DVC Setup and Initialization

INTRODUCTION TO DATA VERSIONING WITH DVC

Ravi Bhadauria
Machine Learning Engineer

Installation

- DVC is a Python package
  - Universally install with `pip`

```
$ pip install dvc
```

- Remember to install in virtual environments
- Ensure Git is installed

Verify Installation

```
$ dvc version
```

```
DVC version: 3.40.1 (pip)
Platform: Python 3.9.16 on macOS-14.2.1-arm64-arm-64bit
Config:
    Global: /Users/<username>/Library/Application Support/dvc
    System: /Library/Application Support/dvc
Repo: dvc, git
```

[1] https://dvc.org/doc/command-reference/version

Initializing DVC

- Ensure Git is initialized

```
$ git init
```

```
Initialized empty Git repository in /path/to/repo/.git/
```

- Initialize DVC in the repository

```
$ dvc init
```

```
Initialized DVC repository.

You can now commit the changes to git.
```

[1] https://dvc.org/doc/command-reference/init

DVC Hidden Files

- Initialization creates internal files that should be tracked with Git

```
$ git status
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .dvc/.gitignore
    new file:   .dvc/config
    new file:   .dvcignore
```

- Commit the changes

```
$ git commit -m "initialized dvc"
```

.dvcignore File

- Similar to `.gitignore` file
    - Follows the same pattern
    - Outline files/directories that DVC will ignore
- Useful when tracking many data files not needed
    - Improves execution time of DVC operations

[1] https://dvc.org/doc/user-guide/project-structure/dvcignore-files [2] https://git-scm.com/docs/gitignore

Example

```
# .dvcignore
# Ignore all files in the 'data' directory
data/*

# But don't ignore 'data/data.csv'
!data/data.csv

# Ignore all .tmp files
*.tmp
```

[1] https://dvc.org/doc/user-guide/project-structure/dvcignore-files

Checking Ignored Files

- Use `dvc check-ignore` command

```
$ dvc check-ignore data/file.txt
```

```
data/file.txt
```

- Use with `-d` flag to get details

```
$ dvc check-ignore -d data/file.txt
```

```
.dvcignore:3:data/*    data/file.txt
```

[1] https://dvc.org/doc/command-reference/check-ignore

Summary

- Install DVC using `pip install dvc`
- Verify version, platform etc. of DVC
  - `dvc version`
- Initializing DVC in workspace
  - `dvc init`
  - Initialize Git first
- `.dvcignore` files are used to specify excluded files
  - Similar to `.gitignore` , follows same syntax
  - Check if a specific file is excluded
    - `dvc check-ignore <filename>`

Let's practice!

