AWS DevOps Engineer Professional Certification Guide

Hands-on guide to understand, analyze, and solve 150 scenario-based questions

Sumit Kapoor

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55516-657

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

www.bpbonline.com

Dedicated to

My beloved wife:
Joohi
and
My daughters ***Khushi*** *and* ***Avni***

About the Author

With over 20 years of experience in software development and cloud infrastructure, **Sumit Kapoor** is a Lead DevOps Engineer at Clarivate, a global leader in providing trusted insights and analytics to accelerate the pace of innovation. Sumit is an AWS Certified Architect, a Certified Kubernetes Application Developer (CKAD), and a skilled practitioner of cloud-native architecture, infrastructure as code (IaC), and continuous integration and delivery (CI/CD).

Sumit has a proven track record of delivering consistently on both large-scale and start-up environments, leveraging his expertise in Kubernetes, Docker, Terraform, Jenkins, and other cutting-edge technologies. Sumit is passionate about enabling and empowering teams to build, deploy, and operate scalable, reliable, and secure applications on the cloud, while optimizing performance, efficiency, and cost. Sumit is always eager to learn new skills, explore new challenges, and contribute to the advancement of the DevOps community.

About the Reviewer

Arvind Singh is an accomplished DevOps engineer with comprehensive expertise in cloud-native solutions and a broad spectrum of tools, including AWS, Azure, Ansible, Kubernetes, and Terraform. His profound knowledge of cloud technologies and Infrastructure as Code (IaC) empowers him to catalyze organizational changes. Arvind is committed to operational excellence, focusing on automating workflows to enhance efficiency and productivity. In addition to his professional endeavors, he actively engages in non-fiction and IT literature, serving as a technical reviewer for publications focusing on DevOps, CI/CD, and Kubernetes.

Acknowledgement

I want to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout this book's writing, especially my wife Joohi and my daughters Khushi and Avni.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues and co-worker during many years working in the tech industry, who have taught me so much and provided valuable feedback on my work. Special thanks to my ex-colleague, Chaitanya Tirumala, and my current colleagues, Mohit Raina and Nilesh Patil, for their support and insights that have been instrumental in my professional growth.

I would also like to acknowledge the valuable contribution of the technical reviewer of this book, Arvind Singh. He has reviewed the book for technical errors and has given important suggestions to improve quality of this book.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

DevOps is the combination of cultural philosophies, engineering practices, and tools that increase an organization's ability to deliver applications and services at high velocity and better quality. Over time, several essential practices have emerged when adopting DevOps: **continuous integration** (**CI**), **continuous delivery** (**CD**), **Infrastructure as Code** (**IaC**), and monitoring and logging.

AWS DevOps Engineer Professional Certification Guide is one of the more elaborate and demanding cloud certifications. Organizations with these qualified professionals can ensure speedy delivery of secure, compliant, systems that are highly available and scalable. Moreover, job listings requiring this certification have significantly increased over the past few years.

This book is designed to provide a comprehensive guide to provisioning, operating, and managing distributed systems and services on AWS. It covers a wide range of topics, including implementing and managing continuous delivery systems and methodologies on AWS, implementing and automating security controls, governance processes, and compliance validation, defining and deploying monitoring, metrics, and logging systems on AWS, and implementing systems that are highly available, scalable, and self-healing on AWS. Additionally, it addresses designing, managing, and maintaining tools to automate operational processes on AWS.

This book is intended for all IT professionals who have a basic understanding and experience of the AWS environment and looking forward to stepping into the DevOps domain as an AWS Certified DevOps Engineer. Its particularly attractive to candidates who have failed in the 1st and 2nd attempts of the exams, which comprise more than 50% of the candidates.

Chapter 1: Continuous Integration with CodeCommit and CodeBuild - The chapter explains everything needed for the reader to create a CodeCommit repository to store the code and files securely for the project, including the usage of different CodeCommit CLI commands such as clone a repository, tag repository, create pull request, list pull requests, creating an approval rule for pull request by taking different examples. The chapter also goes into greater detail about AWS CodeBuild service by taking two real-world use-cases. Furthermore, the chapter highlights the benefits of integrating AWS CodeGuru, AWS App Runner, AWS CloudShell, and AWS CodeArtifact services into our projects to enhance our DevOps capabilities.

