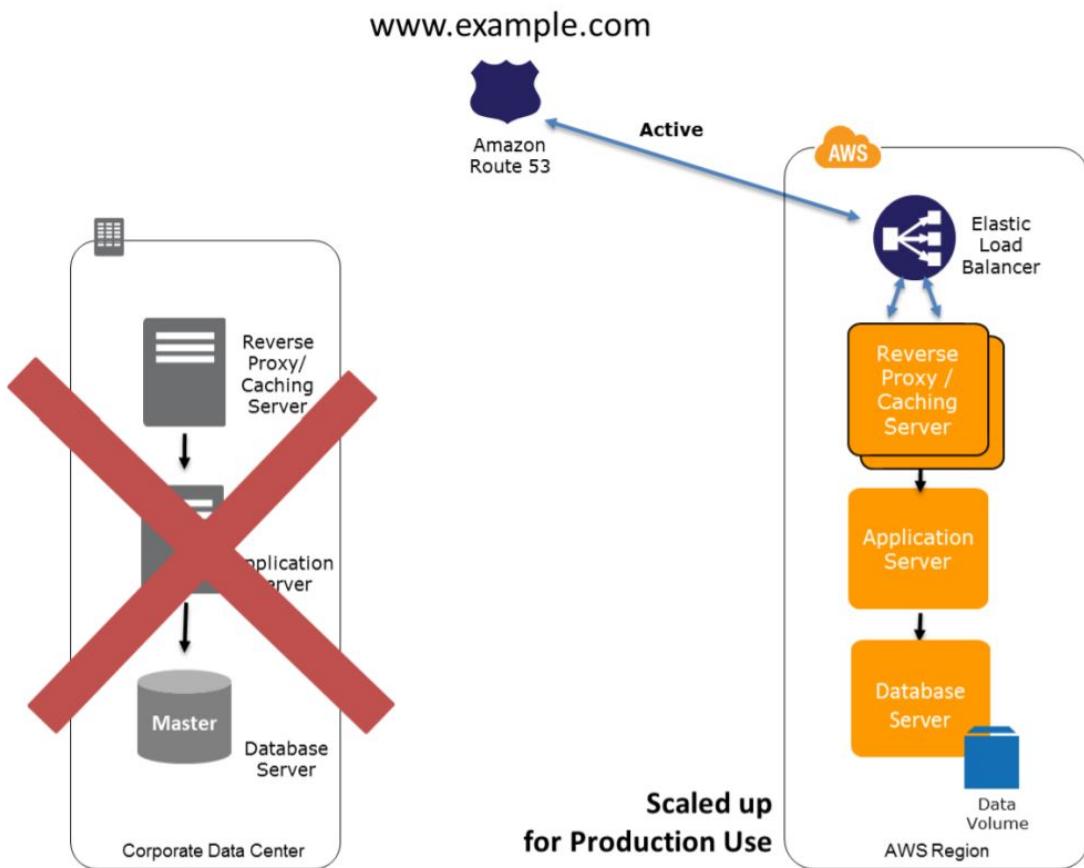


Figure 43. Recovery Phase of the Warm Standby Scenario

Key steps for recovery:

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.

Multi-Site Solution

A multi-site solution runs in AWS as well as on your existing on-site infrastructure, in an active-active configuration. The data replication method that you employ will be determined by the recovery point that you choose. You can use a DNS service that supports weighted routing, such as Amazon Route 53, to route production traffic to different sites that deliver the same application or service. A proportion of traffic will go to your infrastructure in AWS, and the remainder will go to your on-site infrastructure.

In an on-site disaster situation, you can adjust the DNS weighting and send all traffic to the AWS servers. The capacity of the AWS service can be rapidly increased to handle the full production load. You can use Amazon EC2 Auto Scaling to automate this process. You might need some application logic to detect the failure of the primary database services and cut over to the parallel database services running in AWS.

Preparation Phase

The following figure shows how you can use the weighted routing policy of the Amazon Route 53 DNS to route a portion of your traffic to the AWS site. The application on AWS might access data sources in the on-site production system. Data is replicated or mirrored to the AWS infrastructure.

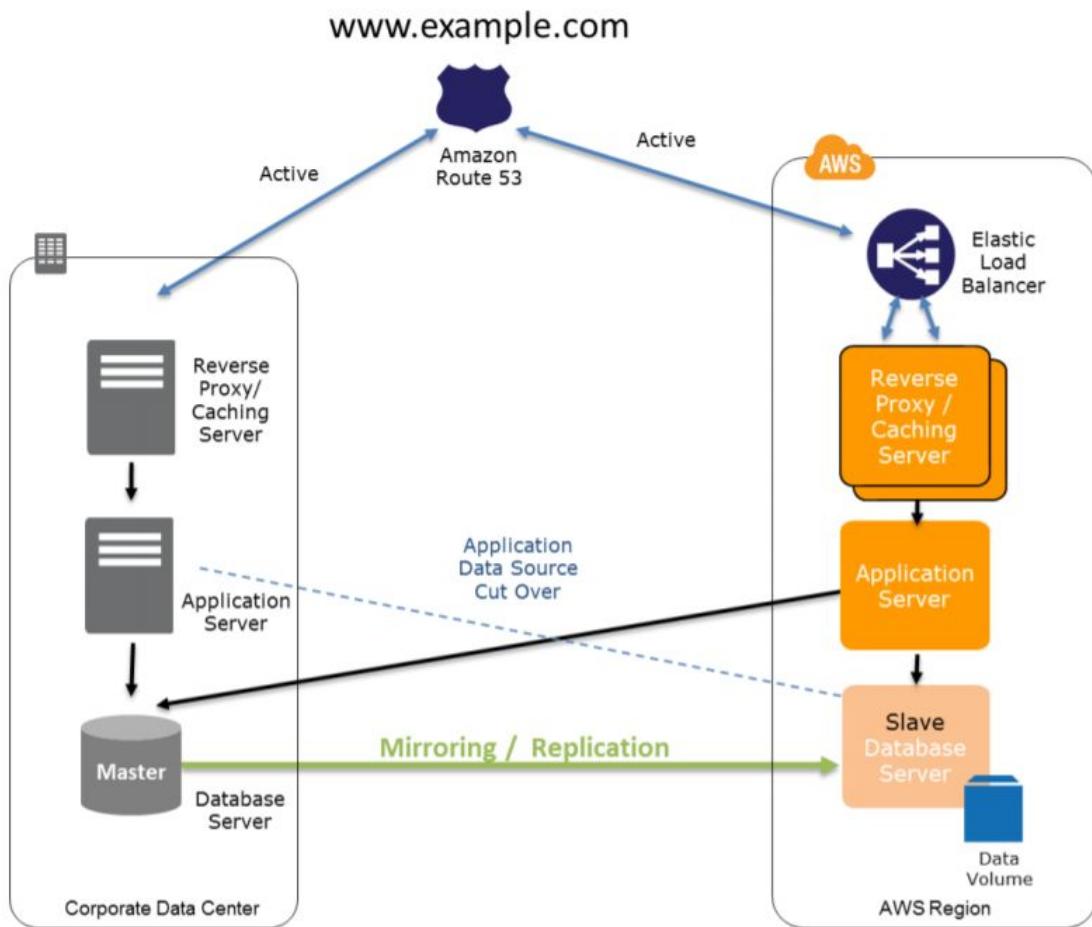


Figure 44. Preparation Phase of the Multi-Site Scenario

Key steps for preparation:

1. Set up your AWS environment to duplicate your production environment.
2. Set up DNS weighting, or similar traffic routing technology, to distribute incoming requests to both sites. Configure automated failover to re-route traffic away from the affected site.

Recovery Phase

The following figure shows the change in traffic routing in the event of an on-site disaster. Traffic is cut over to the AWS infrastructure by updating

DNS, and the AWS infrastructure supports all traffic and supporting data queries.

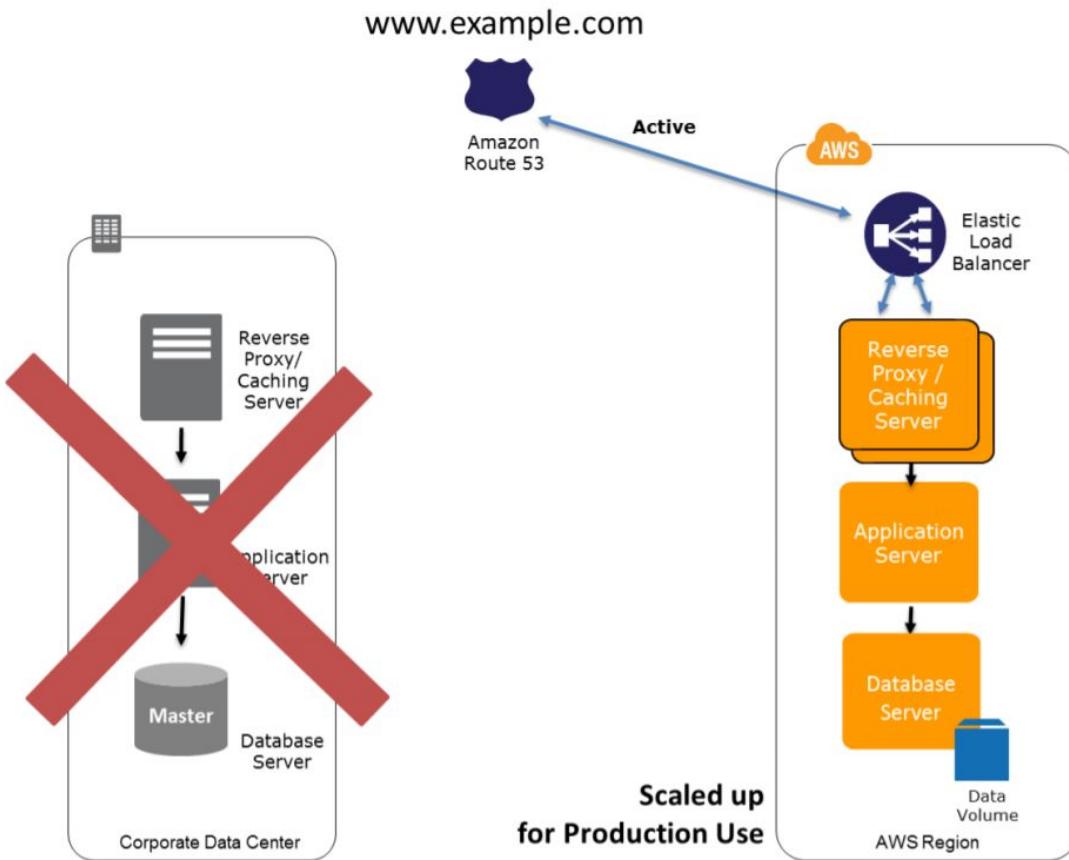


Figure 45. Recovery Phase of the Multi-Site Scenario Involving On-Site and AWS Infrastructure

Key steps for recovery:

1. Either manually or by using DNS failover, change the DNS weighting so that all requests are sent to the AWS site.
2. Have application logic for failover to use the local AWS database servers for all queries.
3. Consider using Auto Scaling to automatically right-size the AWS fleet.

You can further increase the availability of your multi-site solution by designing Multi-AZ architectures.

Failing Back from a Disaster

Once you have restored your primary site to a working state, you will need to regain your regular service, which is often referred to as a “failback.” Depending on your DR strategy, this typically means reversing the flow of data replication so that any data updates received while the primary site was down can be replicated back, without the loss of data. The following steps outline the different fail-back approaches:

Backup and restore

1. Freeze data changes to the DR site.
2. Take a backup.
3. Restore the backup to the primary site.
4. Re-point the users to the primary site.
5. Unfreeze the changes.

Pilot light, warm standby, and multi-site

1. Establish reverse mirroring/replication from the DR site back to the primary site, once the primary site has caught up with the changes.
2. Freeze data changes to the DR site.
3. Re-point the users to the primary site.
4. Unfreeze the changes.

Chapter 6: Security

Security Token Service (STS)

The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users).

Common Scenarios for Temporary Credentials

Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles.

Identity Federation

You can manage your user identities in an external system outside of AWS and grant users who sign in from those systems access to perform AWS tasks and access your AWS resources. IAM supports two types of identity federation.

- **Enterprise identity federation:**

You can authenticate users in your organization's network, and then provide those users access to AWS without creating new AWS identities for them and requiring them to sign in with a separate username and password. This is known as the single sign-on (SSO) approach to temporary access. AWS STS supports open standards like Security Assertion Markup Language (SAML) 2.0, with which you can use Microsoft AD FS to leverage your Microsoft Active Directory. You can also use SAML 2.0 to manage your solution for federating user identities

- **Custom federation broker** – You can use your organization's authentication system to grant access to AWS resources.
- **Federation using SAML 2.0** – You can use your organization's authentication system and SAML to grant access to AWS resources.

- **Web identity federation:**

You can let users sign in using a well-known third-party identity provider such as Login with Amazon, Facebook, Google, or any OpenID Connect (OIDC) 2.0 compatible provider. You can exchange the credentials from that provider for temporary permissions to use resources in your AWS account.

Roles for Cross-account Access

Many organizations maintain more than one AWS account. Using roles and cross-account access, you can define user identities in one account, and use those identities to access AWS resources in other accounts that belong to your organization. This is known as the delegation approach to temporary access.

Roles for Amazon EC2

If you run applications on Amazon EC2 instances and those applications need access to AWS resources, you can provide temporary security credentials to your instances when you launch them. These temporary security credentials are available to all applications that run on the instance, so you don't need to store any long-term credentials on the instance.

Other AWS Services

You can use temporary security credentials to access most AWS services.

Case Scenario

You are hosting a company website on some EC2 web servers in your VPC. Users of the website must log in to the site, which then authenticates against the company's active directory servers that are based on site at the company's headquarters. Your VPC is connected to your company HQ via a secure IPSEC VPN. Once logged in, the user can only have access to their S3 bucket. How do you set this up?

Following are the steps to achieve this:

1. The employee enters their username and password.

2. The application calls an Identity Broker. The broker captures the username and password.
3. The Identity Broker uses the organization's LDAP directory to validate the employee's identity.
4. The Identity Broker calls the new GetFederationToken function using IAM credentials. The call must include an IAM policy and a duration (1 to 36 hours), along with a policy that specifies the permissions to be granted to the temporary security credentials.
5. The Security Token Service confirms that the policy of the IAM user making the call to GetFederationToken gives permission to create new tokens and then returns four values to the application: An access key, a secret access key, a token, and a duration (the token's lifetime)
6. The Identity Broker returns the temporary security credentials to the reporting application.
7. The data storage application uses the temporary security credentials (including the token) to make requests to Amazon S3.
8. Amazon S3 uses IAM to verify that the credentials allow the requested operation on the given S3 bucket and key.
9. IAM provides S3 with the go-ahead to perform the requested operation.

AWS Shared Responsibility Model

The management of the security in the cloud is slightly different from the security in the on-premises data center. Migrating computer systems and data to the cloud requires AWS and customers to work together towards security objectives. The security responsibilities become shared between the user and the cloud service provider. Under this shared responsibility model, AWS is responsible for securing the underlying infrastructure that supports the cloud, and the user is accountable for anything deployed in the cloud or connects to the cloud.

While AWS manages the security of the cloud, security in the cloud is the responsibility of the customer. The control of security implementation for protecting the content, platform, applications systems, and networks, retains with the customer, no different than it would be in an on-site datacenter.

Following is the shared security responsibility model that describes what AWS and the customer is responsible for in this cloud-computing domain.

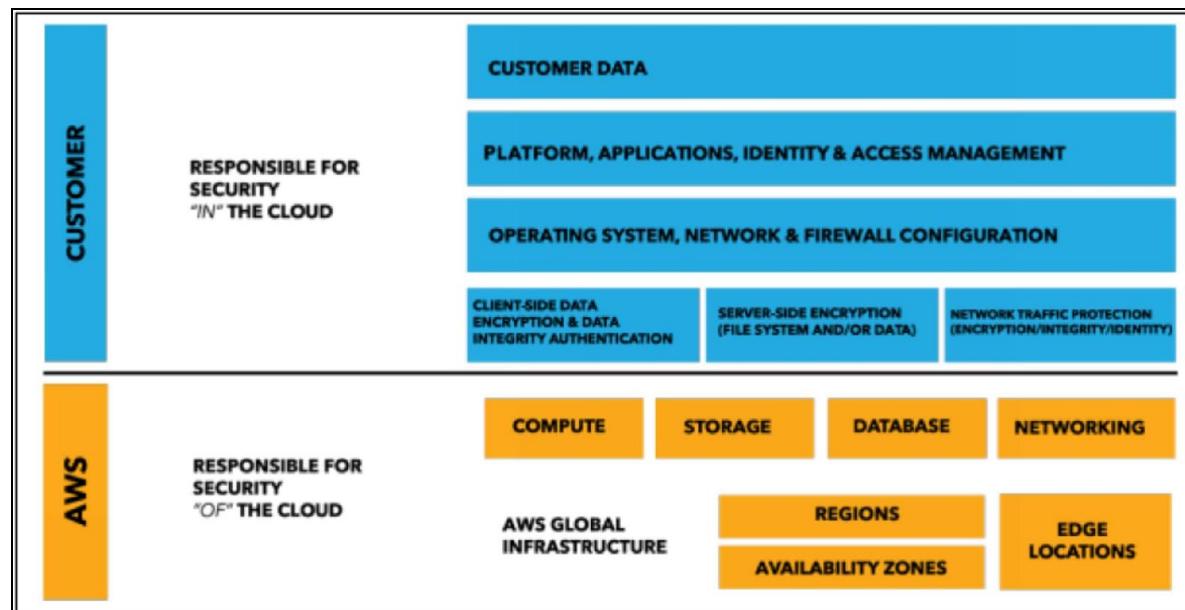


Figure 46. AWS Shared Security Responsibility Model

AWS Security Responsibilities

AWS operates, manages, and controls the components of the host operating system and virtualization layer down to the physical security of the facilities in which the services are operated. Therefore, AWS is responsible for securing their whole global infrastructure including foundational compute, storage, networking and database services, as well as higher-level services.

In addition to the above, AWS is also responsible for the security configuration of its products that are considered managed services. Examples of these types of services include Amazon DynamoDB, Amazon RDS, Amazon Redshift, Amazon Elastic MapReduce, Amazon WorkSpaces, and several other services. For these services, AWS handles basic security tasks like a guest operating system (OS) and database patching, firewall configuration, and disaster recovery.

Customer Security Responsibilities

As AWS customers retain control over their data, they consequently hold the responsibilities relating to that content as part of the AWS “shared responsibility” model. They must protect the confidentiality, integrity, and availability of their data in the cloud. They undertake responsibility for the management of their operating system (including updates and security patches), other associated application software, as well as the configuration of the AWS-provided security group firewall. AWS provides a range of security services and features that AWS customers can use to secure their assets.

The responsibilities and the amount of security configuration work the customer needs to take care of depends on the type of AWS services selected and the sensitivity of the data. If the services fall under the category of Infrastructure as a service (IaaS), such as Amazon EC2 and Amazon VPC, then all the necessary security configuration and management tasks need to be handled entirely by the customer. Whereas for AWS managed services such as Amazon RDS or Amazon Redshift, there is no need to worry about the configuration work as AWS handles it for you.

Irrespective of the AWS services used, you should always configure security by using AWS Account credentials and setting up individual user accounts with Amazon Identity and Access Management (IAM) so that each

of the users has their credentials. Other security features such as using multi-factor authentication (MFA) with each account, requiring the use of SSL/TLS to communicate with your AWS resources, setting up API/user activity logging with AWS CloudTrail, leveraging technology such as host-based firewalls, host-based intrusion detection/ prevention, and encryption are some of the AWS assistive tools provided to the customers to enhance security.

AWS Global Infrastructure Security

The AWS global infrastructure is one of the most flexible and secure cloud computing platform present today. It is designed to offer an exceptionally scalable, highly reliable platform that facilitates customers in deploying applications and data swiftly and securely. The infrastructure includes the services, network, hardware, and operational software (e.g., host OS, virtualization software, etc.) that supports the provisioning and use of computing resources.

AWS runs under a shared security responsibility model, where AWS is in-charge of the cloud infrastructure security and the user is responsible for securing workloads deployed in the cloud. This provides the flexibility and agility to implement appropriate security controls such as strongly restricting access to locations that process sensitive data, or setting up less rigid controls for data admissible to the public.

The AWS global infrastructure utilizes the security best practices along with a range of security compliance standards. AWS monitors and protects the underlying infrastructure 24x7 using redundant and layered controls, continuous validation and testing, and extensive automation. AWS ensures the replication of these controls in each new data center or service.

AWS Compliance Program

AWS computing environments are continuously audited, with certifications from accreditation bodies across geographies and verticals, including ISO 27001, FedRAMP, DoD CSM, and PCI DSS. By operating in an accredited environment, customers reduce the scope and cost of audits they need to perform. AWS continuously undergoes assessments of its underlying infrastructure including the physical and environmental security of its hardware and data centers so customers can take advantage of those certifications and merely inherit those controls. Following are the programs that AWS have regarding Compliance. They are divided into three areas:

	Certifications/Attestation
	<p>DoD SRG - FedRAMP - FIPS - IRAP - ISO 9001 - ISO 27001 ISO 27017 - ISO 27018 - MTCS - PCI DSS Level 1 - SEC Rule 17-a-4(f) - SOC 1 - SOC 2 - SOC 3</p>
	Laws, Regulations, and Privacy
	<p>EAR - EU Model Clauses - FERPA GLBA - HIPAA - HITECH - IRS 1075 - ITAR - My Number Act (Japan) - U.K. DPA 1988 VPAT/ Section 508 - EU Data Protection Directive Privacy Act (Australia) - PDPA - 2010 (Malaysia) - PDPA - 2012 (Singapore)</p>

Alignments/Frameworks

CJIS - CLIA - CMS EDGE - CMSR - CSA - FDA - FedRAMP TIC - FISC - FISMA - G-Cloud - GxP (FDA CFR 21 Part 11)
 IT Grundschutz - MITA 3.0 - MPAA - NERC - NIST - PHR
 UK Cyber Essentials

Figure 47. AWS Assurance Programs

Certifications / Attestations:

A third-party, independent auditor assesses compliance certifications and attestations and results in a certification, audit report, or attestation of compliance. Major ones that you need to be aware of for this course are ISO 27001, PCI DSS Level 1, SOC 1, SOC 2, and SOC 3.

