Recitation 7

Continuous Integration

Continuous Integration

- A sequence of stages through which the system has to go through before it can be deployed; usually followed by continuous deployment stages
- Flow
  - Code commit triggers a new pipeline run
  - Pipeline executes
  - If the CI pipeline passes, CD pipeline starts
- Main goal is to reduce the time taken from code commit to deployment (with CD)
- Another goal is to automate activities (or reduce manual effort as much as possible) that need to be repeated for every code commit, which ties back to the main goal.

CI Pipeline

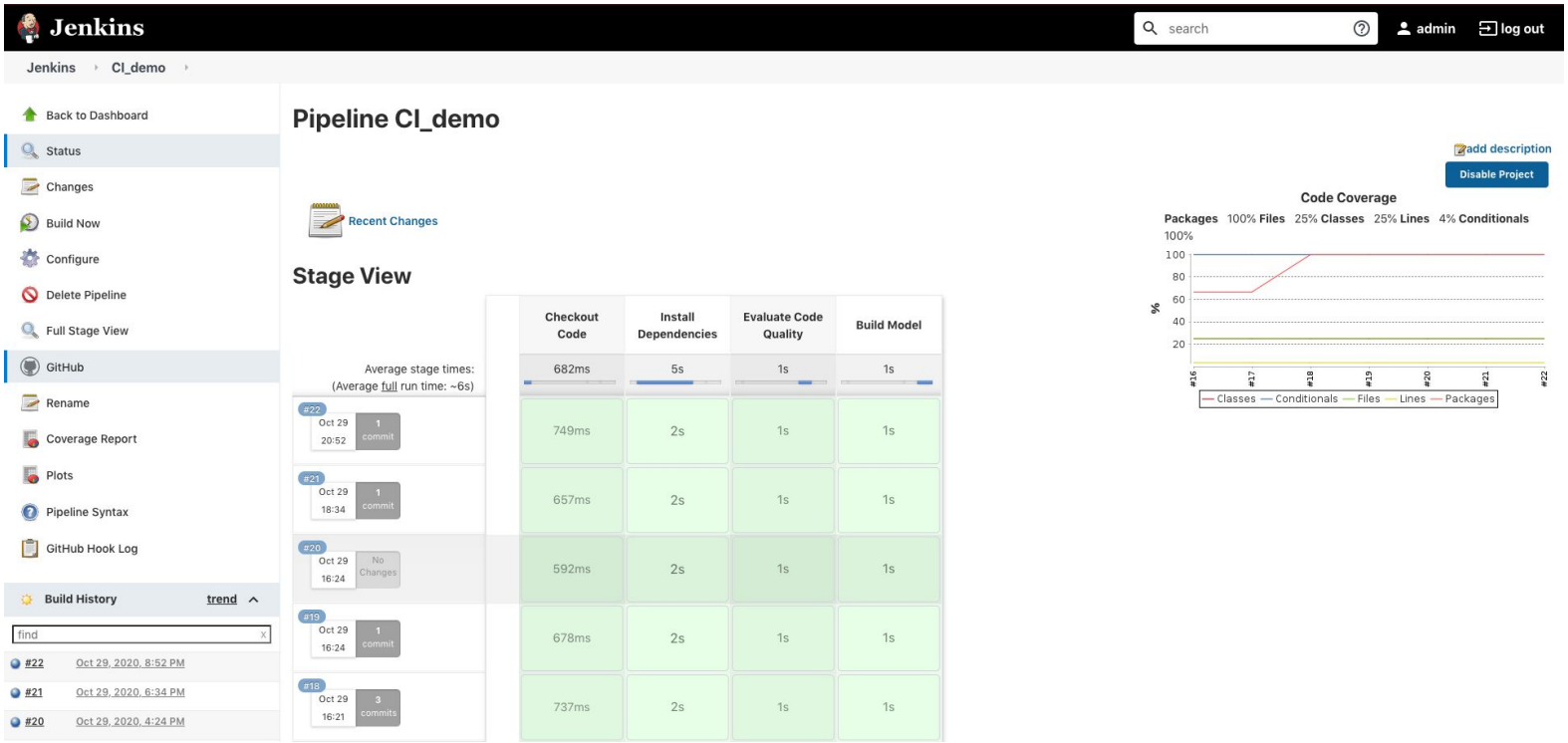
- Defined set of stages which run in an automated fashion once triggered
- Pipeline stages:
  - Checkout code → Set up environment → Build code → Static checks → Unit tests → Integration tests → Packaging the software → ...
- For machine learning, you may have more stages such as:
  - Data quality check, offline model evaluation, data collection, data cleaning/preprocessing, model serialization, telemetry data collection, etc.
- CI/CD tools: Jenkins, TravisCI, GitHub Actions, etc.

