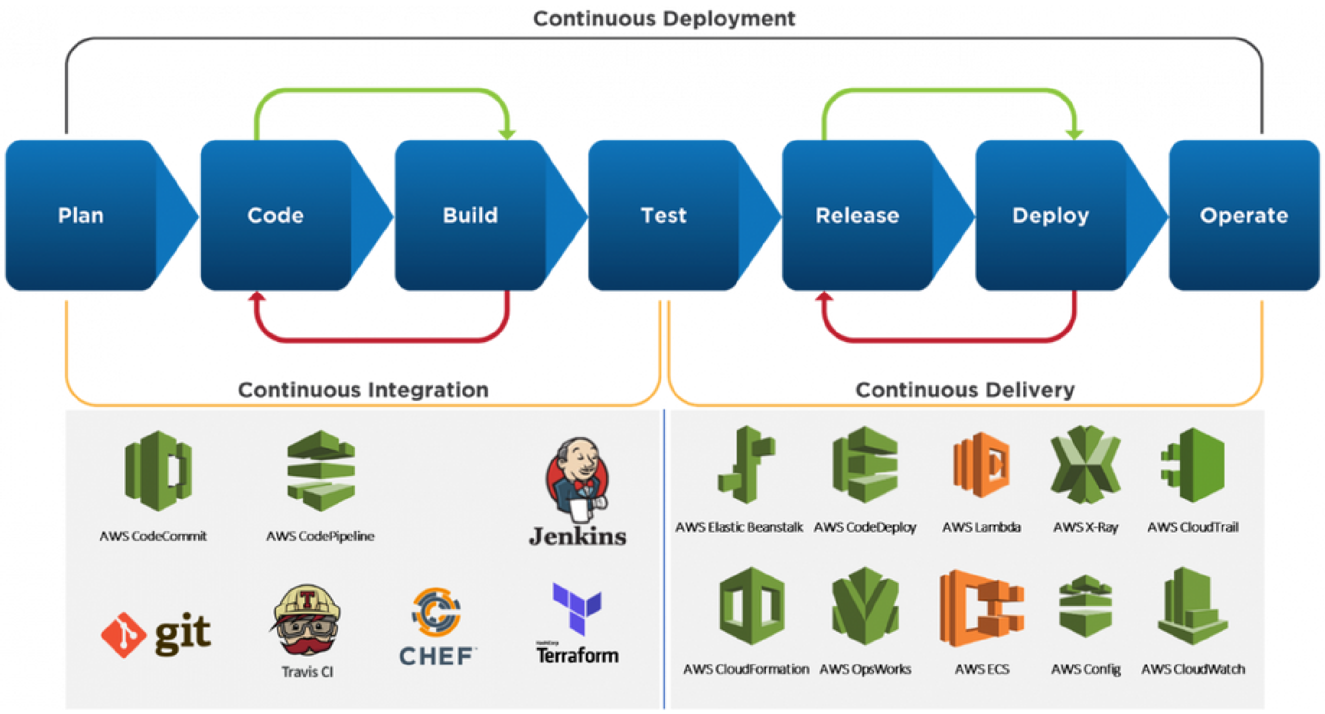
# The Workflow!