- **ISO 27001** - ISO 27001 is a security management standard that specifies security management best practices and comprehensive security controls

- ***PCI DSS Level 1*** - The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary information security standard administered by the PCI Security Standards Council. All entities that deal with online payments using credit cards that involve storing, processing or transmitting cardholder's data, need to be PCI DSS Level 1 compliant.
- ***SOC*** - AWS System & Organization Control (SOC) Reports are independent third-party examination reports that demonstrate how AWS achieves key compliance controls and objectives. AWS platform is compatible with SOC 1, SOC 2, and SOC 3

Laws, Regulations, and Privacy:

AWS customers remain responsible for complying with applicable compliance laws and regulations. The main one you should be aware of is HIPAA.

- ***HIPAA*** - U.S. Health Insurance Portability and Accountability Act (HIPAA) is a set of federal standards intended to protect the security and privacy of PHI Protected Health Information (PHI). AWS enables covered entities and their business associates subject to HIPAA to leverage the secure AWS environment for processing, maintaining, and storing protected health information.

Alignments / Frameworks:

Compliance alignments and frameworks include published security or compliance requirements for a specific purpose, such as a particular industry or function. The one to be looked at is G-Cloud [UK].

G-Cloud [UK] - The G-Cloud framework is an agreement between the UK government and cloud-based service providers. The structure enables public bodies to procure commodity-based, pay-as-you-go cloud services on

government-approved short-term contracts. So to host on AWS, they need to meet the G-Cloud [UK] requirement.

Physical and Environmental Security

AWS's data centers are state of the art, utilizing innovative architectural and engineering approaches. AWS data centers are housed in nondescript facilities. Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video surveillance, intrusion detection systems, and other electronic means.

Storage Device Decommissioning

When a storage device has reached the end of its useful life, AWS procedures include a decommissioning process that is designed to prevent customer data from being exposed to unauthorized individuals. AWS uses the techniques detailed in NIST 800-88 ("Guidelines for Media Sanitization") as part of the decommissioning process. All decommissioned magnetic storage devices are degaussed and physically destroyed by industry-standard practices.

Network Security

The AWS network has been architected to permit you to select the level of security and resiliency appropriate for your workload. To enable you to build geographically dispersed, fault-tolerant web architectures with cloud resources, AWS has implemented a world-class network infrastructure that is carefully monitored and managed.

Transmission Protection

You can connect to an AWS access point via HTTP or HTTPS using Secure Sockets Layer (SSL), a cryptographic protocol that is designed to protect against eavesdropping, tampering, and message forgery.

For customers who require additional layers of network security, AWS offers the Amazon Virtual Private Cloud (VPC), which provides a private subnet within the AWS cloud, and the ability to use an IPsec Virtual Private Network (VPN) device to provide an encrypted tunnel between the Amazon VPC and your data center.

Amazon Corporate Segregation

Logically, the AWS Production network is segregated from the Amazon Corporate network using a complex set of network security/segregation devices.

Network Monitoring and Protection

AWS utilizes a wide variety of automated monitoring systems to provide a high level of service performance and availability. These tools monitor server and network usage, port scanning activities, application usage, and unauthorized intrusion attempts. It protects from the following attacks:

- Denial of service (DDoS)
- Man in the middle (MITM)

- IP Spoofing
- Port Scanning
- Packet Sniffing by other tenants

For protection against IP Spoofing, the AWS-controlled, host-based firewall infrastructure will not permit an instance to send traffic with a source IP or MAC address other than its own.

Unauthorized port scans by Amazon EC2 customers are a violation of the AWS Acceptable Use Policy. You may request permission to conduct vulnerability scans as required to meet your specific compliance requirement. These scans must be limited to your instances and must not violate the AWS Acceptable Use Policy. You must request a vulnerability scan in advance.

AWS Account Security Features

AWS provides a variety of tools and features that you can use to keep your AWS Account and resources safe from unauthorized use. This includes credentials for access control, HTTPS endpoints for encrypted data transmission, the creation of separate IAM user accounts, user activity logging for security monitoring, and Trusted Advisor security checks.

AWS Credentials

To help ensure that only authorized users and processes access your AWS Account and resources, AWS uses several types of credentials for authentication.

Credential Type	Use	Description
Passwords	AWS root account or IAM user account login to the AWS Management Console	A string of characters used to log into your AWS account or IAM account. AWS passwords must be a minimum of 6 characters and may be up to 128 characters.
Multi-Factor Authentication (MFA)	AWS root account or IAM user account login to the AWS Management Console	A six-digit single-use code that is required in addition to your password to log in to your AWS Account or IAM user account.
Access Keys	Digitally signed requests to AWS APIs (using the AWS SDK, CLI, or REST/Query APIs)	It includes an access key ID and a secret access key to sign programmatic requests that you make to AWS digitally.
Key Pairs	SSH login to EC2 instances CloudFront signed URLs	A key pair is required to connect to an EC2 instance launched from a public AMI. The keys

		that Amazon EC2 uses are 1024-bit SSH-2 RSA keys. You can have a key pair generated automatically for you when you launch the instance, or you can upload your own.
X.509 Certificates	Digitally signed SOAP requests to AWS APIs SSL server certificates for HTTPS	X.509 certificates are only used to sign SOAP-based requests (currently used only with Amazon S3). You can have AWS create an X.509 certificate and private key that you can download, or you can upload your certificate by using the Security Credentials page.

Table 19. AWS Authentication Credentials

AWS Trusted Advisor Security Checks

Trusted Advisor inspects your AWS environment and makes recommendations when opportunities may exist to save money, improve system performance, or close security gaps. It provides alerts on several of the most common security misconfigurations that can occur, including leaving specific ports open that make you vulnerable to hacking and unauthorized access, neglecting to create IAM accounts for your internal users, allowing public access to Amazon S3 buckets, not turning on user activity logging (AWS CloudTrail), or not using MFA on your root AWS Account.

Amazon EC2 Security

Amazon Elastic Compute Cloud (EC2) is a key component in Amazon's Infrastructure as a Service (IaaS), providing resizable computing capacity using server instances in AWS's data centers.

Multiple Levels of Security

Security within Amazon EC2 is provided on various levels: the operating system (OS) of the host platform, the virtual instance OS or guest OS, a firewall, and signed API calls.

The Hypervisor

Amazon EC2 currently utilizes a highly customized version of the Xen hypervisor, taking advantage of paravirtualization (in the case of Linux guests). Because para-virtualized guests rely on the hypervisor to provide support for operations that generally require privileged access, the guest OS has no elevated access to the CPU.

Instance Isolation

Different instances running on the same physical machine are isolated from each other via the Xen hypervisor. Amazon is active in the Xen community, which provides awareness of the latest developments. Also, the AWS firewall resides within the hypervisor layer, between the physical network interface and the instance's virtual interface. All packets must pass through this layer. Thus an instance's neighbors have no more access to that instance than any other host on the Internet and can be treated as if they are on separate physical hosts. The physical RAM is separated using similar mechanisms.

Customer instances have no access to raw disk devices but instead, are presented with virtualized disks. The AWS proprietary disk virtualization

layer automatically resets every block of storage used by the customer, so that one customer's data is never unintentionally exposed to another. Also, memory allocated to guests is scrubbed (set to zero) by the hypervisor when it is unallocated to a guest. The memory is not returned to the pool of free memory available for new allocations until the memory scrubbing is complete.

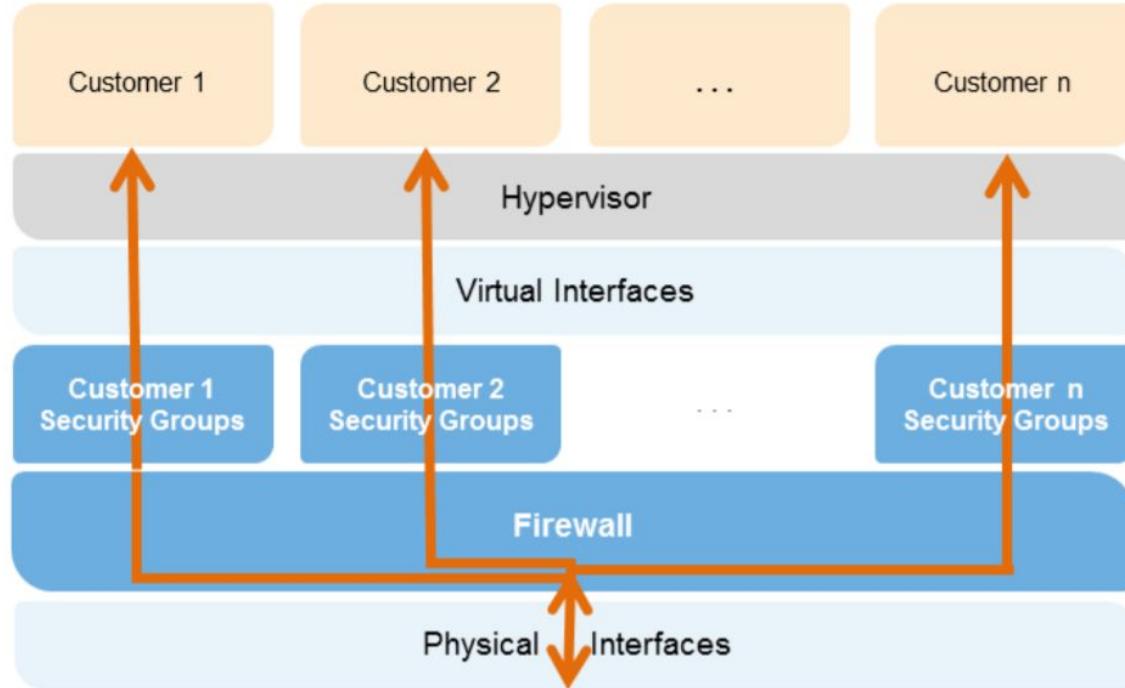


Figure 48. Amazon EC2 Multiple Layers of Security

Guest Operating System

You completely control virtual instances, the customer. You have full root access or administrative control over accounts, services, and applications. AWS does not have any access rights to your instances or the guest OS.

Firewall

Amazon EC2 provides a complete firewall solution; this mandatory inbound firewall is configured in a default deny-all mode, and Amazon EC2 customers must explicitly open the ports needed to allow incoming traffic.

Amazon EBS Security

Encryption of sensitive data is generally a good security practice, and AWS provides the ability to encrypt EBS volumes and their snapshots with AES-256. The encryption occurs on the servers that host the EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. To be able to do this efficiently and with low latency, the EBS encryption feature is only available on EC2's more powerful instance types (e.g., M3, C3, R3, G2).

Amazon ELB Security

Amazon Elastic Load Balancer provides several security benefits:

- Takes over the encryption and decryption work from the Amazon EC2 instances and manages it centrally on the load balancer
- Offers clients a single point of contact, and can also serve as the first line of defense against attacks on your network
- When used in an Amazon VPC, supports the creation and management of security groups associated with your Elastic Load Balancing to provide additional networking and security options
- Promotes end-to-end traffic encryption using TLS (previously SSL) on those networks that use secure HTTP (HTTPS) connections. The TLS server certificate used to terminate client connections can be managed centrally on the load balancer, rather than on every individual instance.

Amazon Elastic Load Balancing allows you to identify the originating IP address of a client connecting to your servers, whether you're using HTTPS or TCP load balancing.

AWS Direct Connect Security

With Direct Connect, you bypass Internet service providers in your network path. You can procure rack space within the facility housing the AWS Direct Connect location and deploy your equipment nearby. Once implemented, you can connect this equipment to AWS Direct Connect using a cross-connect.

The dedicated connection can be partitioned into multiple virtual interfaces using industry standard 802.1q VLANs which allows you to use the same connection to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within an Amazon VPC using private IP space while maintaining network separation between the public and private environments.

Auditing on AWS

You should periodically audit your security configuration to make sure it meets your current business needs. An audit gives you an opportunity to remove unneeded IAM users, roles, groups, and policies, and to make sure that your users and software have only the permissions that are required. Your organization may undergo an audit. It could be for PCI Compliance, ISO 27001, SOC, and others. There is a level of shared responsibility in regards to inspections:

- AWS provides their annual certifications and reports (ISO 27001, PCI-DSS certificates). Amazon is responsible for the global infrastructure including all hardware, datacenters, physical security, and others.
- Customer provides everything they have put on AWS, such as EC2 instances, RDS instances, Applications, Assets in S3. Essentially the organizations AWS assets (this can include the data itself)

Chapter 7: Networking

What is DNS?

Browsing the internet requires DNS service to translate Domain name into their IP addresses. It converts human-friendly domain names such as *ipspecialist.net* into an Internet Protocol (IP) address. These IP addresses are used by the computers and networking devices to identify each other on the network.

The Domain Name System (DNS) is the phonebook of the Internet.

- Humans access information online through domain names, like example.com.
- Web browsers interact through Internet Protocol (IP) addresses.
- DNS translates domain names to IP addresses so browsers can load Internet resources.

The Domain Name System delegates the responsibility of assigning domain names and mapping those names to Internet resources by designating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their allocated namespace to other name servers. This mechanism provides distributed, fault-tolerant service and was designed to avoid a single large central database.

The Internet maintains two principal namespaces, the domain name hierarchy, and the Internet Protocol (IP) address spaces. The Domain Name System supports the domain name hierarchy and provides translation services between it and the address spaces. Internet name servers and a communication protocol implement the Domain Name System. A DNS name server is a server that stores the DNS records for a domain; a DNS name server responds with answers to queries against its database.

Internet Protocol (IP)

Two versions of the Internet Protocol are in frequent use in the Internet today, IPv4 and IPv6:

- An IPv4 address has a size of 32 bits, which limits the address space to 4294967296 (2³²) addresses. Of this number, reserved addresses are for special purposes such as private networks (~18 million addresses) and multicast addressing (~270 million addresses).
- In IPv6, the address size was increased from 32 bits in IPv4 to 128 bits or 16 octets, thus providing up to 2¹²⁸ (approximately 3.403×10³⁸) addresses. This is deemed sufficient for the foreseeable future.

Top Level Domain (TLD)

A domain name consists of one or more parts, technically called labels, that are conventionally concatenated, and delimited by dots, such as example.com. The right-most label conveys the top-level domain; for example, the domain name www.example.com belongs to the top-level domain com.

The hierarchy of domains descends from right to left; each label to the left specifies a subdivision or subdomain of the domain to the right. For example, the label example specifies a subdomain of the com domain, and www is a subdomain of example.com. This tree of subdivisions may have up to 127 levels.

These top-level domain names are controlled by the Internet Assigned Numbers Authority (IANA) in a root zone database, which is mostly a database of all available high-level domains.

Domain Name Registration

The right to use a domain name is delegated by domain name registrars who are accredited by the Internet Corporation for Assigned Names and Numbers (ICANN) or other organizations such as OpenNIC, that is charged with overseeing the name and number systems of the Internet.

In addition to ICANN, each top-level domain (TLD) is maintained and serviced technically by an administrative organization, operating a registry.

A registry is responsible for administering the database of names within its authoritative zone, although the term is most often used for TLDs. A registrant is a person or organization who asked for domain registration. The registry receives registration information from each domain name registrar, which is authorized (accredited) to assign names in the corresponding zone and publishes the information using the WHOIS protocol.

DNS Records

The most common types of records stored in the DNS database are for Start of Authority (SOA), name servers (NS), IP addresses (A Records), domain name aliases (CNAME).

Start of Authority (SOA) – A start of authority (SOA) record is information stored in a domain name system (DNS) zone about that zone and other DNS records. A DNS zone is the part of a domain for which an individual DNS server is responsible. Each zone contains a single SOA record. The SOA record stores information about:

- The name of the server that supplied the data for the DNS zone.
- DNS zone administrator information.
- Current data file version information.
- The number of seconds a secondary name server should wait before checking for updates.
- The number of seconds a secondary name server should wait before retrying a failed zone transfer.
- The maximum number of seconds that a secondary name server can use data before it must either be refreshed or expire.
- The default number of seconds for the time-to-live file on resource records.

Name Servers (NS) Records – NS stands for Name Server records and is used by Top Level Domain servers to direct traffic to the Content DNS server that contains the authoritative DNS records.

A Records – An ‘A’ record is the fundamental type of DNS record, and the ‘A’ basically stands for ‘Address.’ The A record is used by a computer to translate the name of the domain to the IP address. For example, <http://www.ipspecialist.net> might point to <http://123.10.10.80>.

CNAMEs – A Canonical Name (CName) can be used to resolve one domain name to another. For example, you may have a mobile website with the domain name ***http://m.ipspecialist.net*** when the users browse to your domain name on their mobile devices. You may also want the name <http://mobile.ipspecialist.net> to resolve to this same address.

Time to Live (TTL)

The duration of caching a DNS record on either the Resolving Server or the user's own local PC is the ‘Time To Live’ (TTL) in seconds. The lower the Time To Live (TTL), the faster the changes to DNS records take to propagate throughout the internet.

Alias Records

Alias records are used to map resource record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions, or S3 buckets configured as websites. Alias records work like a CNAME record in that you can map one DNS name (www.example.com) to another ‘target’ DNS name (elb123.elb.amazonaws.com). The key difference is that a CNAME cannot be used for naked domain names (zone apex). You cannot have a CNAME for ‘<http://ipspecialist.net>'; it must be either an ‘A Record’ or an ‘Alias Record.’

Alias Resource recordsets can save your time because Amazon Route 53 automatically recognizes changes in the record sets that the alias resource record set refers to. For example, suppose an alias resource record set for ‘ipspecialist.com’ points to an ELB load balancer at ‘lb1-123.us-east-1.elb.amazonaws.com’. If the IP address of the load balancer changes, Amazon Route 53 automatically reflects those changes in DNS answers for ‘ipspecialist.com’ without any changes to the hosted zone that contains resource record sets for ‘ipspecialist.com.’

Introduction to Route 53

Amazon Route 53 provides highly available and scalable cloud DNS web service that effectively connects user requests to infrastructure running in AWS such as EC2 instances, Elastic Load Balancers, or Amazon S3 buckets. It can also be used to route users to infrastructure outside of AWS. DNS (Domain Name System) is a globally distributed service that translates human-readable domain names like `www.example.com` to the numeric machine-readable IP addresses like `192.0.2.1` that computers use to connect to each other.

Amazon Route 53 traffic flow makes it easy for you to manage traffic globally through a variety of routing types, including latency-based routing, Geo DNS, and weighted round robin, all of which can be combined with DNS Failover to enable a variety of low-latency, fault-tolerant architectures.

You can use Amazon Route 53 to register new domains, transfer existing domains, route traffic for your domains to your AWS and external resources, and monitor the health of your resources.

DNS Management

If you already have a domain name, such as `example.com`, Route 53 can tell the Domain Name System (DNS) where on the internet to find web servers, mail servers, and other resources for your domain.

Traffic Management

Route 53 traffic flow provides a visual tool that you can use to create and update sophisticated routing policies to route end users to multiple endpoints for your application, whether in a single AWS Region or distributed around the globe.

Availability Monitoring

Route 53 can monitor the health and performance of your application as well as your web servers and other resources. Route 53 can also redirect traffic to healthy resources and independently monitor the health of your application and its endpoints.

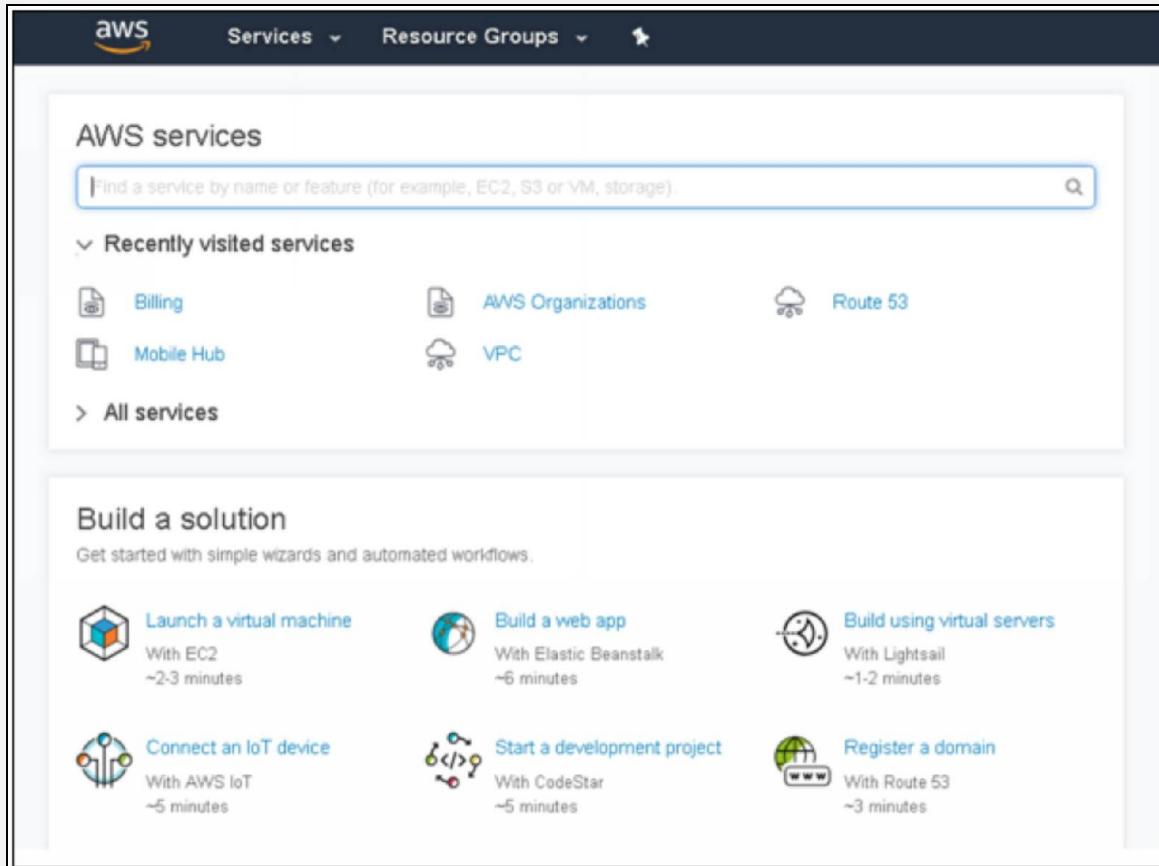
Domain Registration

If you need a domain name, you can find an available name and register it by using Route 53. Amazon Route 53 automatically configures DNS settings for your domains. You can also make Route 53 the registrar for existing domains that you registered with other registrars.

