

# MLOps

Operationalizing Machine learning model to production

**SESSION 2**

## Github Setup

1. Check if the git is installed on your system:
  - a. Open your terminal and run the following command  
`git --version`

```
[→ ~ git --version  
git version 2.37.1 (Apple Git-137.1)
```

2. If not then install git on your system:  
Mac/ Windows: <https://www.linode.com/docs/guides/how-to-install-git-on-linux-mac-and-windows/>



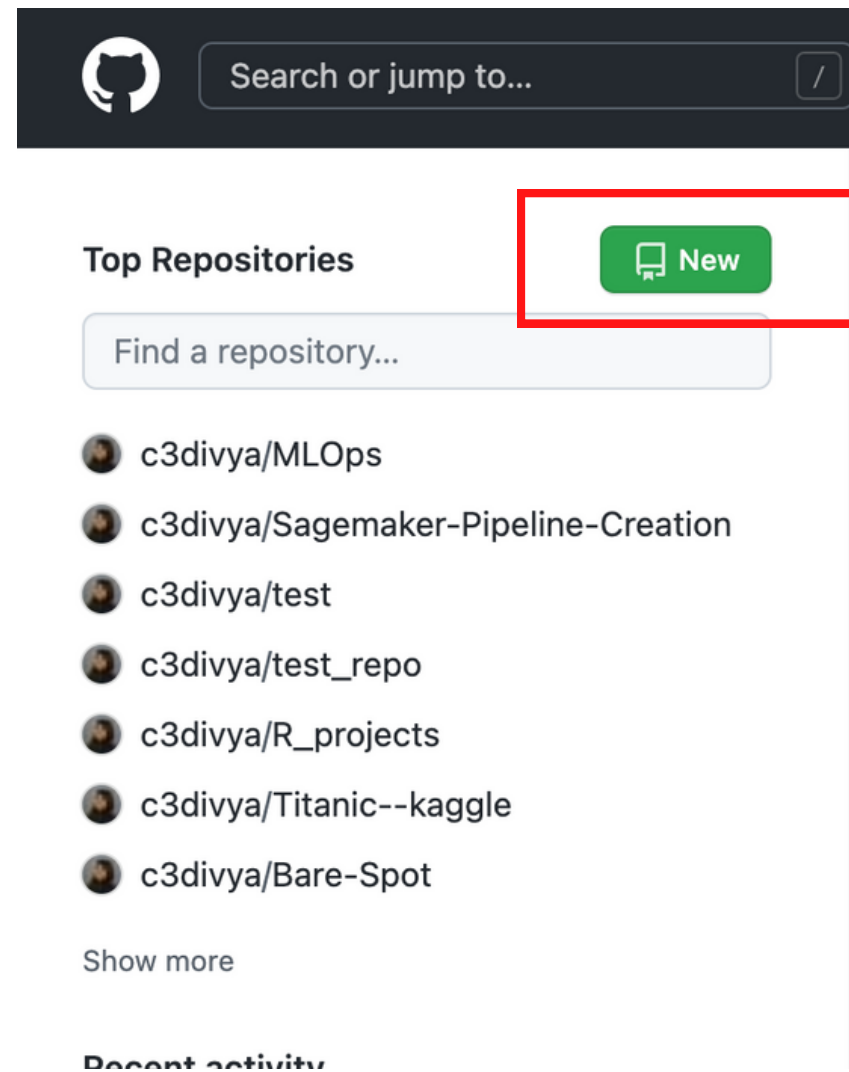
## Let's setup your first git repo

This link will help you provide all the details

<https://docs.github.com/en/get-started/importing-your-projects-to-github/importing-source-code-to-github/adding-locally-hosted-code-to-github>



## First Create a repository on git



1. Navigate to your github account.
2. Click on the "New" tab , this will take you to the create repository page

## First Create a repository on git



### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*



c3divya

Repository name \*

Great repository names are short and memorable. Need inspiration? How about [fantastic-waffle?](#)

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

#### Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

#### Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

#### Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

You are creating a public repository in your personal account.

Add the repository name here that same as what you have on local

Create a public repository

Do not initiaze the repository right now



## First Create a repository on git



Owner <sup>\*</sup> c3divya / Repository name <sup>\*</sup> test1 ✓

Great repository names: test1 is available. ✗morable. Need inspiration? How about [fantastic-waffle?](#)

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

ⓘ You are creating a public repository in your personal account.

