AWS Developer Tools

Code Commit, Code Build, Code Deploy & Code Pipeline

Kalyan Reddy Daida

AWS Developer Tools

AWS CodeCommit

AWS CodeBuild

AWS CodeDeploy

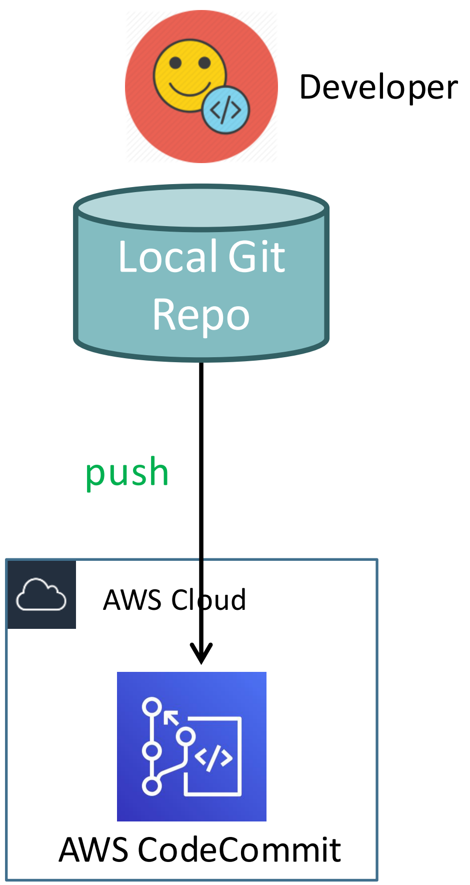
AWS CodePipeline

AWS CodeStar

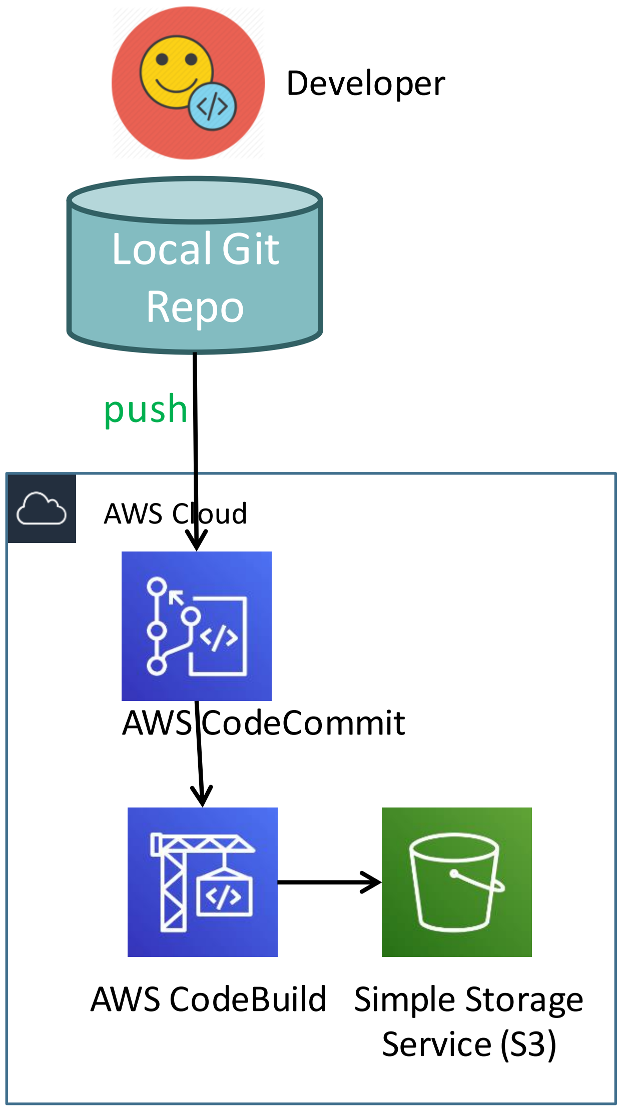
AWS Cloud9

AWS X-Ray

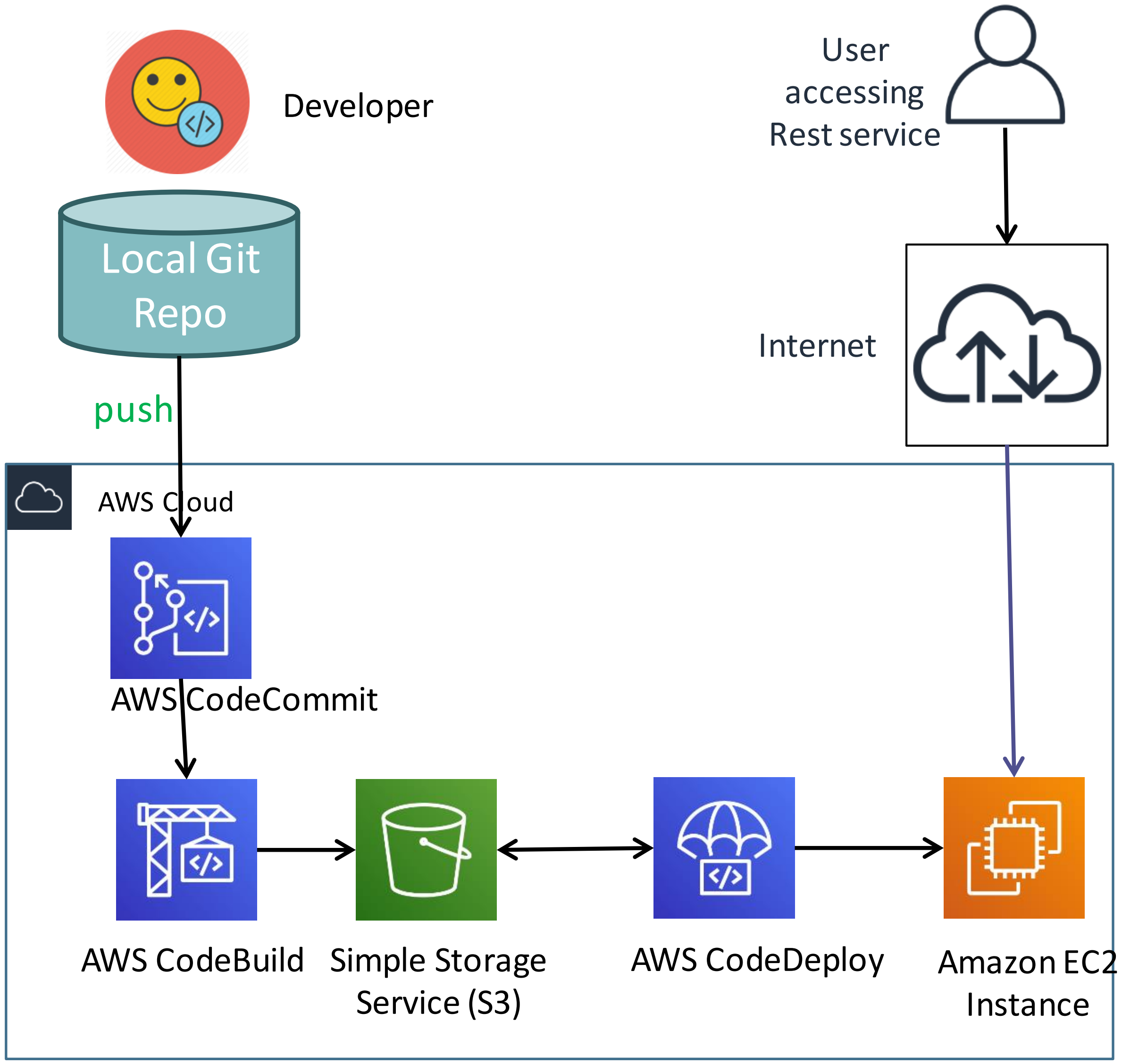
CodeCommit



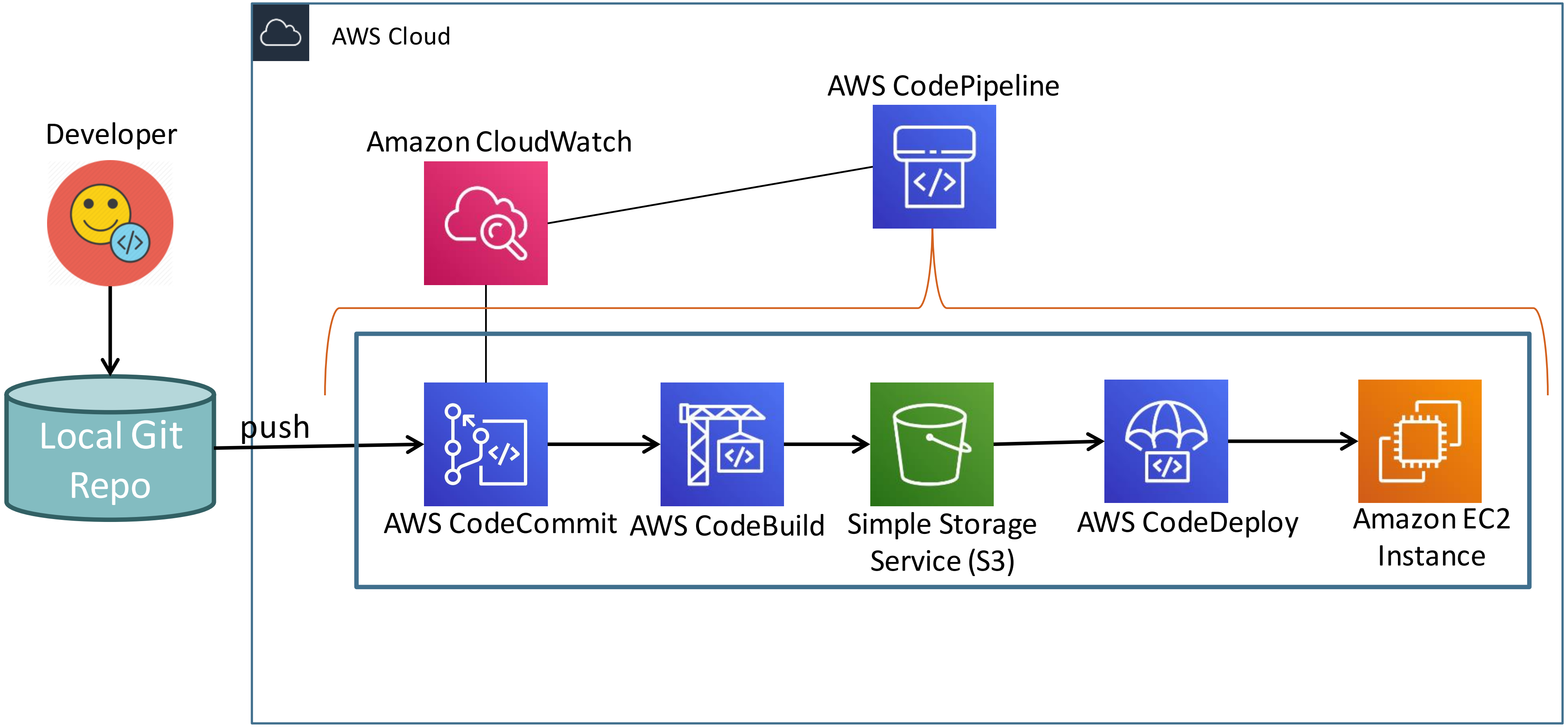
CodeBuild



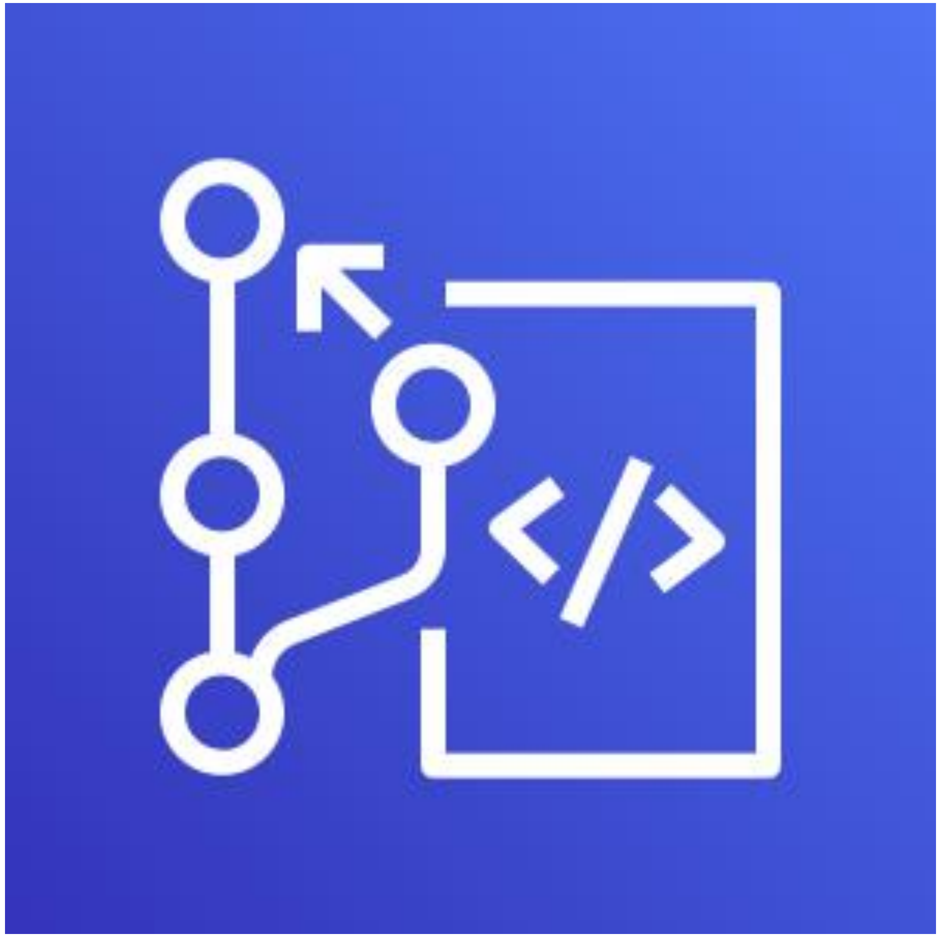
CodeDeploy



CodePipeline



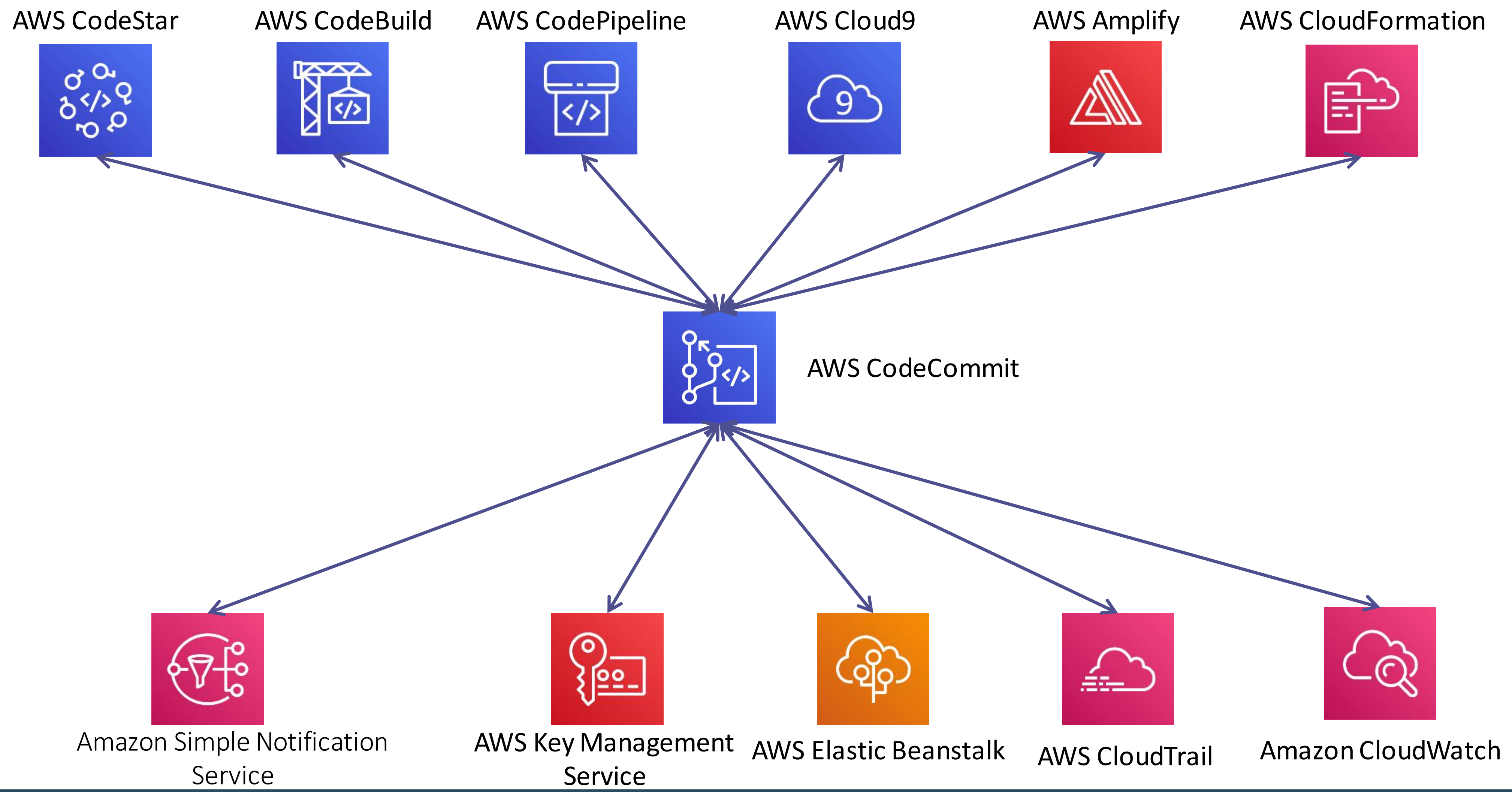
