

Welcome to the **Co**Grammar Tutorial: Git and Docker

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated
moderators answering questions.

Software Engineering Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.

(Fundamental British Values: Mutual Respect and Tolerance)

- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: [**Questions**](#)

Software Engineering Session Housekeeping cont.

- "Please check your spam folders for any important communication from us. If you have accidentally unsubscribed, please reach out to your support team."
- Rationale here: Career Services, Support, etc will send emails that contain NB information as we gear up towards the end of the programme. Students may miss job interview opportunities, etc.

Software Engineering Session Housekeeping cont.

- For all **non-academic questions**, please submit a query:
www.hyperiondev.com/support
- Report a **safeguarding** incident:
www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: [Feedback on Lectures](#)

Skills Bootcamp

8-Week Progression Overview

Fulfil 4 Criteria to Graduation

Criterion 1: Initial Requirements

- **Timeframe:** First 2 Weeks
- **Guided Learning Hours (GLH):** Minimum of 15 hours
- **Task Completion:** First four tasks

Criterion 2: Mid-Course Progress

- **Guided Learning Hours (GLH):** 60
- **Task Completion:** 13 tasks

Skills Bootcamp Progression Overview

✓ Criterion 3: Course Progress

- **Completion:** All mandatory tasks, including Build Your Brand and resubmissions by study period end
- **Interview Invitation:** Within 4 weeks post-course
- **Guided Learning Hours:** Minimum of 112 hours by support end date (10.5 hours average, each week)

CoGrammar

✓ Criterion 4: Demonstrating Employability

- **Final Job or Apprenticeship Outcome:** Document within 12 weeks post-graduation
- **Relevance:** Progression to employment or related opportunity





CoGrammar Tutorial: Git and Docker

June 2024



Learning Objectives

- Explain **Version Control Workflow** with **Git**
- Describe the Git basic concepts like **branches**, **repositories**
- Describe basic **commands** and **operations** in **Git**
- Explain the **purpose** and **benefits** of **version control** systems
- Explain **containerisation** with **Docker**

Learning Objectives

- Describe the basic **commands** and **operations** with **Docker**
- “**Dockerise**” a python application
- Run a docker a **container**
- Update the Git **repository** with the changes

Poll

1. Which command is used to build a Docker image from a Dockerfile?
 - a. docker create
 - b. docker start
 - c. docker build

Poll

2. How do you stage a file for commit in Git?
 - a. git commit <filename>
 - b. git add <filename>
 - c. git push <filename>

Introduction

CoGrammar



Intuition

Imagine you and a teammate are developing a **Python application**. **Git** keeps track of all the **code changes** you both make, like **adding** functionalities or **squashing bugs**. This allows you to see each other's contributions and **revert** back if something goes wrong.

Docker comes into play when you need to **deploy** the application. Docker creates a **self-contained package** with your Python code and all its **dependencies**, guaranteeing that your teammate can run the app on their machine exactly as it runs on yours, even if they have **different** software **versions installed**.

Version Control

CoGrammar



What is Version Control?

- Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.
- The Code base is stored in a central place.
- Format used: deltas.
- This means that only changes between versions are saved.
- You can therefore “roll back” your code to a previous version.

Some Terminology

- **Version:** Code at a particular state.
- **Repository:** The collection of all files at all versions.
- **History:** The list of all changes made to a set of files.
- **Commit:** A wrapper for a set of changes.
- **Staging Area:** A file containing changes to be added to the next commit.

Git

CoGrammar



What is Git?

- Git is a distributed version control system for tracking changes in source code during software development.



Git Clients



Bitbucket



AWS CodeCommit



Azure DevOps



GitLab

Why Git?

- Most widely used version control system.
- Free and open-source. Designed to handle a large variety of systems.
- Distributed architecture:
 - When you download a repository, you download the full history of changes to your local computer.
- Everything is run from the command-line using the git application.

Git Concepts

- **Repository:** is a central storage location for managing and tracking changes in files and directories
- **Commit:** The core conceptual unit of work in Git is the commit. These are snapshots of the files being tracked within your project folder.
- **Branch:** Branching lets you work with multiple chains of commits inside a project.