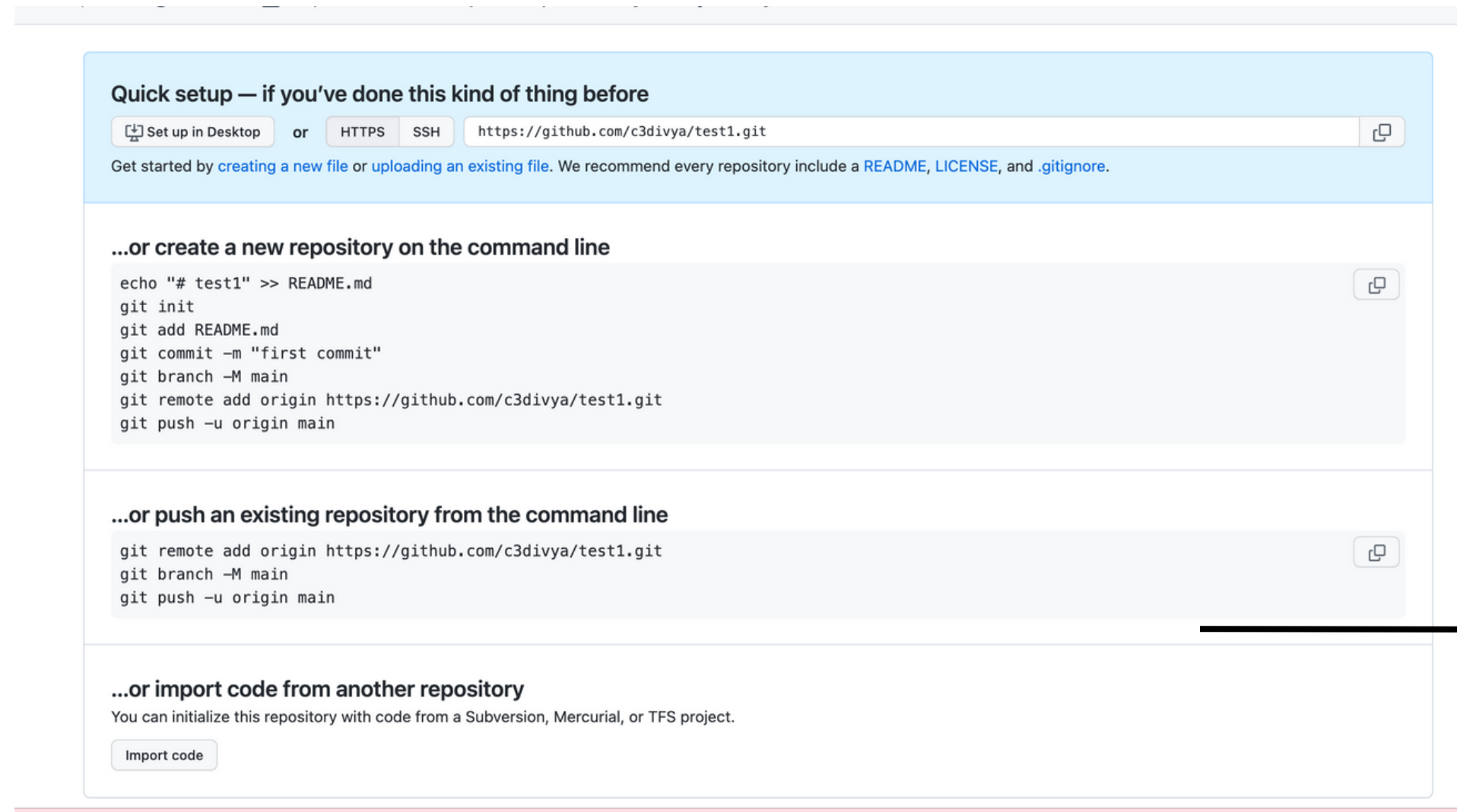
Create repository

Add the repository name here  
.Here is an example where I  
have named the repository as  
test 1