AWS CodeCommit



AWS CodeCommit - Introduction

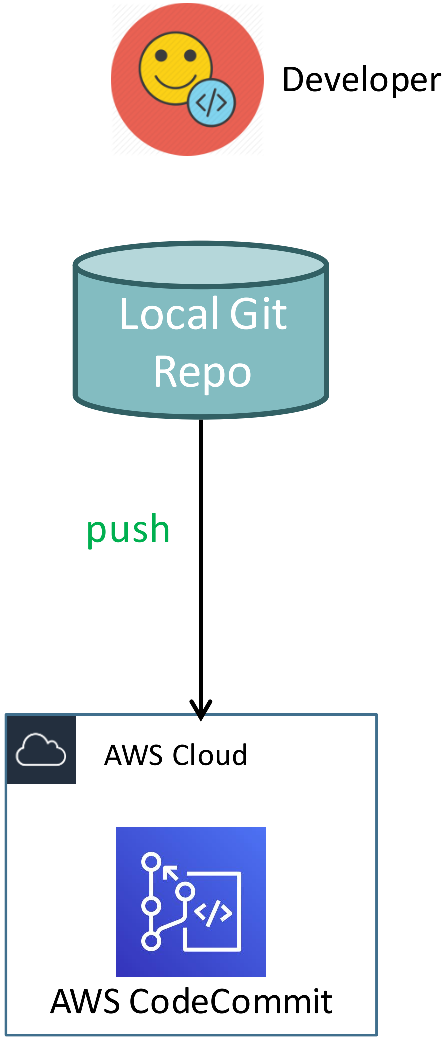
- Version Control Service hosted by AWS
- We can privately store and manage documents, source code, and binary files
- Secure & highly scalable
- Supports standard functionality of Git (CodeCommit supports Git versions 1.7.9 and later.)
- Uses a static user name and password in addition to standard SSH..

CodeCommit – Integration with AWS Services



CodeCommit - Steps

- Step#1: Sample Spring Boot Rest Application
  - Pre-requisites
    - Install STS IDE
  - Create Spring boot rest application.
  - Test it.
- Step#2: GIT Repository
  - Create a local git repository and check-in code.
  - Create a remote git repository in AWS Code Commit.
  - Create Code Commit git credentials to connect.
  - Push the code to remote git repository.
  - Verify code in AWS Code Commit.
- Step#3: CodeCommit Features
  - Code, Commits, Branches
  - Settings: Notifications, Triggers
  - Pull Requests