Chapter 2: Continuous Delivery with CodeDeploy and CodePipeline - The chapter presents a detailed overview of the concepts and principles behind AWS CodeDeploy, its various deployment strategies, and how it simplifies the software deployment process to EC2 instances and AWS ECS services. The chapter emphasizes the importance of implementing a blue/green deployment strategy using AWS CodeDeploy to ensure zero downtime and seamless deployment to Amazon ECS. Additionally, the chapter explains everything to develop a CI/CD pipeline with AWS CodePipeline for S3 websites, integrating AWS CodeCommit, AWS CodeBuild, and AWS CodeDeploy services to produce fast and reliable application and infrastructure software releases.

Chapter 3: Cross-Account CI/CD Pipelines and Testing - The chapter covers how to construct a cross-account CI/CD pipeline, showcasing how AWS services can be utilized in a multi-account setup to streamline deployment processes across distinct environments. The chapter guides us through building golden images with EC2 Image Builder, simplifying the creation and management of Amazon Machine Images (AMIs). Furthermore, the chapter demonstrates automating unit tests and code coverage analysis using AWS CodeBuild and Codecov to ensure our application meets the necessary standards before deployment.

Chapter 4: Infrastructure as Code Using CloudFormation - The chapter allows the readers to learn practical understanding of Infrastructure as Code (IaC) using AWS CloudFormation, enabling them to efficiently create, manage, and update the infrastructure in AWS. Furthermore, the chapter explores various features and techniques, such as nested stacks, Lambda-backed custom resource deployment, differences between CreationPolicy and WaitCondition, cross stack references, stack updates, and drift detection and remediation.

Chapter 5: Automated Account Management and Security in AWS - The chapter provides a comprehensive understanding of deploying automation to create, onboard, and secure AWS accounts in a multi-account/multi-region environment. The readers will gain hands-on experience and in-depth knowledge of various AWS DevOps tools and services, including CloudFormation StackSets, AWS AppConfig, AWS App2Container, AWS Copilot, and AWS Control Tower.

Chapter 6: Automation Using OpsWorks and Elastic Beanstalk - The chapter explains basic concepts of deploying multi-container Docker apps on Elastic Beanstalk, setting them up using .ebextensions, and employing blue/green deployment methods. The readers will also explore AWS API Gateway to create user-friendly interfaces. Through creating an HTTP API using Lambda, DynamoDB, and AWS SAM, the readers will grasp the concept of managing updates via Lambda-based canary deployments. Additionally, the readers will also learn how to handle the deployment of Node.js apps on AWS ECS, refining their skills across a variety of AWS deployment scenarios and boosting your deployment capabilities.

**Chapter 7: Implement High Availability, Scalability, and Fault Tolerance - **The chapter explains with details and numerous practical examples for setting up and testing RDS Multi-AZ, exploring High Availability in Aurora DB, and creating scalable and load-balanced applications. This chapter also allows the reader to learn the basics of AWS EC2 Auto Scaling LifeCycle hooks and its interaction with Lambda functions. By the end of this chapter, the readers will have an in-depth knowledge of these key AWS features and services, empowering them to create more resilient and scalable applications.

Chapter 8: Design and Automate Disaster Recovery Strategies - The chapter is dedicated to the design and automation of disaster recovery strategies on AWS. This chapter covers practical examples of managing data replication across regions with S3 Cross-Region Replication, deploying web applications using AWS EKS, and designing robust disaster recovery strategies with AWS DRS. The chapter also explores how to ensure high availability using CloudFront Origin Failover and Route 53. By the end of this chapter, readers will have developed a comprehensive understanding of these critical AWS features and services, empowering them to create applications that are not only resilient but also capable of quick recovery in the face of disasters.

Chapter 9: Automate Monitoring and Event Management - The chapter focuses on enhancing monitoring and logging in AWS. The chapter explores how to integrate AWS CloudTrail with CloudWatch for improved tracking, and how to use AWS EventBridge with SNS for event-driven applications. Additionally, the readers will learn to publish VPC Flow Logs to S3 and use Athena for querying these logs. By the end of the chapter, the readers will have a solid understanding of these AWS services and how to utilize them for effective log management and event-driven computing in AWS.

Chapter 10: Auditing, Logging and Monitoring Containers and Applications - The chapter covers about Auditing infrastructure using AWS Config, AWS Inspector and assessment templates, analyze logs with CloudWatch logs insights, remediate issues using AWS Config and configure X-Ray for Lambda.

