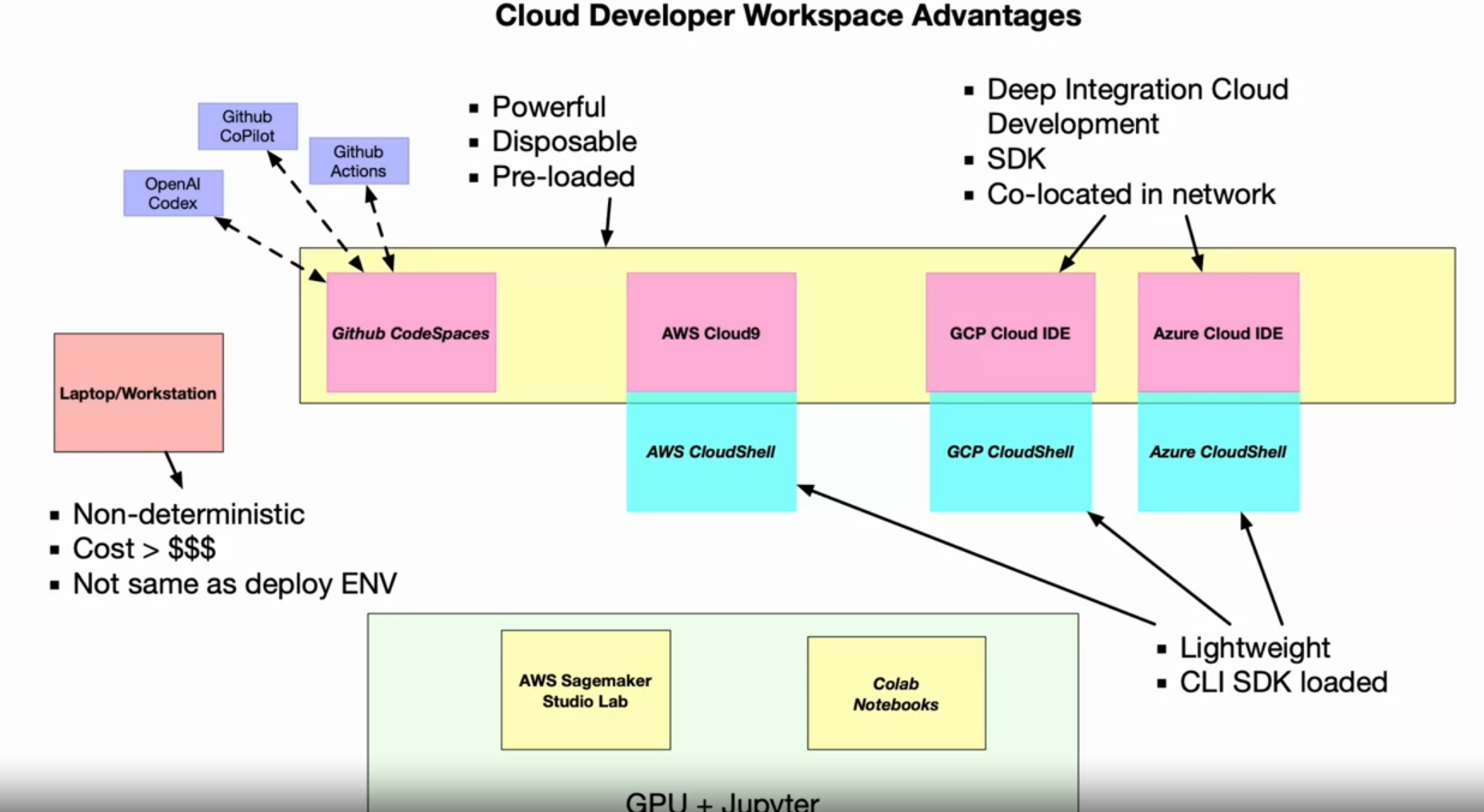
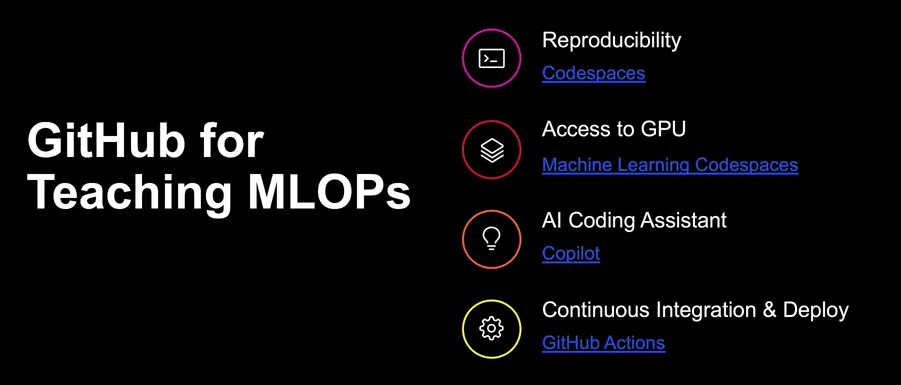
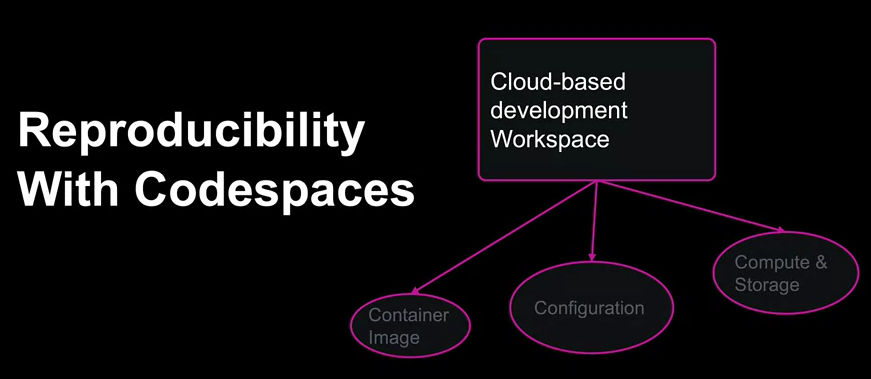
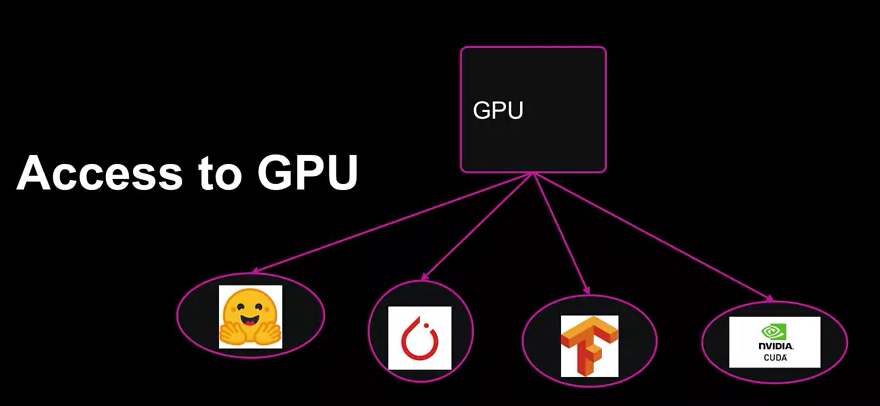
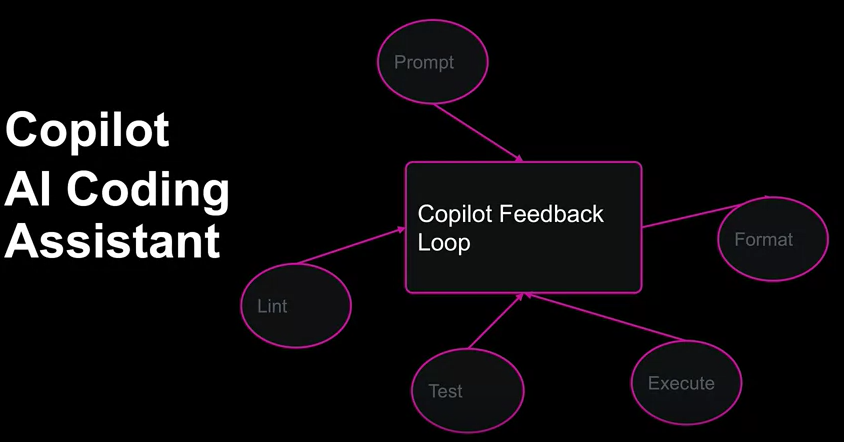
* **Codespaces** - Cloud-based, disposable development environments hosted by GitHub. Provide reproducibility and customization.
* **Reproducibility** - The ability to reliably recreate an environment and obtain the same results. Codespaces ensure this through containers.
* **Container Image** - A lightweight, stand-alone, executable software package allowing code to run quickly and reliably across environments.
* **GPU** - Graphics processing units that accelerate machine learning model building/training with specialized hardware.
* **Copilot** - GitHub's AI pair programmer that suggests code and entire functions to developers as they type.
* **Continuous Integration** - The development practice of frequently merging code changes and validating each change through automated build and test processes.
* **Continuous Delivery** - Software methodology where teams can release new changes to users quickly, reliably and sustainably through automated deployments.