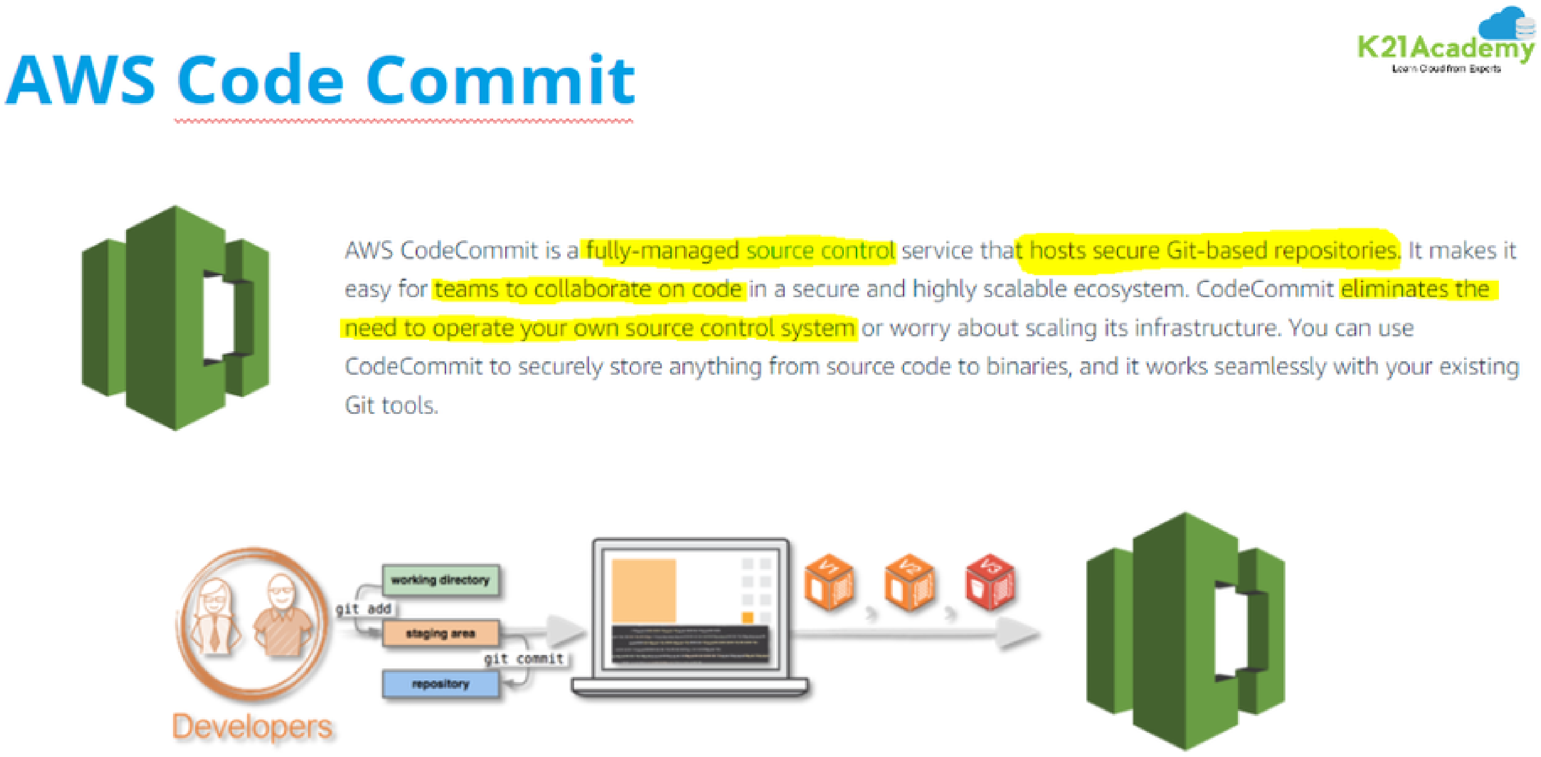
Let's break down the flow of the AWS services based on the software development lifecycle stages:



- **Plan**: In this stage, you define and plan your software project. You outline the features, requirements, and overall strategy. Tools like **AWS CodeStar** can be used for planning (JIRA & Azure Board are used in DevOps) and organizing your project.

- **Integrated Development Environment (IDE**) is a software tool that combines code editing, building, and debugging in one interface. Just like Visual Studio Code (from Microsoft) & **AWS Cloud9** from AWS.