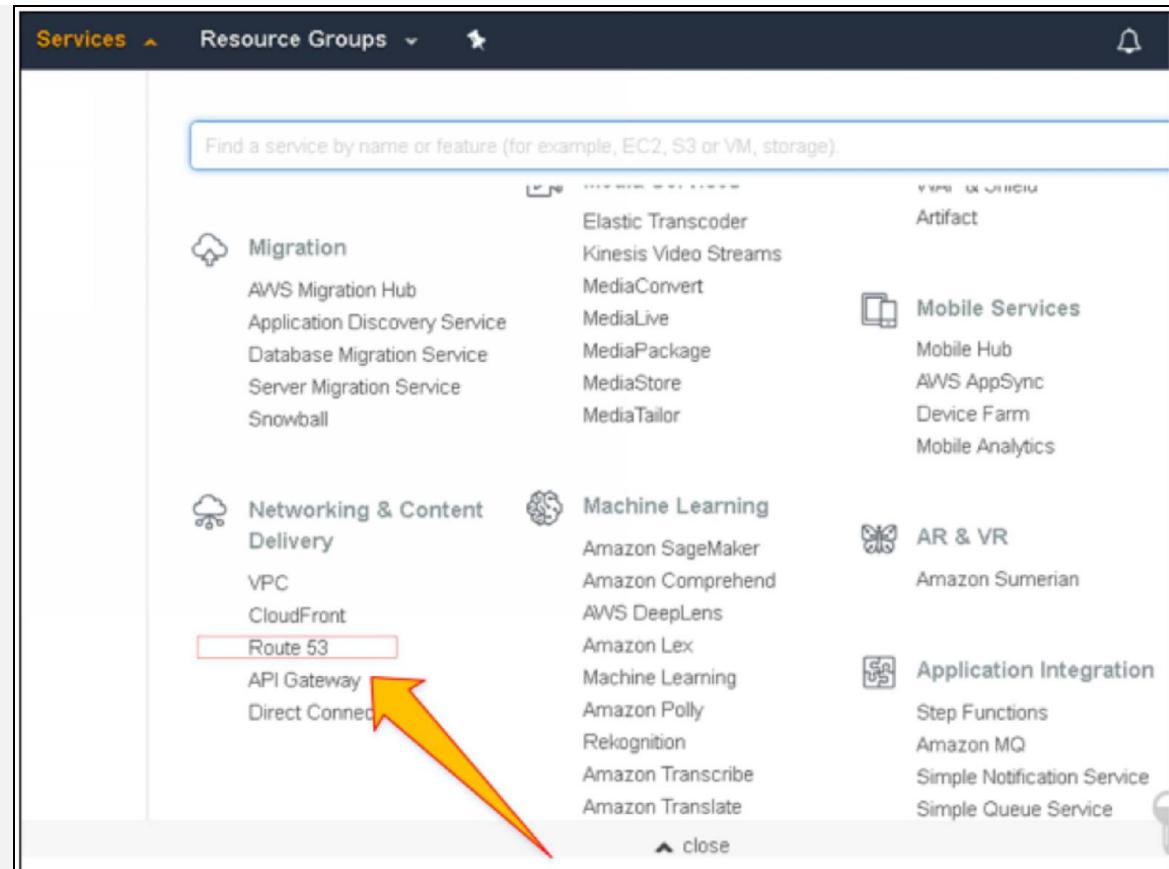
Lab 7.1: Register a domain name – Route 53

In this lab, we are going to register a domain name using route 53 services provided by Amazon web services

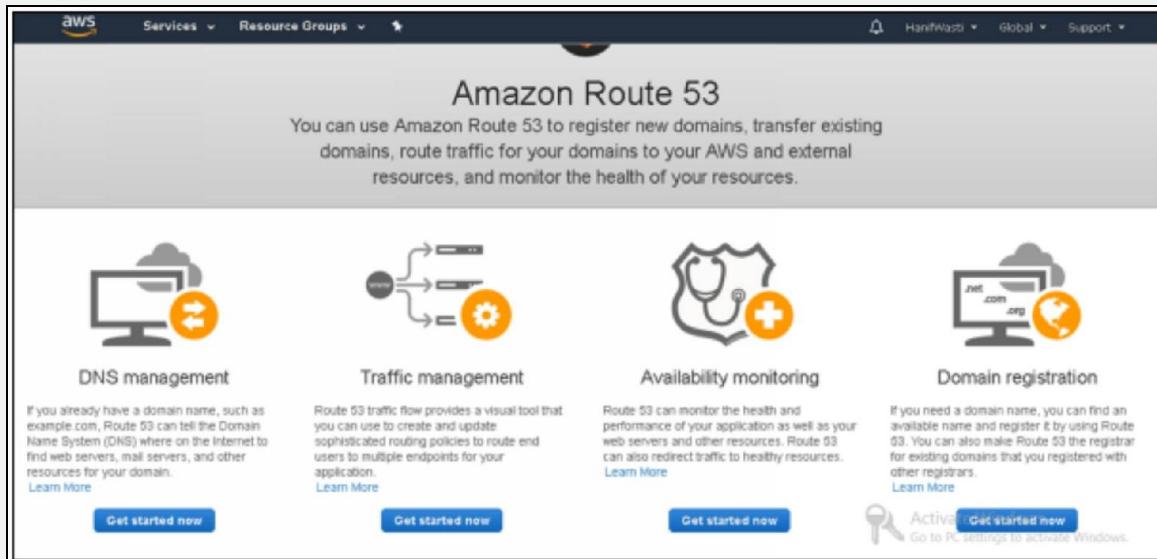
Log in to the AWS management console and click “Services.”



Under Networking & Content Delivery, Click “Route5”

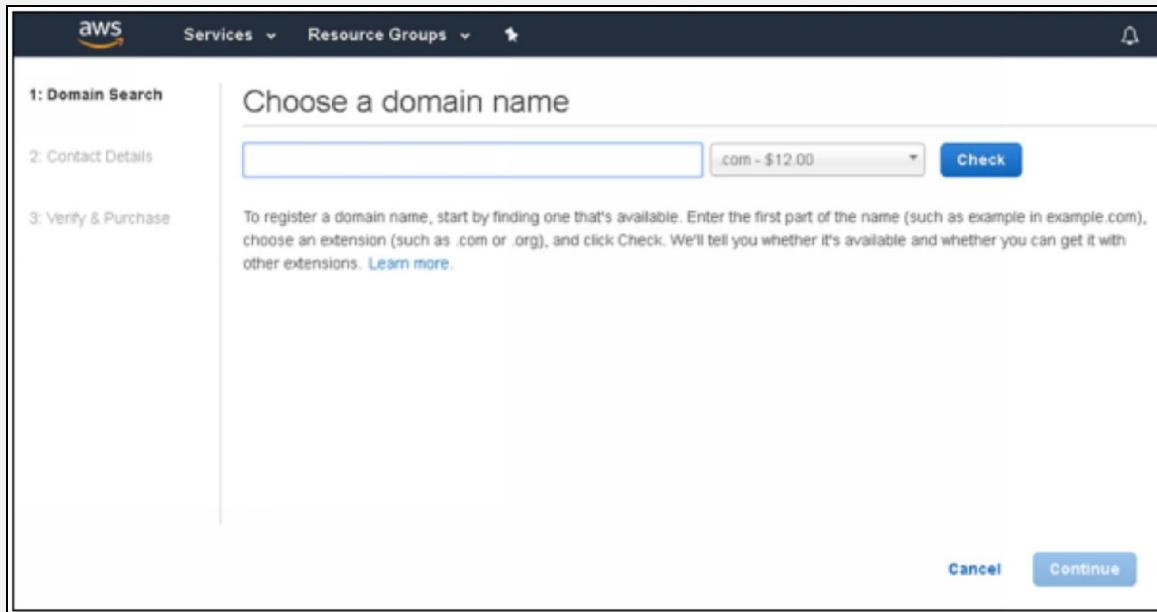


If you are using Route53 for the first time, and observe the following dashboard



Under Domain registration, click “Get started now”

Choose a domain name you want to get registered and check availability



Click Continue after selecting the name of the domain

Services < Services Resource Groups			
			HanifWasti
ips-industries.com	✓ Available	\$12.00	Add to cart
ips-labs.info	✓ Available	\$12.00	Add to cart
ips-labs.net	✓ Available	\$11.00	Add to cart
ips-labs.ninja	✓ Available	\$18.00	Add to cart
ips-labs.org	✓ Available	\$12.00	Add to cart
ips-labs.tv	✓ Available	\$32.00	Add to cart
ips-systems.net	✓ Available	\$11.00	Add to cart
ips-technologies.com	✓ Available	\$12.00	Add to cart
ipslabsonline.com	✓ Available	\$12.00	Add to cart
theipslabs.com	✓ Available	\$12.00	Add to cart

Cancel Continue  Activ Go to F

English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Provide contact details to complete domain registration process

1: Domain Search
2: Contact Details
3: Verify & Purchase

Contact Details for Your 1 Domain

Enter the details for your Registrant, Administrative and Technical contacts below. All fields are required unless specified otherwise. [Learn more](#).

My Registrant, Administrative and Technical Contacts are all the same: Yes No

Registrant Contact

Contact Type <small>?</small>	<input type="text" value="Person"/>
First Name	<input type="text"/>
Last Name	<input type="text"/>
Organization <small>?</small>	<input type="text" value="Not applicable"/>
Email	<input type="text"/>
Phone	+ <input type="text" value="1"/> - <input type="text" value="3115550188"/>
Enter country calling code and phone number	

[Feedback](#) English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Click continue after filling the all the required fields

The screenshot shows a configuration interface for 'Resource Groups'. At the top, there are 'Services' and 'Resource Groups' dropdown menus, and a bell icon. Below these are several input fields:

- Country:** A dropdown menu.
- State:** A dropdown menu with the option 'State not required'.
- City:** An empty input field.
- Postal/Zip Code:** An input field with the placeholder 'Optional'.
- UK Contact Type:** A dropdown menu with the placeholder 'Select an Option'.
- UK Company Number:** An input field with the placeholder 'Not applicable'.

Below these fields, there is a section titled 'Privacy Protection' with the note: 'When the contact type is Company:'.

- A bullet point states: 'Privacy protection is not available for .co.uk domains.'
- Two radio buttons are present: 'Enable' (unchecked) and 'Disable' (checked).

At the bottom of the form are three buttons: 'Cancel', 'Back', and a blue 'Continue' button.

At the very bottom of the screen, there is a footer bar with the text 'English (US)' and '© 2008 - 2018, Amazon Web Services, Inc. or its affiliates'.

Check "I agree to terms & conditions" and click "Compete for purchase."

The screenshot shows a confirmation page for purchasing a domain. At the top, there are navigation links for 'Services' and 'Resource Groups'. On the right, there are icons for notifications, user 'HaniWasti', and 'Global'. The main content area contains text explaining that Route 53 will automatically create a hosted zone for the new domain, mentioning traffic routing and pricing. Below this is a section titled 'Terms and Conditions' with a note about domain registration through registrar associates. A checkbox is checked, indicating agreement to the 'AWS Domain Name Registration Agreement'. At the bottom, there are three buttons: 'Cancel', 'Back', and a blue 'Complete Purchase' button.

To make it easier for you to use Route 53 as the DNS service for your new domain, we'll automatically create a hosted zone. That's where you store information about how to route traffic for your domain, for example, to an Amazon EC2 instance. If you won't use your domain right now, you can delete the hosted zone. If you will use your domain, Route 53 charges for the hosted zone and for the DNS queries that we receive for your domain. For more information, see [Amazon Route 53 Pricing](#).

Terms and Conditions

Amazon Route 53 enables you to register and transfer domain names using your AWS account. However, AWS is not a domain name registrar, so we use registrar associates to perform registration and transfer services. When you purchase domain names through AWS, you are registering your domain with one of our registrar associates. The registrar for your domain will periodically contact the registrant contact that you specified to verify the contact details and renew registration.

I have read and agree to the [AWS Domain Name Registration Agreement](#)

[Cancel](#) [Back](#) [Complete Purchase](#)

Click “Go to domains,” and observe your domain registration is in progress

The screenshot shows the 'Domains' section of the AWS Route 53 console. The left sidebar has links for Dashboard, Hosted zones, Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains (which is selected), and Pending requests. The main content area displays a success message: 'Your registration request for the following 1 domain had been successfully submitted:' followed by a list: 'ips-labs.co.uk'. Below this, under 'Registering a new domain: what's next?', there is an 'Important' note: 'If you don't click the link in the email within 15 days to verify that you provided a valid email address, the registrar will suspend your domain. A suspended domain is not available on the Internet.' There is also a 'Note the following:' section with a list: 'Domain registration might take up to three days to complete.', 'We'll send email to the registrant contact when the domain is successfully registered.', 'We'll also send email to the registrant contact if we aren't able to register the domain for some reason.', and 'You can view the current status of your request on the dashboard in the Route 53 console.' At the bottom, there are buttons for 'Go To Domains' and 'Feedback', along with links for 'Activate Windows', 'Go to PC settings', 'Update Windows', 'Privacy Policy', and 'Terms of Use'.

Your registration request for the following 1 domain had been successfully submitted:

- ips-labs.co.uk

Registering a new domain: what's next?

Important

If you don't click the link in the email within 15 days to verify that you provided a valid email address, the registrar will suspend your domain. A suspended domain is not available on the Internet.

Note the following:

- Domain registration might take up to three days to complete.
- We'll send email to the registrant contact when the domain is successfully registered.
- We'll also send email to the registrant contact if we aren't able to register the domain for some reason.
- You can view the current status of your request on the dashboard in the Route 53 console.

[Go To Domains](#)

[Feedback](#) [English \(US\)](#)

You have a confirmation email after process completion; it might take 10 minutes to three days

AWS Services Resource Groups

HandWasti Global Support

Dashboard Hosted zones Health checks Traffic flow Traffic policies Policy records Domains Registered domains Pending requests

Status of new domain registrations and domain transfers

Domains that we're registering or transferring for you are listed below. When the registration or transfer is complete, the domain appears on the [Registered domains](#) page.

Domain Name	Status	Timestamp
	Domain registration in progress	July 07, 2018 13:42 UTC+5

Displaying 1 to 1 out of 1 domains

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Activate Windows Go to PC settings to activate Windows

The screenshot shows the AWS Route 53 console under the 'Pending requests' section. It displays a single row in a table with columns for 'Domain Name', 'Status', and 'Timestamp'. The status is 'Domain registration in progress' and the timestamp is 'July 07, 2018 13:42 UTC+5'. The page also includes navigation links for 'Feedback' and 'English (US)', and footer information from 2018.

Introduction to VPC

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 a selection of your IP address ranges, the creation of subnets, and configuration of route tables and network gateways.

A Virtual Private Cloud is a cloud computing model which offers an on-demand configurable pool of shared computing resources allocated within a public cloud environment while providing a certain level of isolation from other users of the public cloud. Since the cloud (pool of resources) is only accessible to a single client in a VPC model, it, therefore, offers privacy with greater control and a secure environment where only the specified client can operate.

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 have 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. You can also create a hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS cloud as an extension of your corporate data center.

Features & Benefits

Multiple Connectivity Options:

- Connect directly to the Internet (public subnets)
- Connect to the Internet using Network Address Translation (private subnets)
- Connect securely to your corporate data center
- Connect privately to other VPCs

- Privately connect to AWS Services without using an Internet gateway, NAT or firewall proxy through a VPC Endpoint
- Privately connect to SaaS solutions supported by AWS PrivateLink
- Privately connect your internal services across different accounts and VPCs within your organizations

Secure:

- 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
- Store data in Amazon S3 and restrict access so that it's only accessible from instances in your VPC
- For additional isolation launch dedicated instances which run on hardware dedicated to a single customer

Simple:

- Setup VPC quickly and easily using the AWS Management Console
- Easily select common network setups that best match your needs
- Subnets, IP ranges, route tables, and security groups are automatically created using VPC Wizard

Scalability & Reliability:

- Amazon VPC provides all of the benefits of the AWS platform

Amazon VPC Functionality

With Amazon Virtual Private Cloud (Amazon VPC), you can:

- Create an Amazon VPC on AWS's scalable infrastructure and specify its private IP address range from any range you choose.
- Expand your VPC by adding secondary IP ranges.

- Divide your VPC's private IP address range into one or more public or private subnets to facilitate running applications and services in your VPC.
- Assign multiple IP addresses and attach various elastic network interfaces to instances in your VPC.
- Attach one or more Amazon Elastic IP addresses to any instance in your VPC so it can be reached directly from the Internet.
- Bridge your VPC and your onsite IT infrastructure with an encrypted VPN connection, extending your existing security and management policies to your VPC instances as if they were running within your infrastructure.
- Enable EC2 instances in the EC2-Classic platform to communicate with instances in a VPC using private IP addresses.
- Associate VPC Security Groups with instances on EC2-Classic.
- Use VPC Flow Logs to log information about network traffic going in and out of network interfaces in your VPC.
- Enable both IPv4 and IPv6 in your VPC.

Components of Amazon VPC

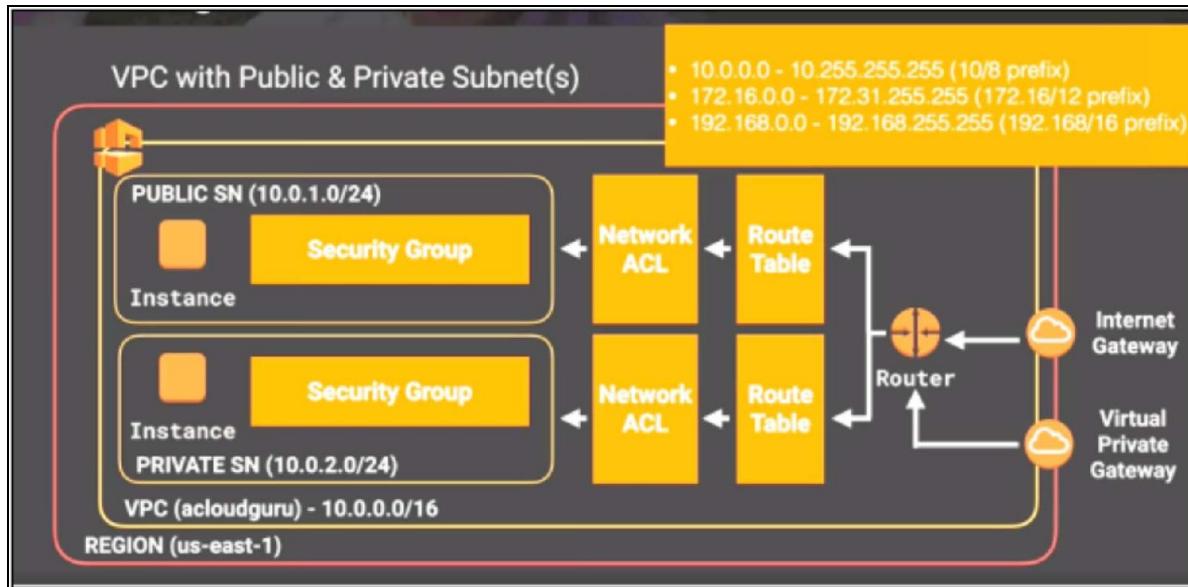


Figure 49. Amazon VPC infrastructure

- **A Virtual Private Cloud:** A logically isolated virtual network in the AWS cloud. You define a VPC's IP address space from ranges you select.
- **Subnet:** A segment of a VPC's IP address range where you can place groups of isolated resources.
- **Internet Gateway:** The Amazon VPC side of a connection to the public Internet.
- **NAT Gateway:** A highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet.
- **Hardware VPN Connection:** A hardware-based VPN connection between your Amazon VPC and your data center, home network, or co-location facility.
- **Virtual Private Gateway:** The Amazon VPC side of a VPN connection.
- **Customer Gateway:** Your side of a VPN connection.
- **Router:** Routers interconnect subnets and direct traffic between Internet gateways, virtual private gateways, NAT gateways, and subnets.
- **Peering Connection:** A peering connection enables you to route traffic via private IP addresses between two peered VPCs.
- **VPC Endpoints:** Enables private connectivity to services hosted in AWS, from within your VPC without using an Internet Gateway, VPN, Network Address Translation (NAT) devices, or firewall proxies.
- **Egress-only Internet Gateway:** A stateful gateway to provide egress only access for IPv6 traffic from the VPC to the Internet.

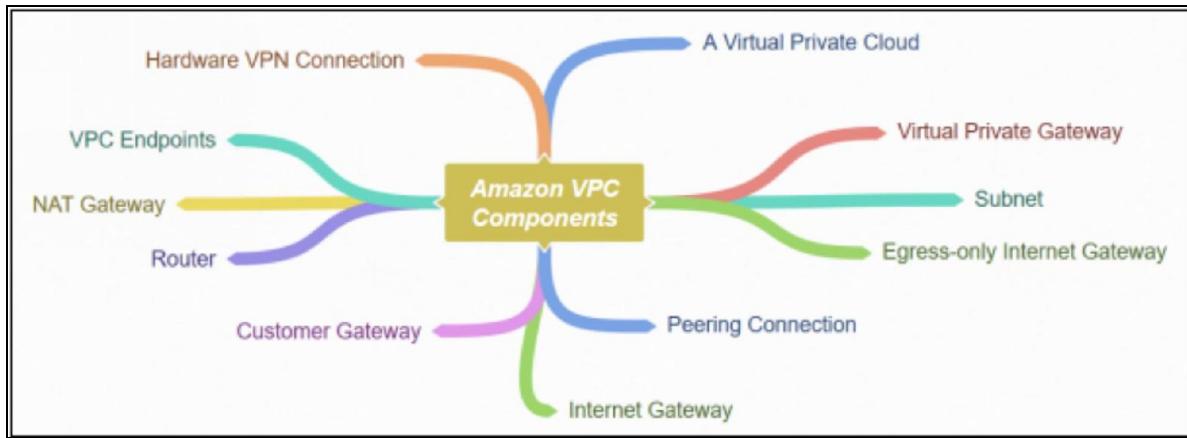


Figure 50. Mind Map of Amazon VPC Components

VPC Configuration Scenarios

Scenario 1: VPC with a Single Public Subnet

This scenario includes a virtual private cloud (VPC) with a single public subnet, and an Internet gateway to enable communication over the Internet. It is a recommended configuration if you need to run a single-tier, public-facing web application, such as a blog or a simple website.