Chapter 11: Troubleshooting and Restoring Operations - The chapter covers strategies for effectively addressing incidents and events across various AWS services. Topics include handling failed deployments, utilizing OpsCenter for streamlined operational tasks, implementing auto-healing in AWS OpsWorks, automating responses to AWS Health events, uncovering root causes using AWS X-Ray, leveraging S3 event notifications, optimizing event distribution with AWS SQS fanout, and establishing robust dead-letter queues in SQS.

Chapter 12: Setup Event-Driven Automated Actions - The chapter explains with details and numerous practical examples about AWS's event-driven automation using Kinesis Firehose, teaching them to integrate CloudWatch logs with DataDog for enhanced monitoring. This chapter also allows the reader to learn about security with AWS Inspector, guiding them through automated vulnerability scans in your AWS account and immediate alerts via AWS SNS. By the end of the chapter, the readers will grasp AWS's automation, security scanning, and threat detection processes.

Chapter 13: Implement Governance Strategies and Cost Optimization - The chapter covers how to perform sensitive data discovery using AWS Macie, integrated AWS WAF integration with AWS CloudFront, automate administrative tasks using AWS Systems Manager and Patch manager. In this chapter, the readers will also learn how to optimize the infrastructure with AWS Trusted advisor and create an AWS organization.

Chapter 14: Advanced Security, Access Control, and Identity Management - The chapter explores advanced security and identity management topics in AWS. The readers will learn how to secure sensitive data using AWS CloudHSM, gain insights into AWS Directory Services, and understand the intricacies of sharing resources through AWS RAM. The readers will also take a deeper dive into the world of intrusion detection using AWS Network Firewall and analyze the differences between RBAC and ABAC. By the end of this chapter, the readers will have a comprehensive grasp of advanced security practices and identity management principles in the AWS cloud environment.

Chapter 15: Mock Exam: 1 – The chapter covers 75 scenario-based questions and their answers with explanations.

Chapter 16: Mock Exam: 2 – The chapter covers 75 scenario-based questions and their answers with explanations.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/sjmb5f8

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/AWS-DevOps-Engineer-Professional-Certification-Guide**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

CHAPTER 1

Continuous Integration with CodeCommit and CodeBuild

Introduction

Continuous Integration and Continuous Delivery/Deployment (**CI/CD**) are two important parts of DevOps. In this process, when commit or changes happen in source code, they go through automated stage gates, all the way from building, testing, deploying the applications from development to production environments, across cloud accounts.

In this chapter, we learn how AWS CodeCommit makes it easy for developers to collaborate on code using secure code repositories and AWS CodeBuild compiles source code, runs tests and produces software packages that are ready to deploy, on a dynamically created build server.

Structure

In this chapter, we will discuss about the following topics in AWS CodeCommit:

- Setup AWS CodeCommit repository
- Security requirements for AWS CodeCommit repository
- Tagging repositories in CodeCommit
- Setting up CodeCommit trigger for SNS Topic
- Setting up CodeCommit trigger for Lambda Function

- Creating a pull request in AWS CodeCommit
- Creating an approver rule for pull request
- AWS CloudShell for DevOps tasks
- Working with private NPM packages in CodeArtifact
- Automated code review with AWS CodeGuru

Here are the topics that we will cover in AWS CodeBuild:

- Building and deploying source code with AWS CodeBuild and S3, building and pushing docker images to AWS **Elastic Container Registry** (**ECR**), and configuring an App Runner service

