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)

CI Pipeline

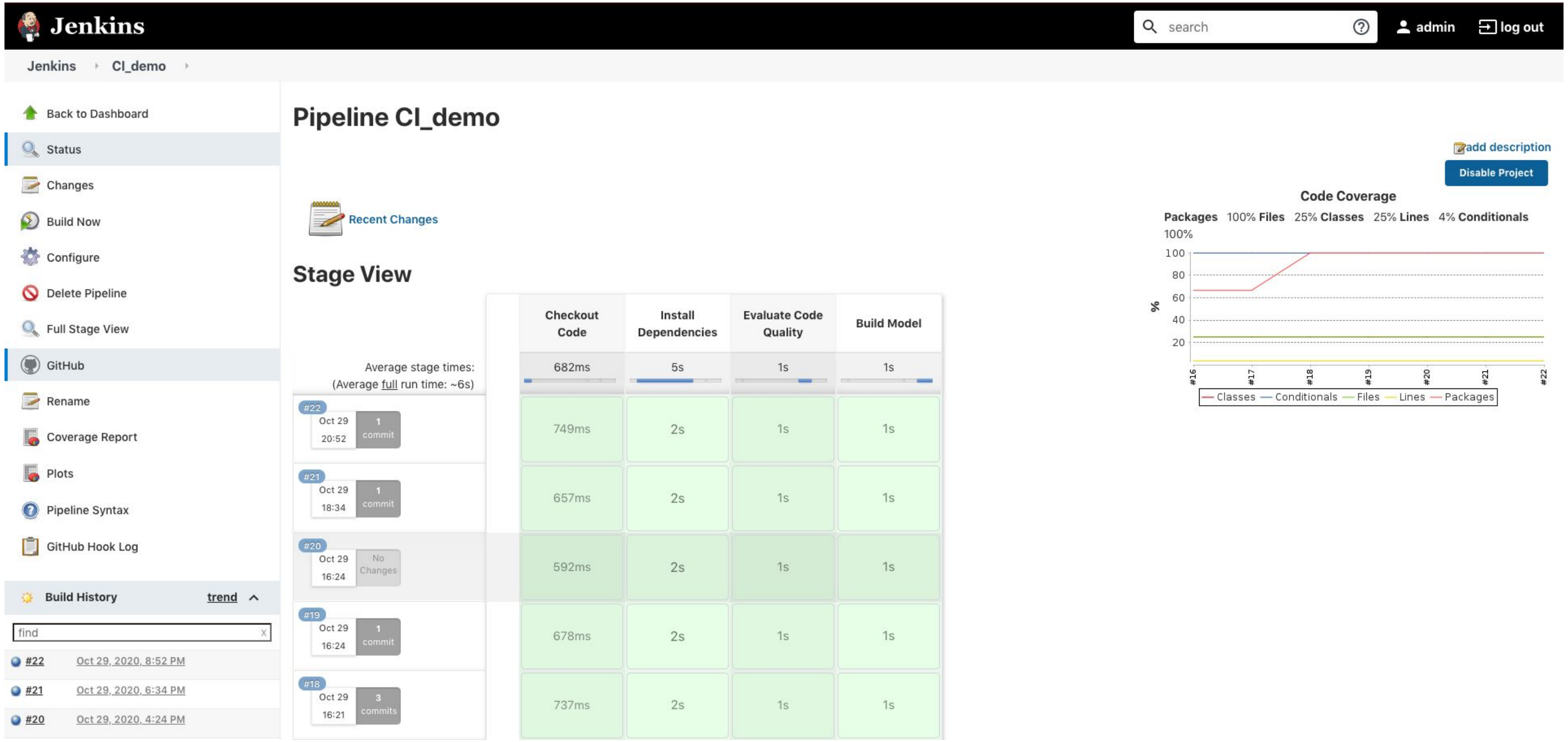
- Defined set of stages which run in an automated fashion once triggered
- Pipeline stages:
  - Checkout code → Set up environment → Build code → Static checks → Unit tests → Integration tests → Packaging the software → …
- For machine learning, you may have more stages such as:
  - Data quality check, 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