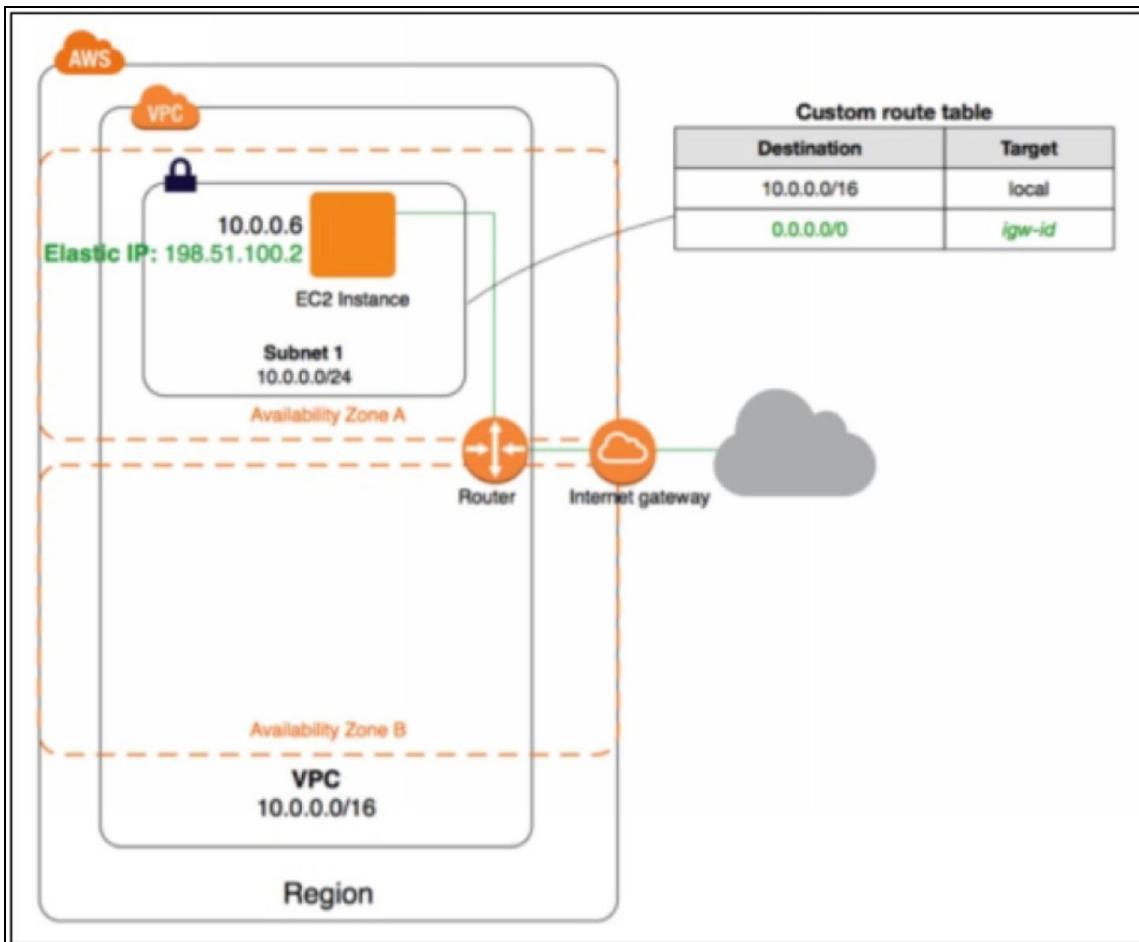


Figure 51. Key Components of the Configuration

Scenario 2: VPC with Public and Private Subnets (NAT)

This scenario includes a virtual private cloud (VPC) with a public subnet and a private subnet. It is the best practice if you want to run a public-facing

web application while maintaining back-end servers that aren't publicly accessible. A typical example is a multi-tier website, with the web servers in a public subnet and the database servers in a private subnet.

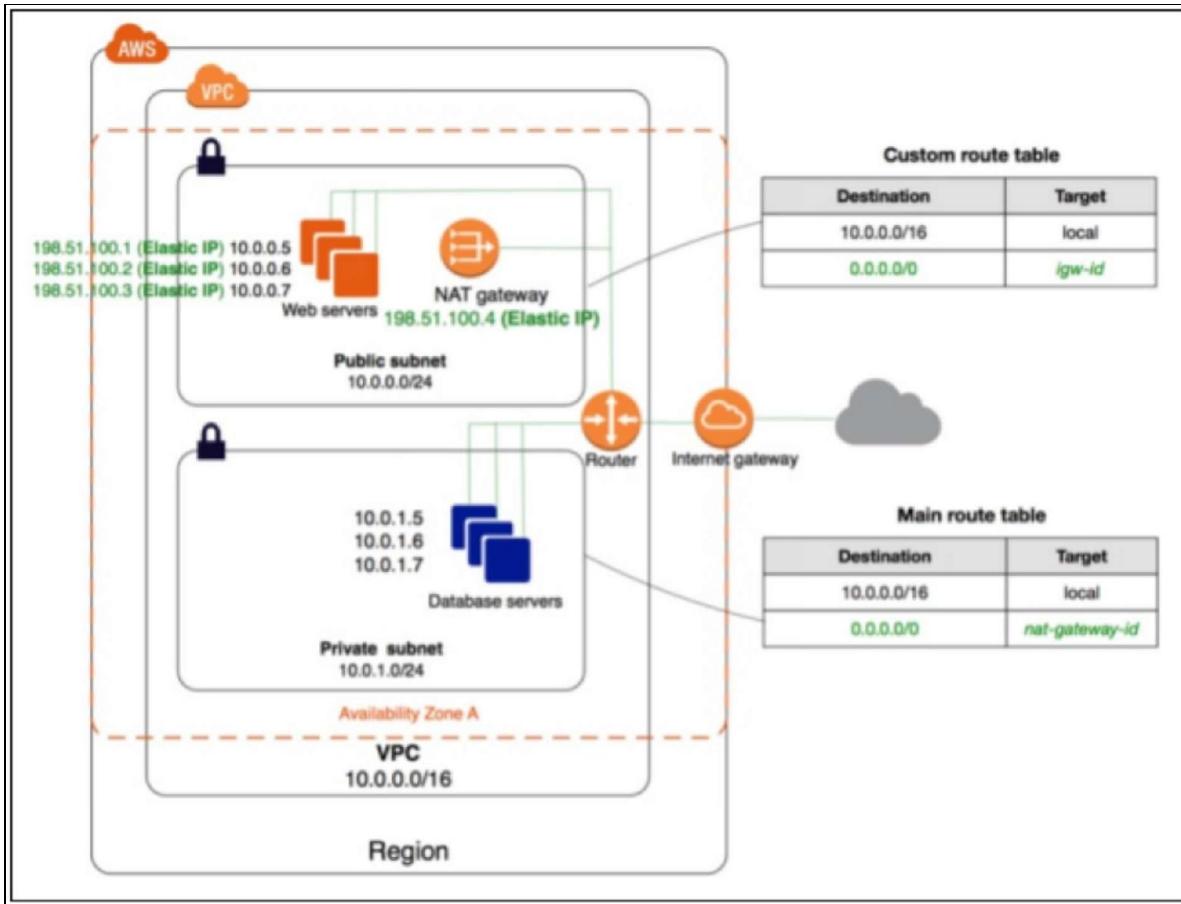


Figure 52. Key Components of the Configuration

Scenario 3: VPC with Public and Private Subnets and Hardware VPN Access

This scenario includes a virtual private cloud (VPC) with a public subnet and a private subnet, and a virtual private gateway to enable communication with your network over an IPsec VPN tunnel. It is a best practice when you want to extend your network into the cloud and also directly access the Internet from your VPC. This scenario enables you to run a multi-tiered application with a scalable web front end in a public subnet and to house your data in a private subnet that is connected to your network by an IPsec VPN connection.

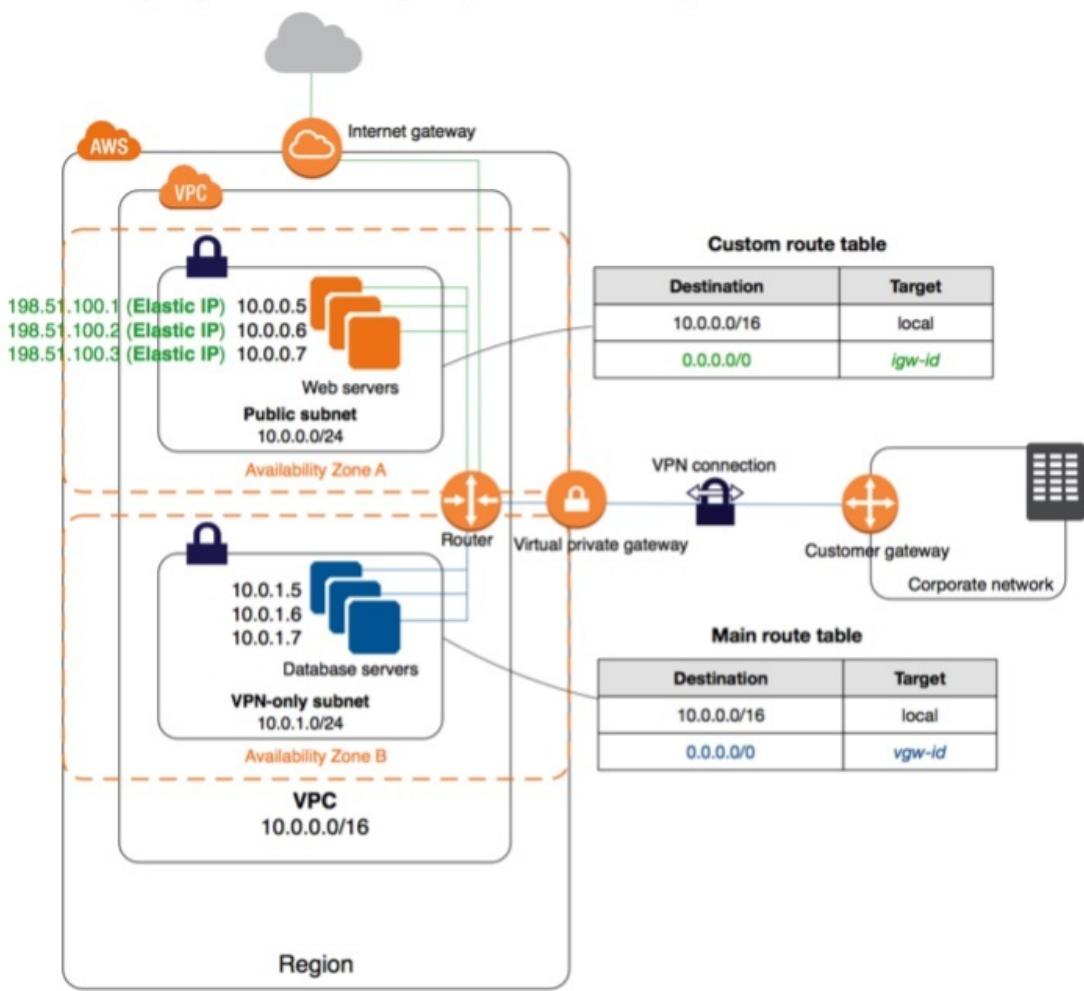


Figure 53. Key Components of the Configuration

Scenario 4: VPC with a Private Subnet and Hardware VPN Access

This scenario includes a virtual private cloud (VPC) with a single private subnet, and a virtual private gateway to enable communication with your network over an IPsec VPN tunnel. There is no Internet gateway to allow communication over the Internet. It is a best practice if you want to extend your network into the cloud using Amazon's infrastructure without exposing your network to the Internet.

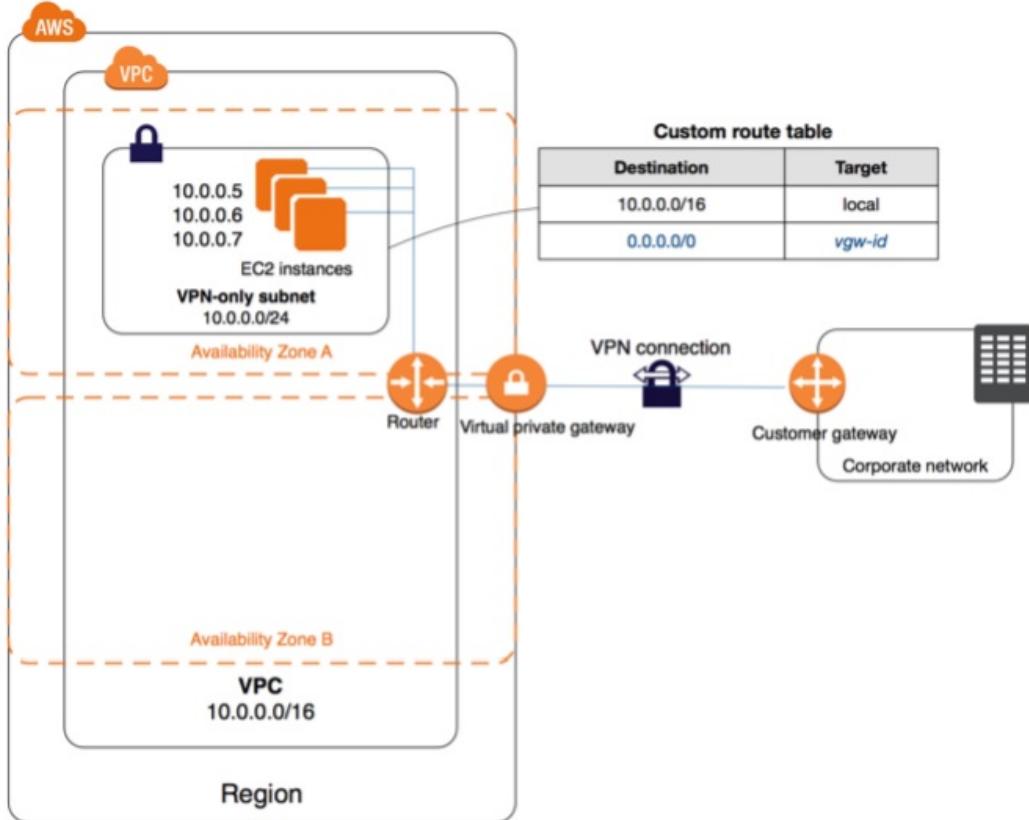


Figure 54. Key Components of the Configuration

VPC Connectivity Options

Amazon VPC provides multiple network connectivity options for you to leverage depending on your current network designs and requirements. These connectivity options include leveraging either the internet or an AWS Direct Connect connection as the network backbone and terminating the connection into either AWS or user-managed network endpoints. Additionally, with AWS, you can choose how network routing is delivered between Amazon VPC and your networks, leveraging either AWS or user-managed network equipment and routes.

Network-to-Amazon VPC Connectivity Options

AWS Managed VPN – Establishing a VPN connection from your network equipment on a remote network to AWS managed network equipment attached to your Amazon VPC.

AWS Direct Connect – Establishing a private, logical connection from your remote network to Amazon VPC, leveraging AWS Direct Connect.

AWS Direct Connect Plus VPN – Establishing a private, encrypted connection from your remote network to Amazon VPC, leveraging AWS Direct Connect.

AWS VPN CloudHub – Establishing a hub-and-spoke model for connecting remote branch offices.

Software VPN – Establishing a VPN connection from your equipment on a remote network to a user-managed software VPN appliance running inside an Amazon VPC.

Transit VPC – Establishing a global transit network on AWS using Software VPN in conjunction with AWS managed VPN.

Amazon VPC-to-Amazon VPC Connectivity Options

VPC Peering – Connecting multiple Amazon VPCs within and across regions.

Software VPN – Connecting multiple Amazon VPCs using VPN connections established between user-managed software VPN appliances running inside of each Amazon VPC.

Software-to-AWS Managed VPN – Connecting multiple Amazon VPCs with a VPN connection established between user-managed software VPN appliance in one Amazon VPC and AWS managed network equipment attached to the other Amazon VPC.

AWS Managed VPN – Connecting multiple Amazon VPCs, leveraging various VPN connections between your remote network and each of your Amazon VPCs.

AWS Direct Connect – Connecting multiple Amazon VPCs, leveraging logical connections on customer-managed AWS Direct Connect routers.

AWS PrivateLink – Connecting multiple Amazon VPCs, leveraging VPC interface endpoints and VPC endpoint services.

Internal User-to-Amazon VPC Connectivity Options

Software Remote-Access VPN – Leveraging a remote-access solution for providing end-user VPN access into an Amazon VPC.

Hardware VPN

Amazon VPC provides the option of creating an IPsec, hardware VPN connection between remote customer networks and their Amazon VPC over the Internet. Consider taking this approach when you want to take advantage of an AWS managed VPN endpoint that includes automated multi-datacenter redundancy and failover built into the AWS side of the VPN connection. Although not shown, the Amazon virtual private gateway (VGW) represents two distinct VPN endpoints, physically located in separate data centers to increase the availability of your VPN connection.

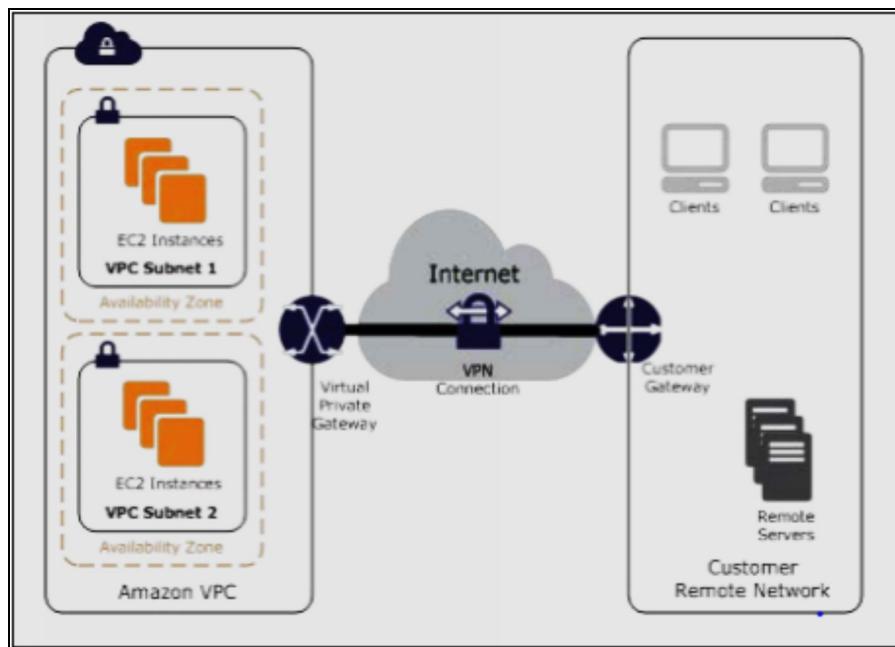


Figure 55. Amazon Virtual Private Gateway

The VGW also supports and encourages multiple user gateway connections so you can implement redundancy and failover on your side of the VPN connection. Both dynamic and static routing options are provided to give you flexibility in your routing configuration. Dynamic routing leverages BGP peering to exchange routing information between AWS and these remote endpoints. With dynamic routing, you can also specify routing priorities, policies, and weights (metrics) in your BGP advertisements and influence the network path between your network(s) and AWS. Remember

while working with BGP to terminate both IPSec and BGP connections on the same user gateway device.

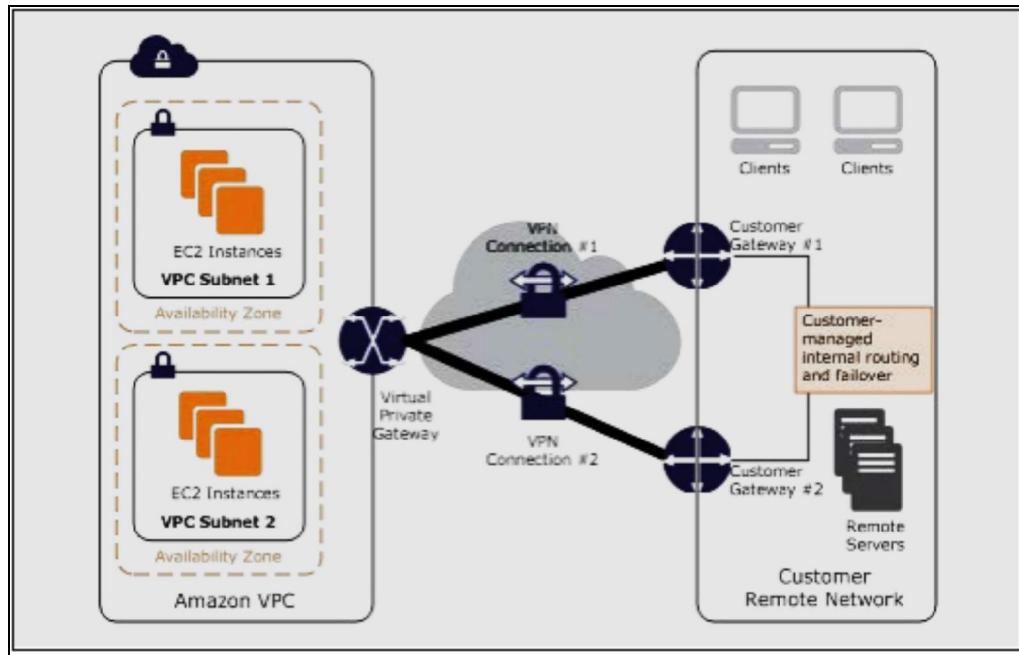


Figure 56. Amazon VPC

AWS Direct Connect

AWS Direct Connect makes it easy to establish a dedicated connection from an on-premises network to Amazon VPC. Using AWS Direct Connect, you can create private connectivity between AWS and your data center, office, or colocation environment. This private connection can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections. AWS Direct Connect lets you establish 1 Gbps or 10 Gbps dedicated network connections (or multiple connections) between AWS networks and one of the AWS Direct Connect locations. It uses industry-standard VLANs to access Amazon Elastic Compute Cloud (Amazon EC2) instances running within an Amazon VPC using private IP addresses. You can choose from an ecosystem of WAN service providers for integrating your AWS Direct Connect endpoint in an AWS Direct Connect location with your remote networks. Figure below illustrates this pattern.