Objectives

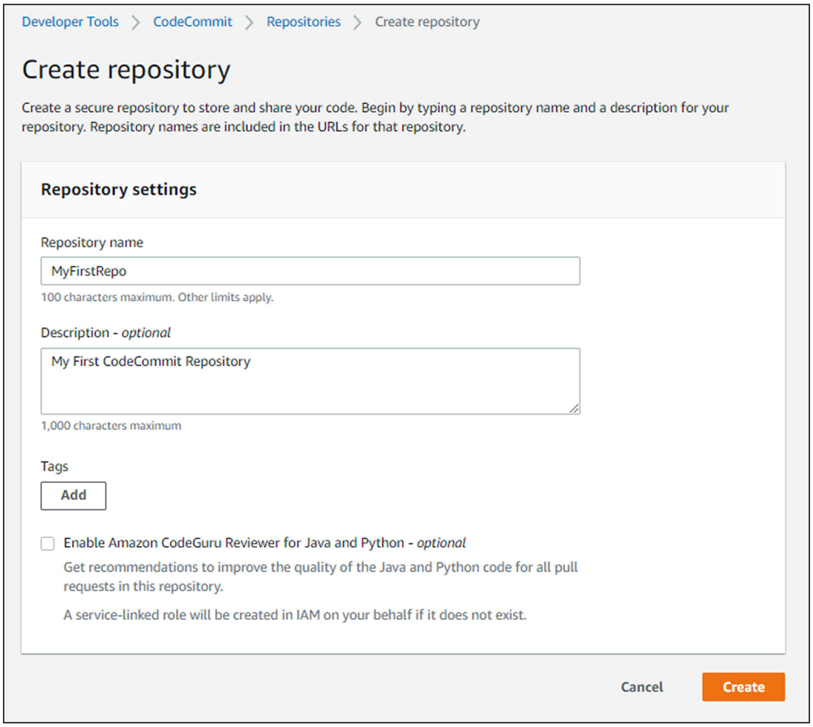
After reading this chapter, we will be able to efficiently set up and manage an AWS CodeCommit repository, implement security measures, and utilize various features like tagging and triggers. We will gain a deep understanding of how to create and manage pull requests and approver rules, as well as how to leverage AWS CloudShell and CodeArtifact for DevOps tasks. Additionally, we will master the art of building and deploying source code with AWS CodeBuild and S3, working with Docker images and AWS ECR, and configuring an App Runner Service. By the end of this chapter, we will have acquired the essential skills and knowledge to implement a seamless CI process using AWS CodeCommit and CodeBuild, ultimately enhancing your proficiency in DevOps and cloud-based development practices.

Setup AWS CodeCommit repository

This section discusses the key features of AWS CodeCommit such as creating a repository, commit files to the repository, cloning a repository, pushing files to the repository, creating pull requests, tagging repositories and adding triggers into the repository.

Creating a Repository using console

We can either use AWS Console or **AWS Command Line Interface** (**AWS CLI**) to create an empty repository. Go to the **Repositories** page from **CodeCommit** console and click on **Create Repository**. Enter the name of **repository**, **description** (optional), add **tags** (optional), select **Enable Amazon CodeGuru Reviewer for Java and Python** (optional), and click **Create** as shown in *Figure 1.1*:



Developer Tools > CodeCommit > Repositories > Create repository

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name

MyFirstRepo

100 characters maximum. Other limits apply.

Description - *optional*

My First CodeCommit Repository

1,000 characters maximum

Tags

□ Enable Amazon CodeGuru Reviewer for Java and Python - *optional*

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

***Figure 1.1**: Create a repository*

Creating a repository using AWS CLI

Use **create-repository** command to create a repository from AWS CLI by providing a unique name for the repository (with **`-repository-name`** option), and some optional comments about the repository (with **`-repository-description`** option):

```
aws codecommit create-repository --repository-name MyFirstRepo \
  --repository-description "My First CodeCommit Repository"
```

If the command is successful, it displays the following output:

```
{
    "repositoryMetadata": {
        "accountId": "958651443844",
        "repositoryId": "8d9914d9-bcb1-4f3a-bbd5-62320273a956",
        "repositoryName": "MyFirstRepo",
        "repositoryDescription": "My First CodeCommit Repository",
        "lastModifiedDate": "2022-10-13T21:32:10.432000+00:00",
        "creationDate": "2022-10-13T21:32:10.432000+00:00",
        "cloneUrlHttp": "https://git-codecommit.us-east-1.amazonaws.com/v1/repos/MyFirstRepo",
        "cloneUrlSsh": "ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/MyFirstRepo",
       "Arn": "arn:aws:codecommit:us-east-1:958651443844:MyFirstRepo"
    }
}
```