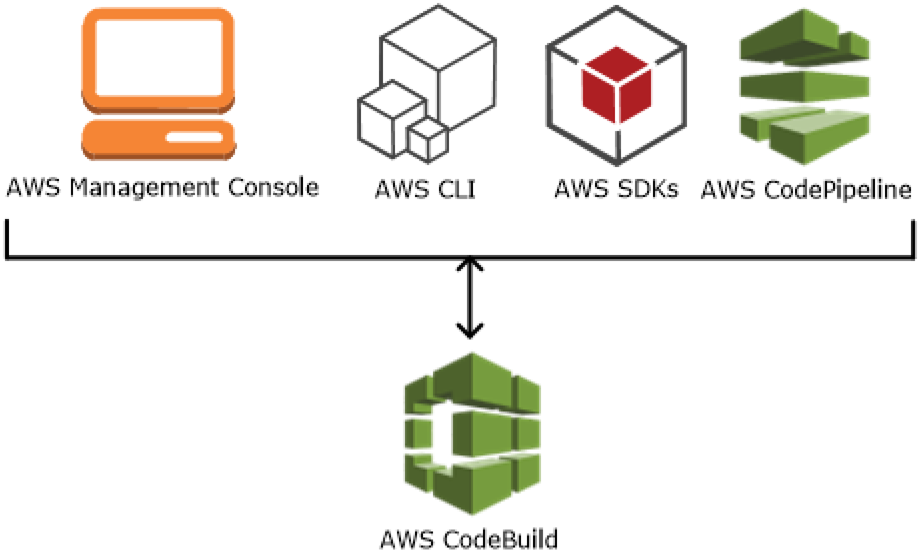


AWS CodeBuild

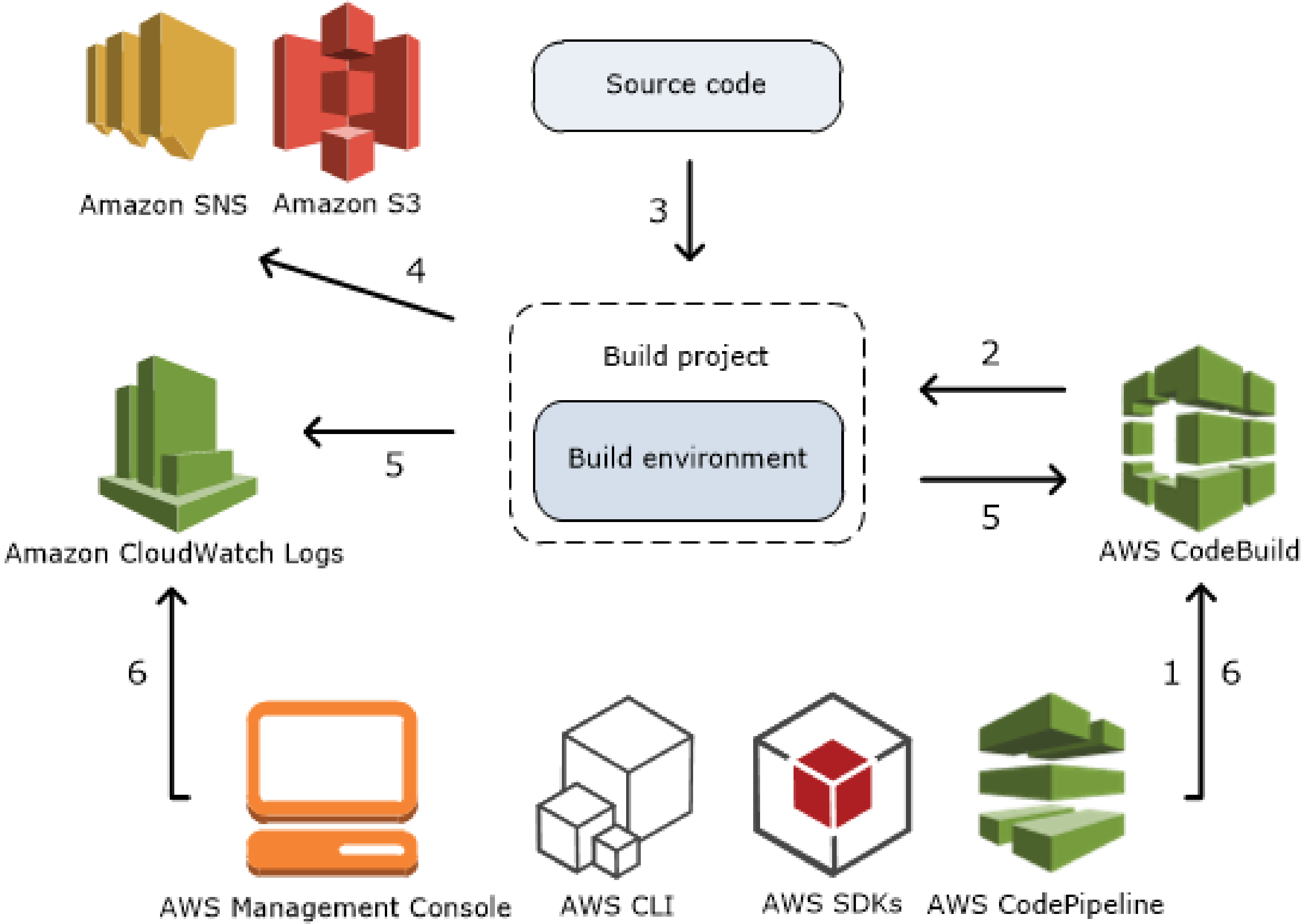
CodeBuild - Introduction

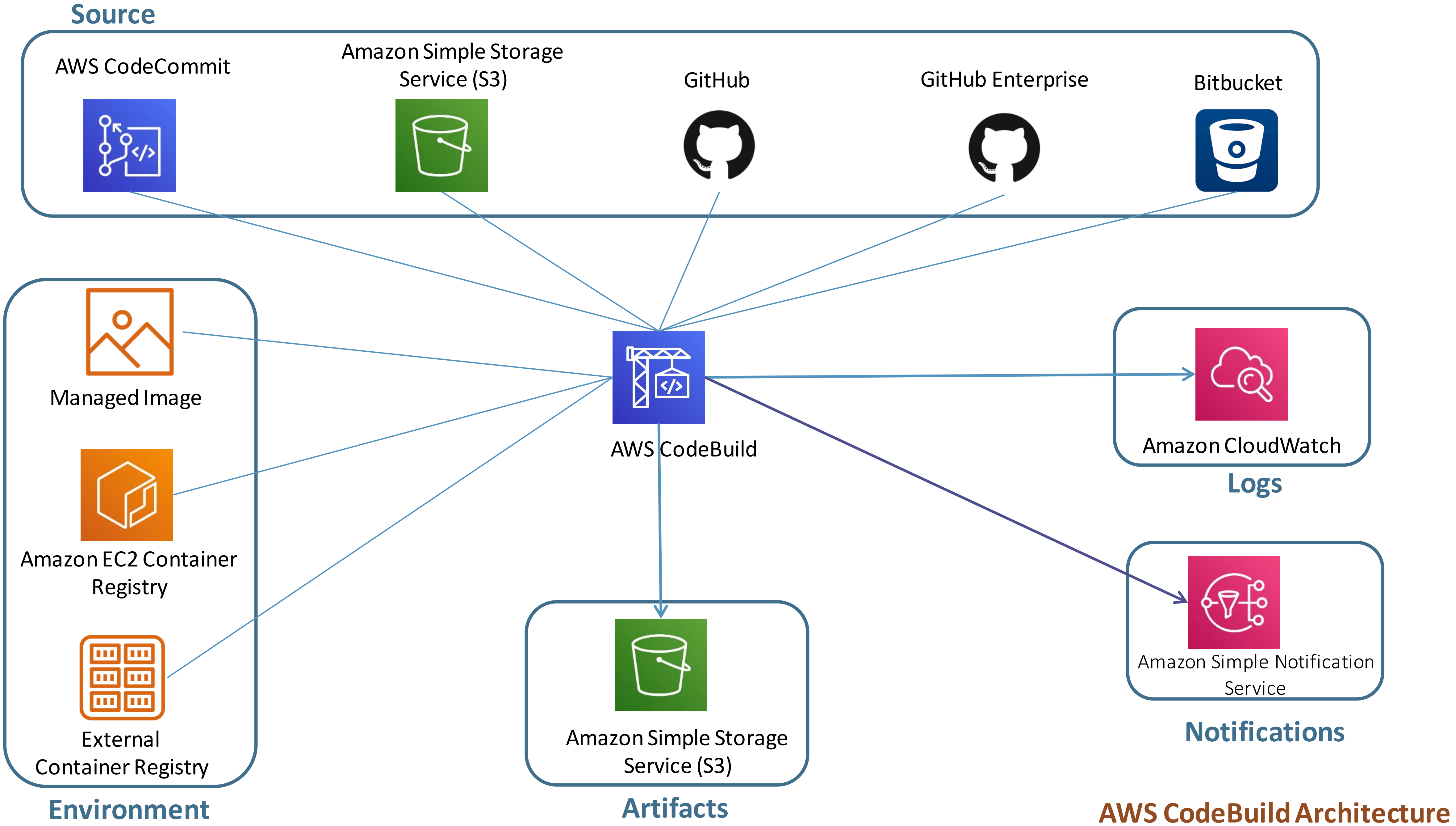
- CodeBuild is a fully managed build service in the cloud.
- Compiles your source code, runs unit tests, and produces artifacts that are ready to deploy.
- Eliminates the need to provision, manage, and scale your own build servers.
- It provides prepackaged build environments for the most popular programming languages and build tools such as Apache Maven, Gradle, and more.
- We can also customize build environments in CodeBuild to use ourown build tools.
- Scales automatically to meet peak build requests.