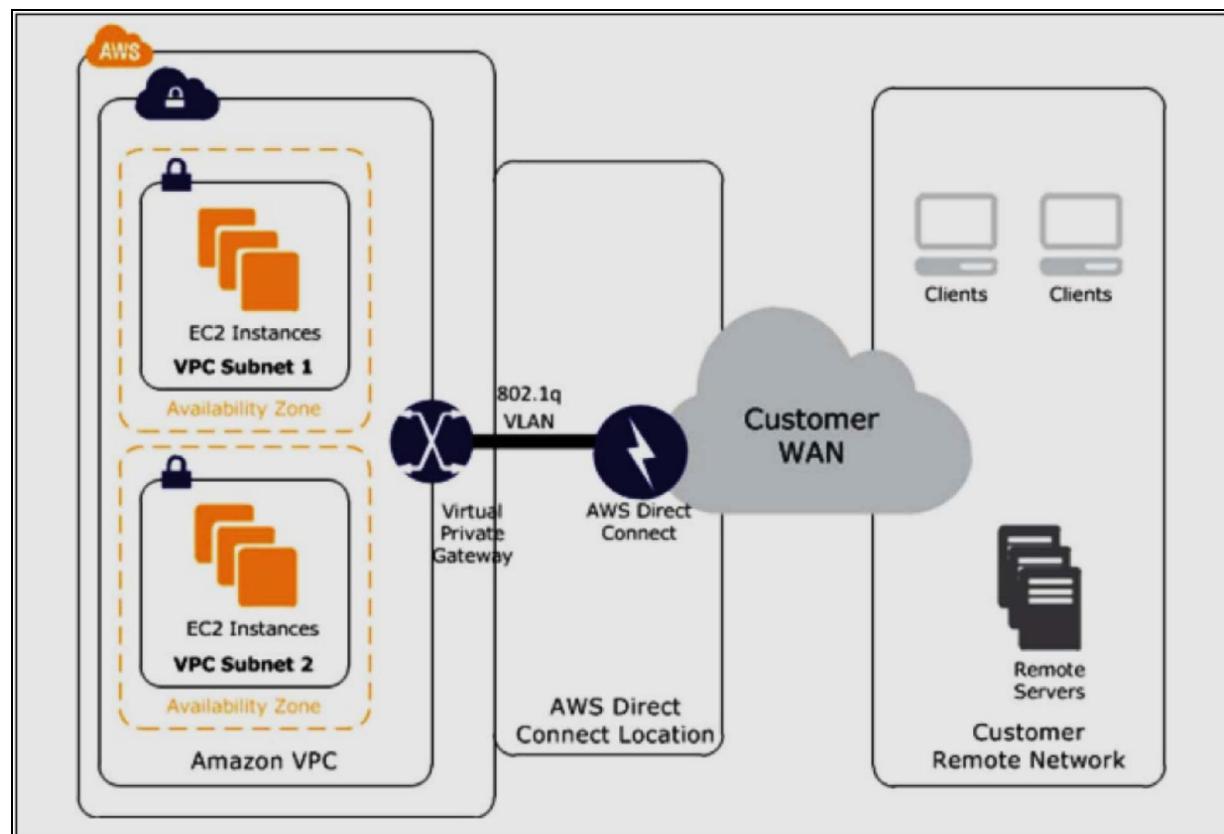


Figure 57. AWS DirectConnect

Software VPN

Amazon VPC offers you the flexibility to fully manage both sides of your Amazon VPC connectivity by creating a VPN connection between your remote network and a software VPN appliance running in your Amazon VPC network. It is a recommended option if you must manage both ends of the VPN connection either for compliance purposes or for leveraging gateway devices that are not currently supported by Amazon VPC's hardware VPN solution.

You can choose from an ecosystem of multiple partners and open source communities that have produced software VPN appliances that run on Amazon EC2. These include products from well-known security companies like Check Point, Astaro, OpenVPN Technologies, and Microsoft, as well as favorite open source tools like OpenVPN, Openswan, and IPsec-Tools. Along with this choice comes the responsibility for you to manage the software appliance, including configuration, patches, and upgrades. Note that this design introduces a potential single point of failure into the network design as the software VPN appliance runs on a single Amazon EC2 instance.

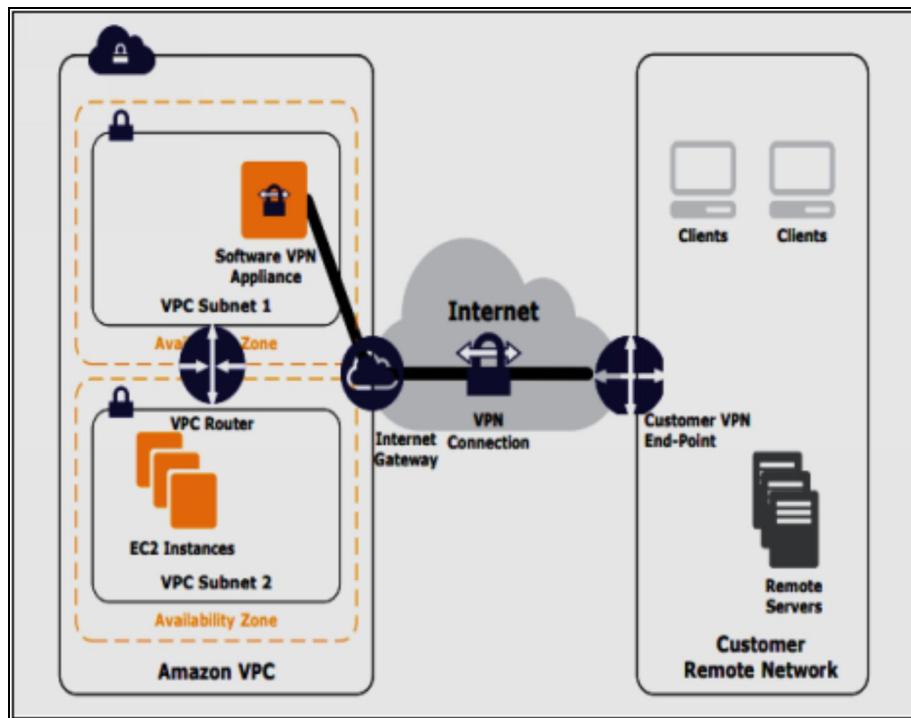


Figure 58. Software VPN

AWS Direct Connect+ VPN

With AWS Direct Connect + VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC hardware VPN. This combination provides an IPsec-encrypted private connection that also reduces network costs, increases bandwidth throughput, and offers a more consistent network experience than Internet-based VPN connections. You can use AWS Direct Connect to establish a dedicated network connection between your network and create a logical connection to public AWS resources, such as an Amazon VGW IPsec endpoint. This solution combines the AWS-managed benefits of the hardware VPN solution with low latency, increased bandwidth, more consistent benefits of the AWS Direct Connect solution, and an end-to-end, secure IPsec connection.

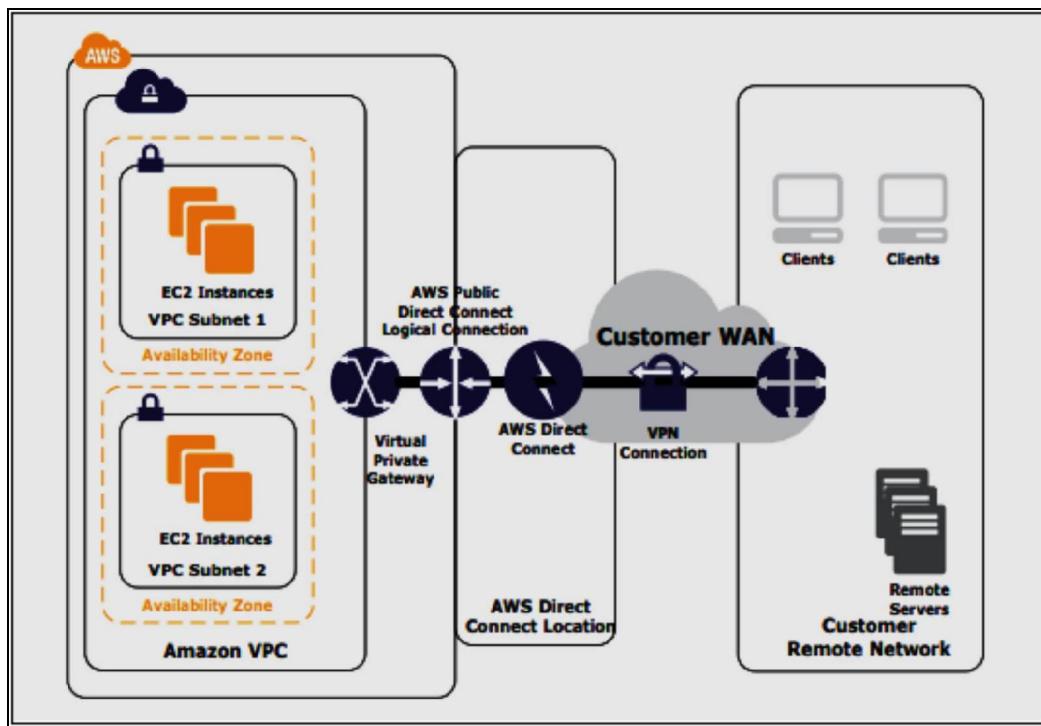


Figure 59. Amazon DirectConnect + VPN

AWS VPN Cloud Hub

Building on the hardware VPN and AWS Direct Connect options described previously; you can securely communicate from one site to another using the AWS VPN CloudHub. The AWS VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC. Use this design if you have multiple branch offices and existing Internet connections and would like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these remote offices. The figure below depicts the AWS VPN CloudHub architecture, with blue dashed lines indicating network traffic between remote sites routed over their AWS VPN connections.

AWS VPN CloudHub leverages an Amazon VPC virtual private gateway with multiple gateways, each using unique BGP autonomous system numbers (ASNs). Your gateways advertise the appropriate routes (BGP prefixes) over their VPN connections. These routing advertisements are received and readvertised to each BGP peer so that each site can send data to and receive data from the other sites. The remote network prefixes for each spoke must have unique ASNs, and the sites must not have overlapping IP ranges. Each site can also send and receive data from the VPC as if they were using a standard VPN connection with AWS Direct Connect or other hardware VPN options (e.g., multiple gateways per site for redundancy or backbone routing that you provide) depending on your requirements.

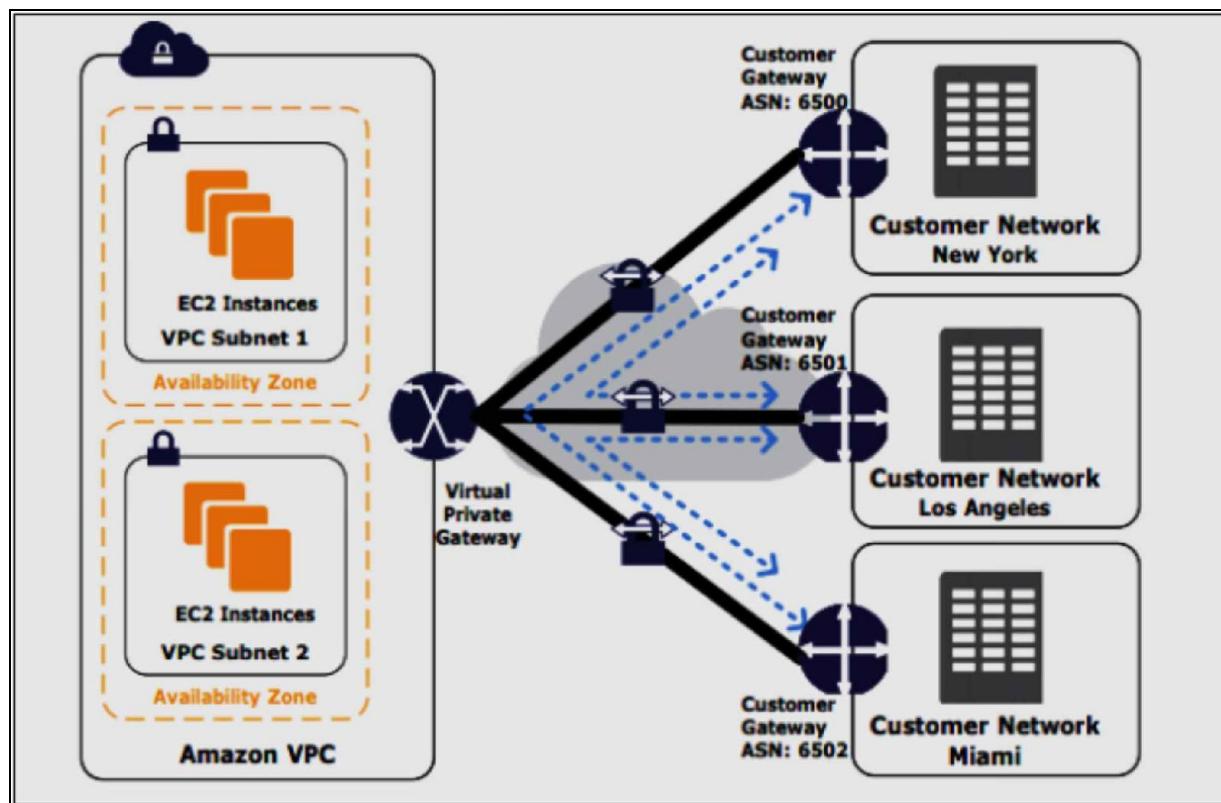
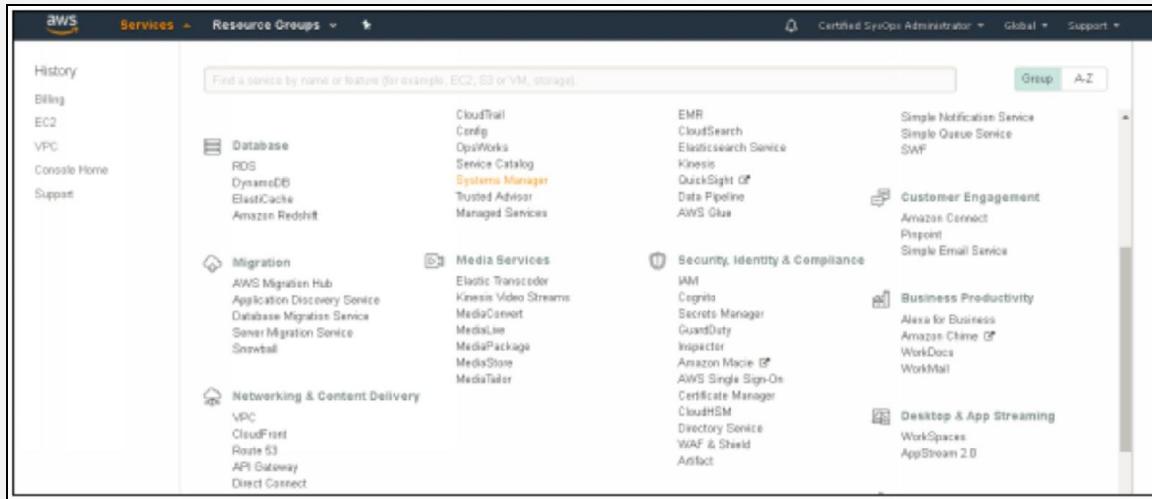


Figure 60. AWS VPN Cloud Hub

Lab 7.2 Build A Custom VPC

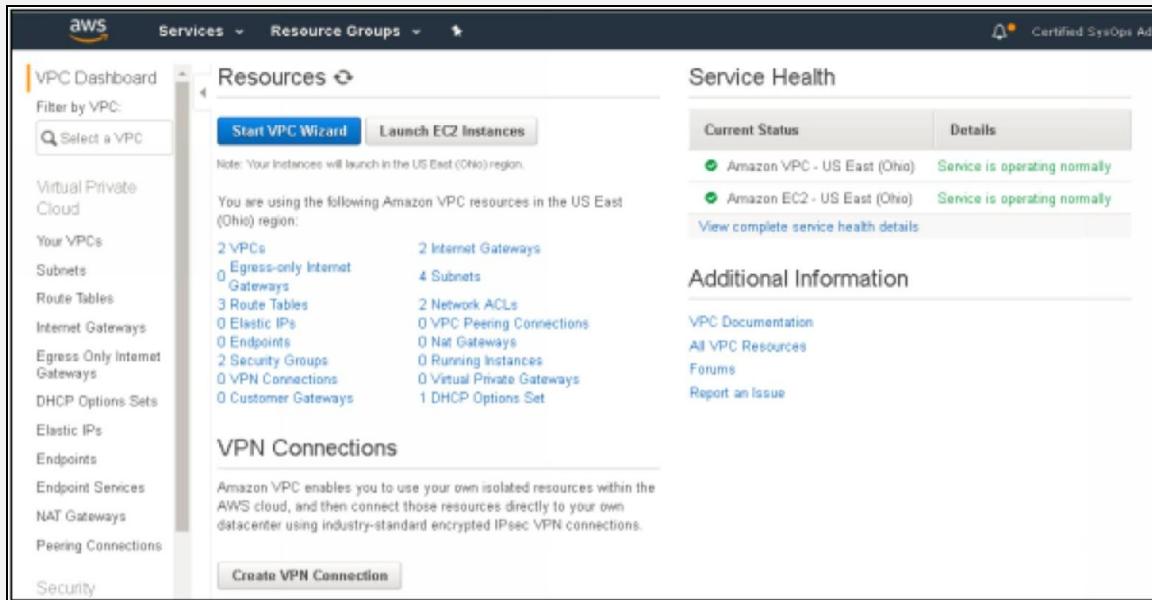
In this lab, we build a Virtual Private Cloud

Login to AWS management console, click “Services,” under “Networking and Content Delivery” click VPC



Before start working, let us look at the default setting

Click subnets from the side menu



These IPv4 addresses are subnets that are used by your default VPCs

Search VPCs and their properties								
Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Route table	Network ACL	Tenancy
vpc-0ab233e2	available	172.31.0.0/16			optd-487ce020	rta-c87108a0	ad-1a72342cf	Default
vpc-09413166	available	10.0.0.0/16			optd-487ce020	rta-d9e9d8b0	ad-1d84d775	Default

Default route tables for your default VPCs

Search Subnets and their properties								
Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4 IP	IPv6 CIDR	Availability Zone	Route Table
subnet-d23108b0	available	vpc-0ab233e2	172.31.0.0/20	4091	us-east-2a	rta-c87108a0		
subnet-1aff0893	available	vpc-0ab233e2	172.31.16.0/20	4090	us-east-2b	rta-c87108a0		
subnet-5e88a213	available	vpc-0ab233e2	172.31.32.0/20	4091	us-east-2c	rta-c87108a0		

The screenshot shows the AWS VPC Dashboard. On the left, a sidebar lists various VPC components: Virtual Private Cloud, Your VPCs, Subnets, Route Tables (which is selected), Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, and Peering Connections. The main content area is titled "Create Route Table" and displays a table of existing route tables. The table has columns for Name, Route Table ID, Explicitly Associated, Main, and VPC. Three route tables are listed:

Name	Route Table ID	Explicitly Associated	Main	VPC
	rtb-c87108a0	0 Subnets	Yes	vpc-8ab232e2
	rtb-5befdf933	0 Subnets	No	vpc-8e413106
	rtb-d8ebdd0	0 Subnets	Yes	vpc-8e413106

A message at the bottom says "Select a route table above".