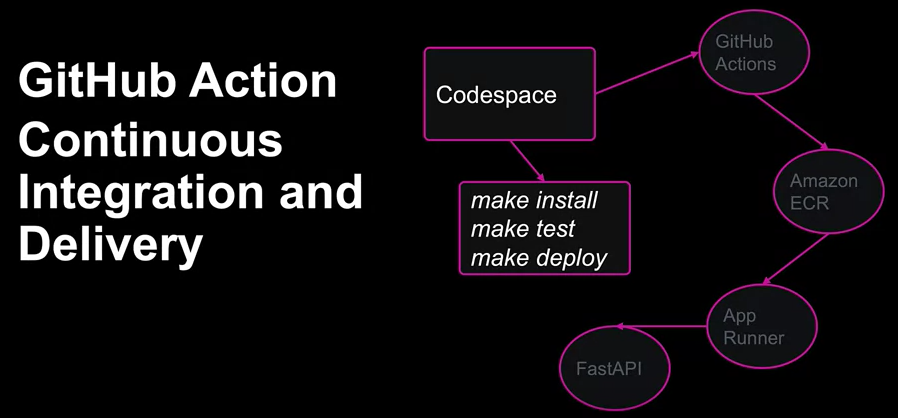




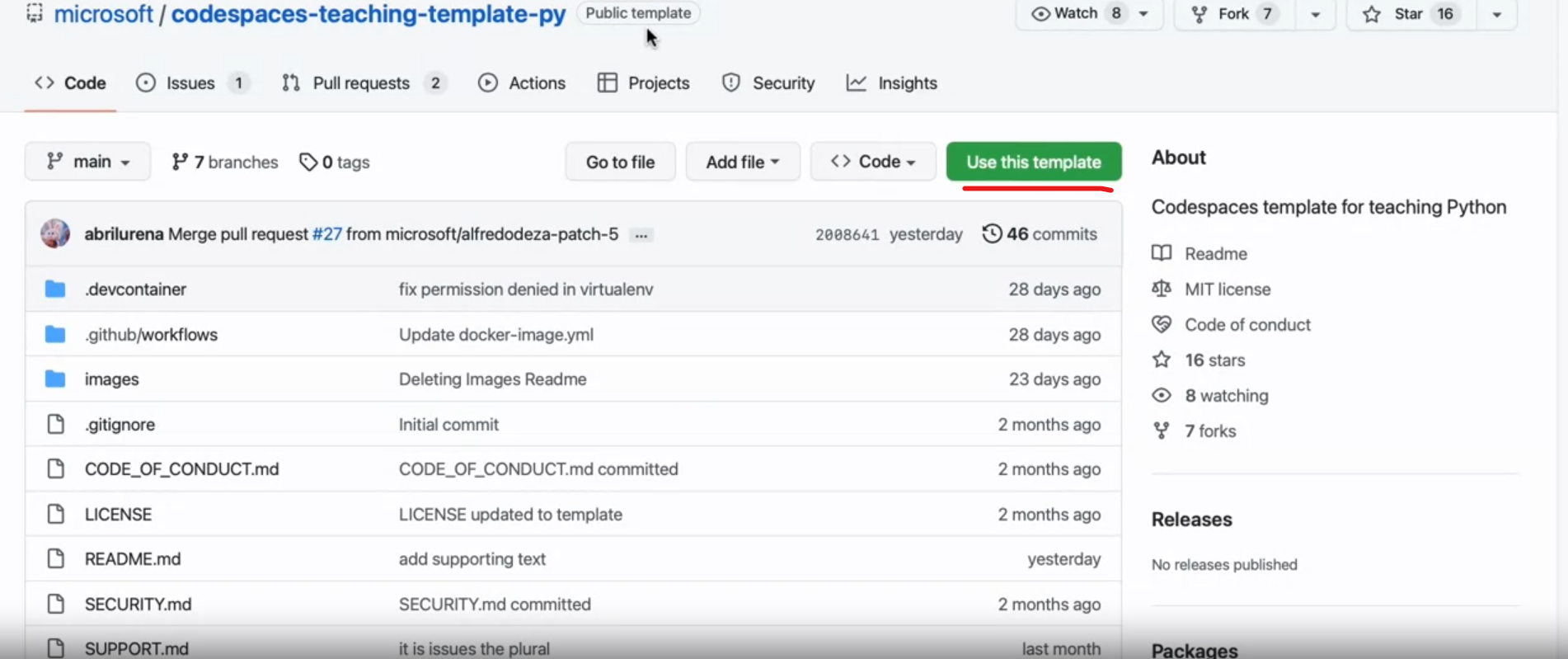








**GitHub Template:**



<https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-template-repository>

| **Title** | **Brief Description** |
| --- | --- |
| [Scaling MLOps Education](https://github.com/readme/guides/mlops-education) | In this Machine Learning Operations (MLOps) video Guide, you'll learn how to utilize GitHub for maximizing developer efficiency and product innovation. The guide covers four major benefits: reproducibility (via GitHub Codespaces), access to machine learning technology, AI coding assistance, and CI/CD capabilities. You'll be walked through building GitHub **Templates**, creating a CI/CD workflow with GitHub **Actions**, and configuring GitHub **Codespaces** environments with .devcontainer. You'll also learn how automation and AI pair programming, such as with GitHub **Copilot**, can help streamline these processes. The guide teaches you how to set up and use GitHub Actions, GitHub Codespaces with GPU-capable MLOps templates, and GitHub Copilot for AI pair programming. |
| [GitHub Codespaces Overview](https://docs.github.com/en/codespaces/overview) | Introduces the concept of GitHub Codespaces to students and shows them the core benefits of developing with them. |
| [Introduction to GitHub Dev Containers](https://docs.github.com/en/codespaces/setting-up-your-project-for-codespaces/adding-a-dev-container-configuration/introduction-to-dev-containers) | Introduces the concept of configurable GitHub Dev Containers and shows them the core benefits of configuring workspaces with them. |
| [GitHub MLOps Template](https://github.com/nogibjj/mlops-template) | This repository provides a template for **MLOps projects** with GPU support. It uses the recommended tools from the Python Software Foundation, such as virtualenv, pip, and Docker, for a streamlined setup. As of December 2022, GitHub Codespaces are free for education, including GPU Codespaces. Upon launching, verify that the virtualenv is sourced.  