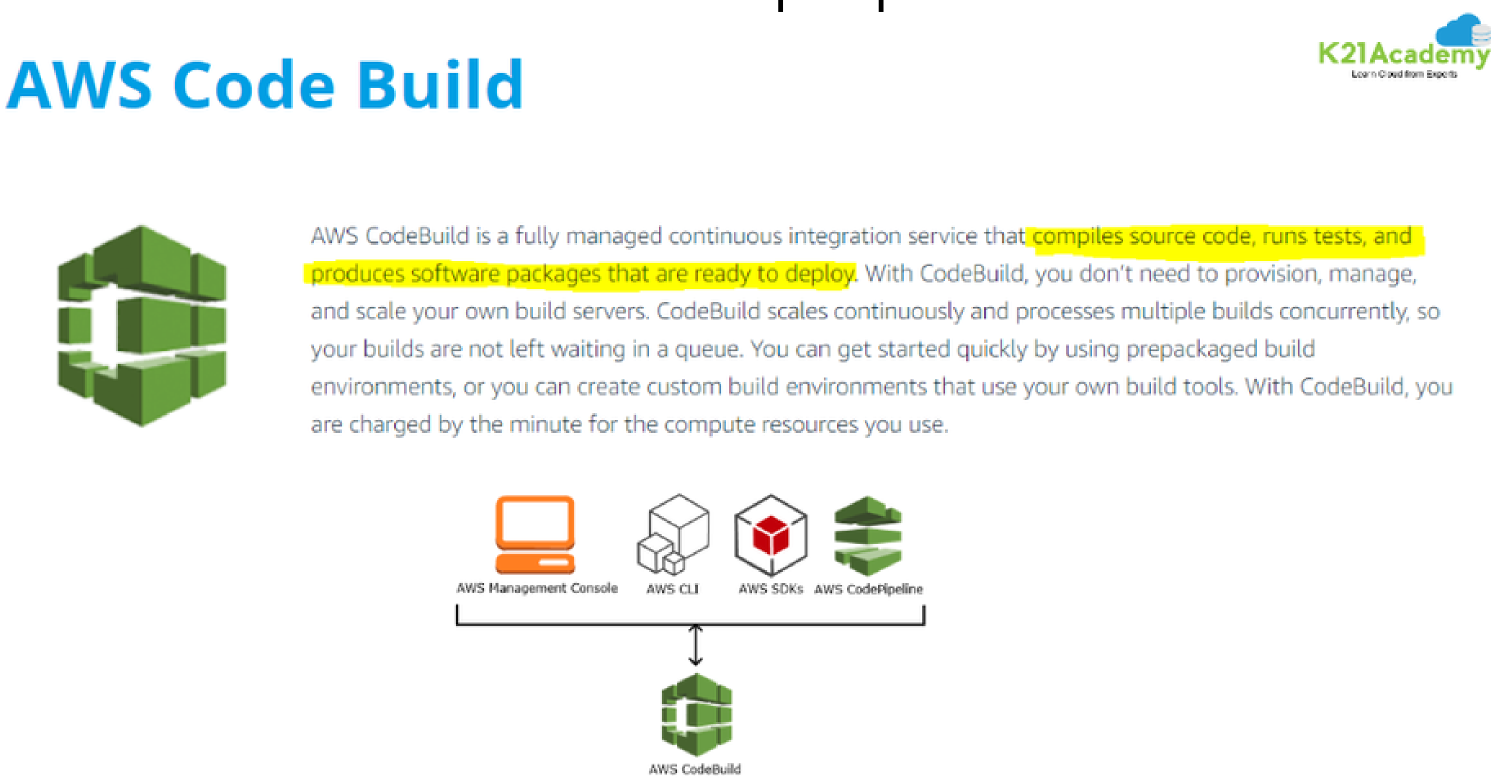
- **Code Repo:** Just like you use GIT, you have **AWS CodeCommit** for code repository to store and manage your code securely in the Cloud.



## AWS Code Commit

AWS CodeCommit is a fully-managed source control service that hosts secure Git-based repositories. It makes it easy for teams to collaborate on code in a secure and highly scalable ecosystem. CodeCommit eliminates the need to operate your own source control system or worry about scaling its infrastructure. You can use CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.

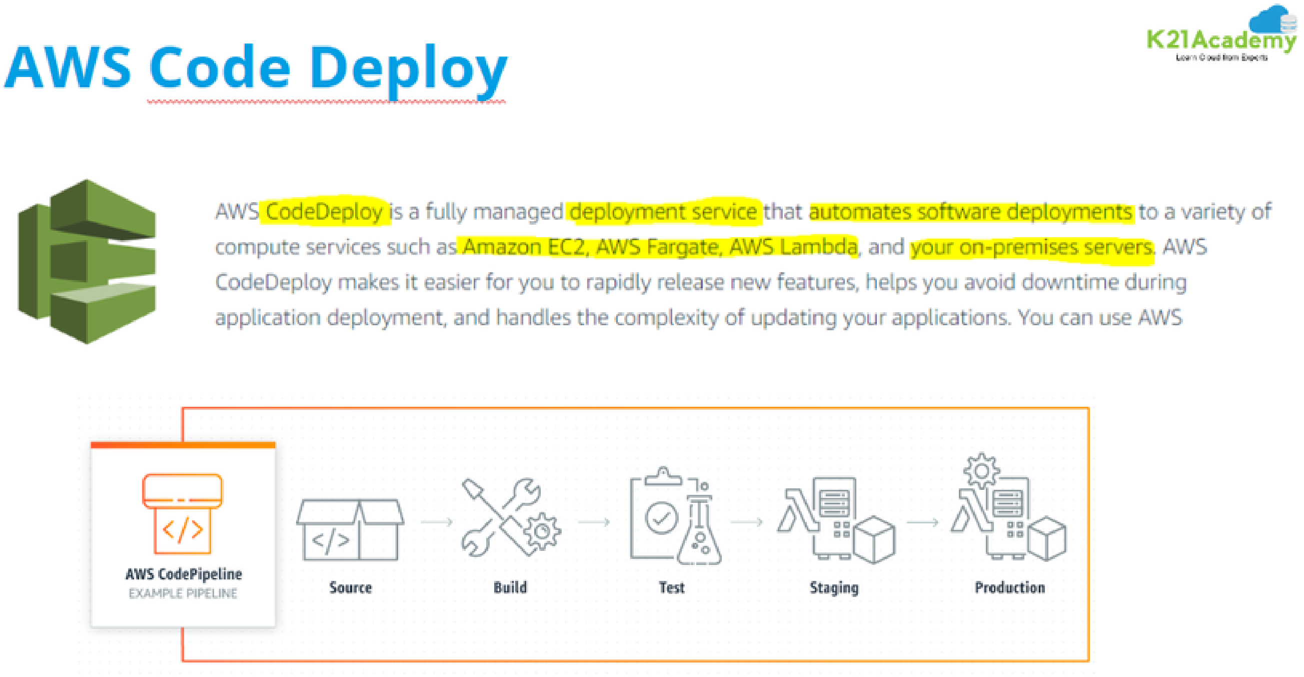
- **Build**: In the build stage, you compile your code, run tests, and create build artifacts. Just like you use Maven and Graddle for building, you have **AWS CodeBuild** for the same purpose.



