aws re/start

Configuration Management

Python Fundamentals

Name of presenter

Date

What you will learn

At the core of the lesson

You will learn how to:

- Define project infrastructure
- Explain why project infrastructure is important to the success of a project
- Define the purpose and function of software configuration management

Project infrastructure

Project infrastructure

Project infrastructure is the way that a project is organized. An architect organizes the infrastructure of a bridge. Software developers organize the infrastructure of code.

Traditional project infrastructure

Traditional project infrastructure is a less-efficient way of developing code between teams.

When one team starts a project, they have their own way of automating their DevOps process. Then, another team comes up with their own way to automate their DevOps process.

+

After multiple teams work on the project, a single team automates the DevOps process. They arrive at one final, cohesive test of the code before they make it available for use.

=

This process can become tedious.

Code organization

What does it mean to have code that is well organized?

Most companies have a certain coding style that their employees should follow. That style includes a way of naming variables in their code and the number of spaces to indent code blocks.

It is important to mention that each company varies in this practice. The details of the style matter less than following the style. It is more important for employees to follow the style so that they can avoid confusion.

There shouldn't be more than one set of tests for the various teams on a single project. That set of tests includes logic tests and compiling code.

Tools and templates

Many tools are available to build a cohesive infrastructure across teams.

For style

Utilities like **pylint** can be run to ensure that code blocks are indented correctly, and to fix the code blocks that are not formatted well.

For logic

Utilities like **pytest** can be used to run tests to make sure that code changes still meet the requirements.

Software configuration management

What is configuration management?

- Tracks versions of the code as it is developed
- Enables developers to work independently on different parts of a project, and then merge changes back into the project
- Version control software (such as Git) tracks what code changed and who made the changes
- When errors occur, configuration management enables fast rollbacks to previous, functioning versions

How does configuration management work?

- Developers *check out* code from a repository like GitHub.
- When they are finished with the code, developers upload their changes to the repository.
- When the new code passes all tests, it can be *merged* back into the main project.
- The process of checking code in and out can be done by:
  - Running **Git** from the command line
  - Using tools that are built into integrated development environments (IDEs), such as PyCharm
- Running tools—such as **pylint** and **pytest**—can also be a part of the check-in process.

Configuration management by example

The following simple example uses Git for configuration management because other steps might be needed, depending on how the repository (repo) is configured.

Get a copy of the remote repo: `$git @<examplerepo.org>:<username>/<sourcecode>.git`

Commit changes locally: `$git –commit –m “Message about the changes.”`

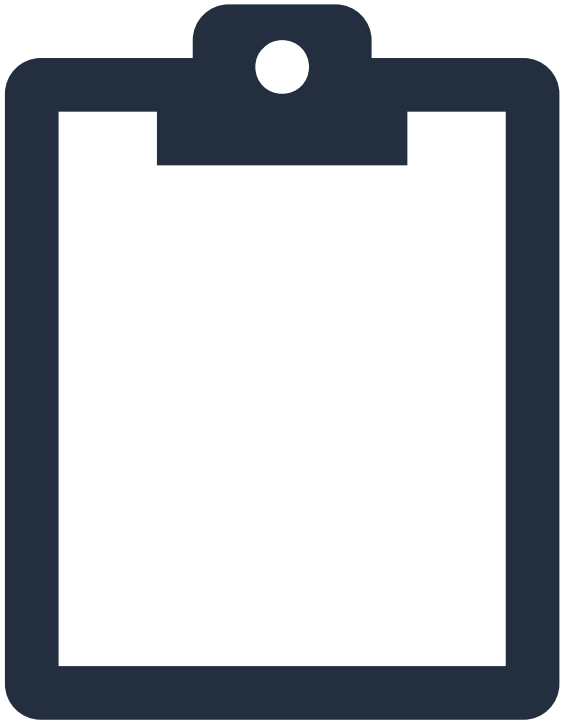
Push change back to the repo: `$git push`

Configuration management versioning

- As developers update the code, the release managers who are responsible for distributing new versions of software can monitor the changes.
- After all tests and functionality are verified, the release manager creates a new distribution of the software based on the contents of the repository.
- Release managers can now *version* the software, which helps manage customer issues when problems occur. (This point is where *rolling back to a previous version* becomes important.)
- In most cases, versioning takes the form of a numeric value (for example: *Version 3.2.1*).

Configuration management accounting

- Sometimes team leads and managers ask for the status of a project.
- With configuration management, the team can quickly report on the ongoing efforts by inspecting check-ins, check-outs, and other activities in the project's repository.



Configuration management security

- Because access to a repository must be granted, it stops unauthorized persons from gaining access to source code

- Because access is logged, it is possible to learn –
  - Who checked out and checked in code
  - When the check-ins and check-outs were done
  - What changes were committed

Key takeaways

- Project infrastructure is a critical discipline for helping to ensure that projects reach their goals.
- Project infrastructure also includes ensuring that Python code is styled properly and that it functions as expected.
- **pylint** checks the style of code, and **pytest** runs tests to ensure that the code works as expected.
- Software configuration management involves the use of code repositories to manage the code that is used in projects.
- To work with code repositories, tools are built into IDEs, and you could also use the command line tool **Git**.