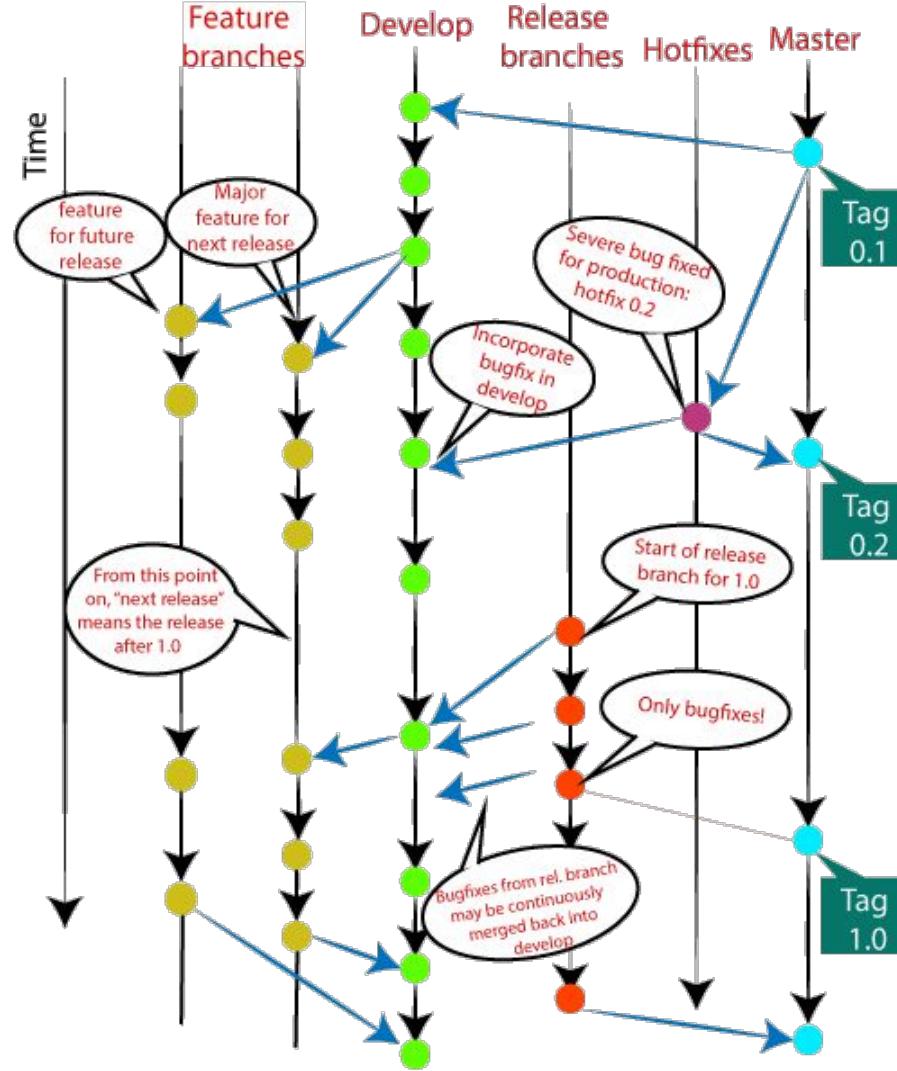
Git Operations

- **Clone:** Copies or downloads the repository from GitHub.com to your local machine
- **Pull:** Used to fetch and download content from a remote repository and immediately update the local repository to match that content.
- **Push:** used to upload local repository content to a remote repository.
- **Status:** shows the current state of your Git working directory and staging area.

Git Flow

CoGrammar

Source

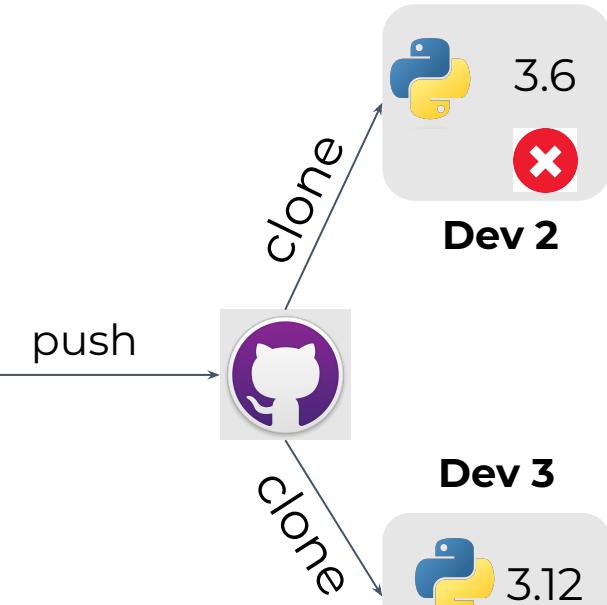
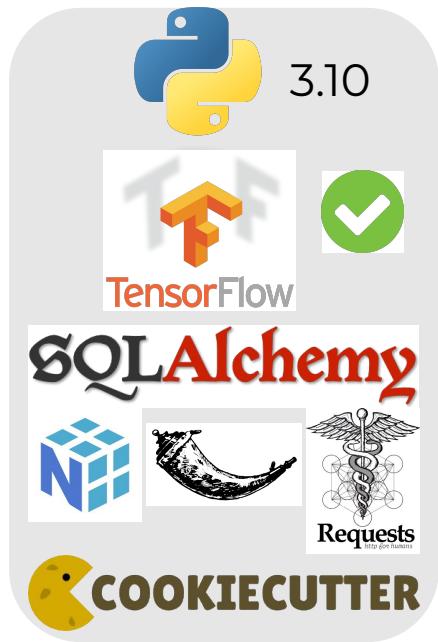