INTRODUCTION TO DATA VERSIONING WITH DVC

DVC Cache and Staging Files

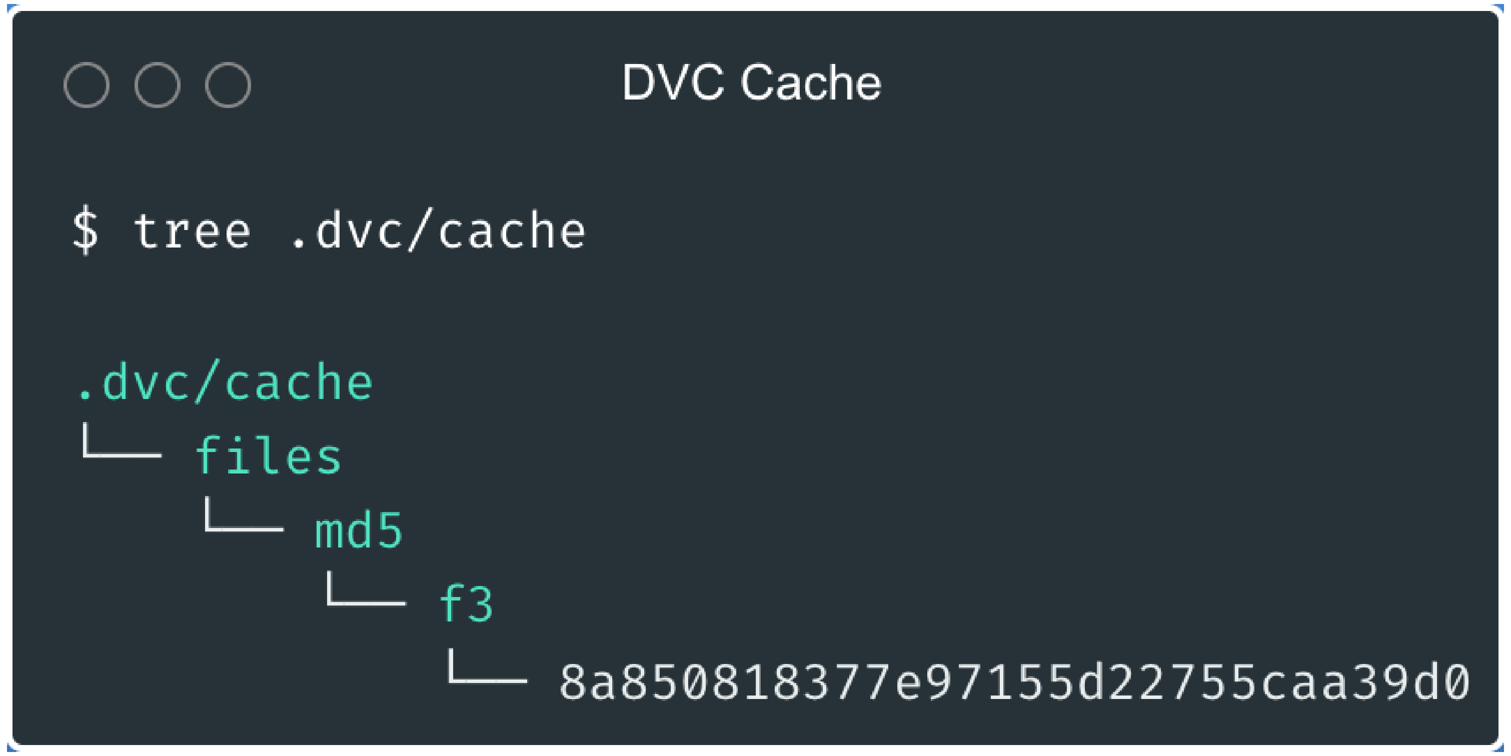
INTRODUCTION TO DATA VERSIONING WITH DVC

Ravi Bhadauria
Machine Learning Engineer

DVC Cache

- Hidden storage for tracked data files and versions
- Stages temporary files until committed
  - Prefer adding large datasets and binary files
- Lives inside the `.dvc` directory in the workspace
  - Configure location

```
$ dvc cache dir ~/mycache
```



DVC Cache

```
$ tree .dvc/cache

.dvc/cache
└── files
    └── md5
        └── f3
            └── 8a850818377e97155d22755caa39d0
```

Adding Files to Cache

- Add data files to dvc

```
$ dvc add data.csv
```

```
100% Adding...|=====================|1/1 [00:00, 53.55file/s]


To track the changes with git, run:
    git add data.csv.dvc
To enable auto staging, run:
    dvc config core.autostage true
```

.dvc files

- Each DVC tracked file has its corresponding `.dvc` file
  - `data.csv -> data.csv.dvc`
- To version the data file, use `git commit -m "data.csv.dvc"`
- Content of `.dvc` files

```
outs:
- md5: f38a850818377e97155d22755caa39d0
  size: 16
  hash: md5
  path: data.csv
```

Interaction with DVC Cache

- The path of cache file uses the MD5 value

```
$ find .dvc/cache -type f
```

```
.dvc/cache/f3/8a850818377e97155d22755caa39d0
```