Click on "Create repository"  
tab

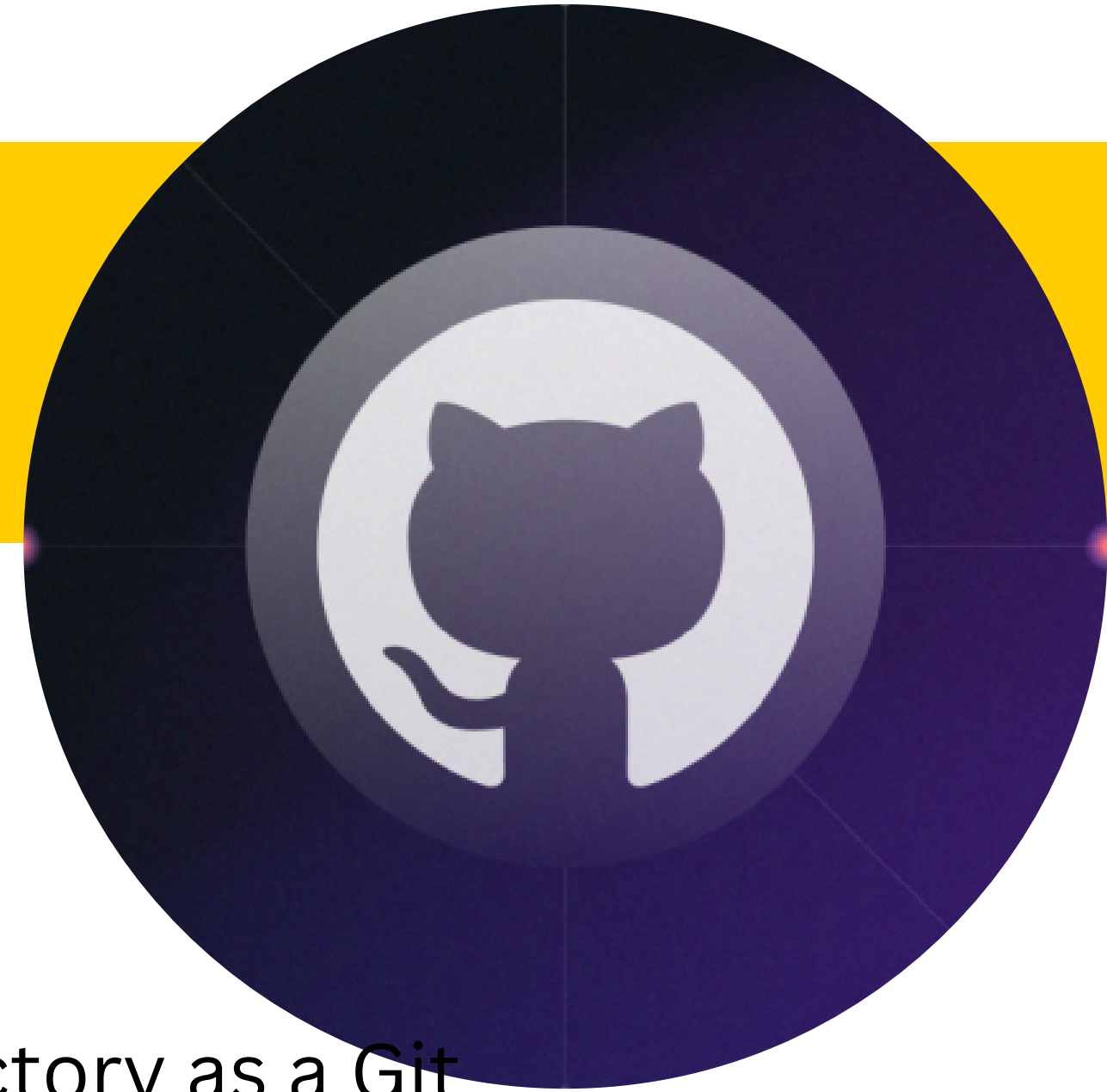
# StatusNeo

## You will land to this page



After this page is ready, go to your terminal

## Git setup for your locally hosted repo



1. Navigate to current working directory

```
→ Documents cd test1  
→ test1 █
```

2. Use the init command to initialise the local directory as a Git repository. By default, the initial branch is called main.

```
→ test1 git init -b main
```

Initialized empty Git repository in /Users/divya\_gandhi/Documents/test1/.git/

```
→ test1 git:(main) █
```



## Git setup for your locally hosted repo



3. Add the global config details if already not added

```
→ test1 git:(main) git config --global user.name "Divya Gandhi"  
test1 git:(main) git config --global user.email "c3divya@gmail.com"
```

4. Create a README.md file and add details into it

```
→ test1 git:(main) touch README.md
```

5. Add the files in your new local repository. This stages them for the first commit.

```
→ test1 git:(main) ✕ git add .
```

## Git setup for your locally hosted repo

6. Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.

```
→ test1 git:(main) ✕ git commit -m "First Commit"
[main (root-commit) 62d99a1] First Commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

7. Copy the URL of your remote registry

**Quick setup — if you've done this kind of thing before**

📄 Set up in Desktop

 or 

HTTPS

SSH

https://github.com/c3divya/test1.git

📄

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).



## Git setup for your locally hosted repo

8. In Terminal, add the URL for the remote repository where your local repository will be pushed.

