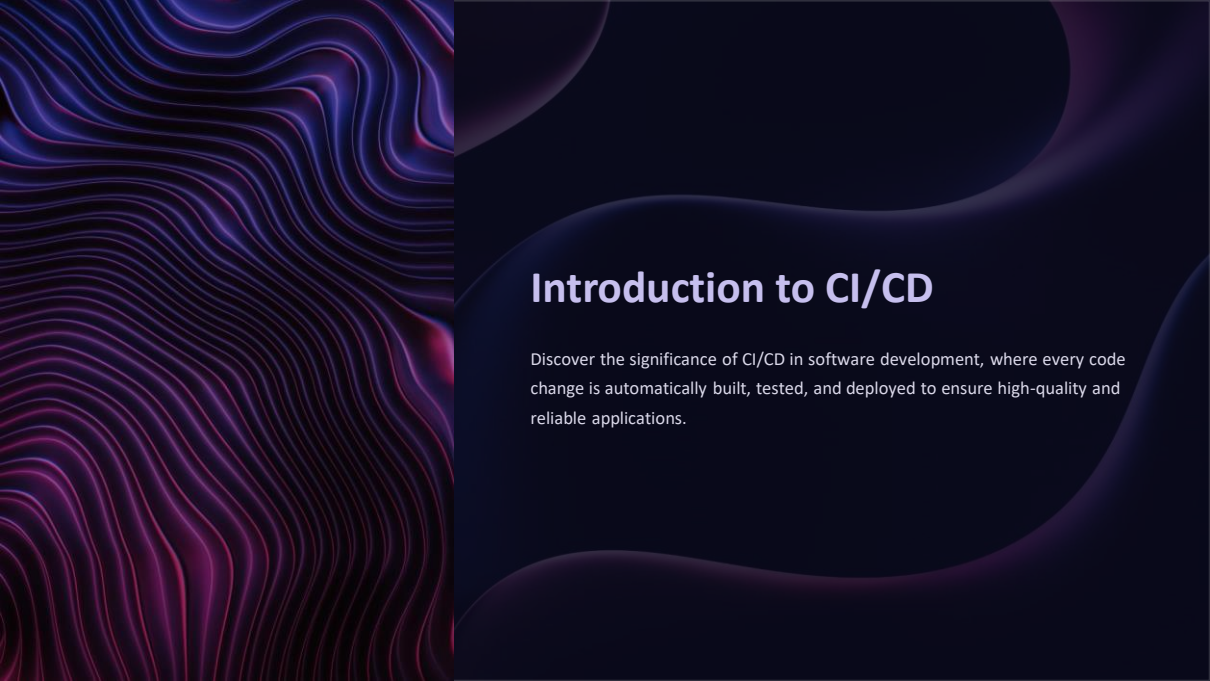


AWS CI/CD Pipeline For Python App

Built-in continuous integration and continuous deployment (CI/CD) features of AWS enable developers to automate the entire software delivery process on the cloud platform.

by Gadagoju Shiva



Introduction to CI/CD

Discover the significance of CI/CD in software development, where every code change is automatically built, tested, and deployed to ensure high-quality and reliable applications.

Overview of AWS CI/CD Pipeline

Components

Understand the key components of an AWS CI/CD pipeline, including CodeCommit, CodeBuild, CodeDeploy, and CodePipeline.

Stages

Follow the sequential stages of the CI/CD pipeline: source, build, test, and deploy, to automate the software delivery process.

Benefits

Explore the advantages of utilizing AWS for CI/CD, such as scalability, reliability, and easy integration with other AWS services.

Creating an AWS CI/CD Pipeline for Python App

1

Set Up

Configure AWS services, including CodeCommit for source control, CodeBuild for building the app, CodeDeploy for deployment, and CodePipeline for pipeline management.

2

Configure Pipeline Steps

Define the pipeline steps and triggers, such as building the app, running tests, and deploying it to the desired environment.

3

Validation & Monitoring

Ensure the pipeline is functioning correctly by validating the successful execution of each step and monitoring for any anomalies or failures.



Testing and Monitoring

Testing Approaches

Implement unit tests and integration tests specifically tailored for Python applications to guarantee the desired functionality and performance.

Monitoring with CloudWatch

Discover how to set up monitoring and alerting using AWS CloudWatch to gain insights into the app's performance and troubleshoot any issues.

Scaling and Deployment Strategies

Deployment Options

Compare blue/green deployment and rolling deployment strategies to minimize downtime and ensure seamless updates of Python apps on AWS.

1

Scaling Strategies

Explore horizontal scaling and auto scaling techniques to handle increased traffic and efficiently utilize AWS resources for optimal performance.

Best Practices and Considerations

Optimizing CI/CD

Learn best practices for optimizing the CI/CD pipeline on AWS, including reducing build times, automating testing, and improving deployment speed.

Security and Compliance

Address security considerations and compliance requirements when utilizing AWS CI/CD services to protect sensitive data and comply with regulations.

Thank You!

Thank you for joining us on this journey through the AWS CI/CD pipeline for Python app development. We hope you found this presentation informative and valuable.