Demo

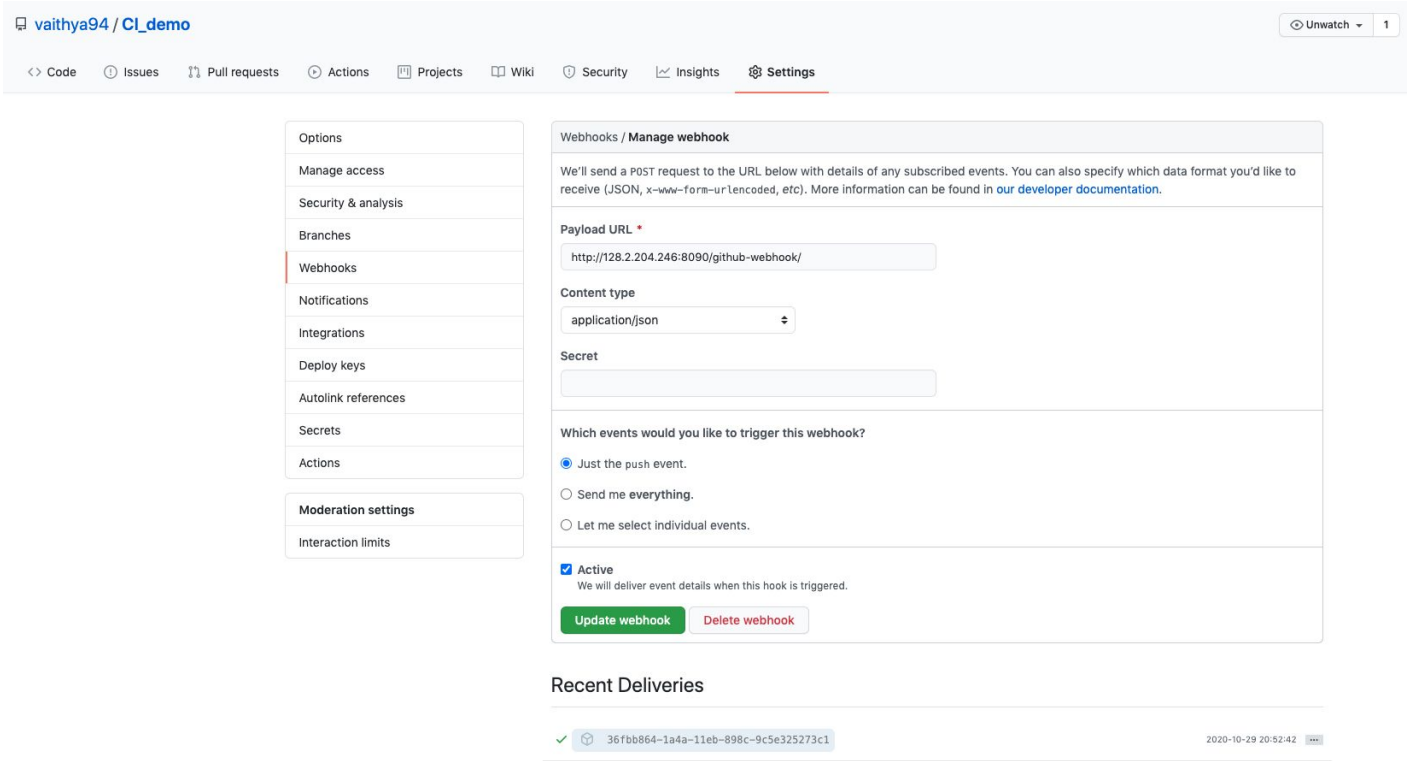
- Goals:
  - Look at some starter code and initial setup of a CI pipeline for a sample ML system
  - Save you some time (hopefully) in setting up your CI pipeline for Milestone 2
- Contents
  - Sample codebase [https://github.com/vaithya94/CI_demo]
  - Jenkins installation and GitHub integration
  - Jenkins pipeline structure
  - Jenkins coverage and plot plugins

NOTE: The Jenkins server from this demo will be taken down after the recitation, but you can refer the recording and the repo