How to run CodeBuild?



How CodeBuild works?

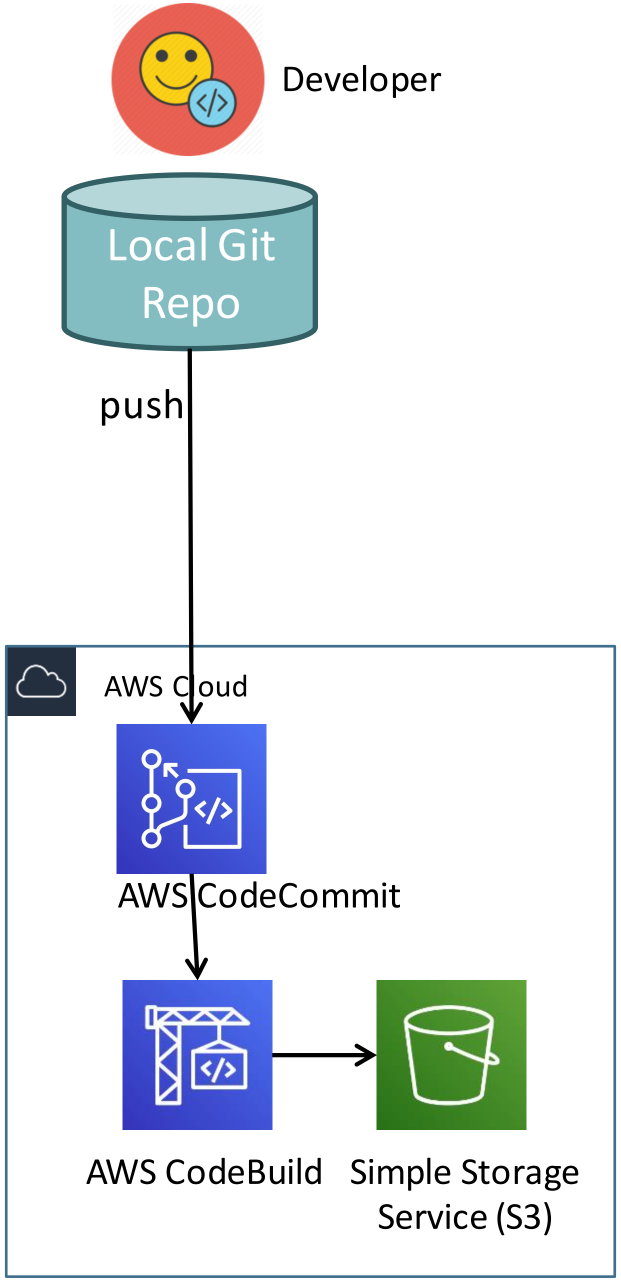



Source
AWS CodeCommit
Amazon Simple Storage Service (S3)
GitHub
GitHub Enterprise
Bitbucket
Managed Image
Amazon EC2 Container Registry
External Container Registry
Environment
AWS CodeBuild
Amazon CloudWatch
Logs
Amazon Simple Storage Service (S3)
Artifacts
Amazon Simple Notification Service
Notifications

AWS CodeBuild Architecture

CodeBuild - Steps

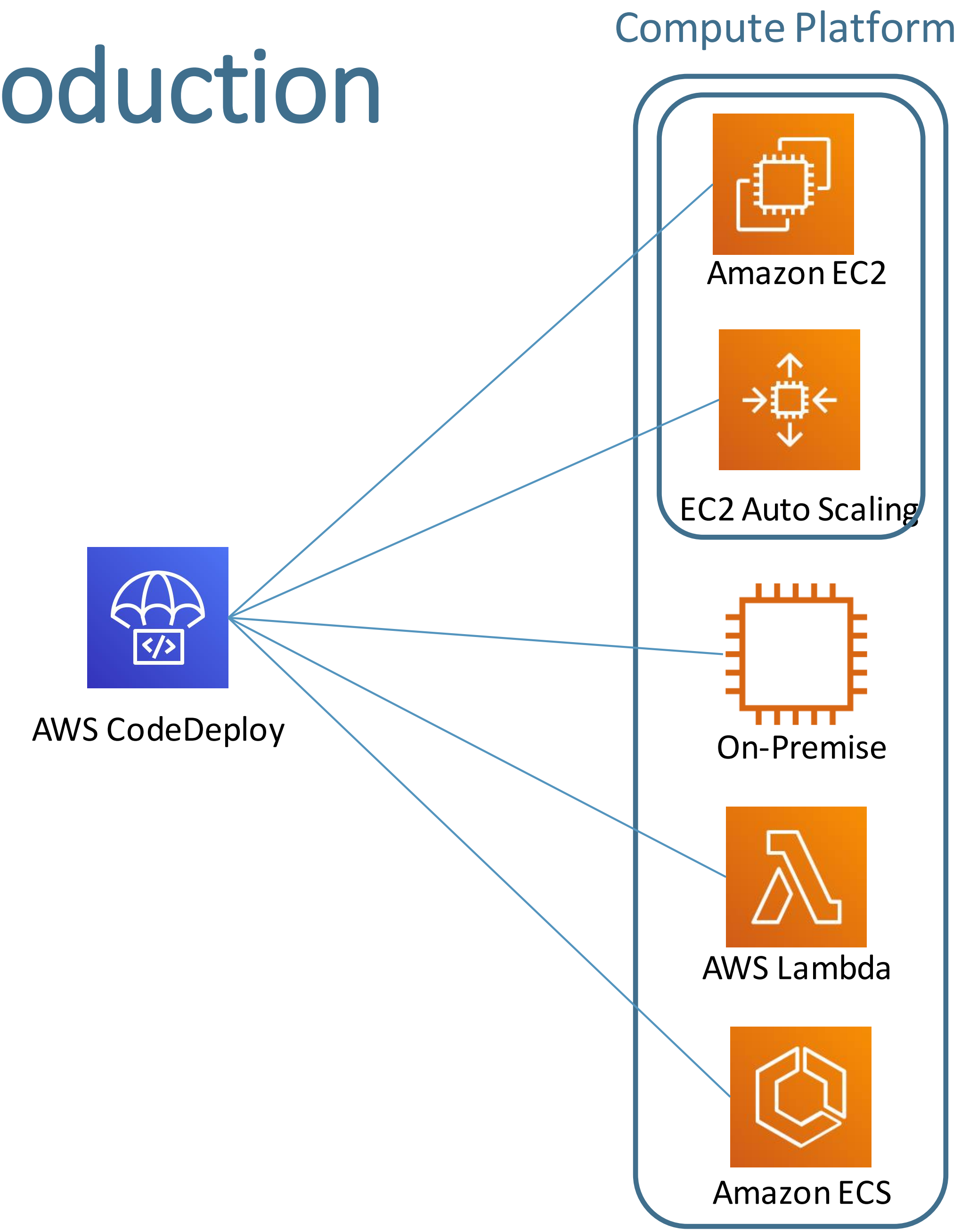
- Step#1: Create CodeBuild Project
  - Create a S3 bucket and folder
  - Create CodeBuild project
  - Start build, Verify build logs, Verify build phase details
- Step#2: buildspec.yml & Start Build
  - Create buildspec.yml and check-in code
  - Start build, Verify build logs, Verify build phase details
  - Download the artifacts from S3, unzip and review
  - Run one more build and see versioning in S3.
- Step#3: Create Build Notifications
  - Create state change notification
  - Create Phase change notification