## AWS Code Build

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use.

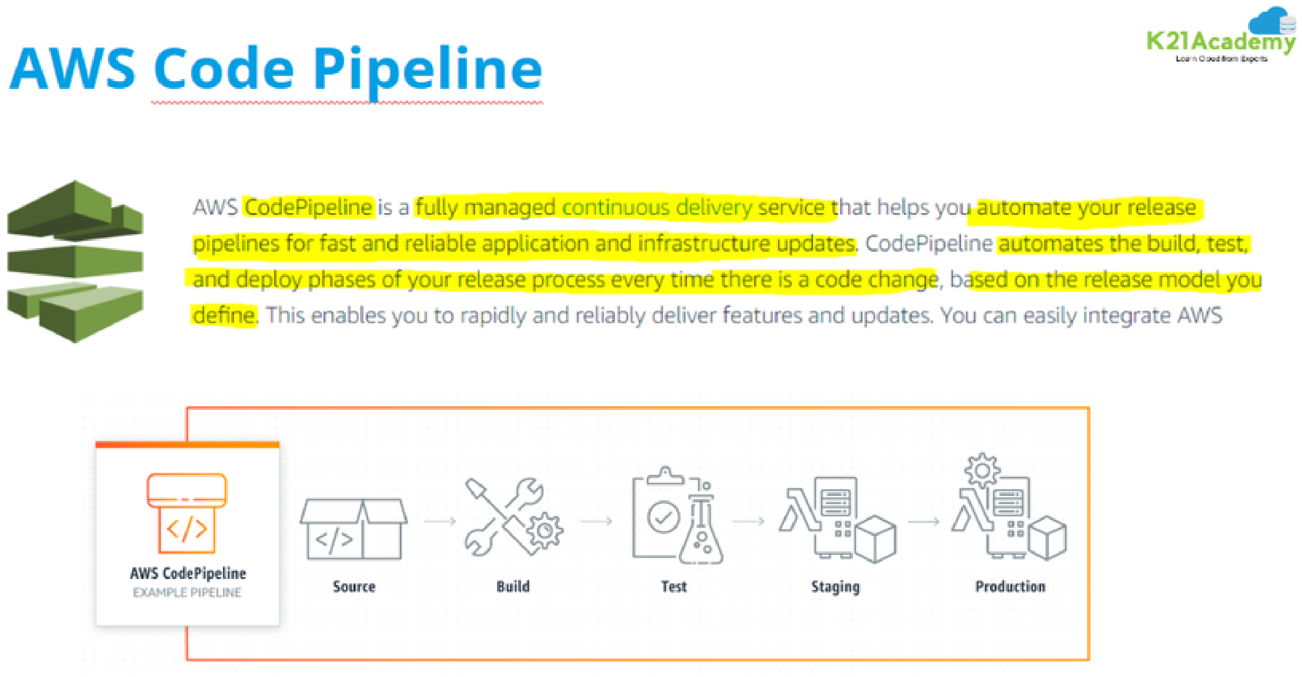
- **Deploy/Release:** You take the packaged code and put it into the environment where it will run. **AWS CodeDeploy** automates the deployment to different services, like EC2, Containers (EKS, AKS, Fargate) & Elastic Beanstalk.



AWS Code Deploy

AWS CodeDeploy is a fully managed deployment service that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications. You can use AWS

- **CI/CD:** It is a process of automating software development and delivery. Just like you use Jenkins to implement CI/CD pipelines, you have AWS **CodePipeline** for the same purpose



AWS Code Pipeline

AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. You can easily integrate AWS

- **Infrastructure-as-Code (IaC):** In DevOps, Terraform is used for infrastructure provisioning. In AWS, **AWS CloudFormation** is used.