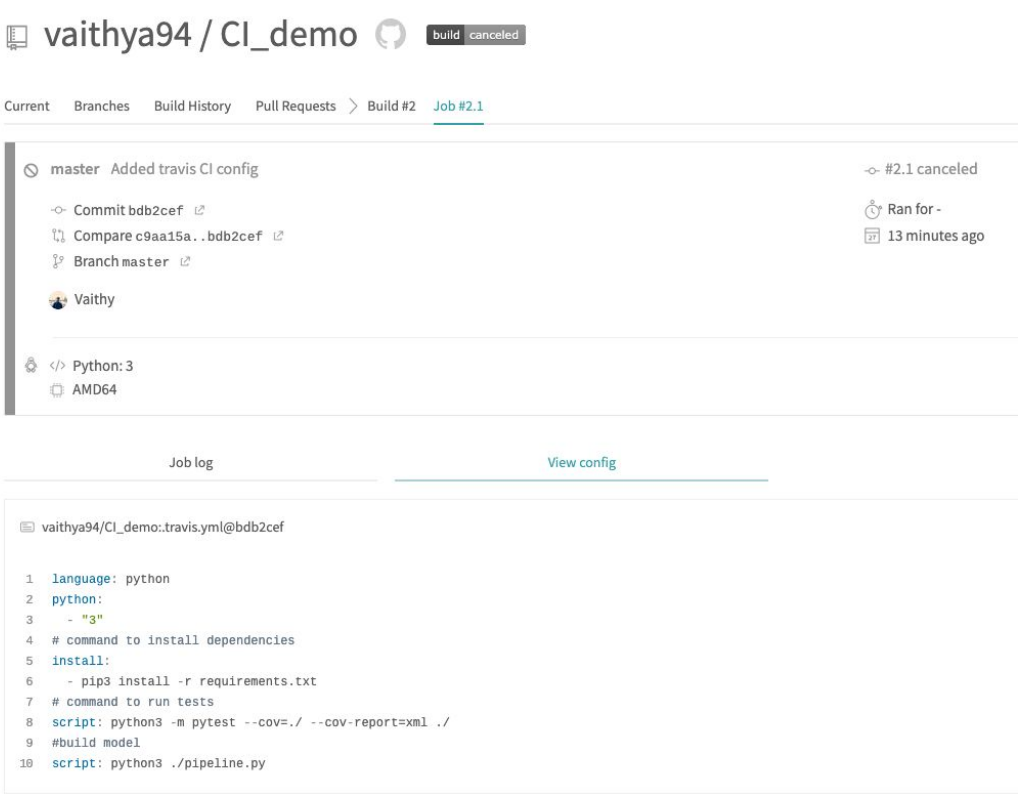
Jenkins Pipeline



Jenkins - GitHub Integration



TravisCI


vaithya94 / CI_demo
build canceled
Current
Branches
Build History
Pull Requests
Build #2
Job #2.1
master Added travis CI config
Commit bdb2cef
Compare c9aa15a..bdb2cef
Branch master
Vaithy
#2.1 canceled
Ran for -
13 minutes ago
Python: 3
AMD64
Job log
View config
vaithya94/CI_demo:.travis.yml@bdb2cef
1 language: python
2 python:
3 - "3"
4 # command to install dependencies
5 install:
6 - pip3 install -r requirements.txt
7 # command to run tests
8 script: python3 -m pytest --cov=./ --cov-report=xml ./
9 #build model
10 script: python3 ./pipeline.py

CI Pipeline Qualities

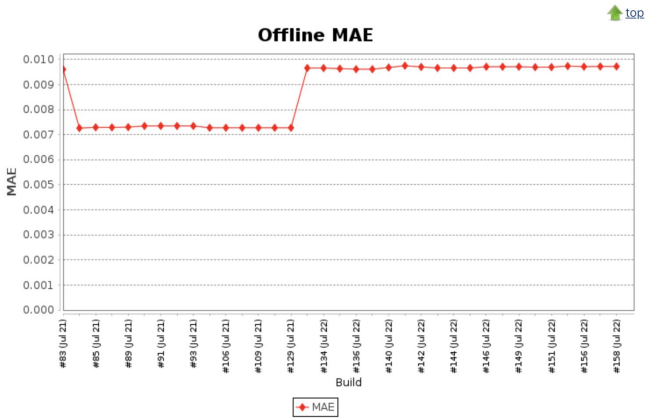
- Repeatable [consistent results across runs; consecutive runs are independent]
- Fault-tolerant [fail gracefully if any stage fails, ie. system remains operational]
- Correct [performs what is expected of it given some inputs]
- Robust [should be able to handle noise in any inputs the pipeline expects]
- Testable [stages of the pipeline should be independently testable]
- Traceable [should be possible to trace any error to its source quickly]
- Performant [should be possible to move through the pipeline quickly]

Testing ML & CI Pipelines

- Unit tests for independent stages of the machine learning pipeline (automated)
  - Adequacy can be measured in terms of statement/branch coverage, etc.
  - Can use equivalence classes, boundary value analysis, etc. to identify test cases
- Integration tests for APIs (automated + manual)
  - Adequacy can be measured in terms of statement/branch coverage, etc.
  - Can use equivalence classes, boundary value analysis, etc. to identify test cases
  - Mock dependencies
- Manual blackbox tests for the CI pipeline
  - Adequacy can be measured in terms of use cases, nodes in activity/flow diagrams, etc.

