



**Scalable and Secure
Deployments with
Cloud-Native AWS
CodePipeline**

-Prithvish K

What is AWS CodePipeline ?

- **AWS CodePipeline is a fully managed CI/CD service.**
- **Cloud-Native solution fully managed by AWS and highly configurable by users.**
- **Helps automate release deployment pipelines for fast and reliable application and infrastructure updates.**
- **Traditionally a large number of organizations use 3rd party tools like Jenkins for software deployment. CodePipeline offers native solutions to organizations that already heavily invested in the AWS ecosystem.**

Advantages of AWS CodePipeline

Integration with AWS Services

- **AWS CodePipeline:** Offers a native, out-of-the-box integration with a plethora of AWS services, such as Lambda, EC2, S3, and CloudFormation.
- **Jenkins:** While integration with cloud services is possible, it usually requires third-party plugins and additional setup, potentially introducing more points of failure or compatibility issues.

Scalability

- **AWS CodePipeline:** Being a part of the AWS suite, it natively scales according to the demands of the deployment pipeline. There's no need for manual intervention, ensuring consistent performance even during peak loads.
- **Jenkins:** Scaling requires manual adjustments, such as adding agent nodes or reallocating resources, which can be both time-consuming and resource-intensive.

Advantages of AWS CodePipeline Contd..

Maintenance

- **AWS CodePipeline:** As a managed service, AWS handles all updates, patches, and backups. This ensures that the latest features and security patches are always in place without user intervention.
- **Jenkins:** Requires periodic manual updates, backups, and patching. Additionally, plugins can introduce compatibility issues or security vulnerabilities, demanding regular monitoring and adjustments.

Security

- **AWS CodePipeline:** One of the key benefits of AWS's comprehensive security model. Features like IAM roles, secret management with AWS Secrets Manager, and fine-grained access controls ensure robust security standards.
- **Jenkins:** Achieving a similar security level necessitates additional configurations, plugins, and tools, which can sometimes introduce more vulnerabilities or complexities.

Pricing and Long-Term Value

- **AWS CodePipeline:** Operates on a pay-as-you-go model, ensuring you only pay for what you use. This can be cost-effective, especially for variable workloads.
- **Jenkins:** While the software itself is open-source, maintaining a Jenkins infrastructure (servers, electricity, backups, etc.) incurs steady costs, which can add up in the long run, especially for larger setups.

Key Components of CodePipeline

- Source
 - Add the application source repository or an artifact
 - E.g. S3, AWS CodeCommit, AWS ECR
- Build
 - Optional stage for compilation, object generation etc.
 - E.g. AWS CodeBuild
- Deploy
 - The main deployment stage
 - E.g. CodeDeploy, ECS, ECS(Blue/Green)

Sample Pipeline