- **Operate/Manage:** After deployment, you monitor and manage your application in a production environment. AWS provides tools like Amazon CloudWatch, AWS X-Ray, AWS CloudTrail, and AWS Systems Manager for this.

    - **Amazon CloudWatch:** Monitors and collects metrics, logs, and events for AWS resources and applications.

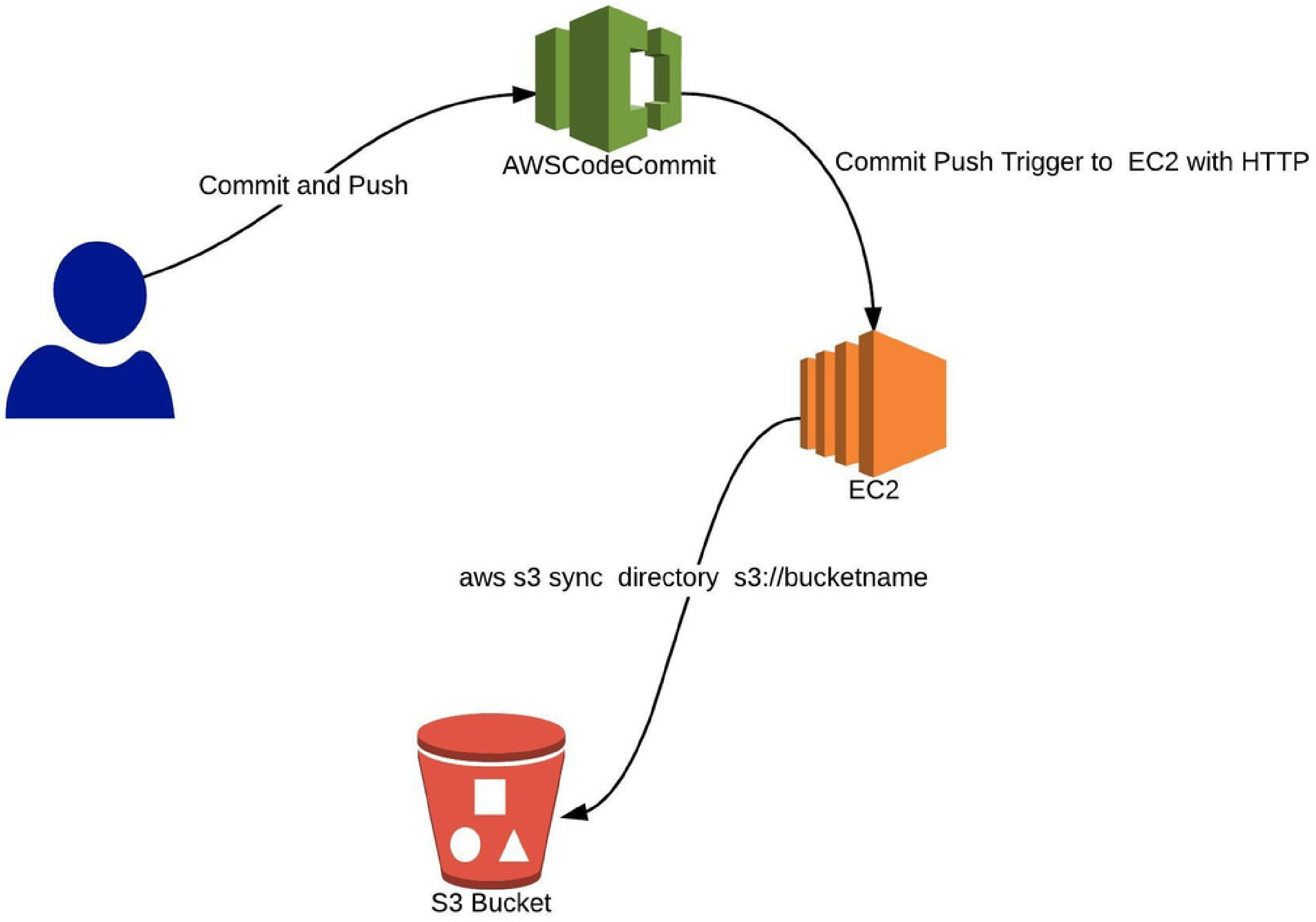
    - **AWS X-Ray:** Provides insights into application performance and behavior, helping with troubleshooting and optimization.

    - **AWS CloudTrail:** Offers visibility into user activity and API usage, aiding in monitoring and auditing AWS API calls.

