

# MASTER

# AWS PIPELINE



For Data Science Interview

# **\*Disclaimer\***

**Everyone learns uniquely.**

**What matters is developing the problem  
solving ability to solve new problems.**

**This Doc will help you with the same.**

# Introduction

The Data Science tool AWS Pipeline enables automation of continuous integration and continuous delivery (CI/CD) procedures. This guide serves professional users who need to learn and improve their implementation of AWS Pipeline as part of their operational processes.

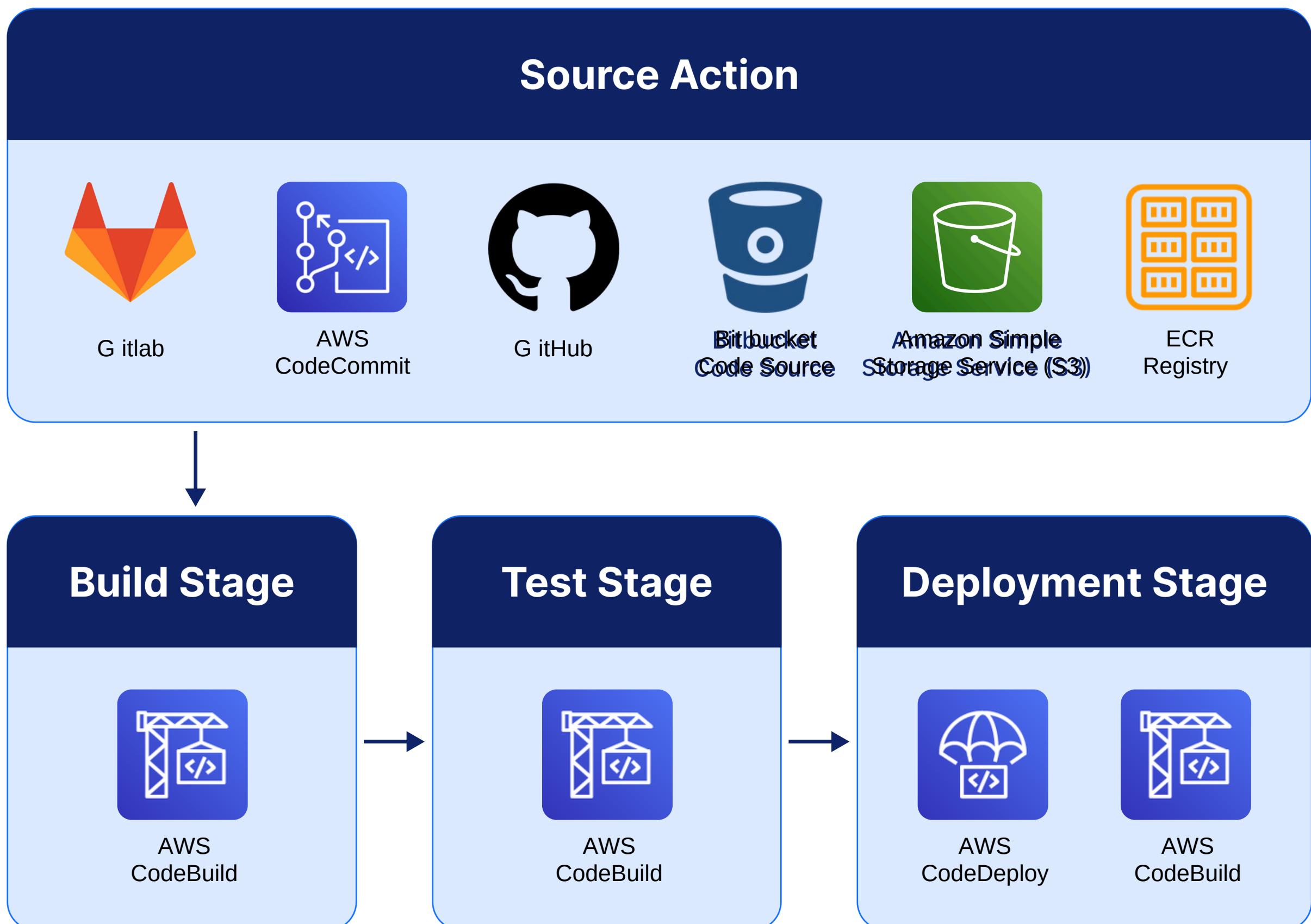
# Understanding AWS Pipeline

The Data Science tool AWS Pipeline enables automation of continuous integration and continuous delivery (CI/CD) procedures. This guide serves professional users who need to learn and improve their implementation of AWS Pipeline as part of their operational processes.

