SATELLITE IMAGE ANALYSIS WITH DVC

PRESENTED BY: NAYANA NAGARAJAPPA
CLASS: MATH 608 DATA SCIENCE FOR GRAD STUDIES

AGENDA

PROJECT OVERVIEW

WHAT IS DVC

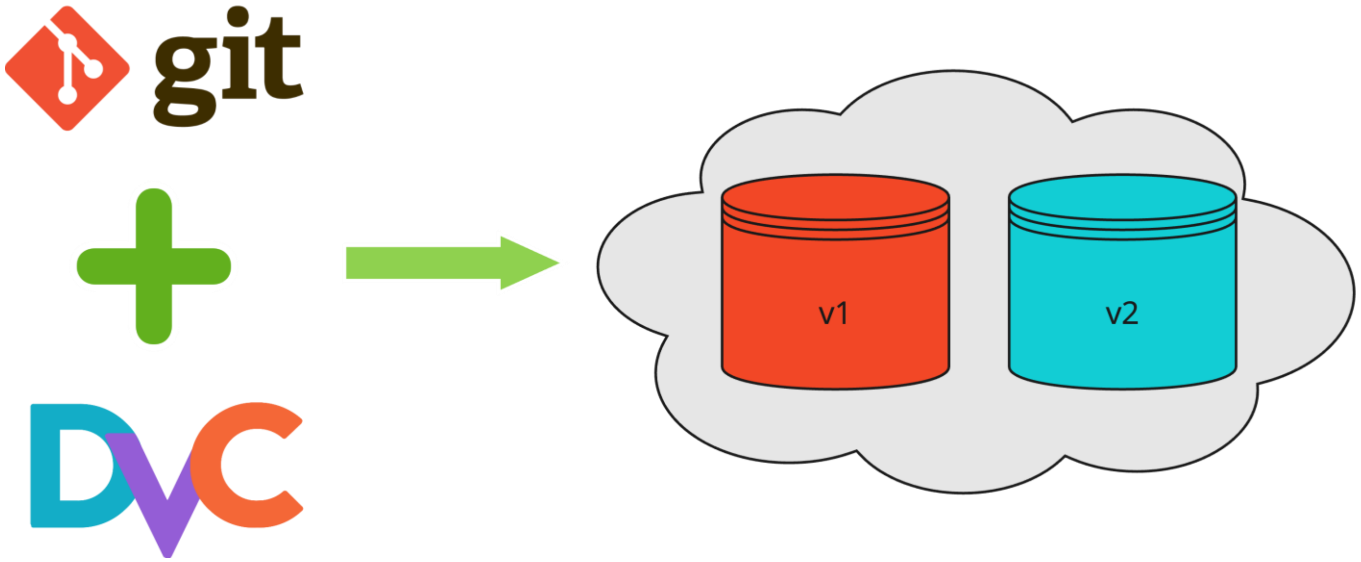
PROJECT WORKFLOW

KEY BENEFITS

Q & A

DATA VERSION CONTROL?

"DVC IS AN OPEN-SOURCE TOOL FOR VERSIONING, MANAGING, AND COLLABORATING ON LARGE DATASETS AND MACHINE LEARNING MODELS."