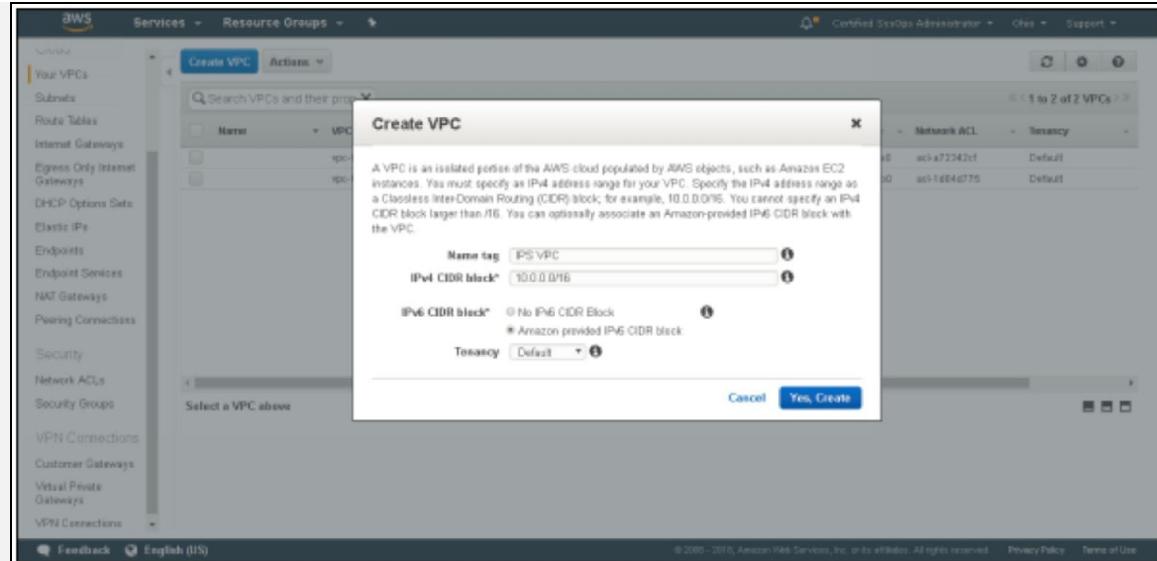
Your security groups

The screenshot shows the AWS Security Groups dashboard. On the left, a sidebar lists components: Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security (which is selected), Network ACLs, Security Groups (selected), and VPN Connections. The main content area is titled "Create Security Group" and displays a table of existing security groups. The table has columns for Name tag, Group ID, Group Name, VPC, and Description. Two security groups are listed:

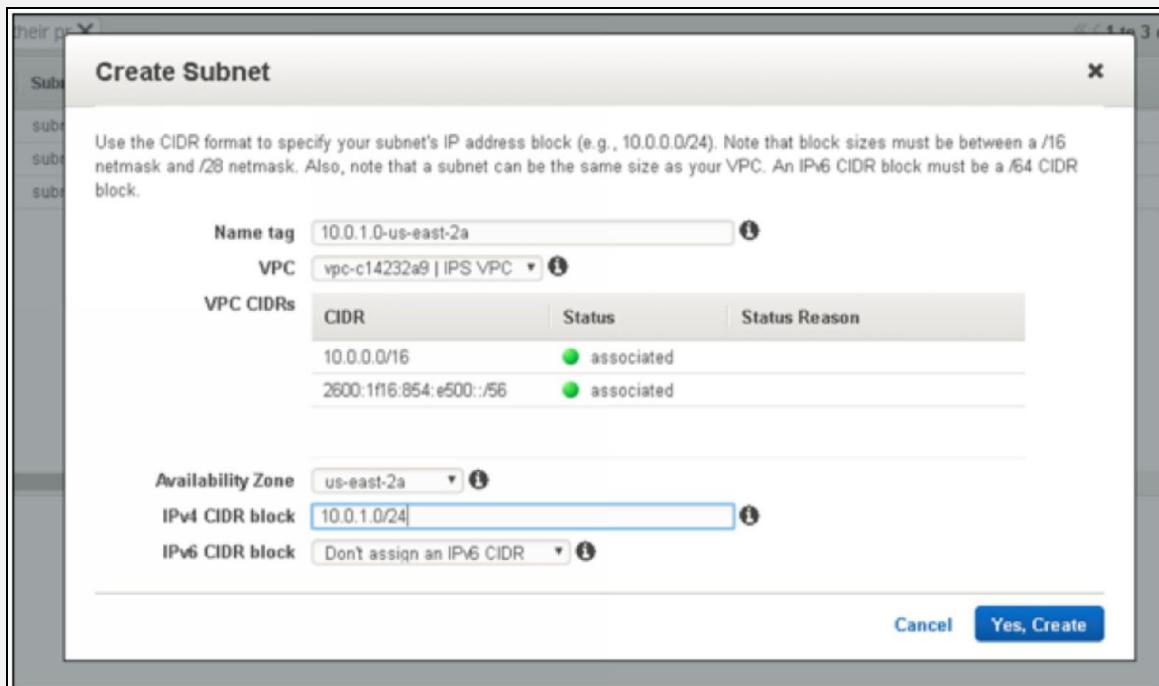
Name tag	Group ID	Group Name	VPC	Description
	sg-1a9c7a70	default	vpc-8e413106	default VPC security group
	sg-954929fe	default	vpc-8ab232e2	default VPC security group

A message at the bottom says "Select a security group above".

Now, click the Create VPC button at the top of VPC dashboard



Name your VPC and provide a CIDR address range. Click “Yes create” button



You can see your VPC is now on the list

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with various VPC-related options like Virtual Private Cloud, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IP, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, and Security Groups. The main area has tabs for 'Create Subnet' and 'Subnet Actions'. A search bar at the top says 'Search Subnets and their pr X'. Below it is a table with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, Availability Zone, and Route Table. There are four subnets listed:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Route Table
10.0.1.0-us-east-2a	subnet-03721d8b	available	vpc-c14232a9 IPS VPC	10.0.1.0/24	251		us-east-2a	rtb-d931eb7
	subnet-d231f6fa	available	vpc-8ab232e2	172.31.0.0/23	4091		us-east-2a	rtb-c8710890
	subnet-1a0d090	available	vpc-8ab232e2	172.31.16.0/23	4093		us-east-2a	rtb-c8710890
	subnet-5e999313	available	vpc-8ab232e2	172.31.32.0/23	4091		us-east-2a	rtb-c8710890

Below the table, a modal window titled 'subnet-03721d8b | 10.0.1.0-us-east-2a' shows the 'Summary' tab with details: Subnet ID: subnet-03721d8b | 10.0.1.0-us-east-2a, Availability Zone: us-east-2a, IPv4 CIDR: 10.0.1.0/24, IPv6 CIDR: 2600:1f16:854:e500::/56, Status: available, Route table: rtb-d931eb7, Network ACL: acl-3532e15d, Default gateway: no.

Now, create a subnet for your VPC

The screenshot shows the 'Create Subnet' dialog box. It has a 'Name tag' field containing '10.0.1.0-us-east-2a' and a 'VPC' dropdown set to 'vpc-c14232a9 | IPS VPC'. Below these, a table titled 'VPC CIDRs' lists two entries: '10.0.0.0/16' and '2600:1f16:854:e500::/56', both marked as 'associated'. Under 'Availability Zone', it shows 'us-east-2b'. The 'IPv4 CIDR block' field is set to '10.0.2.0/24'. The 'IPv6 CIDR block' field is set to 'Don't assign an IPv6 CIDR'. At the bottom right are 'Cancel' and 'Yes, Create' buttons.

You can see your created subnet is now on the list

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4 /-	IPv6 CIDR	Availability Zone	Route Table
10.0.1.0-us-east-2a	subnet-03721deb	available	vpc-c14232a9 IPS VPC	10.0.1.0/24	251		us-east-2a	rtb-d9f8fb7
10.0.1.0-us-east-2a	subnet-d9fd39a3	available	vpc-c14232a9 IPS VPC	10.0.2.0/24	251		us-east-2b	rtb-d9f8fb7
	subnet-d231b9fa	available	vpc-9ab232e2	172.31.0.0/20	4891		us-east-2a	rtb-c87106a0
	subnet-1af1960	available	vpc-9ab232e2	172.31.16.0/20	4890		us-east-2b	rtb-c87106a0
	subnet-54066313	available	vpc-9ab232e2	172.31.32.0/20	4891		us-east-2c	rtb-c87106a0

Click “Internet Gateways” from the side menu

Name	ID	State	VPC
igw-81b747e9	igw-81b747e9	attached	vpc-6e413106
igw-d88091b1	igw-d88091b1	attached	vpc-9ab232e2

Click, “Create Internet Gateway”

Write a name for your Internet Gateway, click “Create.”

Services Resource Groups Certified SysOps Administrator

Internet gateways > Create internet gateway

Create internet gateway

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Name tag (Required)

* Required Cancel Create

Observe the following message

Services Resource Groups

Internet gateways > Create internet gateway

Create internet gateway

The following internet gateway was created.

Internet gateway ID igw-c9a454a1

Close

You can see your created Internet Gateway on the list

Services Resource Groups

VPC Dashboard Filter by VPC:

Virtual Private Cloud Your VPCs Subnets Route Tables Internet Gateways Egress Only Internet Gateways DHCP Options Sets Elastic IPs Endpoints Endpoint Services NAT Gateways Peering Connections

Create internet gateway Actions

Filter by tags and attributes

Name	Name	State	VPC
igw-81b747e9	igw-c9a454a1	attached	vpc-6e413106
igw-d88091b1		attached	vpc-8ab232e2
my gateway	my gateway	detached	-

Internet gateway: igw-c9a454a1

Description Tags

Name	Name	State	VPC
igw-81b747e9	igw-c9a454a1	attached	vpc-6e413106
igw-d88091b1		attached	vpc-8ab232e2
my gateway	my gateway	detached	-

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with links like 'Virtual Private Cloud', 'Your VPCs', 'Subnets', 'Route Tables', 'Internet Gateways' (which is highlighted in orange), 'Egress Only Internet Gateways', 'DHCP Options Sets', 'Elastic IPs', 'Endpoints', 'Endpoint Services', 'NAT Gateways', and 'Peering Connections'. The main area has a title 'Create internet gateway' and a table with columns 'Name', 'ID', 'State', and 'VPC'. It lists three entries: 'igw-81b747e9' (attached to 'vpc-6e413106'), 'my gateway' (selected, with ID 'igw-c9a454a1', attached to 'vpc-c14232a9 | IP...'), and 'igw-d88091b1' (attached to 'vpc-8ab232e2'). Below the table, it says 'Internet gateway: igw-c9a454a1' and has tabs for 'Description' and 'Tags'.

Select “Route Table” from the side menu, click “Create Route Table,” add name and click, “Yes, Create.”

The dialog box is titled 'Create Route Table'. It contains a descriptive text: 'A route table specifies how packets are forwarded between the subnets within your VPC, the Internet, and your VPN connection.' Below this, there are two input fields: 'Name tag' with the value 'My IPS route' and 'VPC' with the value 'vpc-c14232a9 | IPS VPC'. At the bottom right are 'Cancel' and 'Yes, Create' buttons.

To allow internet access, edit the route table and click “Add another route.”

The screenshot shows the AWS VPC Dashboard. On the left, a sidebar lists various VPC-related services: Virtual Private Cloud, Your VPCs, Subnets, Route Tables (which is selected), Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, and Security Groups. A search bar at the top right says "Search Route Tables and th...". Below it is a table with columns: Name, Route Table ID, Explicitly Associated, Main, and VPC. The table lists several route tables, including one named "My IPS route" which is currently selected. At the bottom, there is a modal window titled "Create Route Table" with tabs for "Cancel" and "Save". It shows a table of routes with columns: Destination, Target, Status, Propagated, and Remove. Two routes are listed: "10.0.0.0/16" with target "local" and status "Active", and "2600:1f16:854:e500::/56" with target "local" and status "Active". A new route is being added with a destination of "0.0.0.0/0" and a target of "igw-c9a454a1". The status is set to "No" and the propagate checkbox is checked. A "Save" button is visible at the top of the modal.

Create and Save IPv4 Route.

The screenshot shows the AWS VPC Dashboard with the 'Route Tables' section selected. A route table named 'My IPv6 route' is highlighted. The table lists two routes:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-c9a454a1	Active	No

For consistency, we need to add IPv6 route out as well; It gives us accessibility of both IPv4 and IPv6.

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Create Route Table Delete Route Table Set As Main Table

Search Route Tables and th X

Name	Route Table ID	Explicitly Associated	Main	VPC
	rtb-dfe9dfb7	0 Subnets	Yes	vpc-c14232a9 IPS VPC
	rtb-c87108a0	0 Subnets	Yes	vpc-8ab232e2
	rtb-5befd933	0 Subnets	No	vpc-e413106
My IPS route	rtb-e2c3f58a	0 Subnets	No	vpc-c14232a9 IPS VPC

Edit View: All rules

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
2600:1f16:854:e500::/56	local	Active	No
0.0.0.0/0	igw-c9a454a1	Active	No
::/0	igw-c9a454a1	Active	No

Feedback English (US) © 2006 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Go to subnet, click subnet actions

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Create Subnet Submit Actions

Search Subnets and their pr X

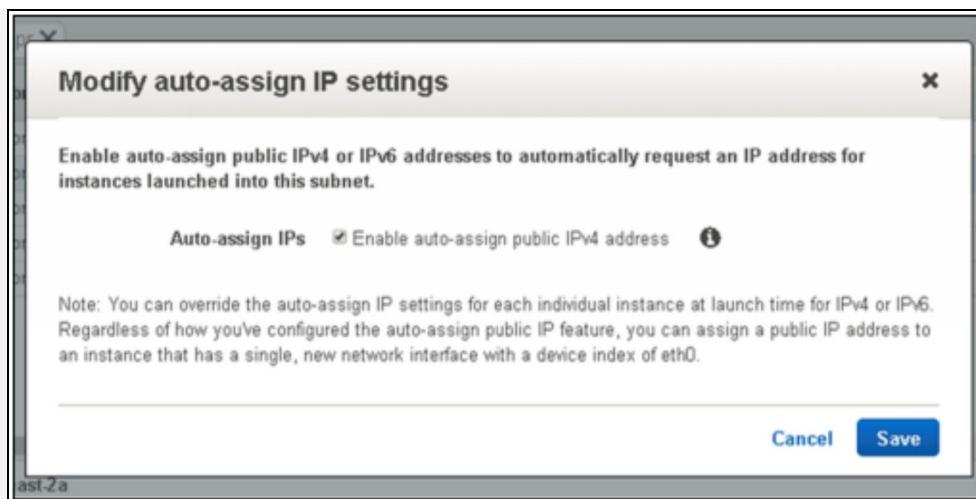
IPv4 CIDR	Available IPv4 /	IPv6 CIDR	Availability Zone	Route Table	Network ACL	Default Subnet	Auto-assign Public IP	Auto-assign IPv6 address
10.0.1.0/24	251		us-east-2a	rtb-dfe9dfb7	acl-35b2e15d	No	No	No
10.0.2.0/24	251		us-east-2b	rtb-dfe9dfb7	acl-35b2e15d	No	No	No
172.31.0.0/20	4091		us-east-2a	rtb-c87108a0	acl-e72342cf	Yes	Yes	No
172.31.16.0/20	4090		us-east-2b	rtb-c87108a0	acl-e72342cf	Yes	Yes	No
172.31.32.0/20	4091		us-east-2c	rtb-c87108a0	acl-e72342cf	Yes	Yes	No

subnet-e99d09a3 | 10.0.1.0/us-east-2a

Summary	Route Table	Network ACL	Flow Logs	Tags
Subnet ID: subnet-e99d09a3 10.0.1.0-us-east-2a	Route table: rtb-dfe9dfb7	Availability Zone: us-east-2b		
IPv4 CIDR: 10.0.1.0/24	Network ACL: acl-35b2e15d			
IPv6 CIDR:	Default subnet: no			
	Auto-assign Public IP: no			
	Auto-assign IPv6 address: no			

Feedback English (US) © 2006 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Check the “Enable auto assign Public IP” checkbox



We are all set to provision our EC2 instances, go to EC2 from AWS console

Click “Launch an instance.”

Choose the AMI

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' section of the AWS CloudFormation console. The top navigation bar includes tabs for 'Services' (selected), 'Resource Groups', and a progress bar from '1. Choose AMI' to '7. Review'. On the left, a sidebar titled 'Quick Start' lists 'My AMIs', 'Amazon Linux' (selected), 'Community AMIs', and 'Free tier only'. The main content area displays four AMI options:

- Amazon Linux AMI 2018.02.0 (HVM), SSD Volume Type** - ami-976153f2: 64-bit. Description: The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
- Amazon Linux 2 LTS Candidate 2 AMI (HVM), SSD Volume Type** - ami-31c71654: 64-bit. Description: Amazon Linux 2 LTS Candidate 2 provides an updated version of the Linux Kernel (4.14) tuned for EC2, systemd snapshot, a newer compiler (gcc 7.3), an updated C runtime (glibc 2.26), newer toolkits (libsodium 2.29.1), and the latest software packages through the extras mechanisms.
- SUSE Linux Enterprise Server 12 SP0 (HVM), SSD Volume Type** - ami-57d3e732: 64-bit. Description: SUSE Linux Enterprise Server 12 Service Pack 3 (HVM, EBS General Purpose (SSD) Volume Type). Public Cloud, Advanced Systems Management, Web and Billing, and Legacy modules available.
- Ubuntu Server 16.04 LTS (HVM), SSD Volume Type** - ami-91615054: 64-bit. Description: Ubuntu Server 16.04 LTS (HVM,EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/technical>)).

At the bottom, there are 'Feedback' and 'English (US)' buttons, and a footer with copyright information and links to Privacy Policy and Terms of Use.

Configure instance details, make sure to associate the instance with your VPC.

The screenshot shows the 'Step 3: Configure Instance Details' section of the AWS CloudFormation console. The top navigation bar includes tabs for 'Services' (selected), 'Resource Groups', and a progress bar from '1. Choose AMI' to '7. Review'. The main content area displays configuration options:

- Number of instances**: 1 (), **Launch into Auto Scaling Group** (checkbox).
- Purchasing option**: Request Spot instances (radio button selected).
- Network**: vpc-c14250a9 | IPS VPC (dropdown), **Create new VPC** (button).
- Subset**: subnet-d9d49a3 | 10.0.1.0-10.0.1.255 | us-east-2a | 251 IP Addresses available (dropdown).
- Auto-assign Public IP**: Use subnet setting (Enable) (dropdown).
- IAM role**: None (dropdown), **Create new IAM role** (button).
- Shutdown behavior**: Stop (dropdown).
- Enable termination protection**: Protect against accidental termination (checkbox).
- Monitoring**: Enable CloudWatch detailed monitoring (checkbox), **Additional charges apply** (link).
- Tenancy**: Shared - Run a shared hardware instance (dropdown), **Additional charges will apply for dedicated tenancy** (link).

At the bottom, there are 'Cancel', 'Previous', 'Review and Launch' (selected), and 'Next: Add Storage' buttons, and a footer with copyright information and links to Privacy Policy and Terms of Use.

Add storage

Screenshot of the AWS EC2 instance creation wizard Step 4: Add Storage. The page shows a table for adding storage volumes. A single row is present for the root volume:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0e60b1641dcfa67ba	8	General Purpose SSD (GP2)	100 / 3000	N/A	No	Not Encrypted

A note below the table states: "Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions."

Buttons at the bottom include: Cancel, Previous, Review and Launch (highlighted in blue), and Next: Add Tags.

Add Tags

Screenshot of the AWS EC2 instance creation wizard Step 5: Add Tags. The page shows a table for adding tags. A single row is present:

Key	(127 characters maximum)	Value	(255 characters maximum)	Instances	Volumes
This resource currently has no tags					

Instructions and notes:
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Buttons at the bottom include: Add Tag (highlighted in blue), Cancel, Previous, Review and Launch (highlighted in blue), and Next: Configure Security Group.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances (1)	Volumes (1)
mykey	my 1st webserver		

Add another tag (Up to 50 tags maximum)

Cancel **Previous** **Review and Launch** **Next: Configure Security Group**

Feedback **English (US)**

© 2006 - 2010, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Configure security group

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Create a new security group
 Select an existing security group

Security group name: Web-DMZ
Description: Web-DMZ

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/-0	e.g. SSH for Admin Desktop

Add Rule

Warning
 Rules with source of 0.0.0.0 allow all IP addresses to access your instance. We recommend updating security group rules to allow access from known IP addresses only.

Cancel **Previous** **Review and Launch**

Feedback **English (US)**

© 2006 - 2010, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Launch your instance

Services ▾ Resource Groups ▾

Certified SysOp Administrator ▾ Ohio ▾ Support ▾

Launch Status

Your instances are now launching.
The following instance launches have been initiated: i-01be49a6d0bd779 [View launch log](#)

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started:

- [How to connect to your Linux instance](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also:

- Create status check [alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

[Feedback](#) [English \(US\)](#)

© 2006 - 2010, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Lab 7.3 Custom VPC with Private Subnet

In this lab, we are going to use EC2 instance as a private subnet. First of all, go to the AWS Management Console.