## Key Features:

- Fully managed CI/CD service
- Automates build, test, and deploy processes
- Seamless integration with AWS and third-party tools
- Parallel execution for fast deployments

# AWS Pipeline Components



## Source Stage

- Stores the code repository (AWS CodeCommit, GitHub, or S3)

## Build Stage

- AWS CodeBuild serves to both compile and test the provided code.

The system performs unit tests while creating necessary artifacts

## Test Stage

- Automates functional and integration testing

## Deploy Stage

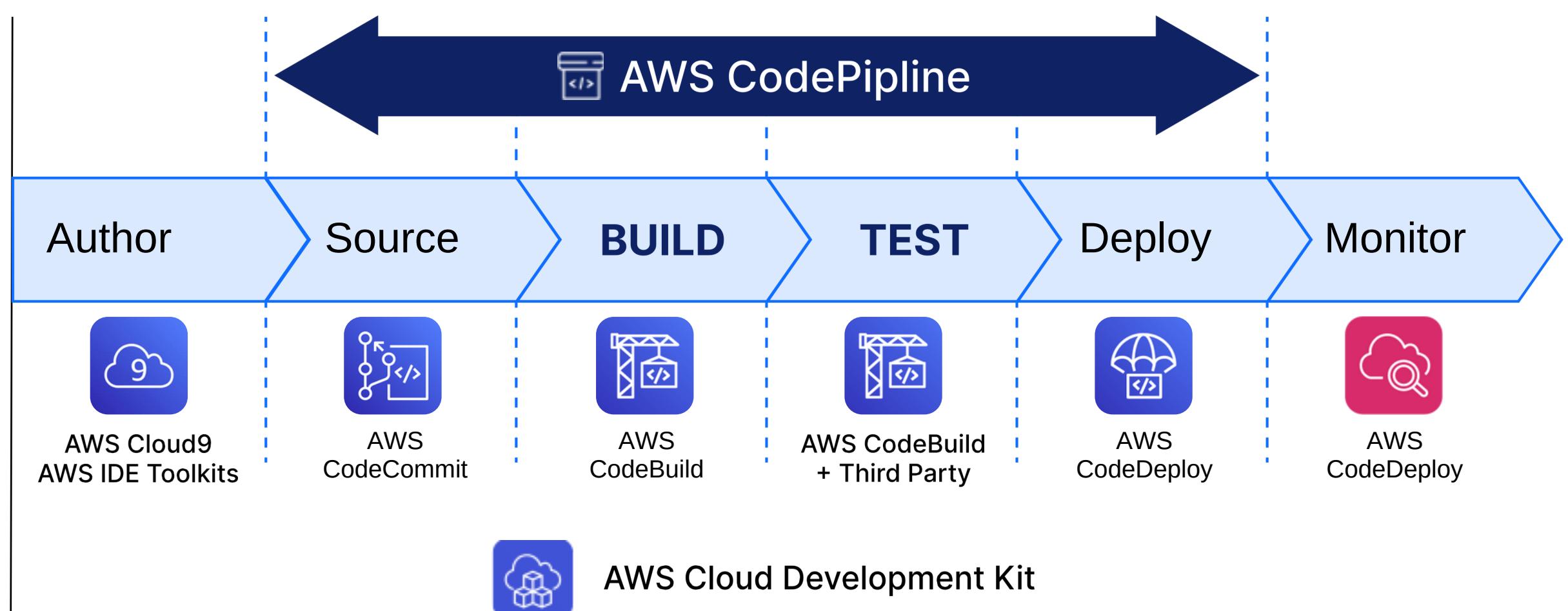
- AWS CodeDeploy automates deployment
- Supports Blue-Green and Rolling deployments