AWS CodeDeploy

CodeDeploy - Introduction

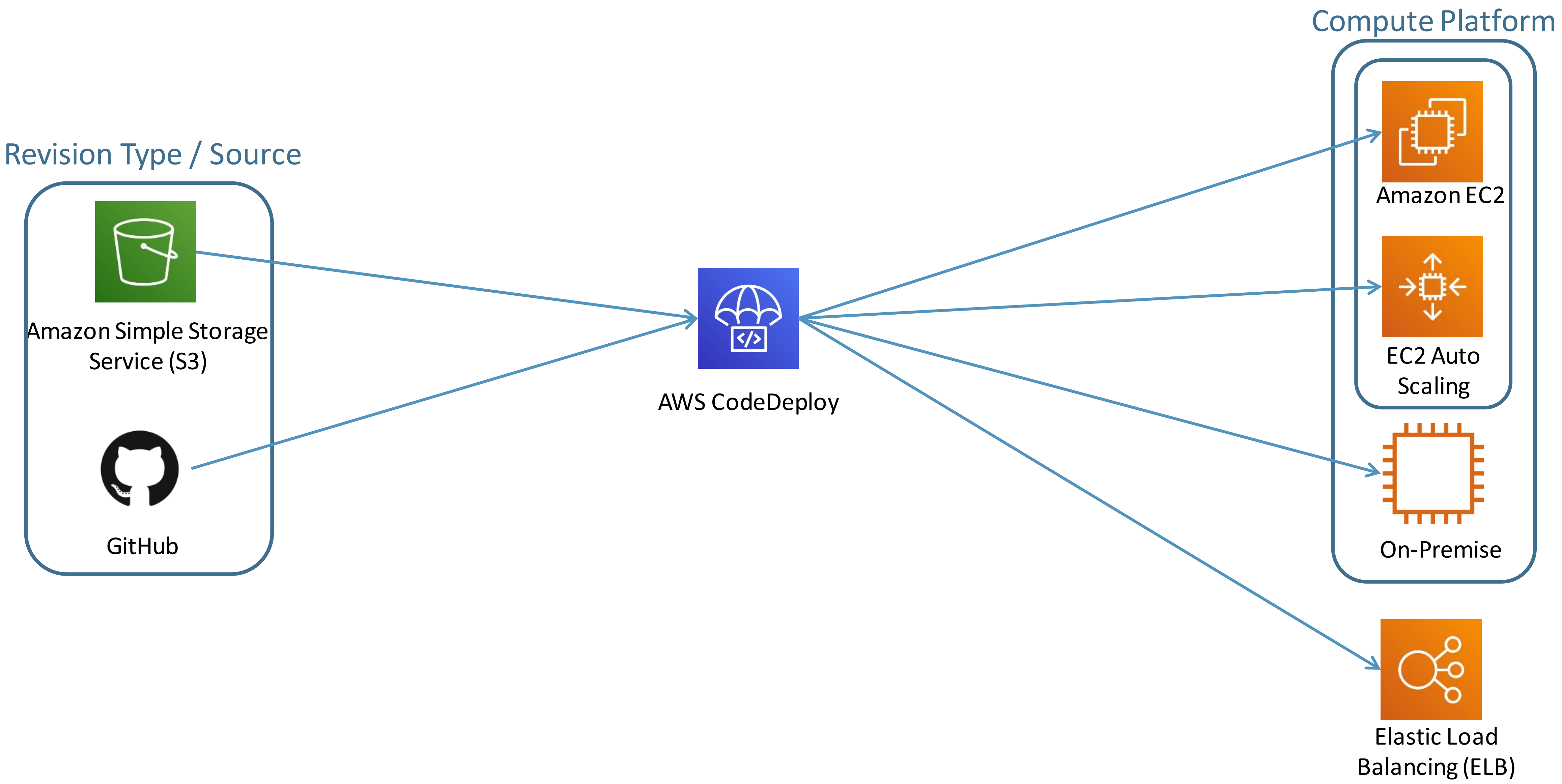
- CodeDeploy is a deployment service that automates application deployments to
  - EC2 instances
  - On-premises instances
  - AWS Lambda
  - AWS ECS
- We can deploy unlimited variety of application content
  - code
  - serverless AWS Lambda functions
  - web and configuration files
  - executables
  - packages
  - scripts
  - multimedia files



CodeDeploy - Introduction

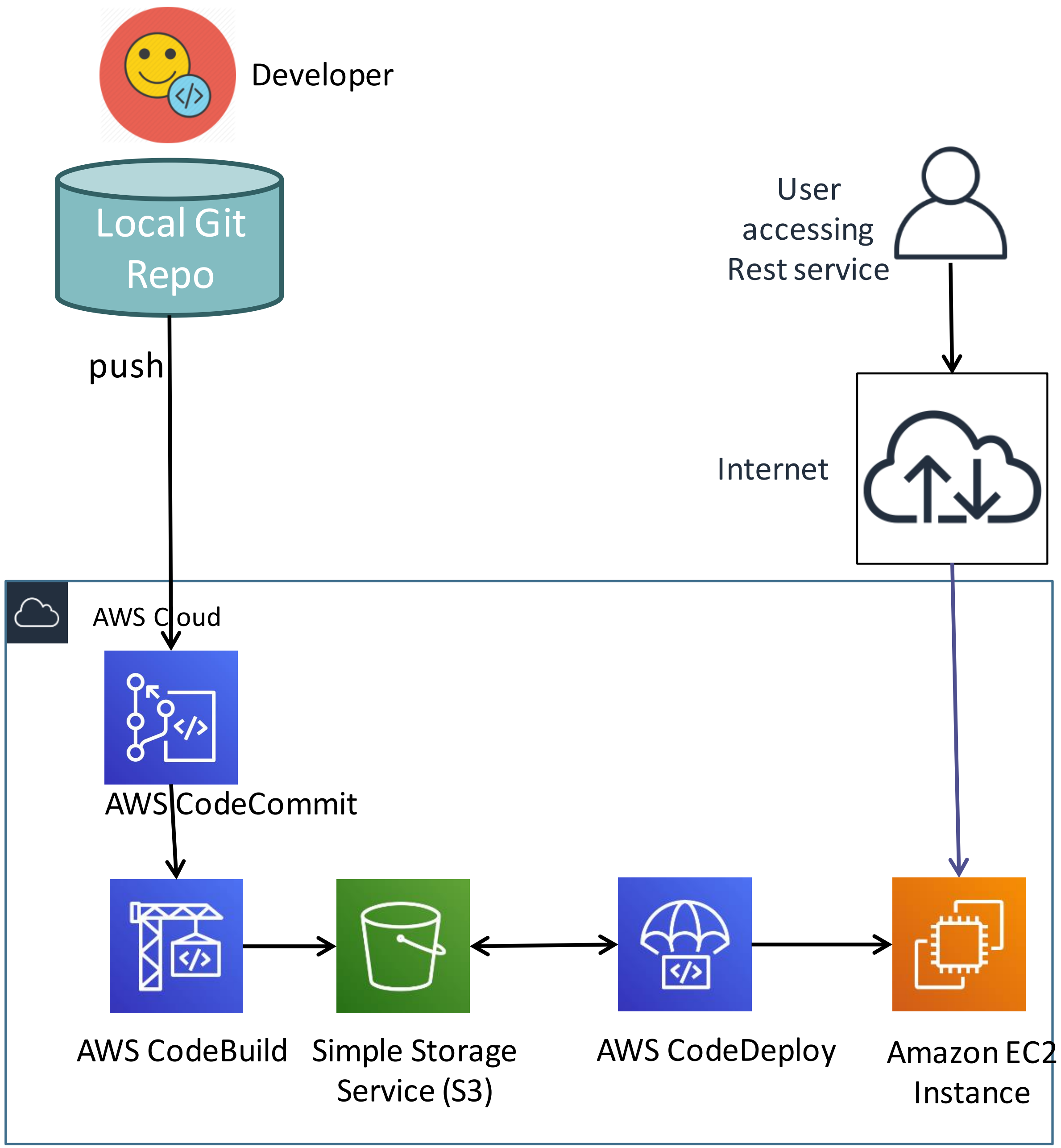
- Benefits
  - We can rapidly release new features.
  - Update AWS Lambda function versions.
  - Avoid downtime during application deployment.
  - Reduces the complexity of updating applications when compared to error-prone manual deployments.
  - Service scales with our infrastructure so we can easily deploy to one instance or thousands.