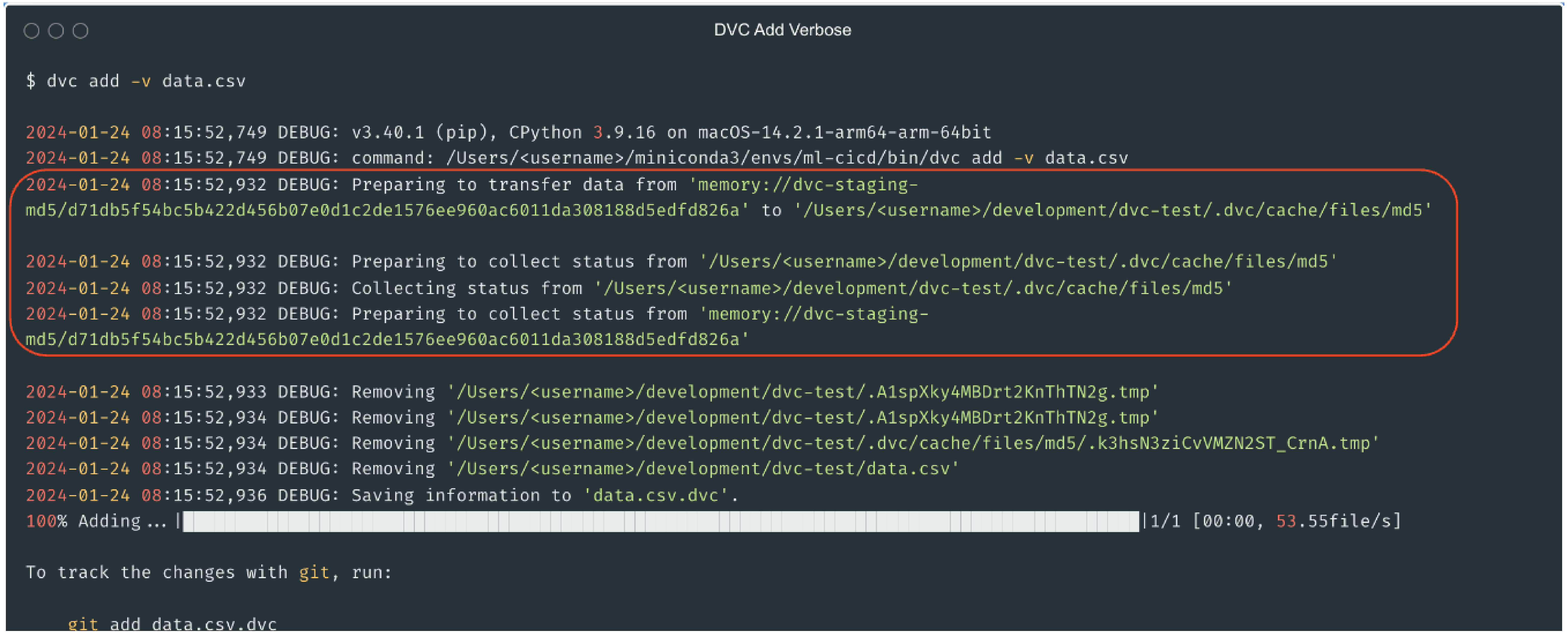
- Compute MD5 of dataset

```
$ md5 data.csv
```

```
MD5 (data.csv) = f38a850818377e97155d22755caa39d0
```

Interaction with DVC Cache

- Use `dvc add -v` for verbose output



DVC Add Verbose

```
$ dvc add -v data.csv

2024-01-24 08:15:52,749 DEBUG: v3.40.1 (pip), CPython 3.9.16 on macOS-14.2.1-arm64-arm-64bit
2024-01-24 08:15:52,749 DEBUG: command: /Users/<username>/miniconda3/envs/ml-cicd/bin/dvc add -v data.csv
2024-01-24 08:15:52,932 DEBUG: Preparing to transfer data from 'memory://dvc-staging-
md5/d71db5f54bc5b422d456b07e0d1c2de1576ee960ac6011da308188d5edfd826a' to '/Users/<username>/development/dvc-test/.dvc/cache/files/md5'

2024-01-24 08:15:52,932 DEBUG: Preparing to collect status from '/Users/<username>/development/dvc-test/.dvc/cache/files/md5'
2024-01-24 08:15:52,932 DEBUG: Collecting status from '/Users/<username>/development/dvc-test/.dvc/cache/files/md5'
2024-01-24 08:15:52,932 DEBUG: Preparing to collect status from 'memory://dvc-staging-
md5/d71db5f54bc5b422d456b07e0d1c2de1576ee960ac6011da308188d5edfd826a'

2024-01-24 08:15:52,933 DEBUG: Removing '/Users/<username>/development/dvc-test/.A1spXky4MBDrt2KnThTN2g.tmp'
2024-01-24 08:15:52,934 DEBUG: Removing '/Users/<username>/development/dvc-test/.A1spXky4MBDrt2KnThTN2g.tmp'
2024-01-24 08:15:52,934 DEBUG: Removing '/Users/<username>/development/dvc-test/.dvc/cache/files/md5/.k3hsN3ziCvVMZN2ST_CrnA.tmp'
2024-01-24 08:15:52,934 DEBUG: Removing '/Users/<username>/development/dvc-test/data.csv'
2024-01-24 08:15:52,936 DEBUG: Saving information to 'data.csv.dvc'.
100% Adding...|████████████████████████████████████████|1/1 [00:00, 53.55file/s]

To track the changes with git, run:

    git add data.csv.dvc
```