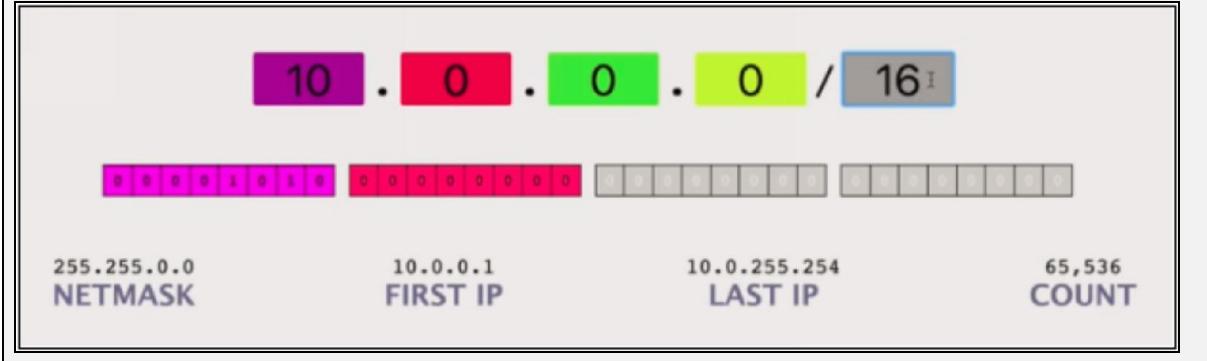
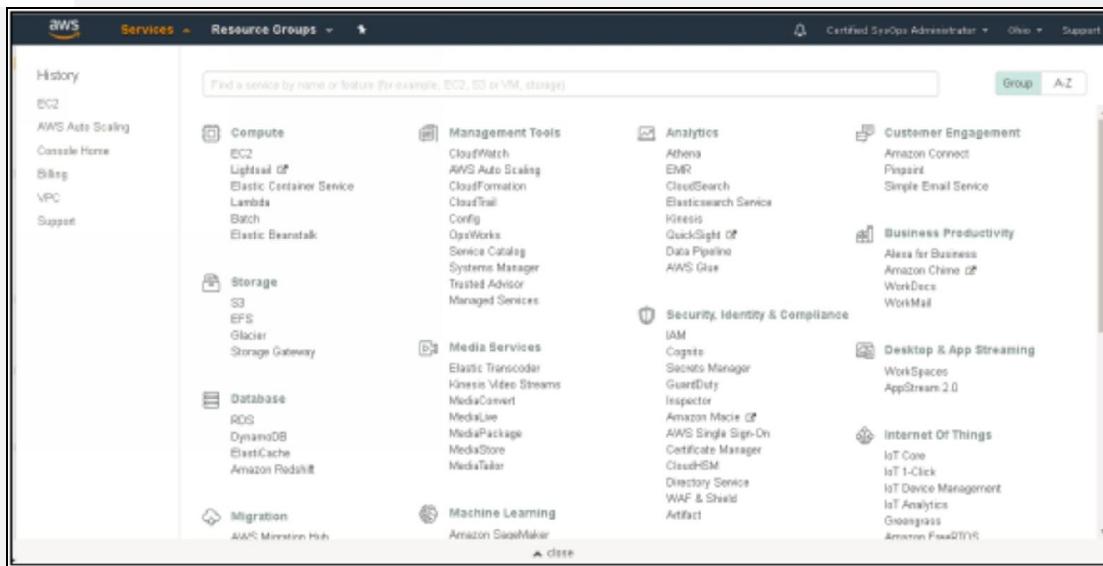
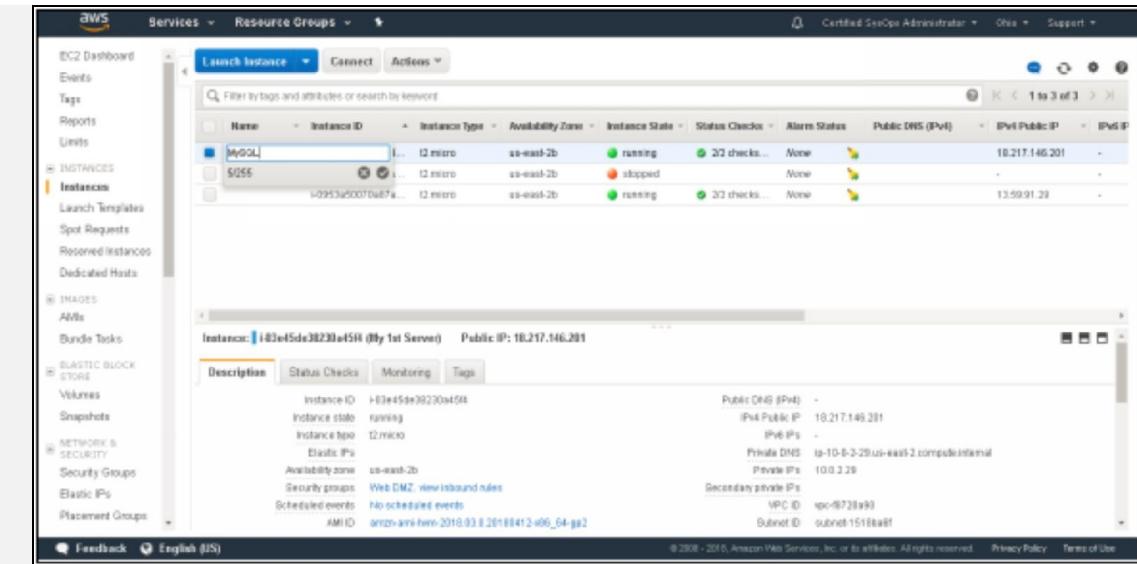


Figure 59. Private Subnet

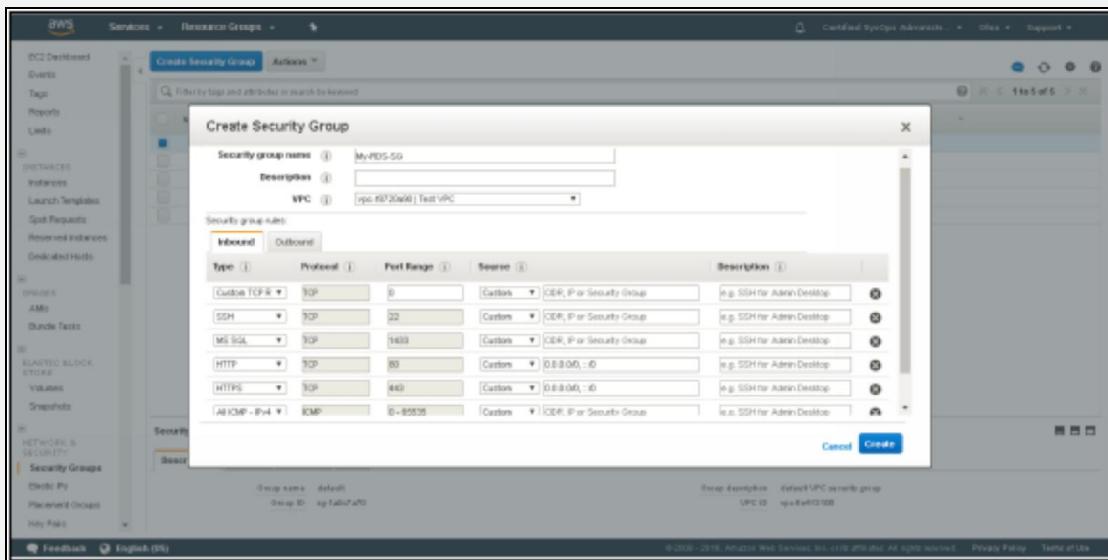
1. Go to services and click on EC2, then click on “instances.”



2. Now select any running instance and change the name of that instance, as shown in the figure.

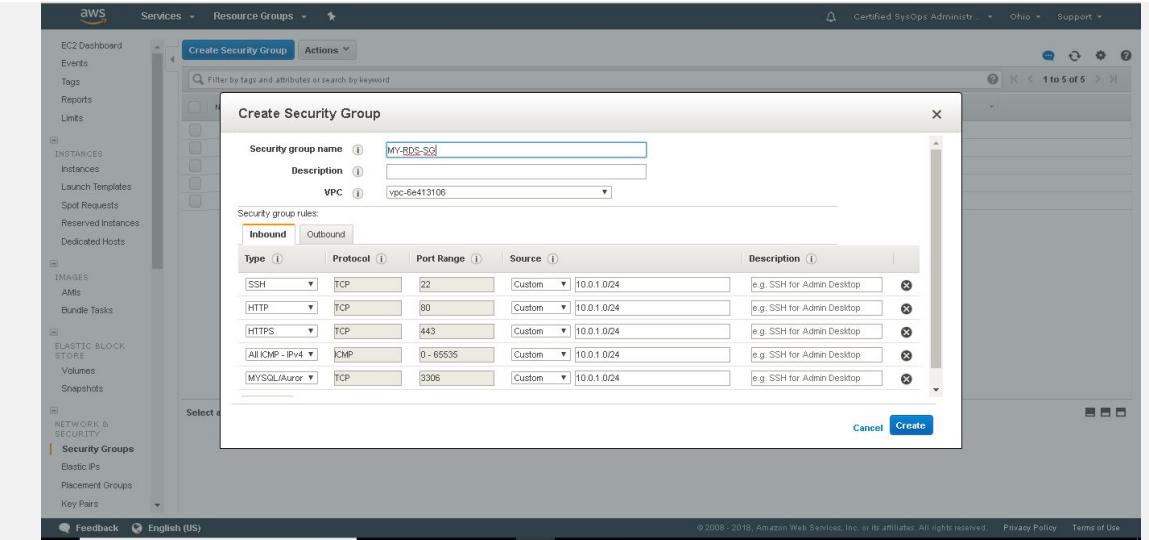


3. No, to associate this instance with the security group, go to security groups and select any default security group and associate that instance with it.

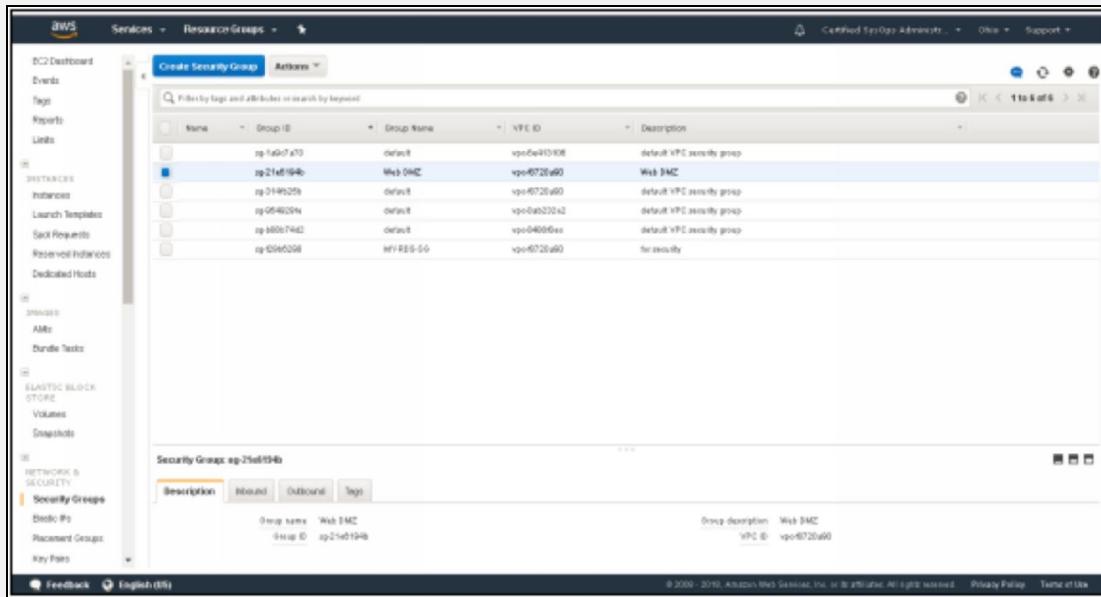


4. Name the security group and allows the type of traffic to allow traffic through this instance from private subnets to public subnets.

(Note: Description field is mandatory)

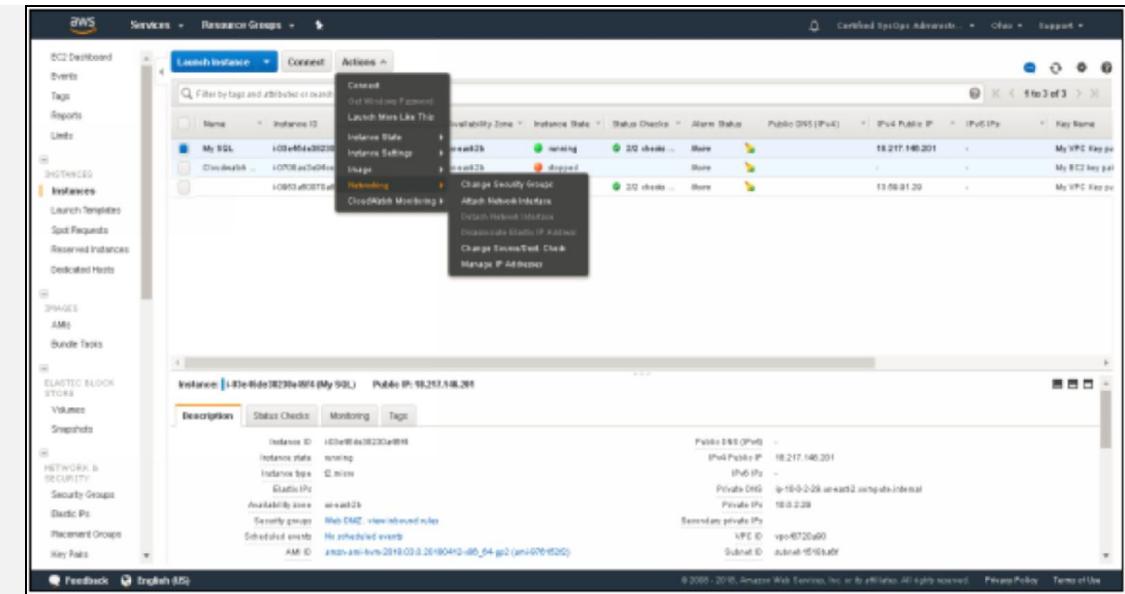


5. These are the security groups.



6. Select the security group that you have created and go back to your EC2 instances.

7. Choose the instance “MySQL” that you have created and then click on “Actions.” You have various options, select networking from those options and then select “Change Security Groups.”



8. This instance has now assigned with a Public IPv4 address and a security group.

```
[root@ip-10-0-1-242 ec2-user]# ping 10.0.2.143
PING 10.0.2.143 (10.0.2.143) 56(84) bytes of data.
64 bytes from 10.0.2.143: icmp_seq=1 ttl=255 time=1.39 ms
64 bytes from 10.0.2.143: icmp_seq=2 ttl=255 time=1.40 ms
64 bytes from 10.0.2.143: icmp_seq=3 ttl=255 time=1.42 ms
64 bytes from 10.0.2.143: icmp_seq=4 ttl=255 time=1.44 ms
64 bytes from 10.0.2.143: icmp_seq=5 ttl=255 time=1.39 ms
64 bytes from 10.0.2.143: icmp_seq=6 ttl=255 time=1.36 ms
64 bytes from 10.0.2.143: icmp_seq=7 ttl=255 time=1.42 ms
^C
--- 10.0.2.143 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 1.367/1.407/1.444/0.041 ms
[root@ip-10-0-1-242 ec2-user]#
```

```
[root@ip-10-0-1-242 ec2-user]# nano mypvk
```

GNU nano 2.5.3 File: mypvk.pem

[New File]

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify
^X Exit **^R** Read File **^V** Replace **^U** Uncut Text **^T** To Spell

GNU nano 2.5.3 File: mypvk.pem Modified

```
A3TCyQ8Q4XMMWMdEXfi1+HfW97DnlgvR5SKLSnTy2wnLTSl6ok+ff7PSrRNXWjAbFc9KC9A$  
AoGALfec72vKp8edycmCA5/21RjkActhFRpJJVm djLDabLiihZRNLBszMGauy1CGsG/Fk+k$  
GUeLqCeRS+3yFcoEEHX3119GfAMCq80hKWKdyiVjVcM1aY0vD3b/PKWCUI6r91XrOsKpG1A$  
mSjf1rEsquPuieN1Cb4ayT0CgYB2bZQ7p9j5mqfE8quFEDvqkJtJ9ZfIALMTYZLgd0/e8hd$  
Vb/ur10ubksSCZFbHfx8Js0S0m73gB8zaLNlzLXlptSK4tsJoYiQEweOobBEbChIzckyDd8$  
jGevUhSNKGiYsFk/2FJG6XUJ+nSJ15PjGhqXfBCZZUre2wKBgQCQrbNqRwrBYOV31w5ZEf4$  
0D3V91Hv+ot1dIW iN7QovxCGKjqhUOSADJ/AHEqqpW6TgjpKk03oHFYtp1+t+LcNNLQzvJ2$  
JM/b7X4rk7z+C/uQ0icm2H/5w5XuE71prA/75BYVBs5EnlGyykk/xA6miw0H20h5m07CwQ==  
-----END RSA PRIVATE KEY-----
```

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?

Yes No Cancel

```
[root@ip-10-0-1-242 ec2-user]# nano mypvk.pem
[root@ip-10-0-1-242 ec2-user]# chmod 400 mypvk.pem
[root@ip-10-0-1-242 ec2-user]# ssh ec2-user@10.0.2.143 -i mypvk.pem
The authenticity of host '10.0.2.143 (10.0.2.143)' can't be established.
ECDSA key fingerprint is SHA256:nKJpkzqM0J/n2YDYg zgS4fWqHMeaaqgi0d56iI0W
1zE.
ECDSA key fingerprint is MD5:4e:73:55:31:72:aa:41:fa:e9:38:e2:c2:7d:9c:7
d:0a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.143' (ECDSA) to the list of known hos
ts.

--| --|- )
-| ( / Amazon Linux AMI
---|\---|---|
```

<https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/>

```
[ec2-user@ip-10-0-2-143 ~]$ sud
```

```
[root@ip-10-0-2-143 ec2-user]# yum update -y
Loaded plugins: priorities, update-motd, upgrade-helper
```

```
yum-config-manager --disable <repoid>

4. Configure the failing repository to be skipped, if it is unavailable.
   Note that yum will try to contact the repo. when it runs most commands,
   so will have to try and fail each time (and thus, yum will be be
   much
   slower). If it is a very temporary problem though, this is often
   a nice
   compromise:

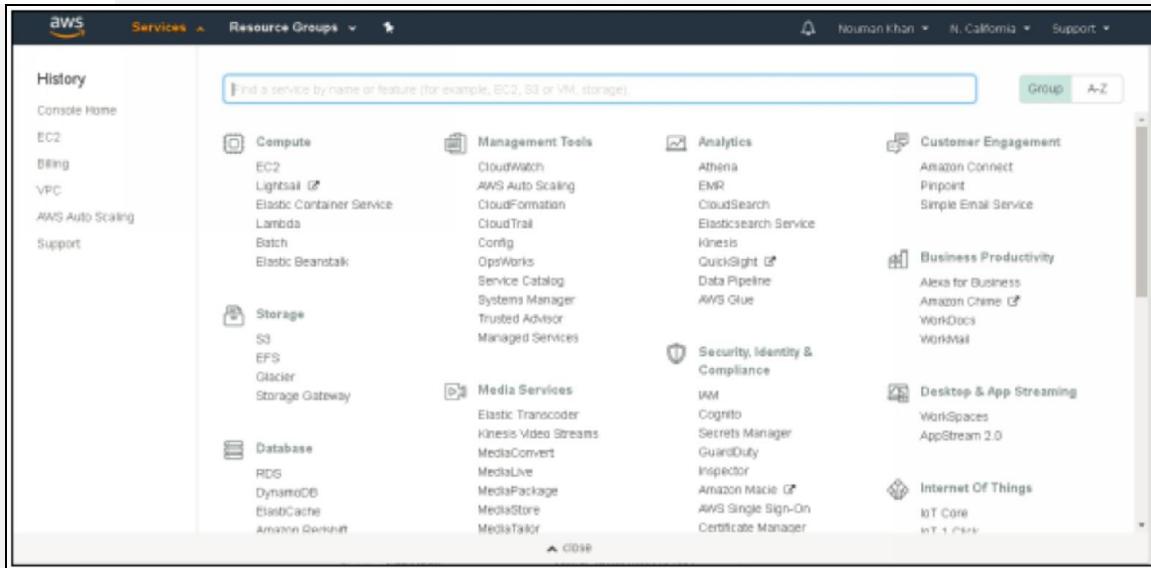
   yum-config-manager --save --setopt=<repoid>.skip_if_unavaila
ble=true

Cannot find a valid baseurl for repo: amzn-main/latest
[root@ip-10-0-2-143 ec2-user]#
```

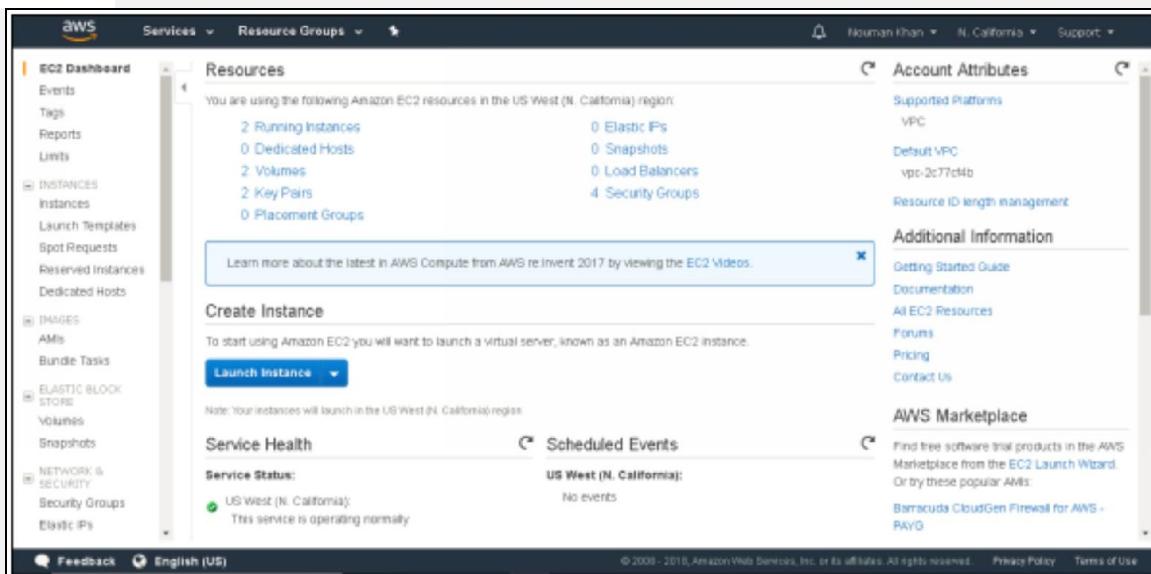
Lab 7.4 Creating a NAT instance

In this lab, we are going to create NAT instances and NAT gateways.

1. Login into AWS console and go to services and click on “EC2”.



2. Click on “Launch an Instance.”



Observe the following screen by clicking on launch an instance.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs
- Free tier only ⓘ

AMI Name	Description	Root device type	Virtualization type	Select
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-2511045	The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.	i3	HVM	64-bit Select
Amazon Linux 2 LTS Candidate 2 AMI (HVM), SSD Volume Type - ami-00d8c880	Amazon Linux 2 LTS Candidate 2 provides an updated version of the Linux Kernel (4.14) tuned for EC2, systemd support, a newer compiler (gcc 7.3), an updated C runtime (glibc 2.26), newer tooling (binutils 2.29.1), and the latest software packages through the extras mechanisms.	i3	HVM	64-bit Select
SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-8342420	SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type: Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.	i3	HVM	64-bit Select

Feedback English (US) © 2006 - 2018, Amazon Web Services, Inc., or its affiliates. All rights reserved. Privacy Policy Terms of Use

3. Choose the Amazon Machine from “Community AMI” by typing “NAT” on the search bar. Select the very first instance from the instance list.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs
- Operating systems

Search: nat [x]

23 results for "nat" on AWS Marketplace

Partner software pre-configured to run an AWS

AMI Name	Description	Root device type	Virtualization type	Select
amazon-ami-vpc-nat-hvm-2016.03.3.x86_64-ebs - ami-004b0f60	Amazon Linux AMI 2016.03.3 x86_64 VPC NAT HVM EBS	i3	HVM	64-bit Select
amazon-ami-vpc-nat-hvm-2017.09.1-testengids.20180307-x86_64-ebs - ami-042f760534b59f20	Amazon Linux AMI 2017.09.1-testengids.20180307 x86_64 VPC NAT HVM EBS	i3	HVM	64-bit Select
amazon-ami-vpc-nat-hvm-2016.03.0.x86_64-ebs - ami-0d087a5d	Amazon Linux AMI 2016.03.0 x86_64 VPC NAT HVM EBS	i3	HVM	64-bit Select
amazon-ami-vpc-nat-hvm-2018.03.0.20180508-x86_64-ebs - ami-1a0e107a	Amazon Linux AMI 2018.03.0.20180508 x86_64 VPC NAT HVM EBS	i3	HVM	64-bit Select

Feedback English (US) © 2006 - 2018, Amazon Web Services, Inc., or its affiliates. All rights reserved. Privacy Policy Terms of Use

4. Click on “Configure Instance Details.”

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GB memory, EBS only)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	No	Low to Moderate	Yes
General purpose	t2.micro <small>Preferred engine</small>	1	1	EBS only	No	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	No	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	No	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	No	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	No	Moderate	Yes
General purpose	t2.2xlarge	8	32	EBS only	No	Moderate	Yes
General purpose	t2.3xlarge	12	48	EBS only	No	Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) © 2006 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Everything remains as default except the network and subnet to configure an instance. You have to change the subnet and the network from default to those which are made by you. Click on “Add Storage.”

