Choose a Topic

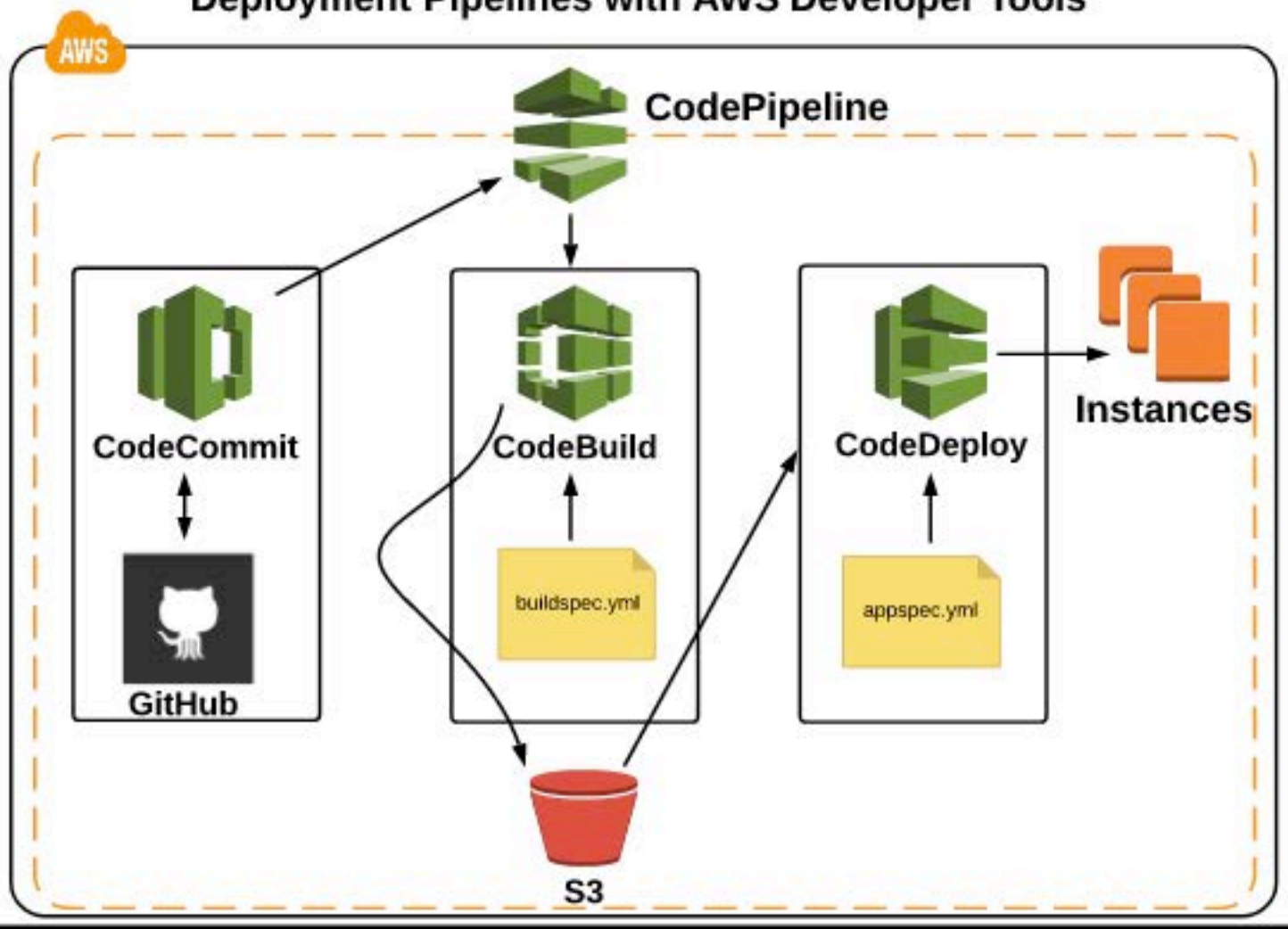
Deployment Pipeline Concepts

Scenario:

A DevOps engineer needs to have a consistent and reliable platform for which they can deploy code. Using the AWS Developer Tools suite and other complementary tools (such as GitHub), the DevOps engineer can build deployment pipelines to perform continuous integration and continuous delivery (CI/CD) for their applications. In this section, we will gain an understanding of these services individually, and then we will see how they work together to provide a reliable pipeline to deploy applications.

- AWS Developer Tools
- CodeBuild
- GitHub
- CodeDeploy
- CodeCommit
- CodePipeline