```
→ test1 git:(main) git remote add origin https://github.com/c3divya/test1.git
```

9. Verifies the new URL added by the below mentioned command

```
→ test1 git:(main) git remote -v
origin https://github.com/c3divya/test1.git (fetch)
origin https://github.com/c3divya/test1.git (push)
```



## Git setup for your locally hosted repo

10. Push the changes in your local repository to GitHub.com. Pushes the changes in your local repository up to the remote repository you specified as the origin

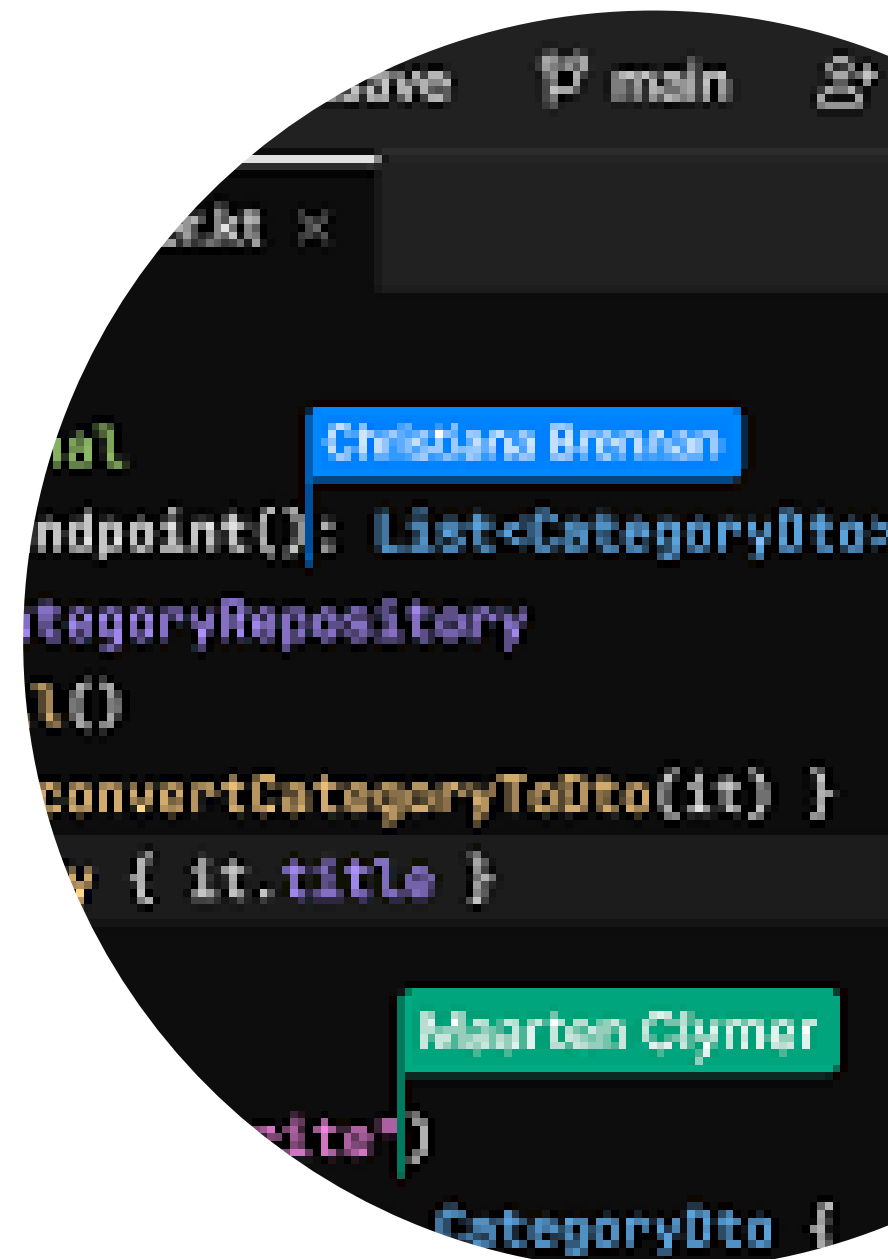
```
→ test1 git:(main) git push -u origin main
```



## Machine Learning System Architecture

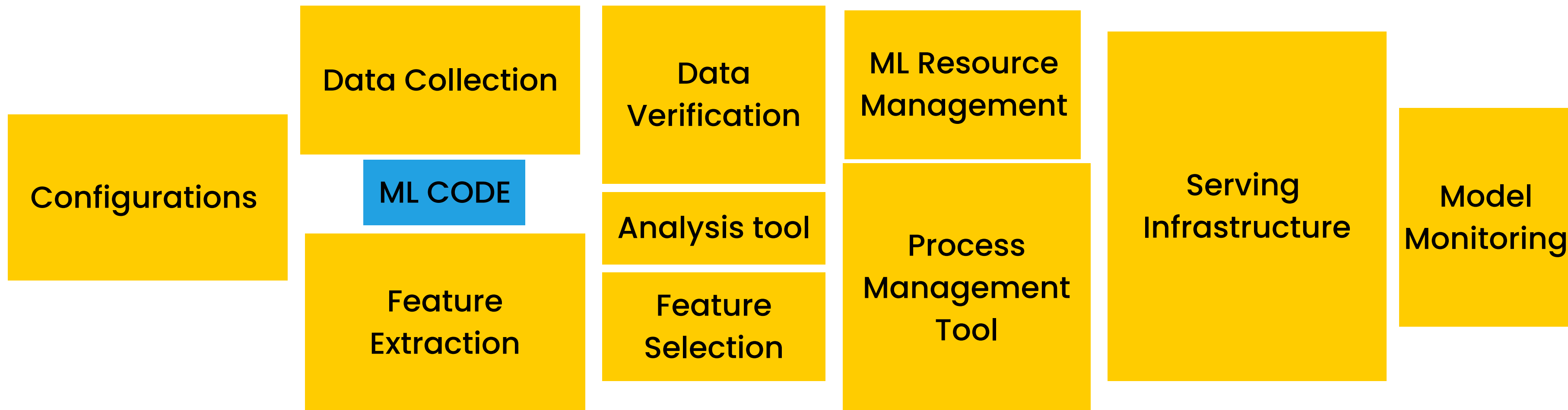
Machine Learning System in Production requires different components in order to work:

- Infrastructure
- Applications
- Data
- Documentation
- Configuration

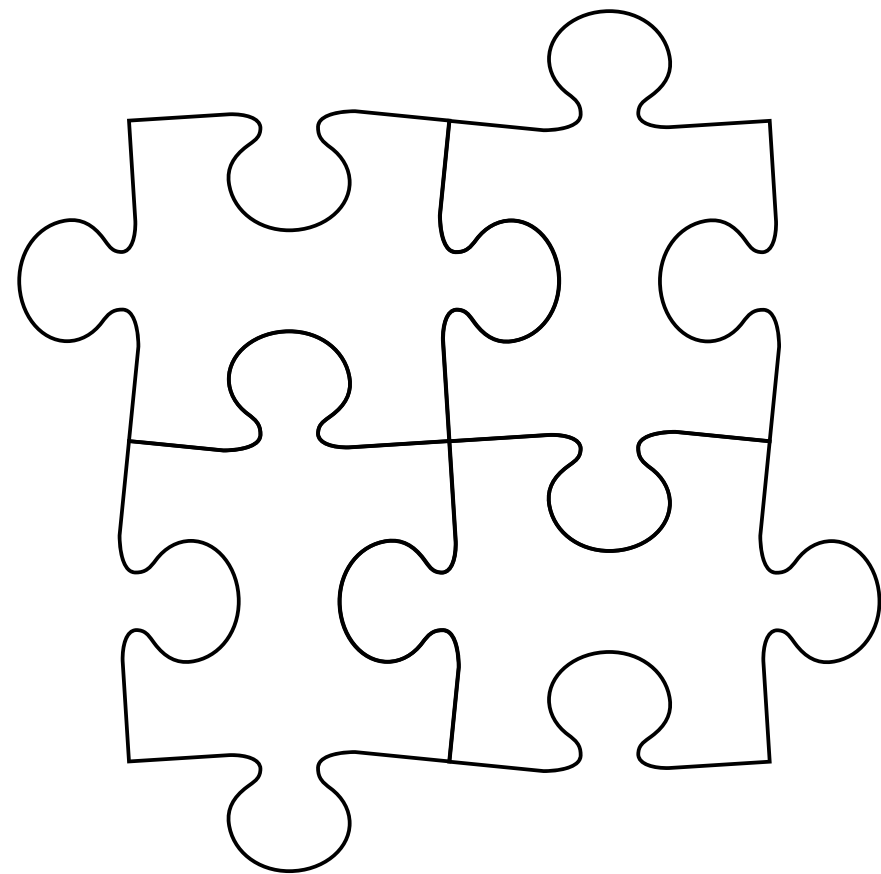




## ML System are complex



## Challenges faced while building ML System



Need of reproducibility (versioning everywhere)