Creating a commit

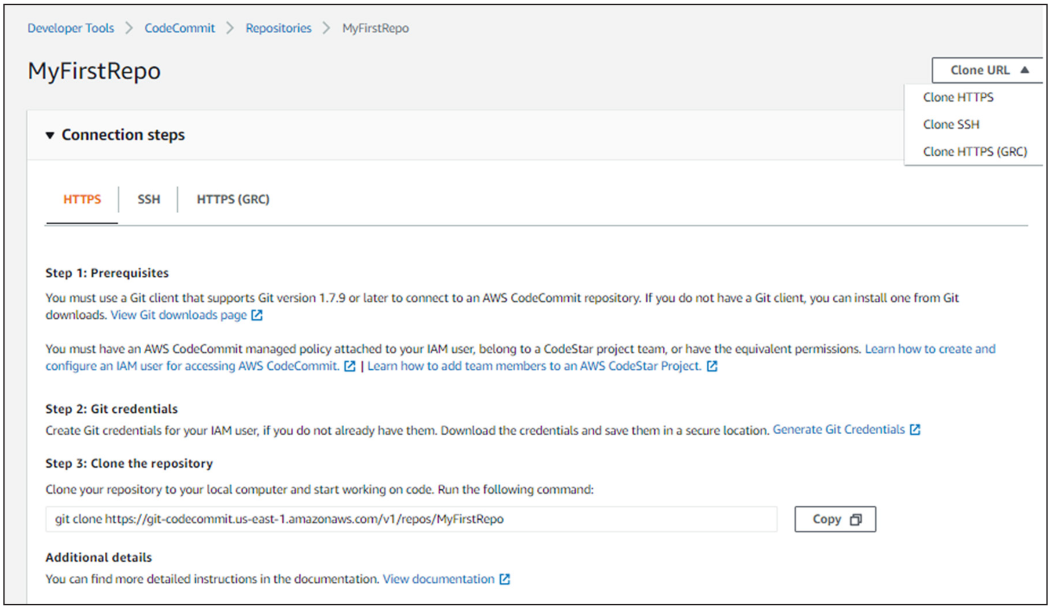
We can use Git client, AWS CLI or CodeCommit Console to create a commit in CodeCommit repository. In this example, we will show how to commit using Git client which is the most preferred method used by the developers to push the changes in AWS CodeCommit repository.

Use the following steps to commit a file in **`MyFirstRepo`** CodeCommit repository:

1. Ensure you are on the right branch otherwise, run **`git branch`** which will display the list of branches. If you are not in the right branch run **`git checkout branch-name`** to switch to the intended branch.
2. Make a change to the branch like adding, updating, or deleting a file. For example, create a test file (**`test.txt`**) with some text: **`Hello to DevOps World!`**.
3. Run **`git status`** command and it will alert you that there are some untracked files.
4. Run **`git add test.txt`**. This command will include a change in the working directory to the staging area.
5. If we run **`git status`**, it will alert that there are some changes to be committed.
6. Finally run **`git commit -m "Some comments to the commit"`**. If we run git status now git will alert that commit is ready to be pushed from local repo to CodeCommit repository.
7. If the changes in the file look good, then run **`git push origin remote-branch`** which will push the file to CodeCommit repository

Cloning a repository

Select the repository which you want to connect and choose one of the following methods shown in the following drop-down populated:



Step 1: Prerequisites

You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. View Git downloads page

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit. | Learn how to add team members to an AWS CodeStar Project.

Step 2: Git credentials

Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. Generate Git Credentials

Step 3: Clone the repository

Clone your repository to your local computer and start working on code. Run the following command:

```
git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/MyFirstRepo
```

Additional details

You can find more detailed instructions in the documentation. View documentation

***Figure 1.2**: Clone a repository*

1. **Clone HTTPS**: Use this option if you want to use your git credentials obtained along with your IAM user details from AWS Administrator:

   `https://git-codecommit.us-east-1.amazonaws.com/v1/repos/MyFirstRepo`

2. **Clone SSH**: Use this option if you want to use SSH public/private key pair with your IAM User:

   `ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/MyFirstRepo`

3. **Clone HTTPS (GRC)**: Use this option if you want to use `git-remote-codecommit` on your local machine. This method is recommended if you want to support connections made with federated access, identity providers, and temporary credentials:

   `codecommit::us-east-1://MyFirstRepo`

Security requirements for AWS CodeCommit repository

CodeCommit repositories are automatically encrypted at rest. No action is needed from the customer in this regard. CodeCommit also encrypts data in transit automatically. The first time we create a repository in CodeCommit, CodeCommit creates an AWS-managed key (aws/codecommit) in the same region and stores it in your AWS account. CodeCommit uses this key to encrypt and decrypt the data in this and other repositories in the same AWS account. Please note that we cannot use a customer-managed key created in AWS KMS Service to encrypt or decrypt data in CodeCommit repositories.