Deployment Pipelines with AWS Developer Tools

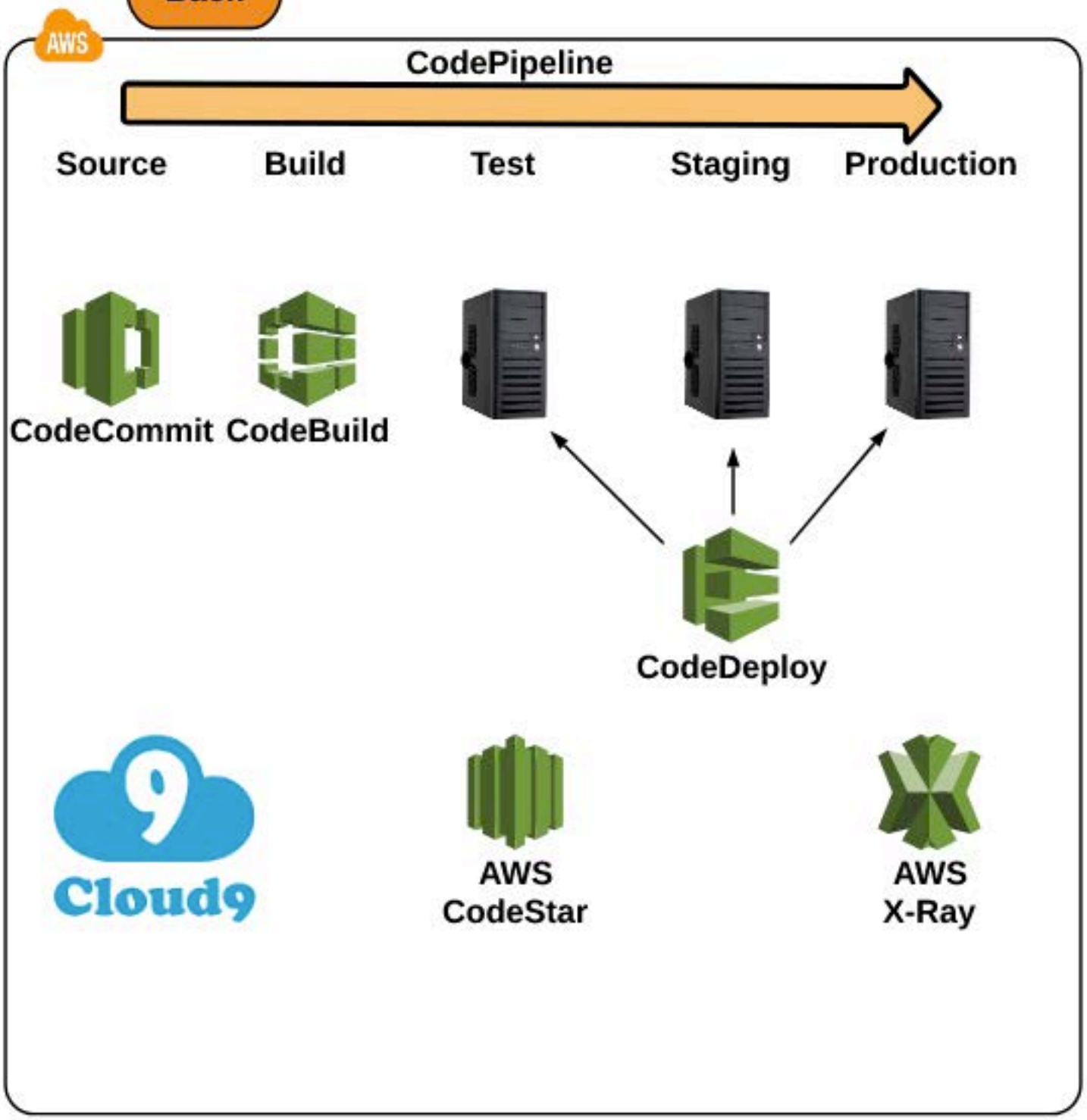


Choose a Topic

AWS Developer Tools

AWS Developer Tools is a set of services designed to enable developers and IT operations professionals practicing DevOps to rapidly and safely deliver software. Together, these services help you securely store and version control your application's source code and automatically build, test, and deploy your application to AWS or your on-premises environment. You can use AWS CodePipeline to orchestrate an end-to-end software release workflow using these services and third-party tools or integrate each service independently with your existing tools.