    - **AWS Systems Manager:** Provides a unified interface for managing and operating AWS resources, including automation, configuration management, and patch management.

*Let's understand why & when to use these services & tools. . .*

# 1. AWS CodeCommit

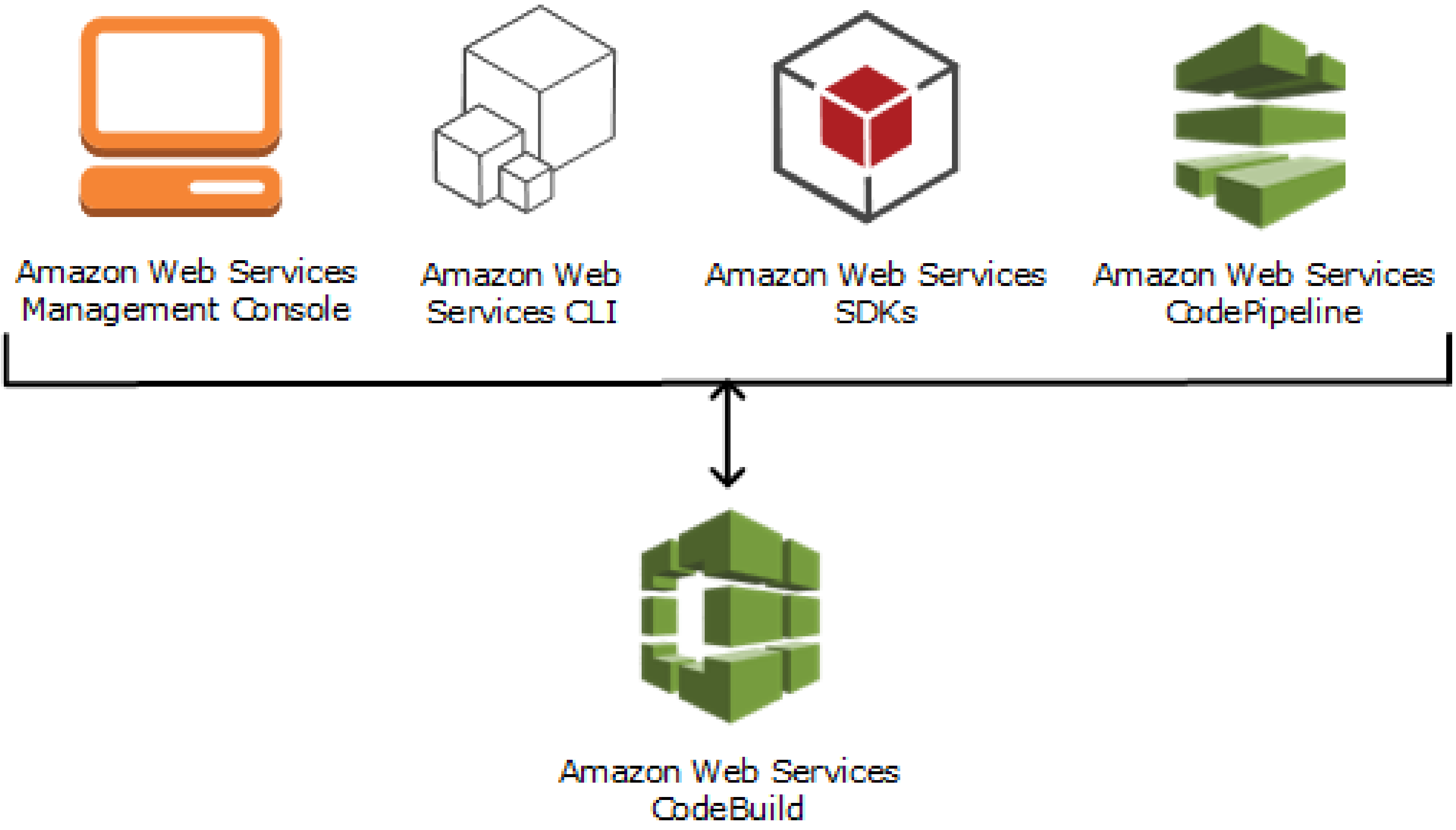

AWSCodeCommit
Commit and Push
Commit Push Trigger to  EC2 with HTTP
EC2
aws s3 sync  directory  s3://bucketname
S3 Bucket

AWS CodeCommit is a secure and managed version control service for your code.

- **Why to use?** It helps teams collaborate, manage different code versions, and integrates smoothly with other AWS services.

- **When to use?** Use CodeCommit when you need a reliable and scalable storage solution for your codebase and want to streamline collaboration among developers.