Pipeline CI_demo

Stage View

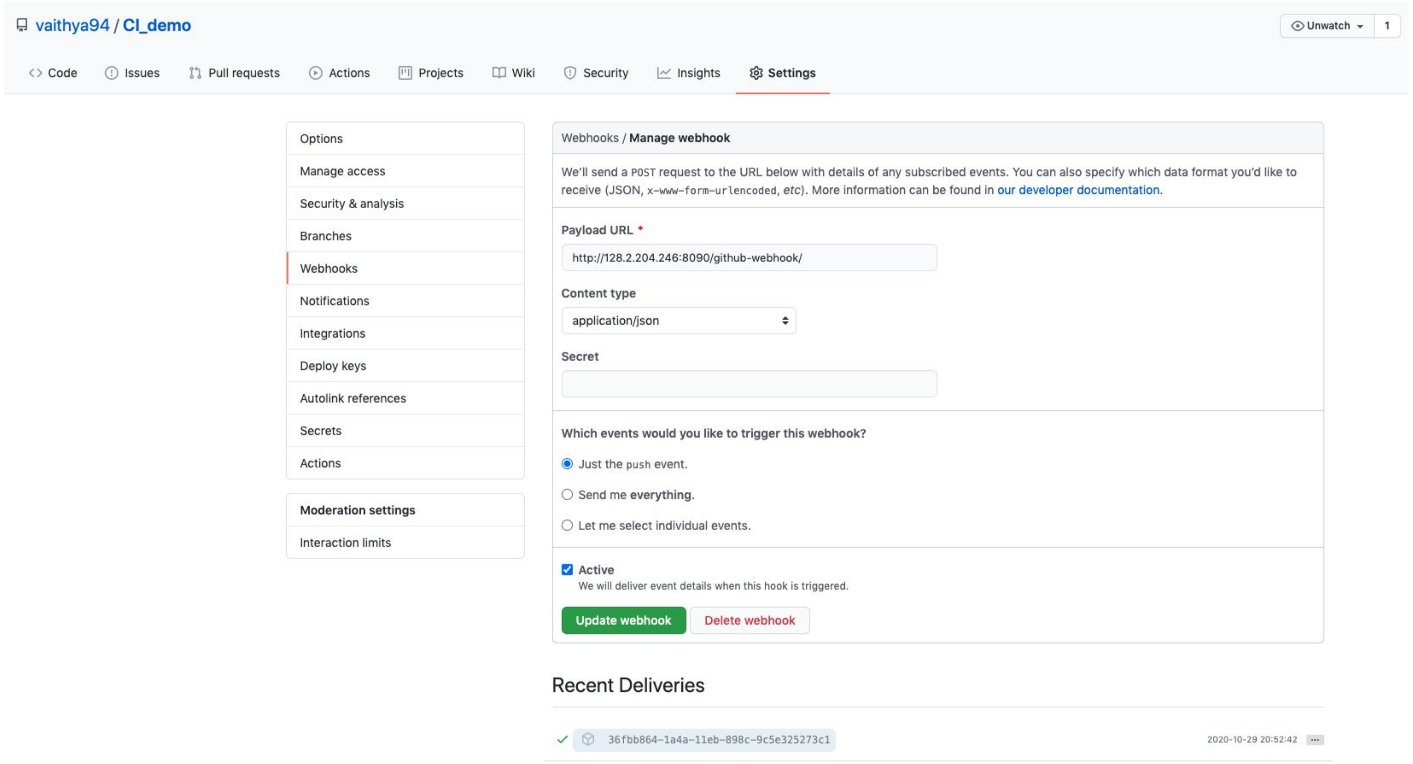
	Checkout Code	Install Dependencies	Evaluate Code Quality	Build Model
Average stage times: (Average full run time: ~6s)	682ms	5s	1s	1s
#22 Oct 29 20:52 — 1 commit	749ms	2s	1s	1s
#21 Oct 29 18:34 — 1 commit	657ms	2s	1s	1s
#20 Oct 29 16:24 — No Changes	592ms	2s	1s	1s
#19 Oct 29 16:24 — 1 commit	678ms	2s	1s	1s
#18 Oct 29 16:21 — 3 commits	737ms	2s	1s	1s

#22	Oct 29, 2020, 8:52 PM
#21	Oct 29, 2020, 6:34 PM
#20	Oct 29, 2020, 4:24 PM

Code Coverage

Packages 100% **Files** 25% **Classes** 25% **Lines** 4% **Conditionals** 100%

Jenkins - GitHub Integration



vaithya94 / CI_demo

Unwatch 1

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

- Options
- Manage access
- Security & analysis
- Branches
- Webhooks
- Notifications
- Integrations
- Deploy keys
- Autolink references
- Secrets
- Actions

Moderation settings

- Interaction limits

Webhooks / **Manage webhook**

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *

http://128.2.204.246:8090/github-webhook/

Content type

application/json

Secret

Which events would you like to trigger this webhook?