Back

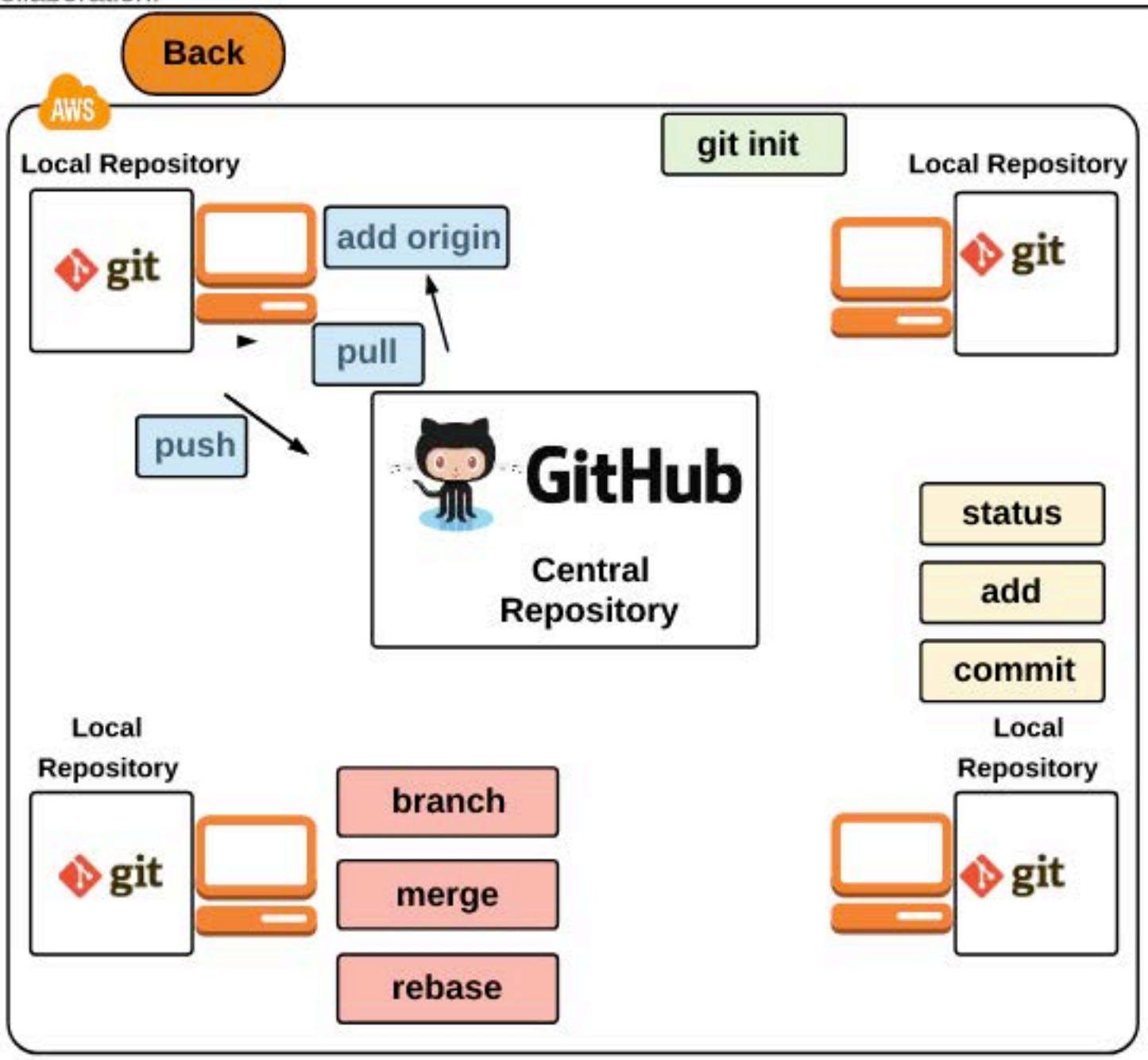


Choose a Topic

Git and GitHub

What is Git? Git is a distributed version control system (free and open source) designed to handle everything from small to very large projects with speed and efficiency. Having a distributed architecture, Git is an example of a DVCS (Distributed Version Control System). Rather than having only one single place for the full version history of the software — as is common in once-popular version control systems like CVS or Subversion (also known as SVN) — in Git, every developer's working copy of the code is also a repository that can contain the full history of all changes.

What is GitHub? GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.

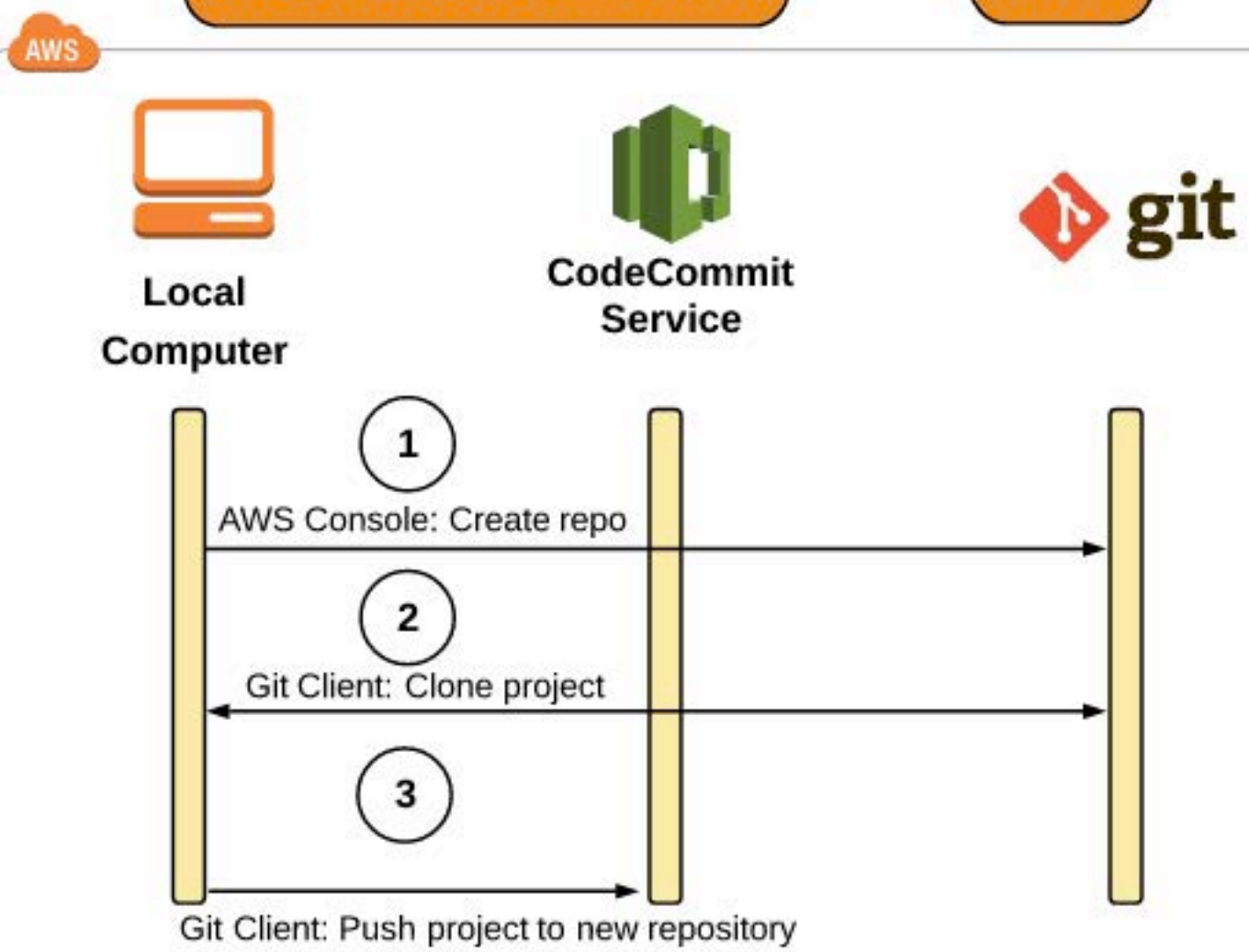


Choose a Topic

Deployment Pipelines

How Does CodeCommit Work?