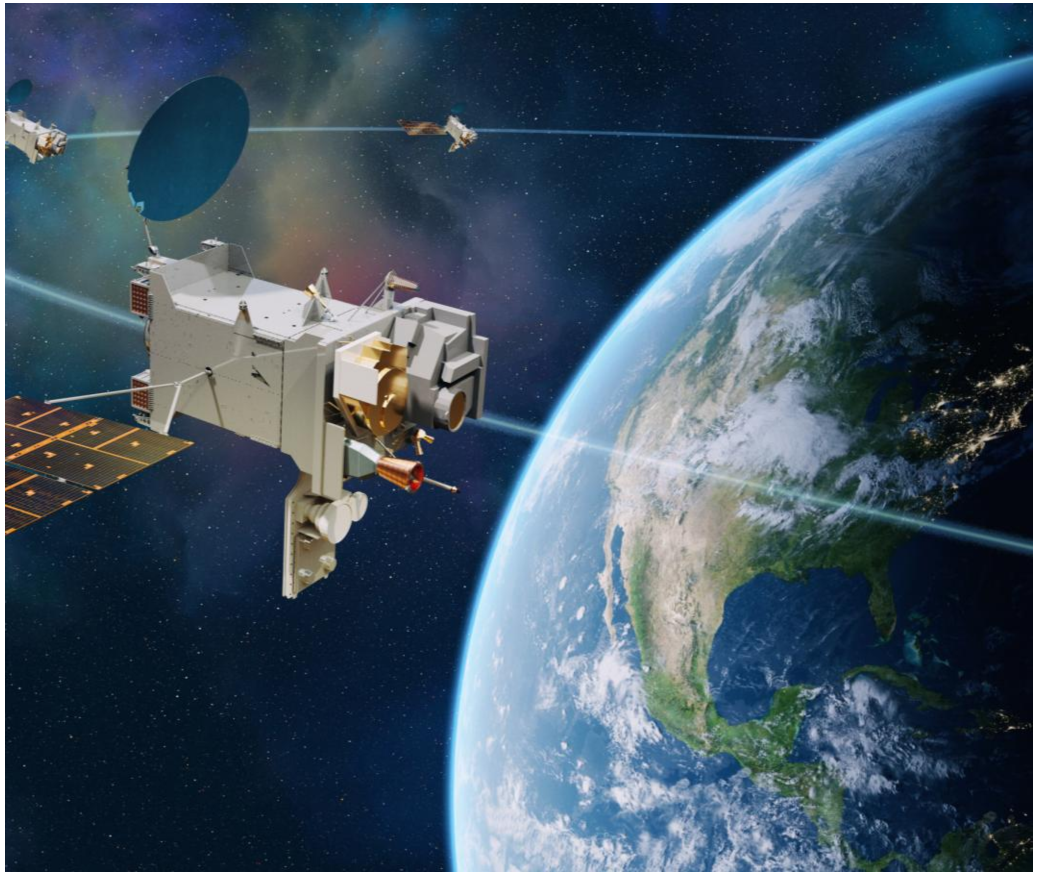
PROJECT WORKFLOW

Step 1: Pulling Satellite Images

Objective: Automate the process of downloading satellite images.

How:

- Developed a Python script that pulls satellite images from free sources like NASA
- Script filters images based on location, date range, or cloud cover.
- The images are retrieved using a **REST API** at scheduled intervals, managed through a **Cron expression** for automation.



PROJECT WORKFLOW

Step 2: Managing Datasets with DVC

Initialization:

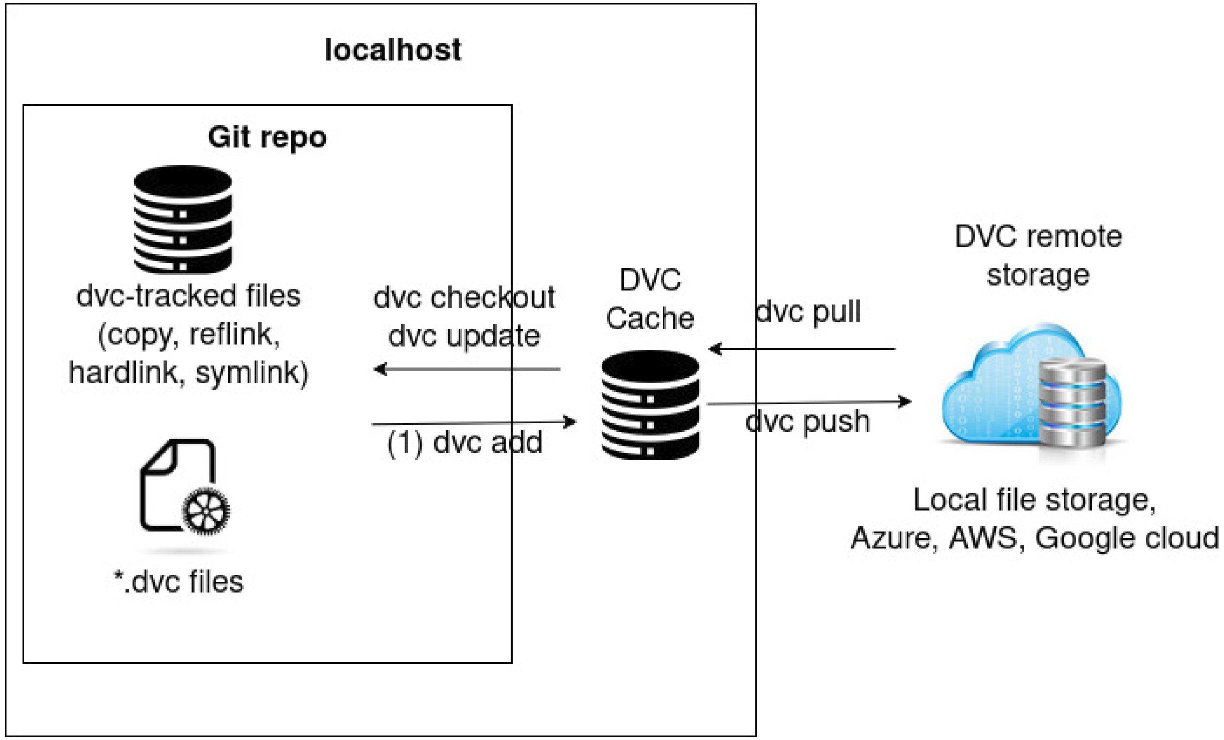
- DVC is initialized in the project directory using dvc init.
- This sets up the DVC environment and creates the necessary .dvc files and configurations.