CodeDeploy - When compute is EC2/On-Premise

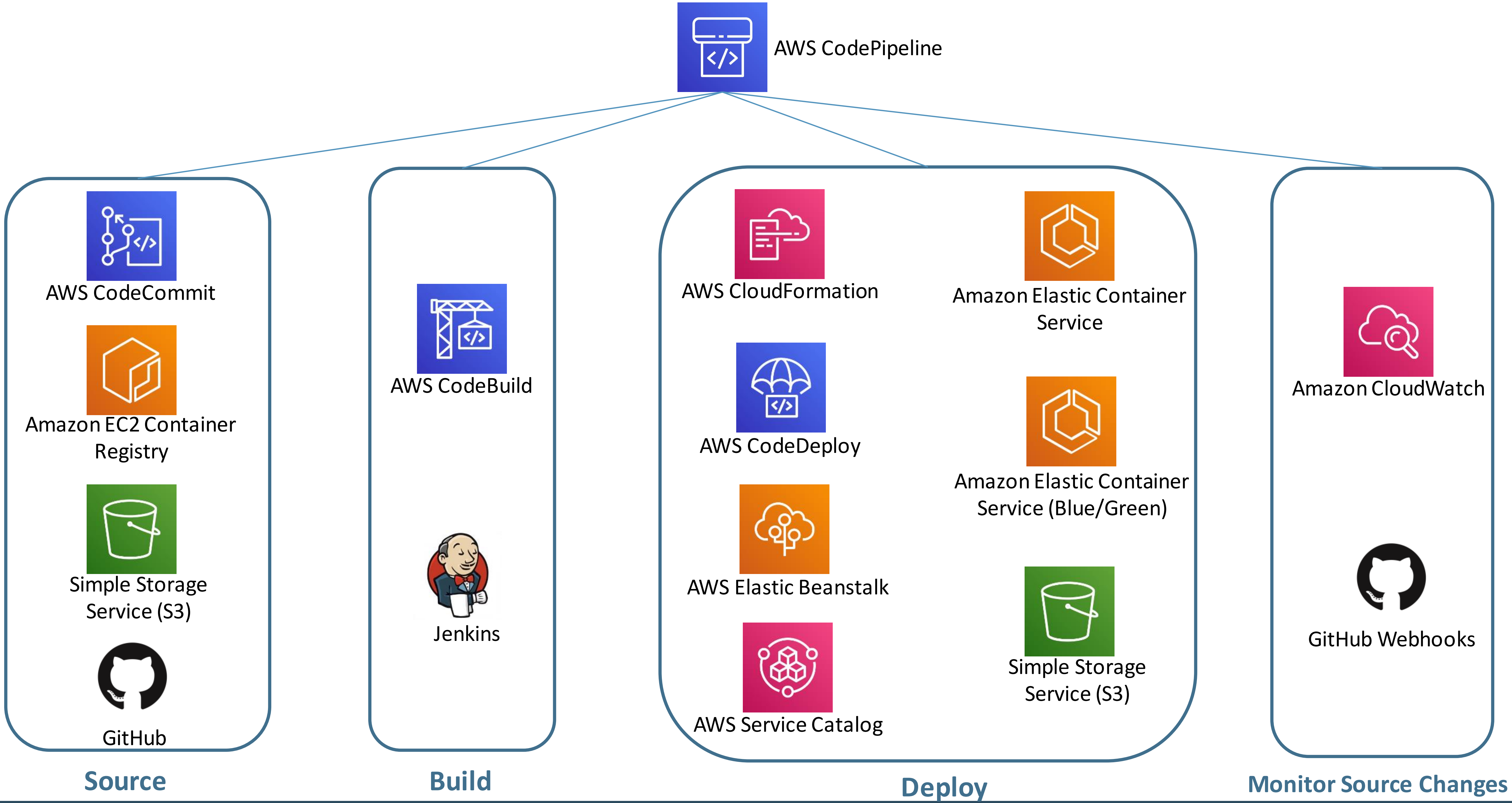


CodeDeploy - Steps

- Step#1: Create CodeDeploy pre-requisite roles
  - Create a service role for codeDeploy.
  - Create an IAM Instance profile.
- Step#2: Create a EC2 VM
  - Create EC2 VM
  - During creation associate IAM instance profile.
  - Discuss about "Userdata" containing tomcat and codeDeploy Agent
- Step#3: Create codeDeploy objects
  - Create Application
  - Create Deployment Group
  - Create Deployment
- Step#4: Create codeDeploy files and scripts
  - Create appspec.yml
  - Create scripts (before_install script, after_install script, Start up script, Shutdown script) and check-in
- Step#5: Run CodeBuild and Create Deployment
- Step#6: Verify Deployment
  - Verify the deployment Events
  - Verify the tomcat deployment
  - Verify the codeDeploy agent log
  - Verify by accessing app
- Step#7: New App Release: Make change to Application and re-deploy



AWS CodePipeline



AWS CodePipeline

AWS CodeCommit

Amazon EC2 Container Registry

Simple Storage Service (S3)

GitHub

Source

AWS CodeBuild

Jenkins

Build

AWS CloudFormation

AWS CodeDeploy

AWS Elastic Beanstalk

AWS Service Catalog

Amazon Elastic Container Service

Amazon Elastic Container Service (Blue/Green)

Simple Storage Service (S3)