Back



CodeCommit Features

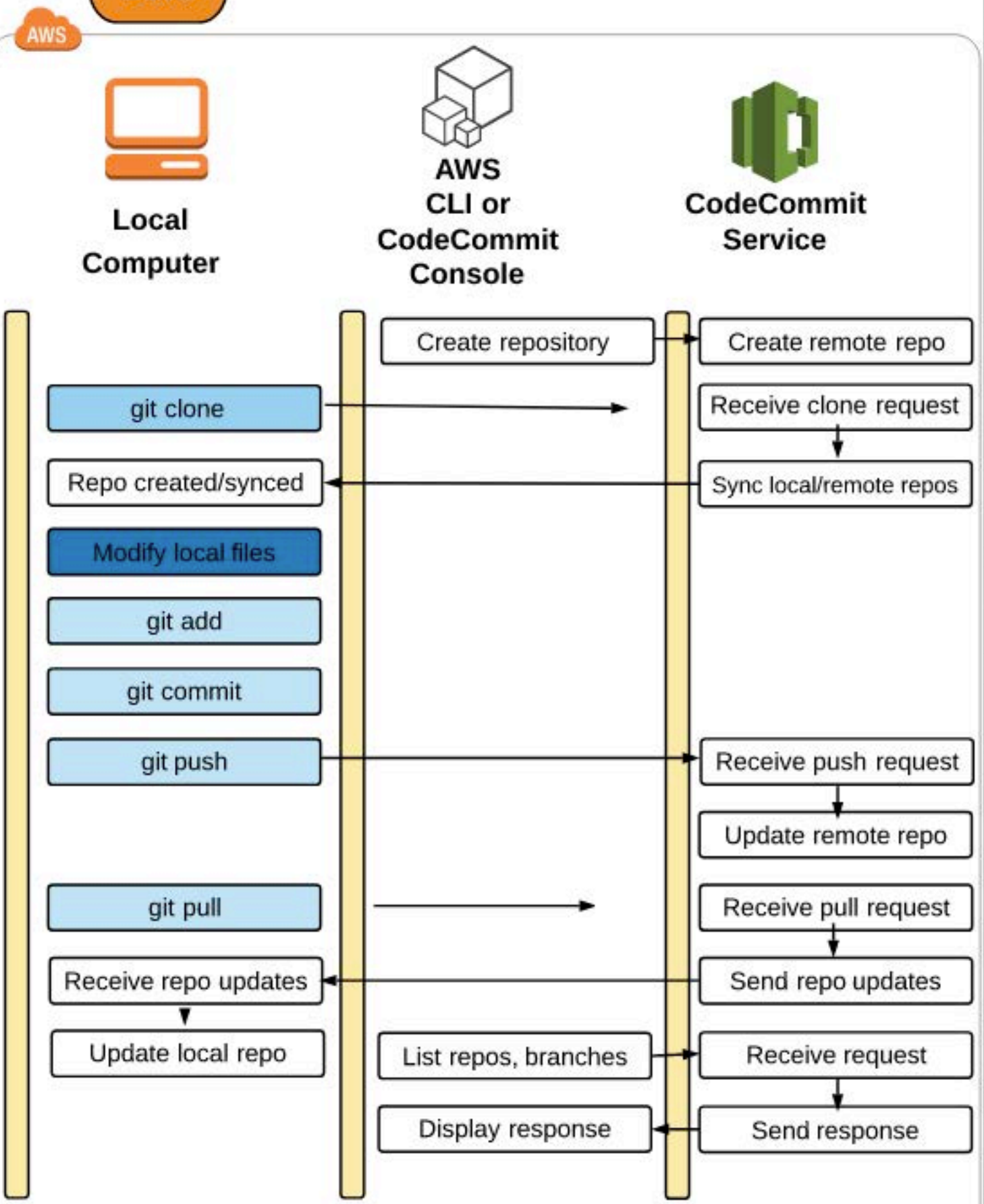
- What is CodeCommit? A version control service hosted by Amazon Web Services that you can use to privately store and manage assets (such as documents, source code, and binary files) in the cloud.
- A secure, highly scalable, managed source control service that hosts private Git repositories. Eliminates the need for you to manage your own source control system or worry about scaling its infrastructure. It supports the standard functionality of Git, so it works seamlessly with your existing Git-based tools.
- Benefit from a fully managed service hosted by AWS.
- Repositories are encrypted at rest as well as in transit.
- Supports pull requests, where users can review and comment on each other's code changes before merging them to branches.
- Integrated with IAM and can be used with other AWS services and in parallel with other repositories.
- You can migrate to AWS CodeCommit from any Git-based repository.