Removing from and Cleaning Cache

- Remove added files using `dvc remove`

```
$ dvc remove data.csv.dvc
```

- To clear the cache, use the `dvc gc`
  - Use with `-w` flag to remove workspace cache

```
$ dvc gc -w
```

```
WARNING: This will remove all cache except items used in the workspace of the current repo.
Are you sure you want to proceed? [y/n]: y
Removed 1 objects from repo cache.
```

Summary

- DVC cache stages data files before commit
- Configure cache location
  - `dvc cache dir ~/mycache`
- Add files to cache
  - `dvc add data.csv`
  - Creates a `.dvc` file with metadata
- Remove added files `dvc remove data.csv.dvc`
  - Clean workspace cache with `dvc gc -w`

Let's practice!

Configuring DVC Remotes

INTRODUCTION TO DATA VERSIONING WITH DVC

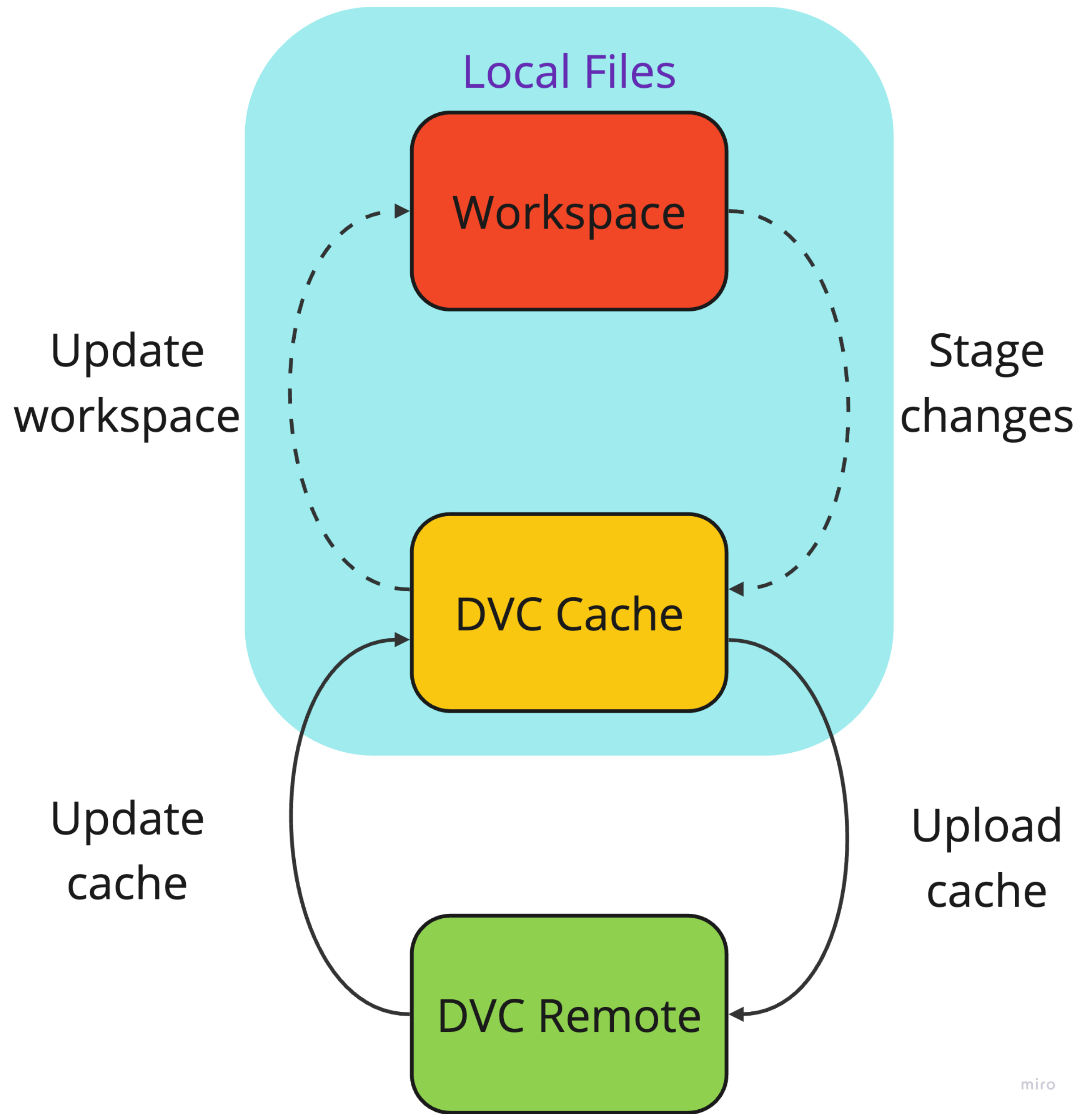
Ravi Bhadauria
Machine Learning Engineer

