## **Review and finalize all documentation**

**1. AWS CodePipeline**

AWS provides **CodePipeline** for automating CI/CD workflows. Here’s a basic setup:

**Components:**

* **AWS CodeCommit**: Source repository
* **AWS CodeBuild**: Build and test automation
* **AWS CodeDeploy (Optional)**: Deployment service

**Setup Steps:**

1. **Create a CodeCommit Repository**
   * Use AWS Console or CLI to create and push code.
2. **Configure CodeBuild**
   * Define a buildspec.yml file with build steps.
3. **Set Up CodePipeline**
   * Define stages: Source -> Build -> Deploy (if needed).
   * Integrate with CodeCommit, CodeBuild, and optional deployment options.

**2. Azure DevOps Pipeline**

Azure provides **Azure Pipelines** as part of Azure DevOps for CI/CD.

**Components:**

* **Azure Repos**: Git repository
* **Azure Pipelines**: CI/CD service

**Setup Steps:**

1. **Create an Azure DevOps Project**
   * Add a repository (Azure Repos or GitHub).
2. **Define a YAML Pipeline** (azure-pipelines.yml)
3. **Run and Monitor Pipeline**
   * Validate build status and logs in Azure DevOps.

**3. Google Cloud Build**

GCP offers **Cloud Build** to automate builds, tests, and deployments.

**Components:**

* **Cloud Source Repositories / GitHub**: Source repository
* **Cloud Build**: CI/CD service

**Setup Steps:**

1. **Enable Cloud Build API**
   * Enable it via GCP Console or CLI.
2. **Define Build Configuration (cloudbuild.yaml)**
3. **Trigger Builds Automatically**
   * Set up a trigger linked to a Git repository.
   * Configure via GCP Console or gcloud CLI.

**Conclusion**

Each cloud provider offers a robust CI pipeline solution, with similar configuration steps:

* **AWS CodePipeline** integrates well with AWS services.
* **Azure DevOps Pipelines** provides extensive flexibility and Azure integrations.
* **Google Cloud Build** offers a managed and scalable build system.