The DevOps Doctrine

Choose a Topic

Deployment Pipelines

Back



Choose a Topic

Deployment Pipelines

Back

AWS CodeBuild

Next

AWS

Local Computer

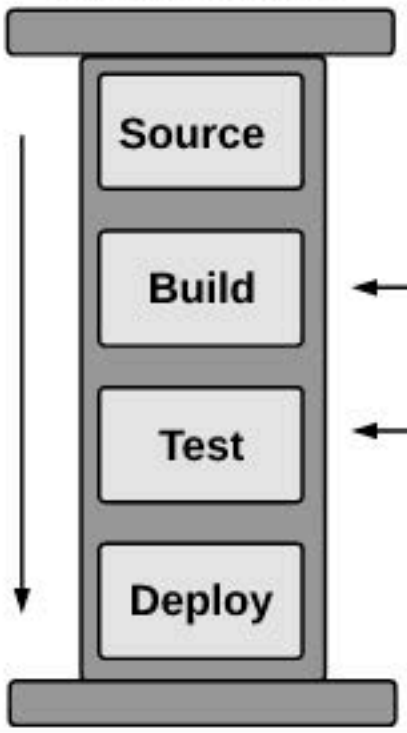
AWS CLI

AWS SDK

AWS CodePipeline

AWS CodeBuild

CodePipeline



CodeBuild

Add CodeBuild as a build or test action to the build or test stage of a pipeline in CodePipeline.

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts with EC2
- CloudWatch for DevOps
- CloudFormation for DevOps
- Elastic Beanstalk for DevOps
- Application Deployments with OpsWorks
- DynamoDB Concepts
- S3 Concepts for DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda

Deployment Pipelines

Back

AWS CodeBuild

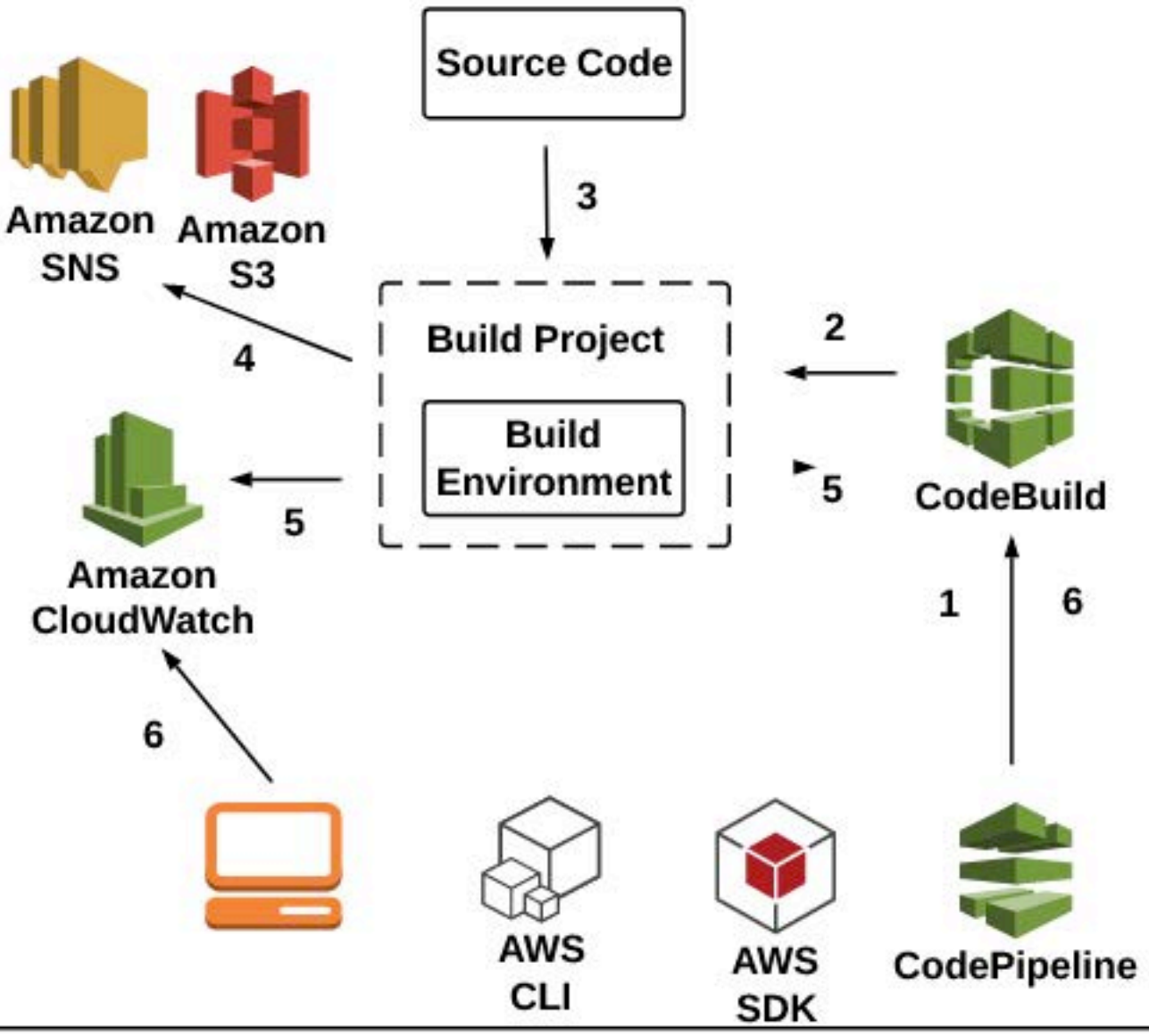
Next

- **What is CodeBuild?** A fully managed build service in the cloud that 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.
- Provides prepackaged build environments for the most popular programming languages and build tools, such as Apache Maven, Gradle, and more.
- You can also customize build environments in AWS CodeBuild to use your own build tools.
- You can run AWS CodeBuild by using the AWS CodeBuild or AWS CodePipeline console.
- You can also automate the running of AWS CodeBuild by using the AWS Command Line Interface (AWS CLI) or the AWS SDKs.