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-5dc4b13a | IPS_VPC Create new VPC

Subnet: subnet-4f99014 | 10.0.1.0/24 | us-west-1a | us-west-1a Create new subnet
250 IP Addresses available

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: None Create new IAM role

Shutdown behavior: Stop

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy

Cancel Previous Review and Launch Next: Add Storage

Feedback English (US) © 2006 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Now click on “Add Tags.”

The screenshot shows the AWS EC2 instance creation process at Step 4: Add Storage. The page title is "Step 4: Add Storage". Below it, a note states: "Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)"

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (Mbps)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0fe6f730	8	Magnetic	N/A	N/A	Yes	Not Encrypted

A button labeled "Add New Volume" is visible. A callout box contains the following text:
General Purpose (SSD) volumes provide the ability to burst to 3000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB. Set my root volume to General Purpose (SSD).
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more about free usage tier eligibility and usage restrictions.](#)

At the bottom are buttons for "Cancel", "Previous", "Review and Launch" (which is highlighted in blue), and "Next: Add Tags".

7. Name the key and value of instance by your desire. Now click on “Configure Security groups.”

The screenshot shows the AWS EC2 instance creation process at Step 5: Add Tags. The page title is "Step 5: Add Tags". Below it, a note states: "A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more about tagging your Amazon EC2 resources.](#)"

Key	Value	Instances	Volumes
NAT name	NAT instance	0	0

A button labeled "Add another tag" is visible. A note below it says "(Up to 50 tags maximum)". At the bottom are buttons for "Cancel", "Previous", "Review and Launch" (which is highlighted in blue), and "Next: Configure Security Group".

8. Select the security groups from existing security groups and click on the security group which you have made already in the previous lab. In this security group, we have configured SSH and HTTP traffic. So click on “Review and Launch.”

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-5fa79227	default	default VPC security group	Copy to new
sg-07a1601	IPB_80	Launch-wizard-1 created 2018-05-19T16:42:56.300+00:00	Copy to new
sg-3fb4747	RDS_SG	RDS_SG	Copy to new

Inbound rules for sg-3fb4747 (Selected security groups: sg-3fb4747)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	10.0.1.0/24	
SSH	TCP	22	10.0.1.0/24	
Custom TCP Rule	TCP	3306	10.0.1.0/24	

[Cancel](#) [Previous](#) [Review and Launch](#)

[Feedback](#) [English \(US\)](#)

© 2006 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

- After clicking on Review and launch, you have the following screen. Click on “Launch.”

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.

AMI Details

arn:aws:ami:ami-033a36c64cfb4 - ami-964b9f99
Amazon Linux AMI 2018.03.3 (x86_64) VPC NAT (x86_64)
Root Block Device: /dev/xvda1

Instance Type

Instance Type	EC2s	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security Group ID	Name	Description
sg-3fb4747	RDS_SG	RDS_SG

All selected security groups inbound rules

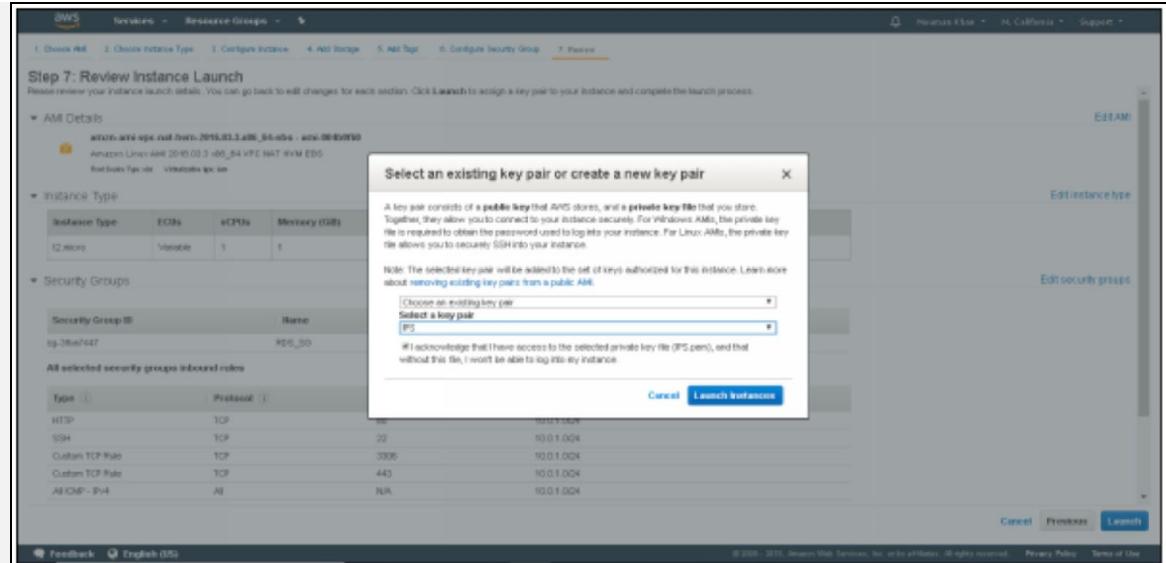
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	10.0.1.0/24	
SSH	TCP	22	10.0.1.0/24	
Custom TCP Rule	TCP	3306	10.0.1.0/24	
Custom TCP Rule	TCP	443	10.0.1.0/24	
All ICMP - IPv4	All	N/A	10.0.1.0/24	

[Cancel](#) [Previous](#) [Launch](#)

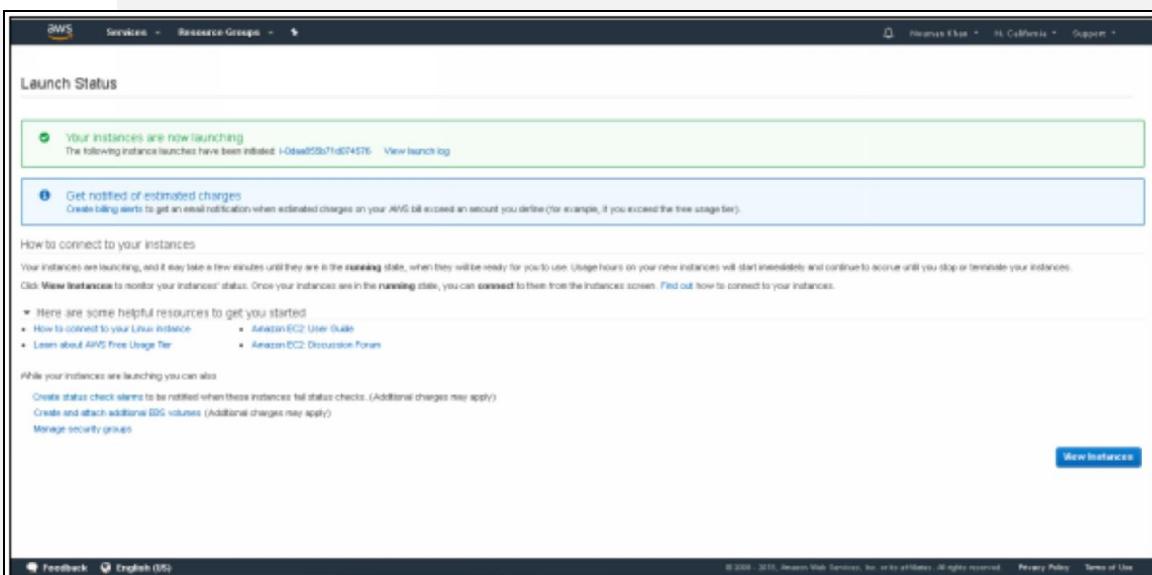
[Feedback](#) [English \(US\)](#)

© 2006 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

- By clicking on “Launch,” you have an option to choose a key pair or create a new key pair. By selecting on “Choose an existing key pair, choose the previous key pair that you had used in the previous lab. Now click on “Launch Instances.”



11. You have the following screen, now click on “view instances.”



12. Now go to security groups and select the security group that you had made.

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. A table lists four security groups:

Name	Group ID	Group Name	VPC ID	Description
sg-07af8f1	sg-07af8f1	IPB_0_P	vpc-0646152a	manually created 2018-09-18T10:42:36.300+03:00
sg-09ea74d7	sg-09ea74d7	IPB_1_P	vpc-0646152a	IPB_1_P
sg-09a70927	sg-09a70927	datast1	vpc-077c464b	datast1 VPC security group
sg-a02494d	sg-a02494d	datast1	vpc-077c464b	datast1 VPC security group

Below the table, a modal window titled 'Edit' displays the 'Inbound Rules' for the selected security group (IPB_0_P). It contains four rules:

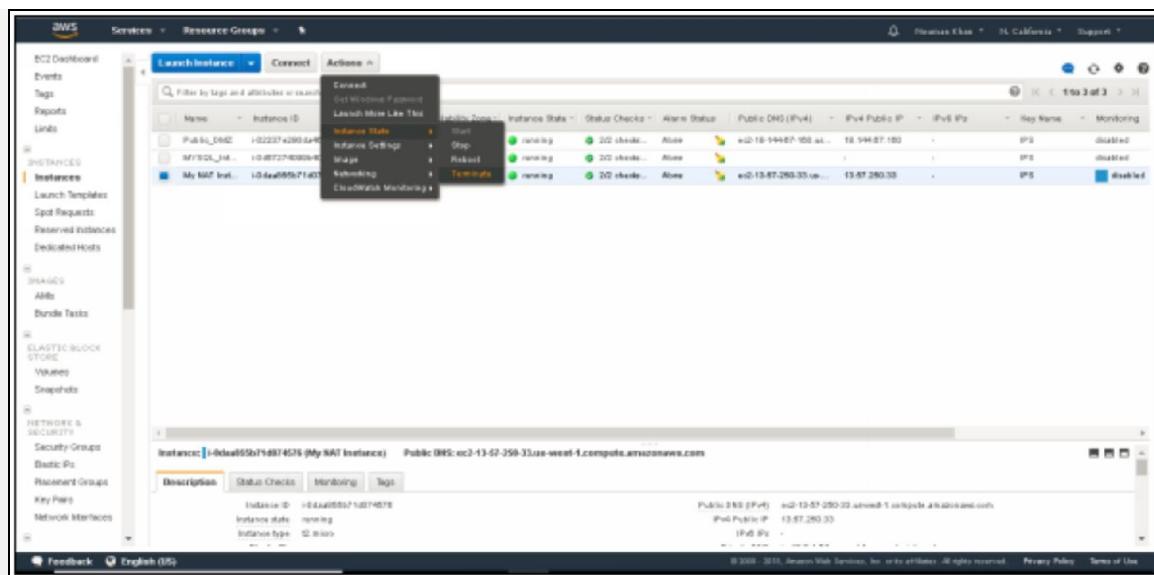
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
SSH	TCP	22	0.0.0.0/0	
HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Click on “Edit” and then allow the “HTTPS” traffic, because you don’t have access to “HTTPS” traffic. Click on save.

The screenshot shows the same EC2 Dashboard and security group list as the previous image. The 'Edit' modal is now open, showing the modified inbound rules for the 'IPB_0_P' security group. The 'HTTPS' rule has been added with port 443 and a custom source of '0.0.0.0/0'.

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	0.0.0.0/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

13. Now click on EC2 Dashboard. Select the instance and don't open it, click actions and go to networking and then attach this instance to the security group.



14. After terminating the instance, login to the putty server by using the way in the previous lab.

15. Now go to the AWS console and create the NAT gateways. For this purpose, you have to select “NAT” for passing internet traffic for IPV4 and select “Egress only NAT gateways” for passing internet traffic for IPV6.

Lab 7.5 Network ACLs Vs. Security groups

In this lab, we are going to tell you how private subnets and public subnets associate with NAT and Security Groups. For this purpose, you have to go through the following steps as mentioned below;

1. Log in to the AWS Management Console and click on VPC dashboard.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Your VPCs' section, 'Network ACLs' is selected. The main content area displays various VPC resources: 2 VPCs, 0 Egress-only Internet Gateways, 4 Route Tables, 1 Elastic IP, 4 Security Groups, 0 Endpoints, 0 NAT Gateways, 0 Peering Connections, 4 Subnets, 2 Internet Gateways, 0 VPC Peering Connections, 1 Nat Gateway, 2 Running Instances, 0 Virtual Private Gateways, and 1 DHCP Options Set. A 'VPN Connections' section is also present. On the right side, there's a 'Service Health' panel showing 'Amazon VPC - US West (N. California)' and 'Amazon EC2 - US West (N. California)' both operating normally. Below that is an 'Additional Information' section with links to VPC Documentation, All VPC Resources, Forums, and Report an Issue.

2. Now, click on Network ACLs from the list of services. Select any default Network ACL.

The screenshot shows the 'Network ACLs' page. The left sidebar has 'Network ACLs' selected. The main area lists two Network ACLs: 'acl-867e7991' (selected) and 'acl-9809bb'. Both are associated with '2 Subnets' and marked as 'Default'. The 'acl-867e7991' row is expanded, showing its summary: 'Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.' Below this are tabs for 'Summary', 'Inbound Rules', 'Outbound Rules', 'Subset Associations', and 'Tags'. At the bottom, there are 'Edit' and 'Delete' buttons, and a dropdown menu set to 'All rules'.

3. After selecting the ACL, click on Inbound Rules and Outbound rules, and check that traffic is allowed on this subnet.

Rule #	Type	Protocol	Port Range	Source	Action / Deny
100	All Traffic	All	All	0.0.0.0/0	ALLOW
101	All Traffic	All	All	0.0.0.0/0	DENY

4. Now click on “Create Network ACL.” Click on “Yes Create.”

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: My test ACL
VPC: vpc-3c77cf4c

Cancel Next Step

5. Following are the Network Access lists. Select the Network ACL.

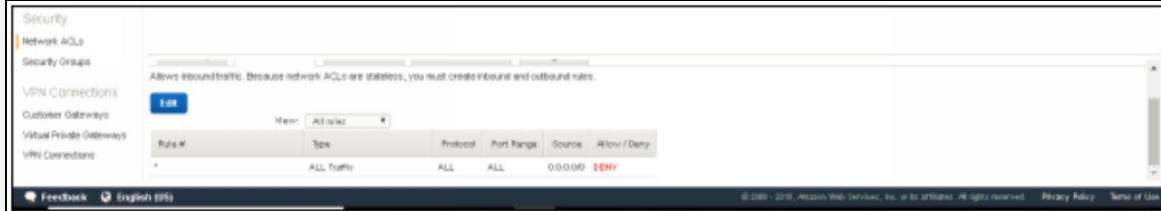
Name	Network ACL ID	Associated With	Default	VPC
aut-0f7jw81	aut-0f7jw81	2 Subnets	Yes	vpc-049612a US_VPC
aut-095949b	aut-095949b	2 Subnets	Yes	vpc-2177cf4b
aut-034a64b	aut-034a64b	0 Subnets	No	vpc-2177cf4b
My test ACL	aut-20dad4fc	0 Subnets	No	vpc-2177cf4b

My test ACL

Summary Inbound Rules Outbound Rules Subnet Associations Tags

Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

6. Click on inbound traffic, and observe that private subnets are no more allowing traffic, its denying. So when you create a private subnet, it denies all the services including inbound traffic, outbound traffic.



The screenshot shows the AWS Network ACLs configuration page. Under the 'Network ACLs' tab, there is one rule listed:

Rule #	Type	Protocol	Port Range	Source	Action
1	Allow	All	All	0.0.0.0	DENY

A note above the table states: "Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules."

7. Now login to your Public Web Server. Type “**yum install httpd -y**” and press enter.

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu May 24 18:19:52 2018 from 113.203.158.89

           _\   _/ 
          _\ \ /_ 
Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
6 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-168 ~] $ yum install httpd -y
```

8. Observe the following windows.

```
amzn-updates | 2.5 kB 00:00
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.2.34-1.16.amzn1 will be installed
---> Processing Dependency: httpd-tools = 2.2.34-1.16.amzn1 for package:
httpd-2.2.34-1.16.amzn1.x86_64
---> Processing Dependency: apr-util-ldap for package: httpd-2.2.34-1.16.
amzn1.x86_64
---> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-
2.2.34-1.16.amzn1.x86_64
---> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.2
.34-1.16.amzn1.x86_64
---> Running transaction check
---> Package apr.x86_64 0:1.5.1-1.12.amzn1 will be installed
---> Package apr-util.x86_64 0:1.4.1-4.17.amzn1 will be installed
---> Package apr-util-ldap.x86_64 0:1.4.1-4.17.amzn1 will be installed
---> Package httpd-tools.x86_64 0:2.2.34-1.16.amzn1 will be installed
```

Package	Arch	Version	Repository	Size
httpd	x86_64	2.2.34-1.16.amzn1	amzn-updates	1.2 M
Installing for dependencies:				
apr	x86_64	1.5.1-1.12.amzn1	amzn-main	116 k
apr-util	x86_64	1.4.1-4.17.amzn1	amzn-main	87 k
apr-util-ldap	x86_64	1.4.1-4.17.amzn1	amzn-main	17 k
httpd-tools	x86_64	2.2.34-1.16.amzn1	amzn-updates	80 k

Transaction Summary

```
=====
Install 1 Package (+4 Dependent packages)
```

```
Total download size: 1.5 M
```

```
Installed size: 3.6 M
```

```
Downloading packages:
```

```
httpd-tools      x86_64      2.2.34-1.16.amzn1      amzn-updates      80 k

Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 1.5 M
Installed size: 3.6 M
Downloading packages:
(1/5): httpd-tools-2.2.34-1.16.amzn1.x86_64.rpm | 80 kB  00:00
(2/5): apr-util-1.4.1-4.17.amzn1.x86_64.rpm       | 87 kB   00:00
(3/5): apr-util-ldap-1.4.1-4.17.amzn1.x86_64.rpm  | 17 kB   00:00
(4/5): apr-1.5.1-1.12.amzn1.x86_64.rpm           | 116 kB  00:00
(5/5): httpd-2.2.34-1.16.amzn1.x86_64.rpm        | 1.2 MB  00:02
-----
Total                                         723 kB/s | 1.5 MB  00:02
Running transaction check
[
```

```
amzn-updates                                         | 2.5 kB    00:00
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.2.34-1.16.amzn1 will be installed
---> Processing Dependency: httpd-tools = 2.2.34-1.16.amzn1 for package:
httpd-2.2.34-1.16.amzn1.x86_64
---> Processing Dependency: apr-util-ldap for package: httpd-2.2.34-1.16.
amzn1.x86_64
---> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd
-2.2.34-1.16.amzn1.x86_64
---> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.2
.34-1.16.amzn1.x86_64
---> Running transaction check
---> Package apr.x86_64 0:1.5.1-1.12.amzn1 will be installed
---> Package apr-util.x86_64 0:1.4.1-4.17.amzn1 will be installed
---> Package apr-util-ldap.x86_64 0:1.4.1-4.17.amzn1 will be installed
---> Package httpd-tools.x86_64 0:2.2.34-1.16.amzn1 will be installed
[
```

9. Now type service httpd start.

```
Complete!
[root@ip-10-0-1-242 ec2-user]# service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for ip-10-0-1-242
httpd: Could not reliably determine the server's fully qualified domain
name, using 127.0.0.1 for ServerName
[ OK ]
[root@ip-10-0-1-242 ec2-user]# cd /var/www/html
```

10. Now clear the screen.

```
[root@ip-10-0-1-242 html]# chkconfig httpd on  
[root@ip-10-0-1-242 html]# nano index.html
```

References

AWS Cloud Certifications

- <https://aws.amazon.com/certification/>
- <https://cloudacademy.com/blog/choosing-the-right-aws-certification/>

AWS Certified SysOps Admin Associate

- <https://aws.amazon.com/certification/certified-sysops-admin-associate/>

Cloud Concepts

- <https://aws.amazon.com/what-is-cloud-computing/>
- <https://aws.amazon.com/types-of-cloud-computing/>

Cloud Compliance

- <https://aws.amazon.com/compliance/>

Identity and Access Management

- <https://aws.amazon.com/iam/>

Security Support

- <https://aws.amazon.com/products/security/>

Cloud Deployment and Management

- <https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf>

AWS Global Infrastructure

- <https://cloudacademy.com/blog/aws-global-infrastructure/>

AWS Compute

- <https://aws.amazon.com/products/compute/>

AWS Storage

- <https://aws.amazon.com/products/storage/>

AWS Database

- <https://aws.amazon.com/products/databases/>

Amazon Virtual Private Cloud

- https://en.wikipedia.org/wiki/Virtual_private_cloud
- <https://aws.amazon.com/vpc/>

Network & Content Delivery

- <https://aws.amazon.com/cloudfront/details/>
- <https://aws.amazon.com/elasticloadbalancing/>
- <https://aws.amazon.com/route53/>

AWS Free Tier

- <https://aws.amazon.com/free/>

AWS Support Plans

- <https://aws.amazon.com/premiumsupport/compare-plans/>

AWS Organizations

- <https://aws.amazon.com/organizations/>

AWS Cost Calculators

- <https://calculator.s3.amazonaws.com/index.html>
- <https://awstcocalculator.com/>