

# Week 2 Reading/Resources

## Testing

To ensure quality, it's important for teams to incorporate testing into the software development lifecycle at different stages of the continuous integration and continuous delivery (CI/CD) pipeline. Overall, testing should start as early as possible.

While there are many kinds of application tests, the three mentioned in this week are:

- **Regression testing** - Tests to ensure that previously developed applications don't break with new changes
- **Integration testing** - Tests to ensure separately developed modules work together as expected
- **Unit testing** - Tests a specific section of code to ensure the code does what it is expected to do

The exercises in this course used pytest for unit testing. For more information, see: [pytest: helps you write better programs](#)

## Automate testing

AWS CodeBuild will provision a clean and consistent environment to perform various application tests. You can also create reports in CodeBuild that contain details about tests that are run during builds.

For more information about CodeBuild and test reporting, see: [Working with test reporting in AWS CodeBuild](#)

If you're interested, check out this GitHub repository, which contains Dockerfiles of official, curated CodeBuild Docker images: [aws-codebuild-docker-images](#)

## AWS CodePipeline

You can automate your release process by using AWS CodePipeline to test your code and run your builds with CodeBuild.

CodePipeline is a continuous delivery (CD) service that you can use to model, visualize, and automate the steps required to release your code. This process includes building your code. A *pipeline* is a workflow construct that describes how code changes go through a release process.

## AWS CodeDeploy

If you want to read more about in-place and blue/green deployments with AWS CodeDeploy, see: [Working with deployments in CodeDeploy](#)

To use CodeDeploy on Amazon Elastic Compute Cloud (Amazon EC2) instances or on-premises servers, the CodeDeploy agent must be installed first. For more information, see: [Install the CodeDeploy agent](#)

## **Troubleshooting deployment details and errors**

You can view the log data created by a CodeDeploy deployment by setting up the Amazon CloudWatch Logs agent to view aggregated data in the CloudWatch console or by logging into an individual instance to review the log file. For more information, see: [View log data for CodeDeploy EC2/On-Premises deployments](#)

For general troubleshooting tips for CodeDeploy, see the following resources:

- [General troubleshooting issues](#)
- [Troubleshoot EC2/On-Premises deployment issues](#)
- [Troubleshoot AWS Lambda deployment issues](#)
- [Error codes for AWS CodeDeploy](#)