Recap

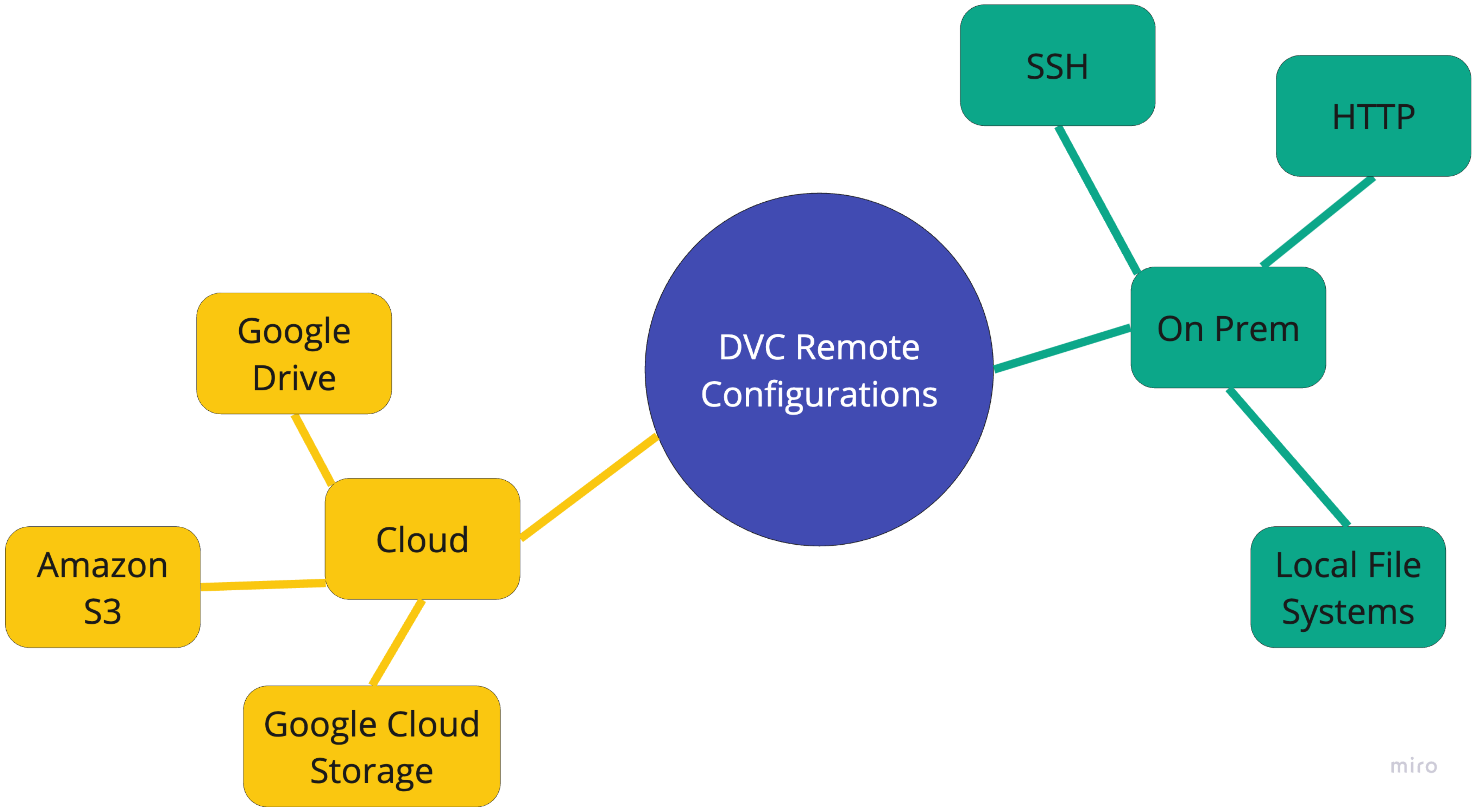
- Initializing DVC repository
  - Run `dvc init`
  - Repo inside workspace (`/path/to/my-project`)
- Setting up DVC cache
  - Temporary staging area within `.dvc` directory
    - `/path/to/my-project/.dvc/cache`
  - Stage temporary files using `dvc add`
- Now: **DVC Remotes**
  - External storage
  - Track and share assets