NOTE: Adequacy criteria can be defined in terms of criticality of the component to your system (for example)

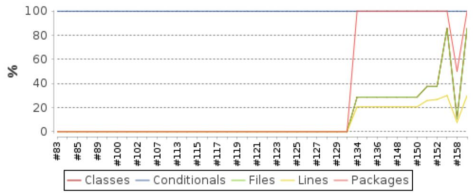
Automated Model Evaluation & Testing



Code Coverage

Cobertura Coverage Report

Trend



Project Coverage summary

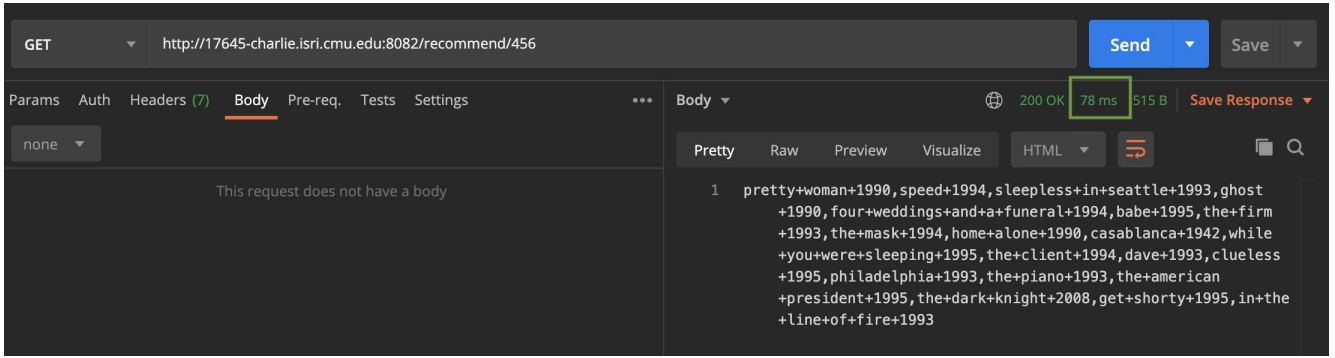
Name	Packages	Files	Classes	Lines
Cobertura Coverage Report	100% 1/1	86% 6/7	86% 6/7	30% 64/214

Coverage Breakdown by Package

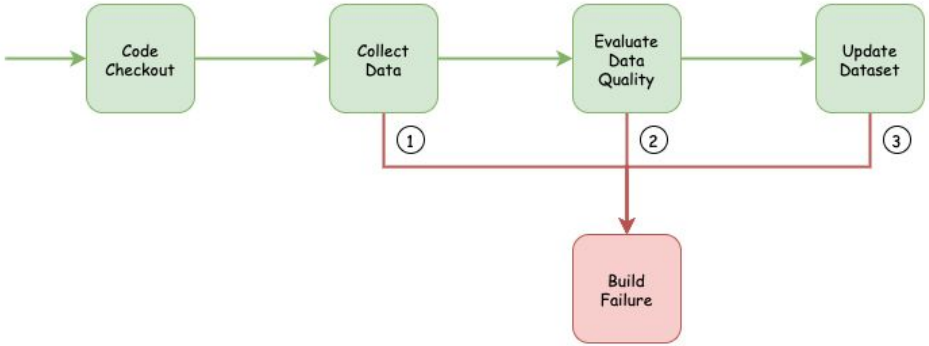
Name	Files	Classes	Lines
.	86% 6/7	86% 6/7	30% 64/214

Manual Testing

Blackbox Integration Testing - Postman



Blackbox Testing - Activity Diagram



Links

- Install Jenkins [https://www.jenkins.io/doc/book/installing/linux/]
- Jenkins plugins [https://plugins.jenkins.io/plot/, https://plugins.jenkins.io/cobertura/]
- Git to Jenkins integration [https://www.blazemeter.com/blog/how-to-integrate-your-github-repository-to-your-jenkins-project]
- Creating a pipeline in Jenkins [https://www.jenkins.io/doc/pipeline/tour/hello-world/]
- Example codebase [https://github.com/vaithya94/CI_demo]
- TravisCI [https://travis-ci.org/, https://docs.travis-ci.com/user/tutorial/#to-get-started-with-travis-ci-using-github]
- Creating a pipeline in TravisCI [https://docs.travis-ci.com/user/languages/python/]
- TravisCI - Plot using Coverall [https://docs.travis-ci.com/user/coveralls/]
- PyBuilder [https://pybuilder.io/, https://pythonhosted.org/pybuilder/walkthrough-new.html]

NOTE: There are a lot more useful resources online, and a lot more plugins for plotting - feel free to choose whatever works for you!

Thank You!