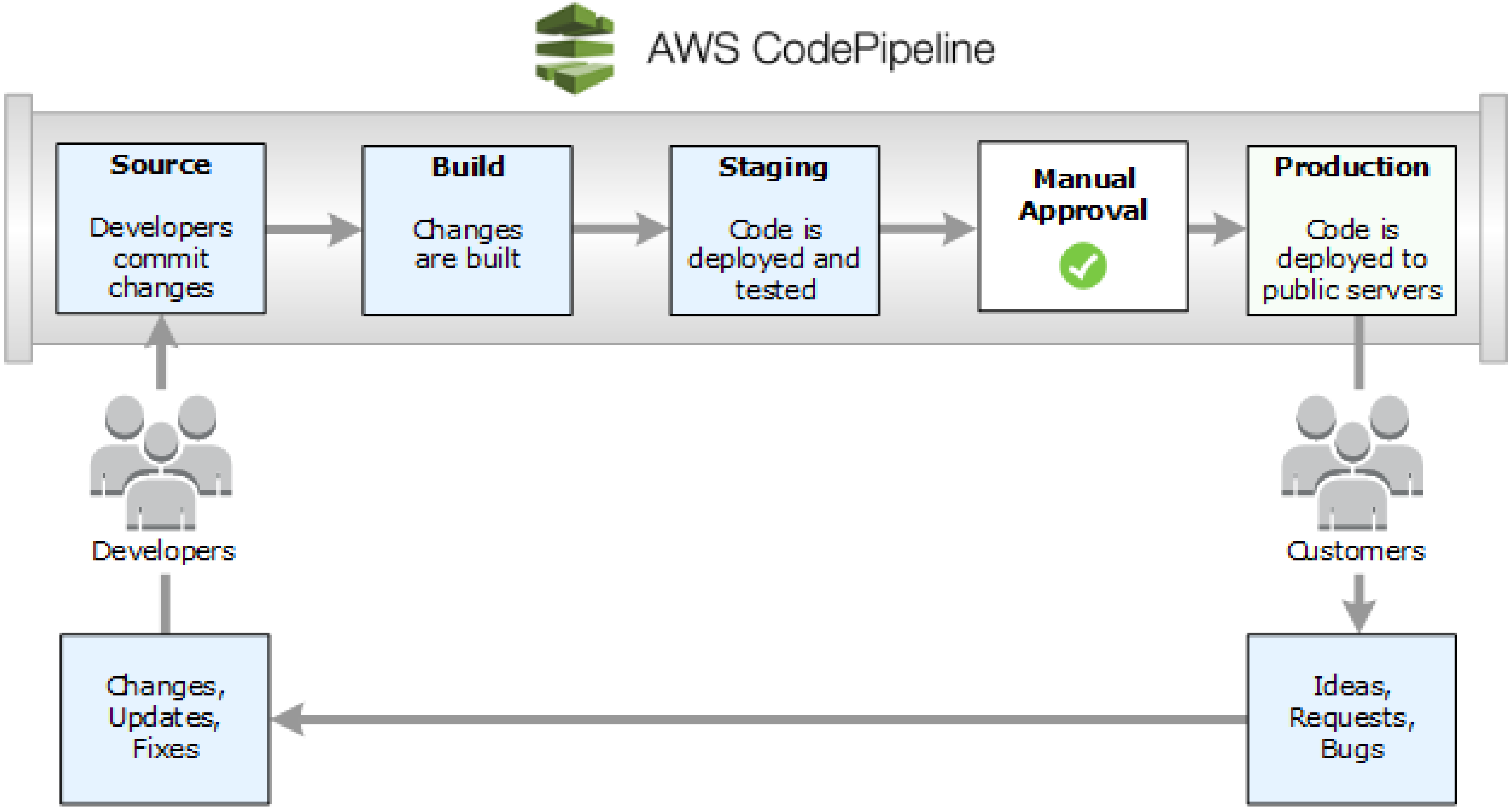
Deploy

Amazon CloudWatch

GitHub Webhooks

Monitor Source Changes

Continuous Delivery



AWS CodePipeline

Source

Developers commit changes

Build

Changes are built

Staging

Code is deployed and tested

Manual Approval

✓

Production

Code is deployed to public servers

Developers

Customers

Changes, Updates, Fixes

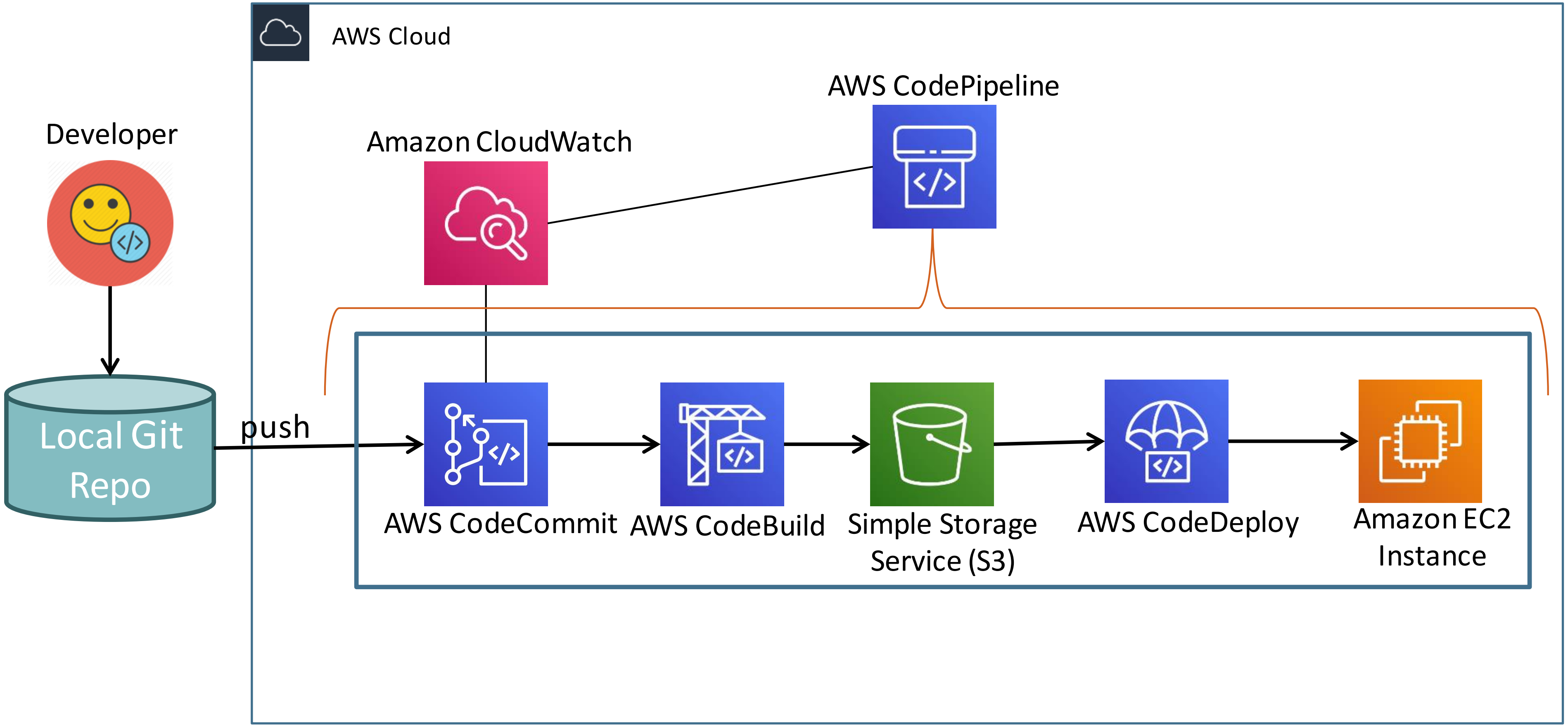
Ideas, Requests, Bugs

CodePipeline - Introduction

- AWS CodePipeline is a continuous delivery service to model, visualize, and automate the steps required to release your software.
- Benefits
  - Automate your release processes.
  - Establish a consistent release process.
  - Speed up delivery while improving quality.
  - Supports external tools integration for source, build and deploy.
  - View progress at a glance
  - View pipeline history details.

CodePipeline - Steps

- Step#1: Create Pipeline
  - Artifacts: S3
  - Source: CodeCommit
  - Build: CodeBuild
  - Deploy: CodeDeploy
  - Server: EC2 Instance
- Step#2: Make changes & Check-In Code
  - Make changes to rest app and check-in
  - Pipeline should trigger the build automatically.



CodePipeline – Manual Approval & Prod Deployment

- Step#1: Create new EC2 Instance with tag name as prod
- Step#2: Create new deployment group for prod
- Step#3: Create Manual Approval stage in CodePipeline
- Step#4: Create Prod Deployment stage in CodePipeline .
- Step#5: Check-in changed code to trigger pipeline and monitor the pipeline process.