Containerisation with Docker



The "Works on My Machine" Problem

Dev environment



Environment Mismatches:

- Python 3.7 vs. Python 3.9
- numpy 1.18 vs. 1.21
- Windows vs. Linux

Dependency Hell:

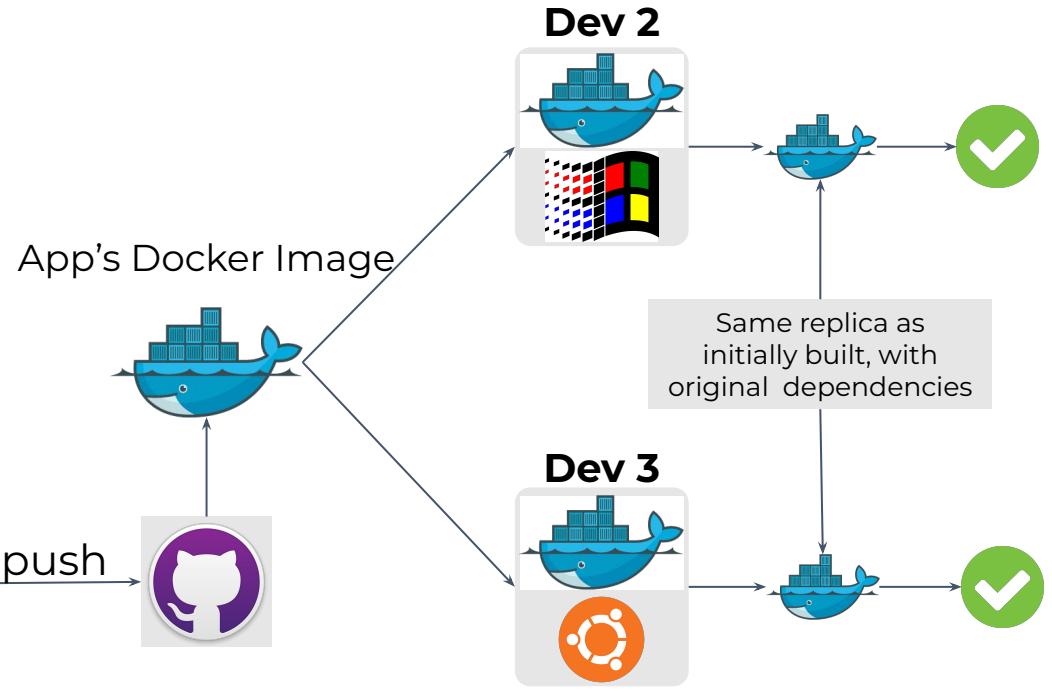
- packageA needs libraryX version 1.2, but packageB needs libraryX version 2.0
- Missing dependencies that work fine on your machine because you installed them manually months ago.

Configuration Issues:

- Different environment variables, paths, or settings files.