Deployment Pipelines

Back

How Does CodeBuild Work?



1. As input, you must provide AWS CodeBuild with a build project.
2. CodeBuild uses the build project to create the build environment.
3. CodeBuild downloads the source code into the build environment and then uses the build specification (build spec), as defined in the build project or included directly in the source code.
4. If there is any build output, the build environment uploads its output to an Amazon S3 bucket.
5. While the build is running, the build environment sends information to AWS CodeBuild and Amazon CloudWatch Logs.
6. You can get summarized info from CodeBuild and CloudWatch.

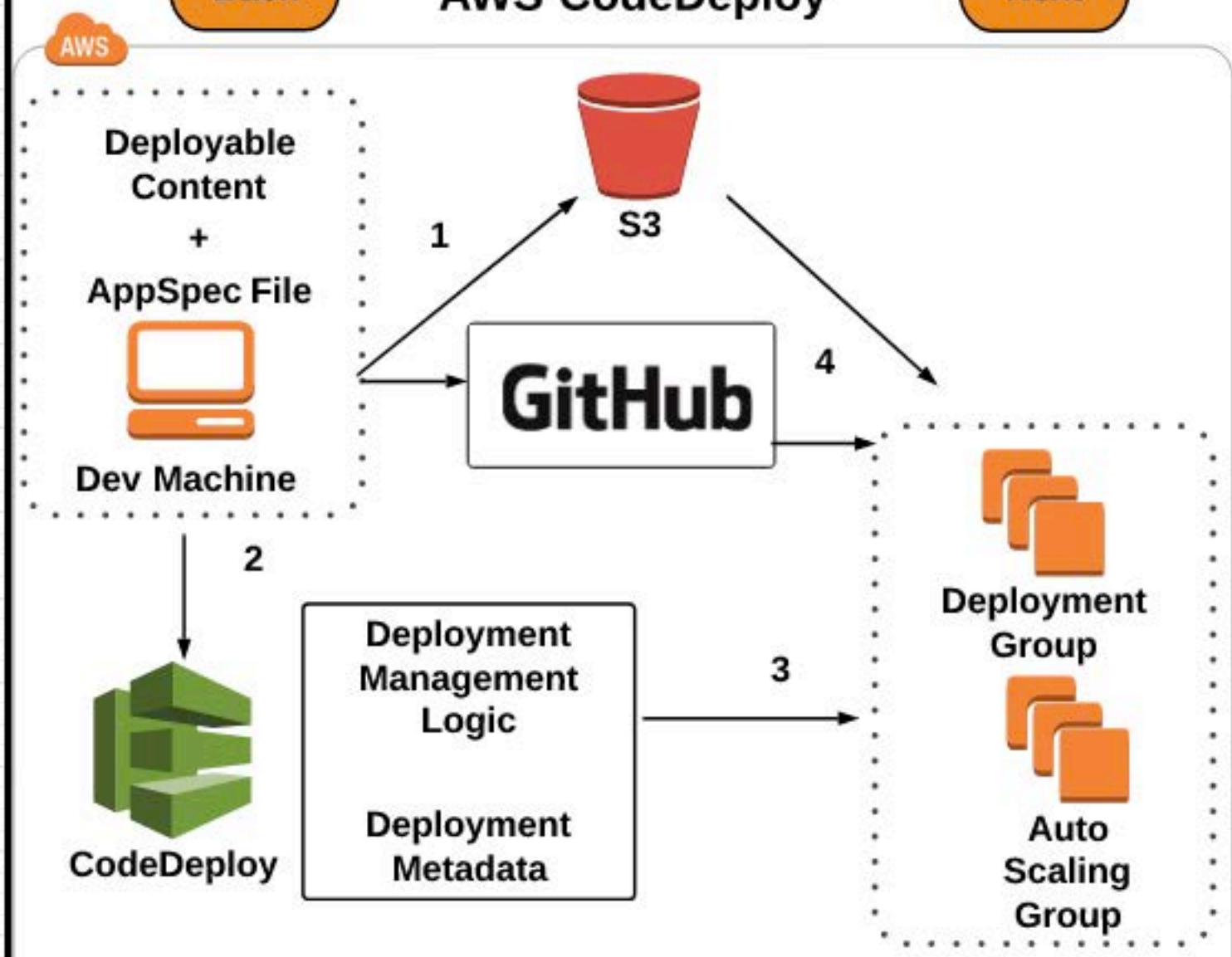
Choose a Topic

Deployment Pipelines

Back

AWS CodeDeploy

Next



- What is CodeDeploy? A deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.
- What can you deploy? Code, Lambda functions, web and config files, executables, packages, scripts, multimedia files.
- Can deploy application content that runs on a server and is stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories.
- Rapidly release new features and avoid downtime during application deployment.
- Automated deployments and stop and rollback capability.
- Centralized control
- Concurrent deployments