The Need for DVC Remotes

- DVC Remotes: Location for Data Storage
- Similar to Git remotes, but for *cached* data
- Benefits of using remotes
  - Synchronize large files and directories
  - Centralize or distribute data storage
  - Save local space



Supported Storage Types



Setting up Remotes

- Setting remotes
  - `dvc remote add <name> <location>`
- S3 bucket

```
$ dvc remote add s3_remote \
   s3://mys3bucket
```

- DVC config changes

```
['remote "s3_remote"']
    url = s3://mys3bucket
```

- GCP bucket

```
$ dvc remote add gcp_remote \
   gs://myGCPbucket
```

- Azure

```
$ dvc remote add azure_remote \
   azure://mycontainer/path
```

Local Remotes

- Local remotes are used for rapid prototyping
- Use system directories or Network Attached Storage

```
$ dvc remote add mylocalremote /tmp/dvc
```

- Set default remotes with `-d` flag

```
$ dvc remote add -d mylocalremote /tmp/dvc
```

- Default remote assigned in the `core` section of `.dvc/config`

```
[core]
remote = mylocalremote
```

Listing Remotes

- Listing remotes

```
$ dvc remote list
```

```
s3_remote    s3://mys3bucket
local_remote /tmp/dvcremote
```

- Reads from `.dvc/config`

```
['remote "s3_remote"']
    url = s3://mys3bucket
['remote "local_remote"']
    url = /tmp/dvcremote
```

Modifying Remote Configuration

- Customizations can be done with `dvc remote modify`

```
$ dvc remote modify s3_remote connect_timeout 300
```

- DVC config file change

```
['remote "s3_remote"']
    url = s3://mys3bucket
    connect_timeout = 300
```

Summary

- DVC remotes are used to share data and ML models
- Variety of local and cloud based storage locations are supported
- Add remotes: `dvc remote add`
  - Use `-d` flag to specify default
- List remotes: `dvc remote list`
- Modify remotes: `dvc remote modify`

Let's practice!

INTRODUCTION TO DATA VERSIONING WITH DVC

Interacting with DVC Remotes

INTRODUCTION TO DATA VERSIONING WITH DVC

Ravi Bhadauria
Machine Learning Engineer

Uploading and Retrieving Data

- Moving data from cache to DVC remote

```
$ dvc push <target>
$ dvc pull <target>
```

- Targets are individual files

```
$ dvc push data.csv
```

Uploading and Retrieving Data

- Push entire cache

```
$ dvc push
```

- Update the cache without changing workspace contents

```
$ dvc fetch
```

- Override default remote with `-r` flag

```
$ dvc push -r aws_remote data.csv
```

Similarities with Git

`dvc pull`

- **Function:** Downloads remote data to DVC workspace
- **Use Case:** Large datasets or model artifacts

`dvc push`

- **Function:** Uploads data to remote storage
- **Use Case:** Sharing or storing data artifacts

`git pull`

- **Function:** Fetch/Merge data remote Git repo
- **Use Case:** Local branch in sync with remote

`git push`

- **Function:** Uploads local changes to remote
- **Use Case:** Share changes to Git remote

Versioning data

- `.dvc` is tracked by Git, not DVC
- Leverage this to checkout specific version of data file
- Checkout `.dvc` file

```
$ git checkout <commit_hash|tag|branch>
```

- Retrieve data with MD5 specified in `.dvc` file

```
$ dvc checkout <target>
```

Tracking Data Changes

- Change data file contents, then add dataset changes

```
$ dvc add <target>
```

- Commit changed `.dvc` file to Git

```
$ git add <target>.dvc
$ git commit <target>.dvc \
    -m "Dataset updates"
```

- Push metadata to Git

```
$ git push origin main
```

- Upload changed data file

```
$ dvc push
```

Let's practice!

INTRODUCTION TO DATA VERSIONING WITH DVC