The template includes a **Makefile, Pytest, pandas, Pylint or ruff, Dockerfile, GitHub Copilot, Jupyter and IPython,** common **Python libraries** for ML/DL**, Hugging Face,** and **GitHub Actions.** Additionally, it features two interesting tools: Zero-shot classification (./hugging-face/zero\_shot\_classification.py classify) and Yake for candidate label creation (./utils/kw\_extract.py). This template is designed to help students quickly set up MLOps projects while exploring essential tools and libraries. |

**Summary of Lesson** This lesson covered leveraging GitHub's cloud-based Codespaces for reproducibility via containers, GPU acceleration, AI-assisted development with Copilot, and CI/CD pipelines

**Top 3 Key Points**

* Codespaces provides customizable, disposable dev environments
* Copilot suggests code interactively to boost productivity
* Reproducibility and CI/CD improve collaboration

# Key Terms

* **AWS Step Functions** - A serverless orchestration service that lets you coordinate components of distributed applications and microservices using visual workflows.
* **AWS Batch** - A managed computing service that schedules and runs batch computing workloads of any scale on AWS. Automatically provisions resources.
* **AWS Glue** - A serverless ETL and data integration service that prepares and transforms data for analytics and machine learning.
* **ETL Pipeline** - Extract, transform, load pipelines that pull data from sources, process/clean it, and load it into a destination database or data warehouse.
* **Serverless Pipeline** - Using cloud services like Lambda, Glue, Batch and Step Functions to build data pipelines that automatically manage infrastructure.
* **DataOps** - Collaborative data management practices focused on improving the quality and speed of data delivery by automating infrastructure.

Tools :-

1. AWS Step Function
2. Azure Databricks