Deployment Pipelines

Back

AWS CodeDeploy Key Concepts

Next

- CodeDeploy is able to deploy applications to three compute platforms: EC2/On-premises, Lambda, ECS.
- CodeDeploy components:
  - **Deployment group**: Deploys a revision to a set of instances.
  - **Deployment**: Deploys a new revision that consists of an application and AppSpec file.
  - **Deployment configuration**: Settings that determine the deployment speed and the minimum number of instances that must be healthy at any point during a deployment.
  - **Revision**: A combination of an AppSpec file and application files, such as executables, configuration files, and so on.
  - **Application**: A collection of deployment groups and revisions.
- CodeDeploy deployment types:
  - **In-place deployment**: The application on each instance in the deployment group is stopped, the latest application revision is installed, and the new version of the application is started and validated.
  - **Blue/green deployment**: The behavior of your deployment depends on which compute platform you use:
    - EC2/on-prem: The instances in a deployment group (the original environment) are replaced by a different set of instances.
    - Lambda: Traffic is shifted from your current serverless environment to one with your updated Lambda function versions.
    - ECS: Traffic is shifted from the task set with the original version of a containerized application in an Amazon ECS service to a replacement task set in the same service. The protocol and port of a specified load balancer listener is used to reroute production traffic. During a deployment, a test listener can be used to serve traffic to the replacement task set while validation tests are run.

Deployment Pipelines

Back

AWS CodeDeploy Pre-Configuration

- Step 1: Provision an IAM user.
- Step 2: Install or upgrade and then configure the AWS CLI.
- Step 3: Create a service role for AWS CodeDeploy.
- Step 4: Create an IAM instance profile for your EC2 instances.

