# Git and Docker Commands with Initialization Steps

## 🧩 Git Commands

* • git init

➤ Initializes a new Git repository in the current directory.

* • git add <file>

➤ Stages a file to be included in the next commit.

* • git commit -m "message"

➤ Records the staged changes with a message.

* • git push origin <branch>

➤ Uploads local commits to a remote repository branch.

* • git clone <repo\_url>

➤ Downloads a copy of a remote repository to your system.

### How to Initialize Git

1. Check if Git is installed: `git --version`
2. Install Git if not installed: `sudo apt install git -y`
3. Set username and email:
4. git config --global user.name "Your Name"
5. git config --global user.email "your\_email@example.com"
6. Initialize a repository: `git init`
7. Add and commit files:
8. git add <filename>
9. git commit -m "Initial commit"
10. Connect to GitHub and push:
11. git remote add origin <repo\_url>
12. git branch -M main
13. git push -u origin main

## 🐳 Docker Commands

* • docker pull <image>

➤ Downloads an image from Docker Hub to your local system.

* • docker run <options> <image>

➤ Creates and starts a container from an image.

* • docker ps -a

➤ Lists all containers (running and stopped).

* • docker build -t <name> .

➤ Builds a Docker image from a Dockerfile in the current directory.

* • docker exec -it <container\_id> bash

➤ Opens a terminal session inside a running container.

### How to Initialize Docker

1. Check if Docker is installed: `docker --version`
2. Install Docker if not installed: `sudo apt install docker.io -y`
3. Start and enable Docker service:
4. sudo systemctl start docker
5. sudo systemctl enable docker
6. Run a test container to verify: `sudo docker run hello-world`
7. List running containers: `sudo docker ps -a`
8. To avoid using sudo each time:
9. sudo usermod -aG docker $USER
10. newgrp docker