# Setting Up AWS Pipeline

## Step-by-Step Guide:

1. Create a CodeCommit Repository
2. Define a Build Project in CodeBuild
3. Configure Deployment in CodeDeploy
4. Create and Configure the Pipeline
5. Monitor and Debug Issues

## Diagram: AWS CodePipeline Workflow



# Best Practices for Working Professionals

## 1. Use IAM Roles and Policies

- Assign least privilege access to AWS services

## 2. Implement Automated Testing

- Use AWS CodeBuild and third-party testing tools

## 3. Monitor and Log Pipeline Activity

- Use AWS CloudWatch and AWS CloudTrail for logging

## 4. Secure Your Artifacts

- Store build artifacts securely in S3 with encryption

# Common Issues and Troubleshooting

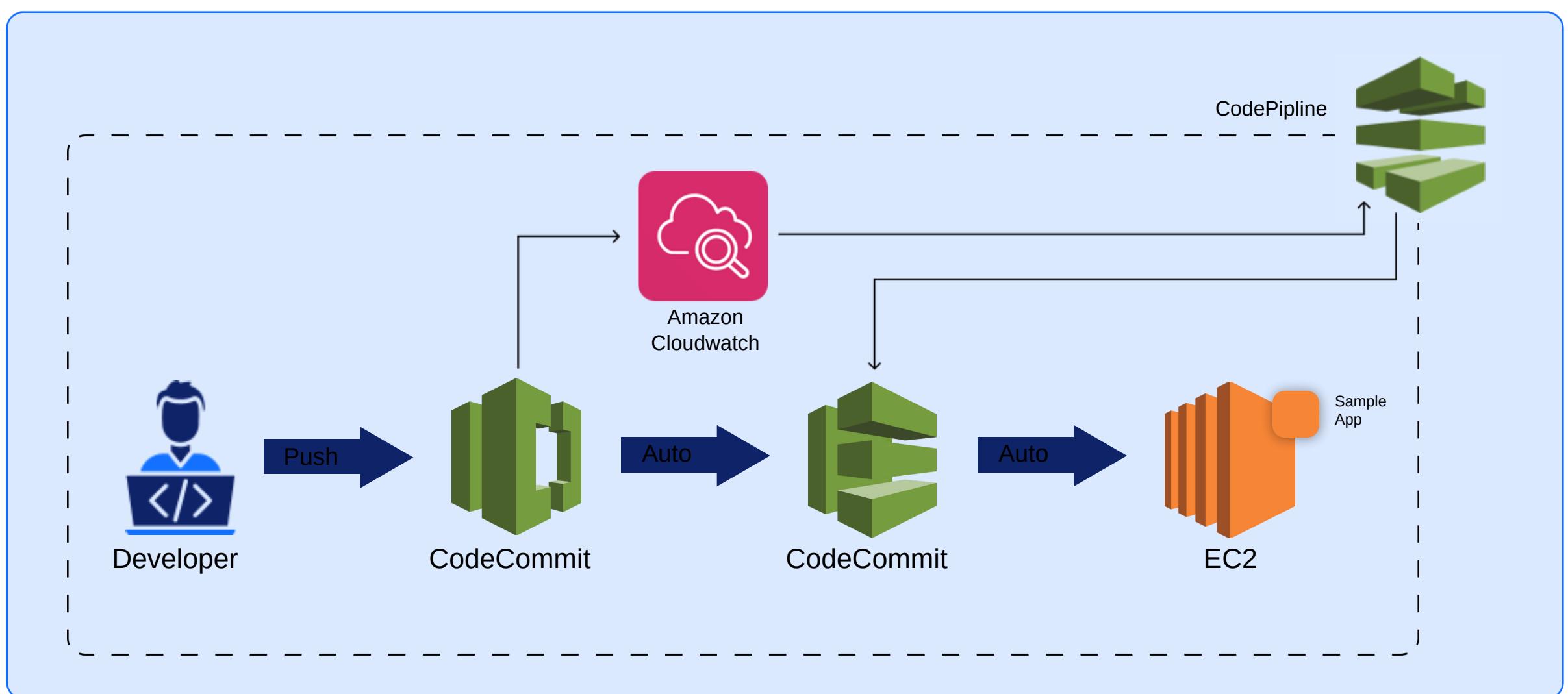
Issue	
Pipeline execution fails	Check IAM permissions & source repo integration
Build stage errors	Verify buildspec.yml configuration
Deployment failures	Review CodeDeploy logs & rollback if needed
Environment variable issues	Ensure correct values & secure storage
Insufficient permissions	Update IAM roles & policies for required access
Artifact upload failures	Check S3 bucket permissions & storage limits