# 2. AWS CodeBuild


Amazon Web Services Management Console
Amazon Web Services CLI
Amazon Web Services SDKs
Amazon Web Services CodePipeline
Amazon Web Services CodeBuild

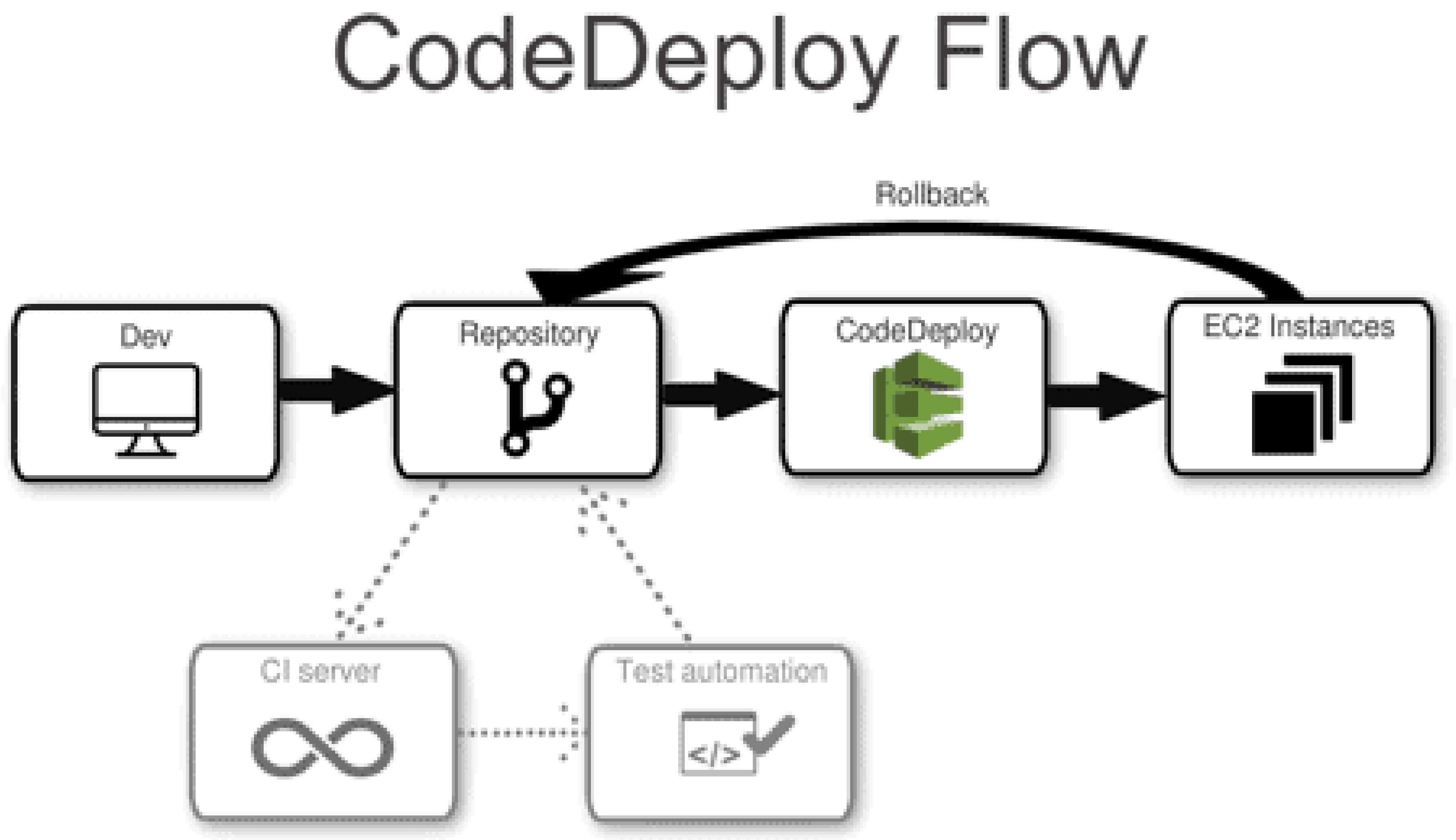
AWS CodeBuild is a service that builds your code automatically in the cloud.

- **Why to use?** It saves you time and effort by handling the entire build process for your software projects.

- **When to use?** You should use CodeBuild when you want to automate your code builds, ensure consistency, and easily integrate with other AWS services. It's great for collaborative projects and helps streamline the development and deployment of your applications.