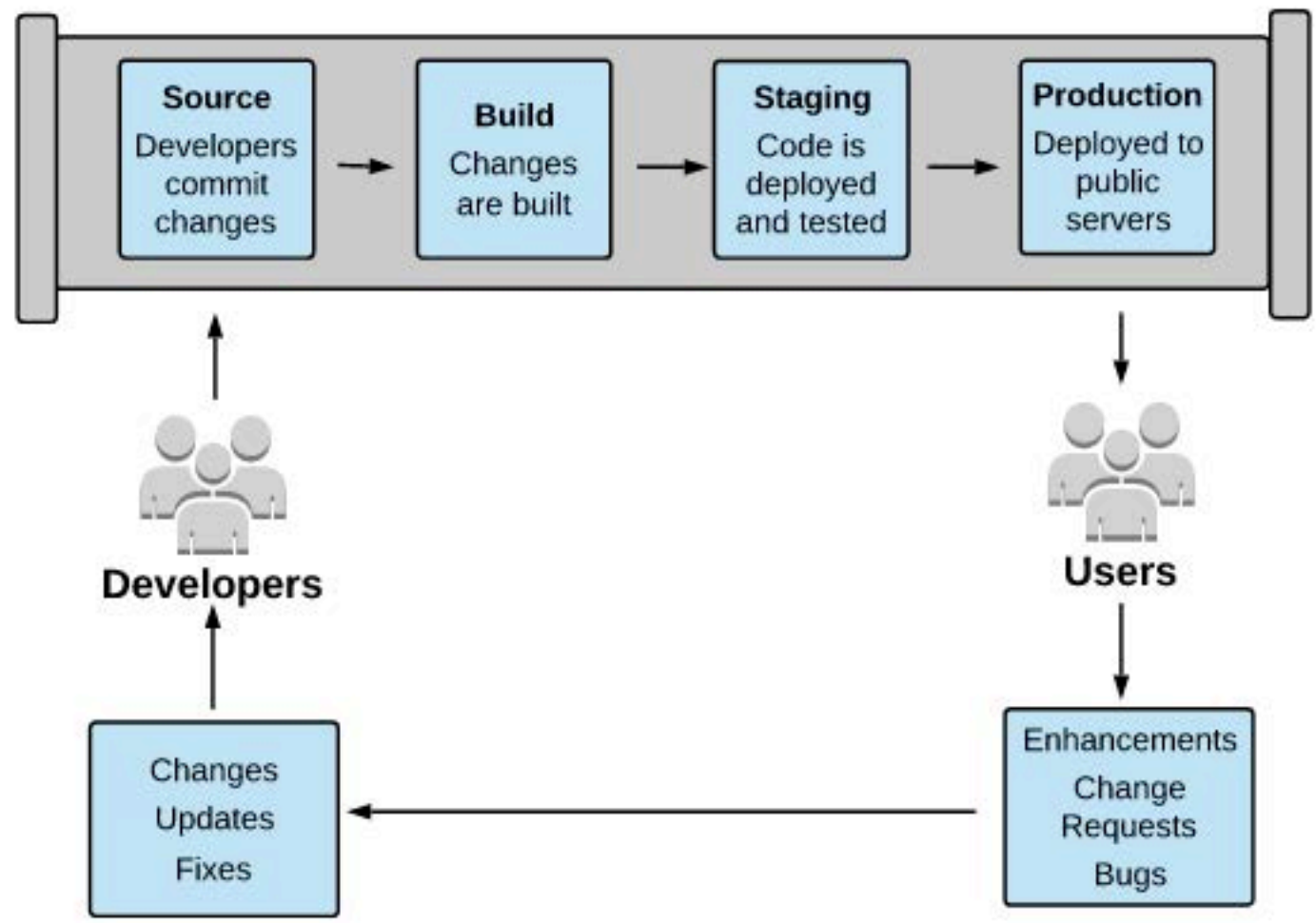
Choose a Topic

Deployment Pipelines

Back

AWS CodePipeline

Next



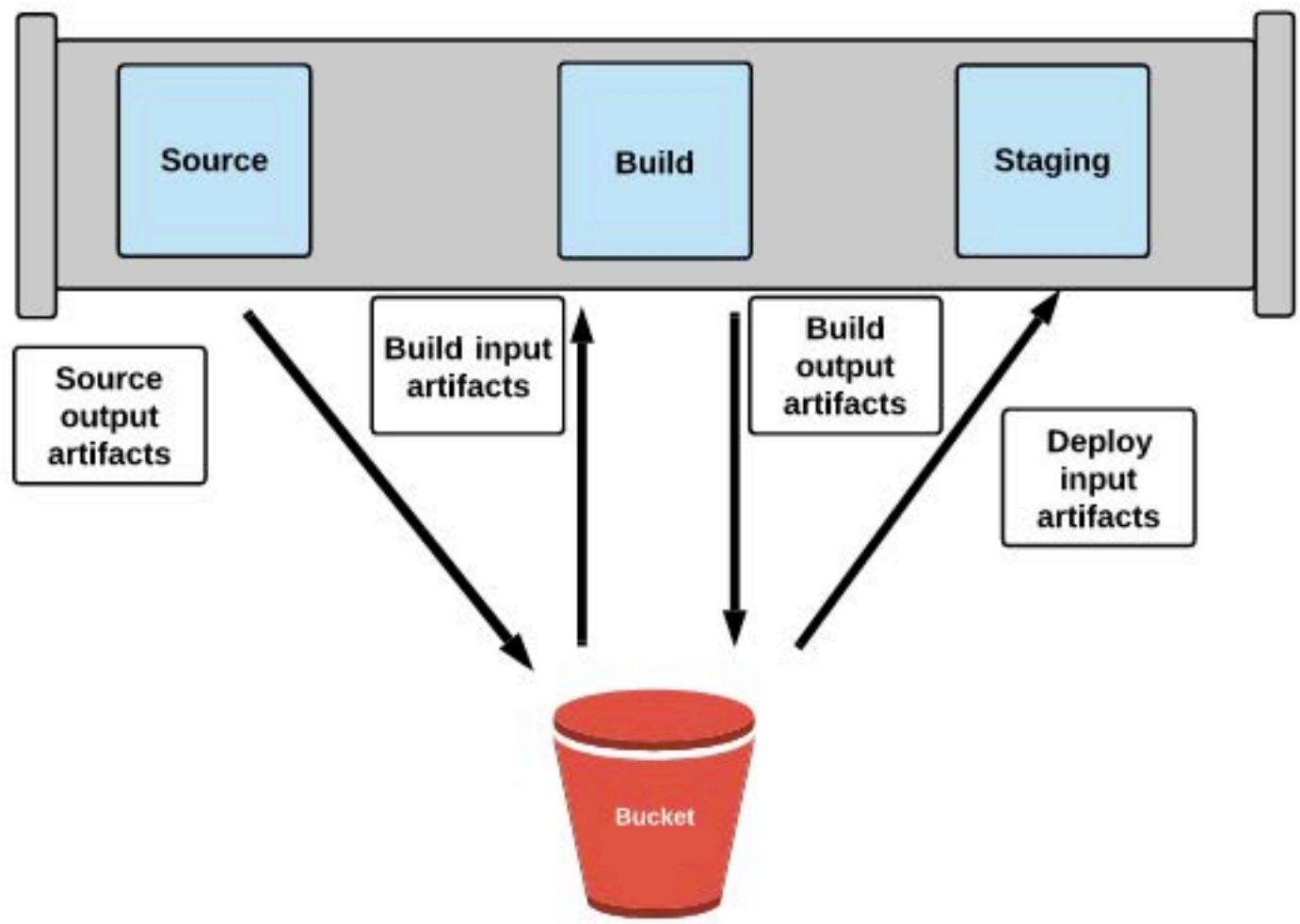
- What is CodePipeline? A continuous delivery service you can use to model, visualize, and automate the steps required to release your software.
- It automates the steps required to release your software changes continuously. Automatically build, test, and deploy your applications in the cloud.
- Steps: 1) Developers commit changes to a source repository. 2) CodePipeline automatically detects those changes. 3) Changes are built, and if any tests are configured, those tests are run. 4) After the tests are complete, the built code is deployed to staging servers for testing. 5) From the staging server, AWS CodePipeline runs additional tests, such as integration or load tests. 6) After a manual approval action that was added to the pipeline is approved, AWS CodePipeline deploys the tested and approved code to production instances.
- Can deploy applications to Amazon EC2 instances by using AWS CodeDeploy, AWS Elastic Beanstalk, or AWS OpsWorks Stacks.
- Can also deploy container-based applications to services by using Amazon ECS.

Choose a Topic

Deployment Pipelines

Back

AWS CodePipeline Artifacts



Amazon S3 Artifact Bucket

- CodePipeline integrates with development tools to check for code changes and then build and deploy through all stages of the continuous delivery process.
- Stages use input and output artifacts that are stored in the artifact store for your pipeline.
- Artifact store is in the same region as the pipeline to store items for all pipelines in that region associated with your account.
- Every time you use the console to create another pipeline in that region, AWS CodePipeline creates a folder for that pipeline in the bucket.
- It uses that folder to store artifacts for your pipeline as the automated release process runs.
- If you add a cross-region action to your pipeline, you provide an artifact bucket for each region where you have actions.
- CodePipeline zips and transfers the files for input or output artifacts as appropriate for the action type in the stage.