Adding and Tracking Images:

- The satellite images (e.g., stored in a data/images/ folder) are tracked using the dvc add command. This creates a .dvc file (e.g., data/images.dvc), which acts as metadata for the tracked files.
- The large files are not committed to Git. Instead, DVC stores their versioned metadata, while the actual files are pushed to remote storage (e.g., GCP).

Reproducibility:

- By committing the .dvc files to Git, the dataset versions are locked in the repository.
- Anyone can reproduce the same version of the dataset using: dvc pull
- This ensures that all team members or environments can access identical datasets.



PROJECT WORKFLOW

Step 3: Storing Data in GCP Cloud Storage

Objective: Efficiently store and retrieve large datasets

How:

- Configured GCP Cloud Storage as the DVC remote.
- Linked DVC to GCP using authentication credentials.

Key Steps:

- dvc remote add -d gcpremote gs://bucket-name
- DVC handles data synchronization between the local project and GCP Cloud Storage.

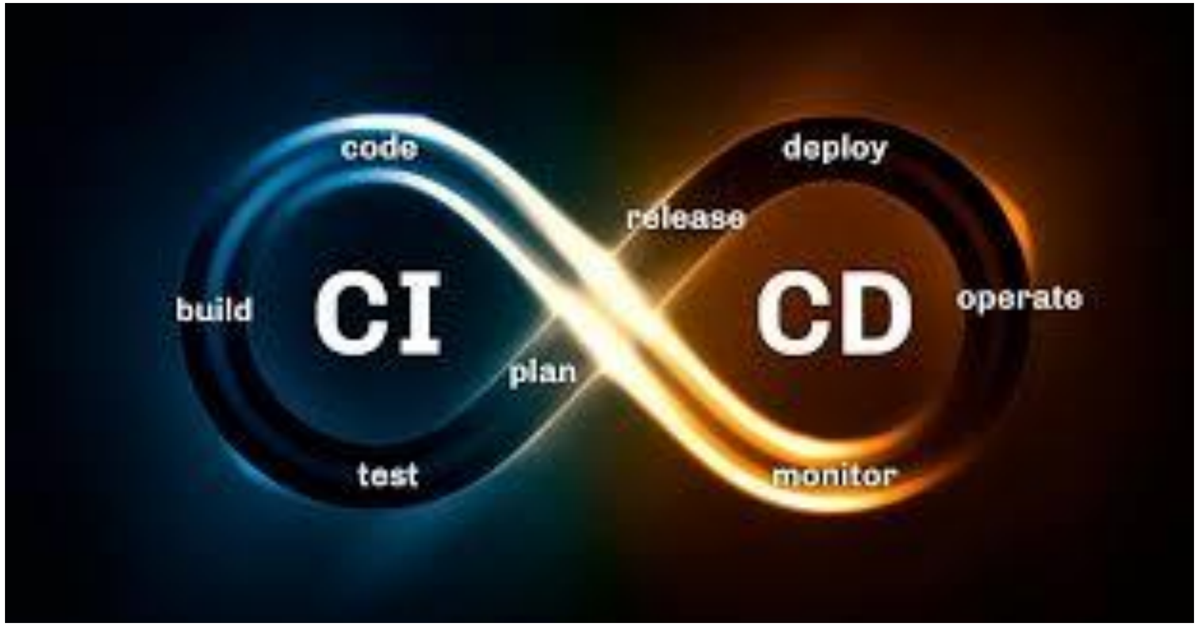
PROJECT WORKFLOW

Step 4: Automation with GitHub Actions

Objective: Automate the end-to-end workflow (image pull → version → push).

How:

- Set up GitHub Actions to trigger:
- Python script execution to pull new images.
- DVC commands (add, commit, push) to manage and store updated data.



KEY BENEFITS

- **Improved version control for large datasets.**
- **Streamlined data storage and retrieval.**
- **End-to-end automation reduces manual effort.**

Q&A

THANK YOU