K21Academy
Learn Cloud from Experts

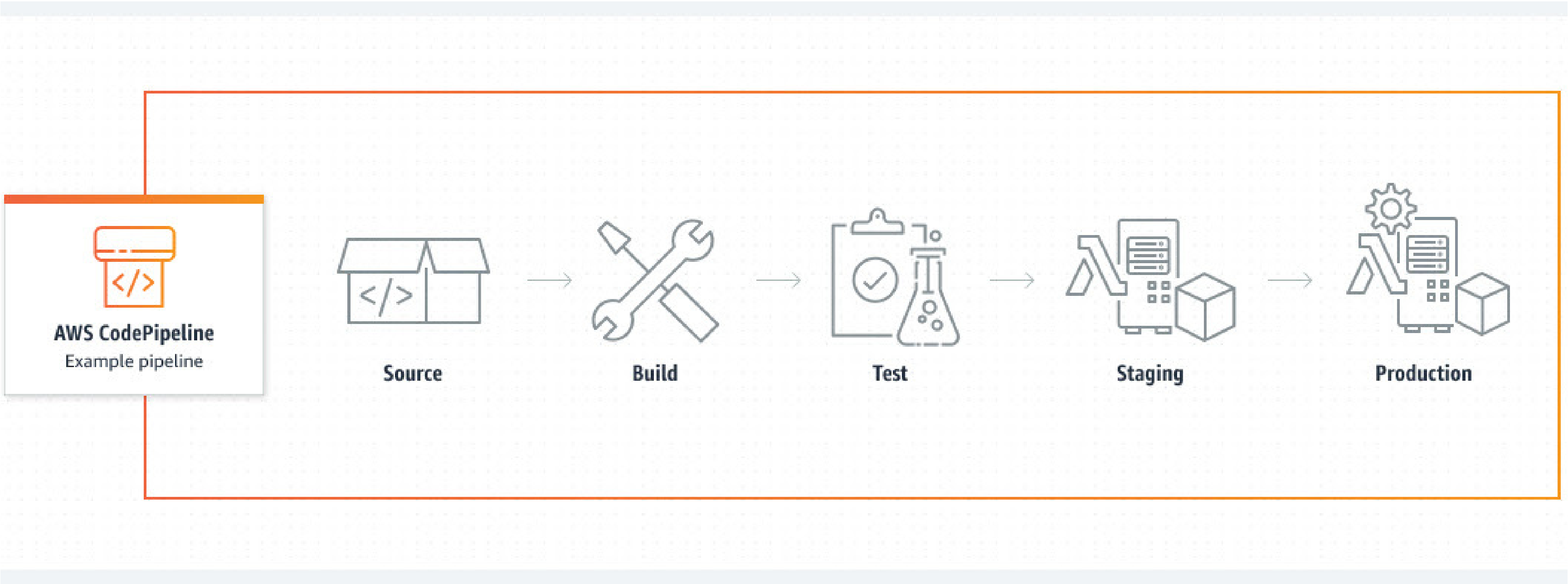
## 3. AWS CodeDeploy



AWS CodeDeploy is a service that automates your application deployments to different AWS resources.

- **Why to use?** It saves time, reduces errors, and allows for gradual updates.
- **When to use?** Use it when you want to streamline and simplify your deployment process on AWS.

# 4. AWS CodePipeline



AWS CodePipeline is an automated tool that handles the entire process of building, testing, and deploying your code.

- **Why to use?** It saves you time and reduces errors by streamlining these steps.

- **When to use?** You should use CodePipeline when you want to automate software delivery and ensure consistent and efficient deployment, especially for projects that require frequent updates or involve team collaboration.

# Join Free Class!

If you are interested in learning more about DevOps on AWS Cloud, I recommend you to join me for this free class.

We're going to thoroughly discuss these tools in the class & more...

✓ Why Learn DevOps on AWS Cloud?
✓ What is AWS DevOps Who should Learn?
✓ CI/CD: CodeCommit, CodeBuid, CodeDeploy, CodePipeline, Test Plans
✓ Infra & Config Management: EKS, ECS, VM, BeanStalk, Chef, Ansible, CloudFormation, Terraform
✓ DevOps for Security (Rugged or DevSecOps) & Testers/QA
✓ Demo: Blue-Green Deployment to App on AWS
✓ Hands-On Labs to clear AWS DevOps Certification
✓ Job Opportunities & Get Better Paid Job in AWS DevOps on Cloud
✓ Docker & Kubernetes on AWS (ECR, ECS, EKS & FarGate)
✓ A Q/A Session With Experts & A Fast Action Gift

**Plus A Special Gift:**
You'll get 30 AWS DevOps Exam Questions as a FREE Bonus when you attend the FREE Class till the end.

K21Academy
Learn Cloud from Experts