# Hands-on Exercise: Build Your First AWS CodePipeline

**Scenario: Deploy a simple web application using AWS CodePipeline.**

**Steps :**

1. Create a repository in AWS CodeCommit.
2. Set up a build project in AWS CodeBuild.
3. Deploy the application using AWS CodeDeploy.
4. Monitor pipeline execution in AWS CodePipeline.
5. Verify successful deployment on EC2.



# AWS Pipeline Interview Questions (MAANG-Level)

## Question - 1

**How does AWS CodePipeline ensure continuous integration and deployment?**

It automates the software release process by integrating with CodeCommit, CodeBuild, and CodeDeploy, ensuring a streamlined CI/CD workflow.

## Question - 2

**Explain Blue-Green Deployment in AWS CodeDeploy.**

Blue-Green Deployment reduces downtime by switching traffic from an old version (blue) to a new version (green) without disruption.

## **Question - 3**

### **How do you optimize AWS CodePipeline for high availability?**

Use multi-region deployments, set up rollback strategies, and monitor with CloudWatch alarms.

## **Question - 4**

### **What are the benefits of using AWS CodeBuild over Jenkins?**

AWS CodeBuild is fully managed, scalable, integrates with AWS IAM, and doesn't require manual maintenance like Jenkins.

## **Question - 5**

### **How can you secure AWS CodePipeline?**

Use IAM roles, encrypt S3 artifacts, restrict access with AWS KMS, and monitor activity with CloudTrail.

Here are five more AWS CodePipeline interview questions commonly asked in product-based companies, along with brief answers and a diagram for better understanding.

## Question - 6

### How does AWS CodePipeline integrate with third-party tools?

CodePipeline enables AWS customers to integrate GitHub and Bitbucket with additional services through custom actions or AWS Lambda. The third-party integration capabilities of AWS CodePipeline enable team members to use CI/CD processes that exceed AWS service limits.

## Question - 7

### What is the role of AWS CodeDeploy in a CI/CD pipeline?

AWS CodeDeploy implements automation for deployment tasks by managing updates across EC2 instances as well as Lambda functions and ECS. It provides zero-downtime deployment management through three methods including **Rolling deployment, Canary and Blue-Green deployment.**

Diagram :

```
Source (GitHub/CodeCommit) -> Build (CodeBuild)  
-> Deploy (CodeDeploy)
```

## Question - 8

### What are the differences between AWS CodePipeline and Jenkins?

Feature	AWS CodePipeline	Jenkins
Management	Fully managed	Requires manual setup
Scalability	Auto-scales	Manual configuration
Security	Integrated with AWS IAM	Requires plugins
Pricing	Pay-as-you-go	Free but needs infra

AWS CodePipeline is recommended for AWS-centric environments, while Jenkins provides more flexibility in hybrid/multi-cloud setups.

## Question - 9

### How do you implement rollback in AWS CodePipeline?

AWS CodeDeploy enables rollbacks through automatic trigger mechanisms based on CloudWatch alarms which you can set up to implement. The automatic rollback feature allows the pipeline to restore the previous stable version in case of detected issues

Deploy New Version -> Monitor Health -> Revert to Previous Version (if failure detected)

## Question - 10

**What are the different deployment strategies in AWS CodeDeploy?**

Strategy	Description
Rolling	Deploys to a few instances at a time, reducing risk.
Blue-Green	Switches traffic from the old to the new version without downtime
Canary	Deploys to a small subset first before rolling out completely.

Each strategy helps reduce downtime and improves deployment safety.

# Conclusion

AWS Pipeline has numerous advantages related to deploying applications, maintaining efficiency, and cutting down on manual work. Knowing the baseline components, recommendations, and potential interview questions, working professional will be ready to integrate or improve the CI/CD in AWS.