- [✓ Just the `push` event.]
- Send me **everything**.
- Let me select individual events.

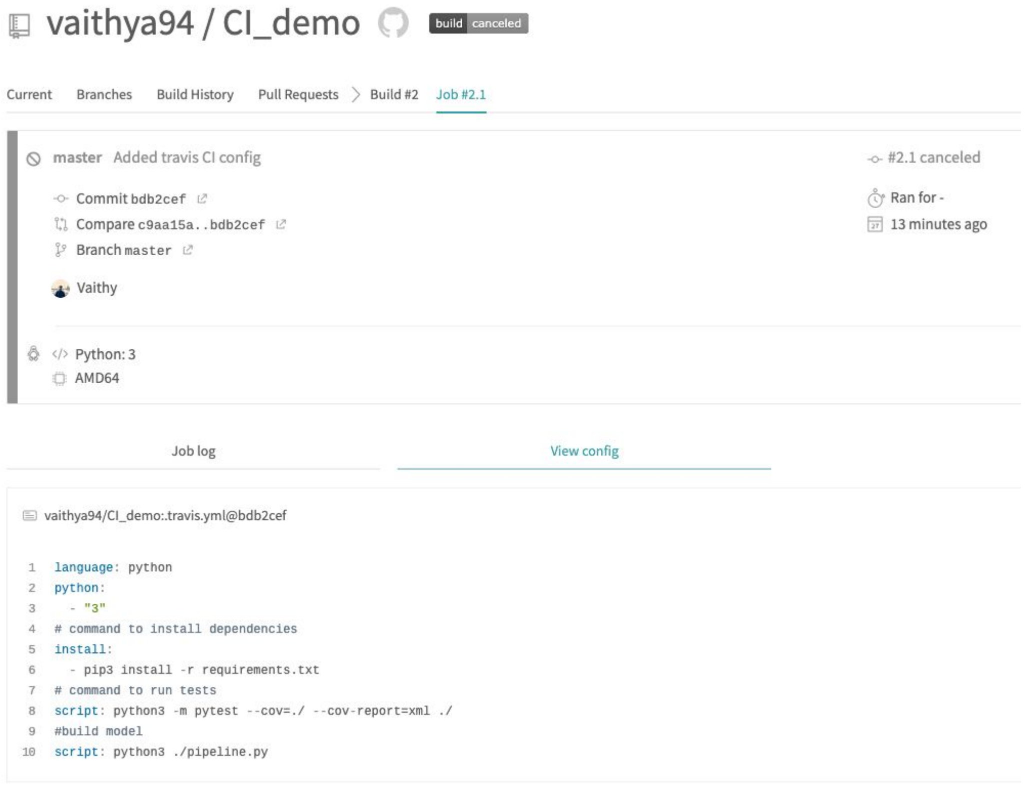
[✓ **Active**]
We will deliver event details when this hook is triggered.

Update webhook | Delete webhook

Recent Deliveries

✓ 36fbb864-1a4a-11eb-898c-9c5e325273c1 2020-10-29 20:52:42

TravisCI



vaithya94 / CI_demo

master Added travis CI config

Commit bdb2cef

Compare c9aa15a..bdb2cef

Branch master

Vaithy

#2.1 canceled

Ran for -

13 minutes ago

Python: 3

AMD64

vaithya94/CI_demo:.travis.yml@bdb2cef

```yaml
language: python
python:
  - "3"
# command to install dependencies
install:
  - pip3 install -r requirements.txt
# command to run tests
script: python3 -m pytest --cov=./ --cov-report=xml ./
#build model
script: python3 ./pipeline.py
```

CI Pipeline Qualities

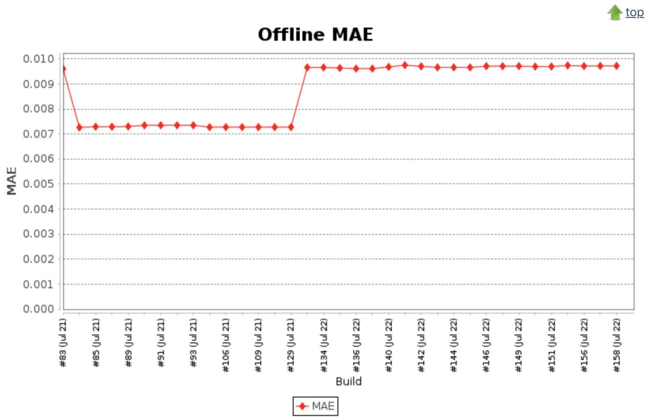
Repeatable	[consistent results across runs; consecutive runs are independent]
Fault-tolerant	[fail gracefully if any stage fails, ie. system remains operational]
Correct	[performs what is expected of it given some inputs]
Robust	[should be able to handle noise in any inputs the pipeline expects]
Testable	[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)