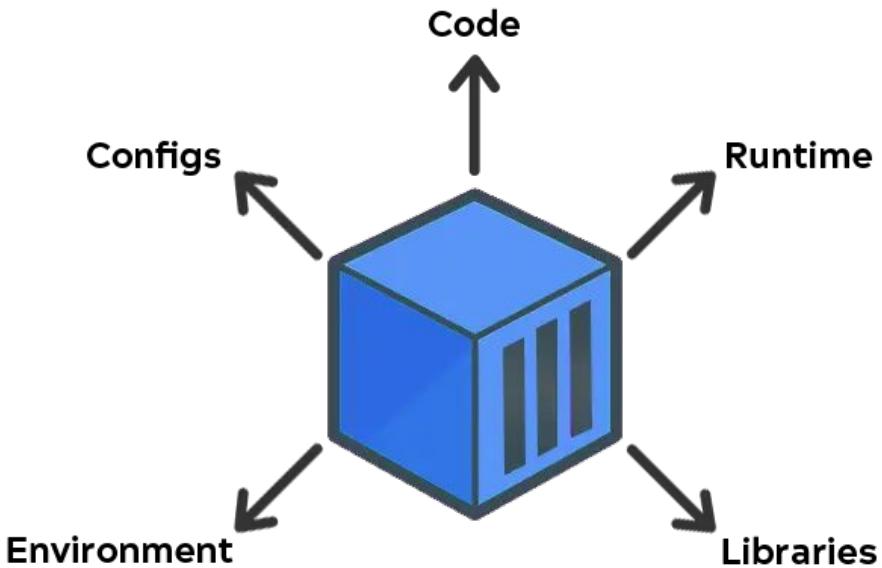
Docker

Dev environment

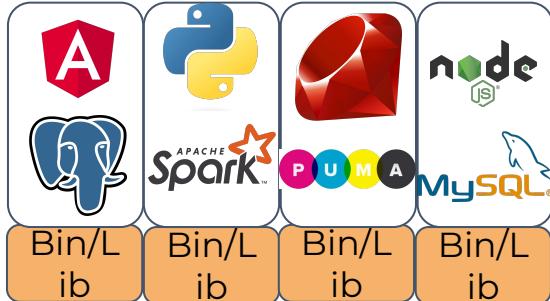


Introduction to Modern Application Deployment

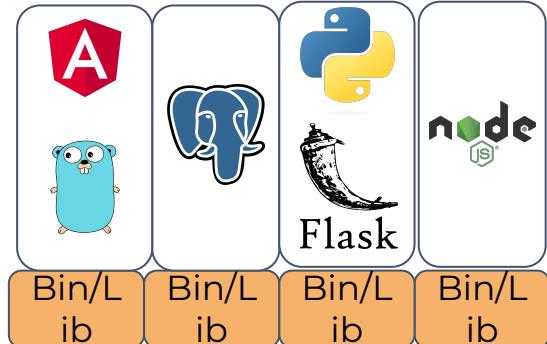
What's included in a Container?



Introduction to Modern Application Deployment

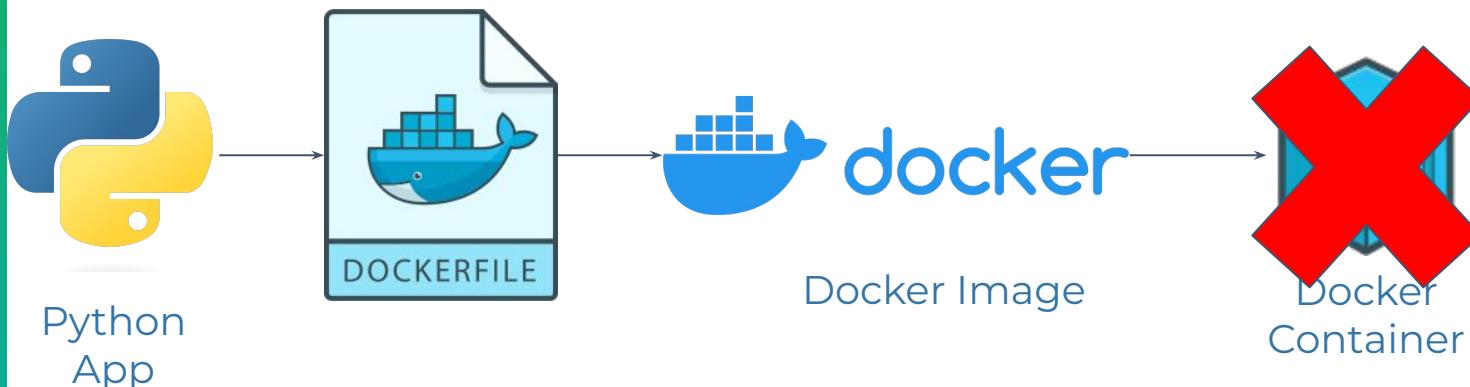


Virtual Machine



Docker Container

Docker Container Lifecycle



Docker Core Concepts

- **Images:** Blueprints for containers, containing application code, libraries, and dependencies.
- **Containers:** Instances of images, isolated environments running applications.
- **Dockerfile:** A text file with instructions to build images (**FROM, RUN, EXPOSE, CMD**). It specifies:
 - **Base image:** The starting point for the image.
 - **Instructions:** Commands to install dependencies, copy files, and configure the environment.

Docker Core Concepts

```
1 FROM python:3.9-slim
2
3 WORKDIR /app
4
5 COPY . /app
6
7 RUN if [ -f requirements.txt ]; then pip install --no-cache-dir -r requirements.txt; fi
8
9 EXPOSE 80
10
11 CMD ["python", "app.py"]
```



Let's take a break



Let's get coding

CoGrammar

Poll

1. How can you view running Docker containers and their details?
 - a. docker list
 - b. docker status
 - c. docker ps

Poll

2. What is the purpose of the git clone command?
 - a. To create a new Git repository.
 - b. To download a copy of an existing Git repository from a remote server to your local machine.
 - c. To delete a branch from a Git repository.

Summary

- **Version Control with Git:** Git allows teams to track changes in their codebase over time through commits, branches, and merges
- **Basic Git Commands:** add, commit, status, push
- **Containerization with Docker:** Docker enables application containerization, bundling an application with its dependencies into a standardised unit for development, shipment, and deployment.
- **Key Docker Concepts:** Image, Container, Dockerfile
- **Dockerise a python application:** Docker allows the creation of a self-contained environment for deployment.

Thank you for attending



Department
for Education