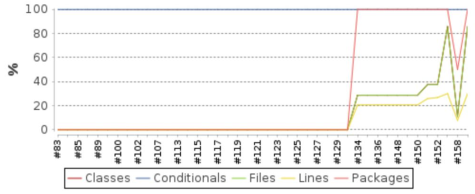
Automated Model Evaluation & Testing



Code Coverage

Cobertura Coverage Report

Trend



Project Coverage summary

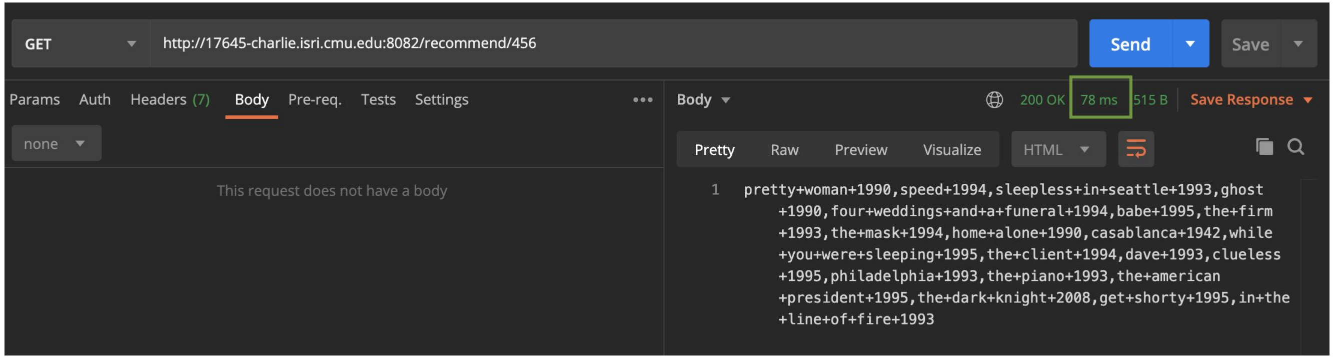
Name	Packages		Files		Classes		Lines	
Cobertura Coverage Report	100%	1/1	86%	6/7	86%	6/7	30%	64/214

Coverage Breakdown by Package

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

Manual Testing

Blackbox Integration Testing - Postman

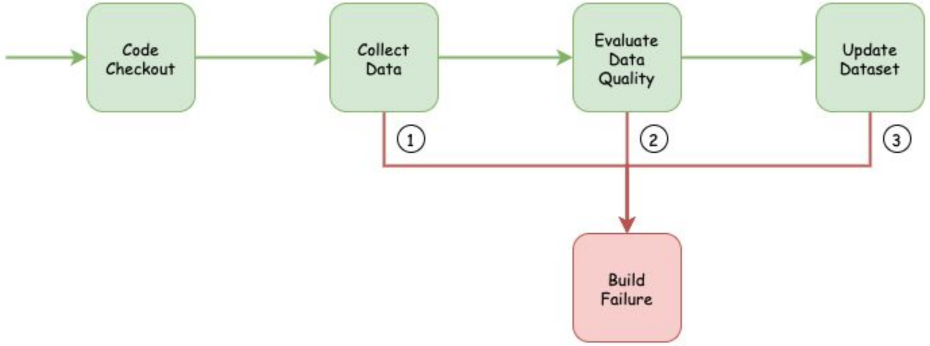


http://17645-charlie.isri.cmu.edu:8082/recommend/456

This request does not have a body

```
pretty+woman+1990,speed+1994,sleepless+in+seattle+1993,ghost
    +1990,four+weddings+and+a+funeral+1994,babe+1995,the+firm
    +1993,the+mask+1994,home+alone+1990,casablanca+1942,while
    +you+were+sleeping+1995,the+client+1994,dave+1993,clueless
    +1995,philadelphia+1993,the+piano+1993,the+american
    +president+1995,the+dark+knight+2008,get+shorty+1995,in+the
    +line+of+fire+1993
```

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!