Good Morning

JSD#7 !

AGENDA

กำหนดการ

09:15 - 10:00 am	**Theory** (45 mins)
10:00 - 10:50 am	**Practice 1 & 2** (50 mins)
10:50 - 11:00 am	Break (10 mins)
11:00 - 11:40 am	**Practice 3 & 4** (40 mins)
11:40 - 12:00 pm	Discuss with cohort & NJMKL

Total: 2h 45m

INTRO-PROG2: Git & GitHub Source Control

Version Control,

Remote Repository and Deploy

OBJECTIVE

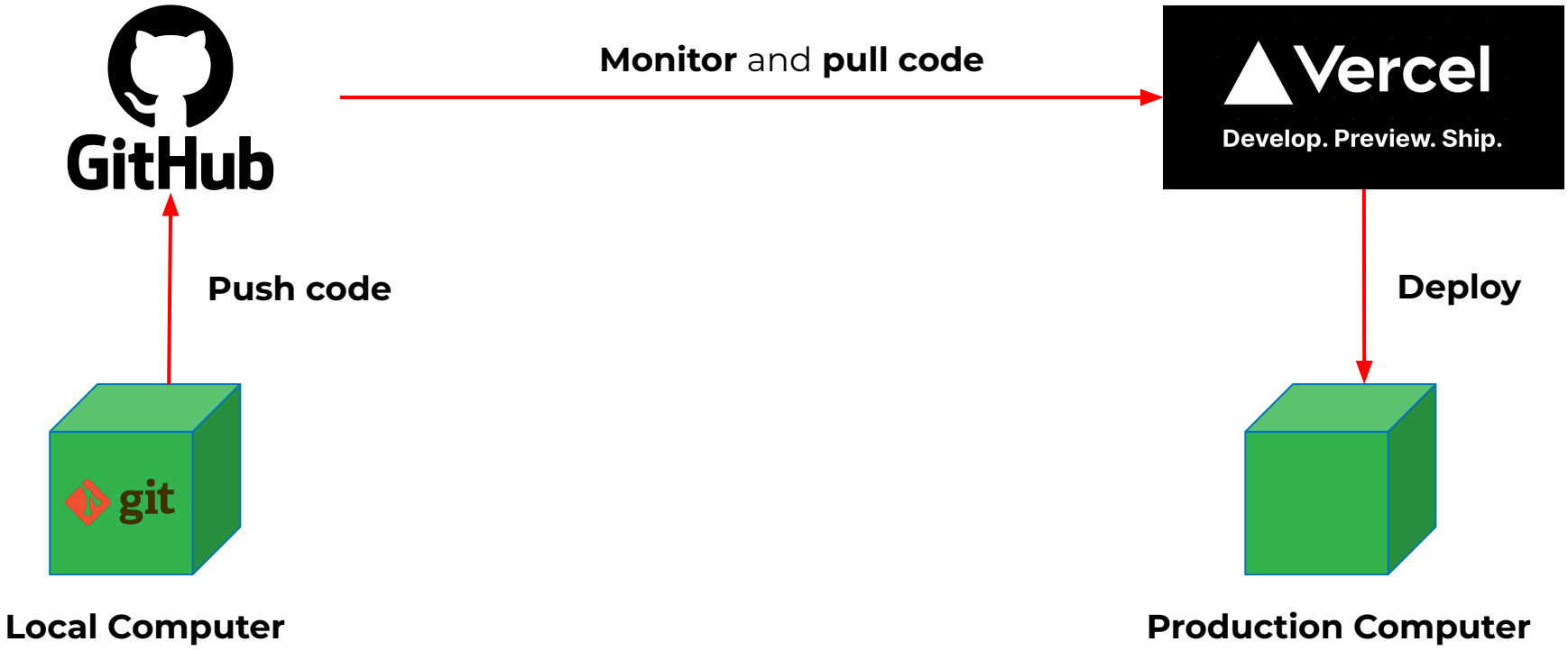
วัตถุประสงค์

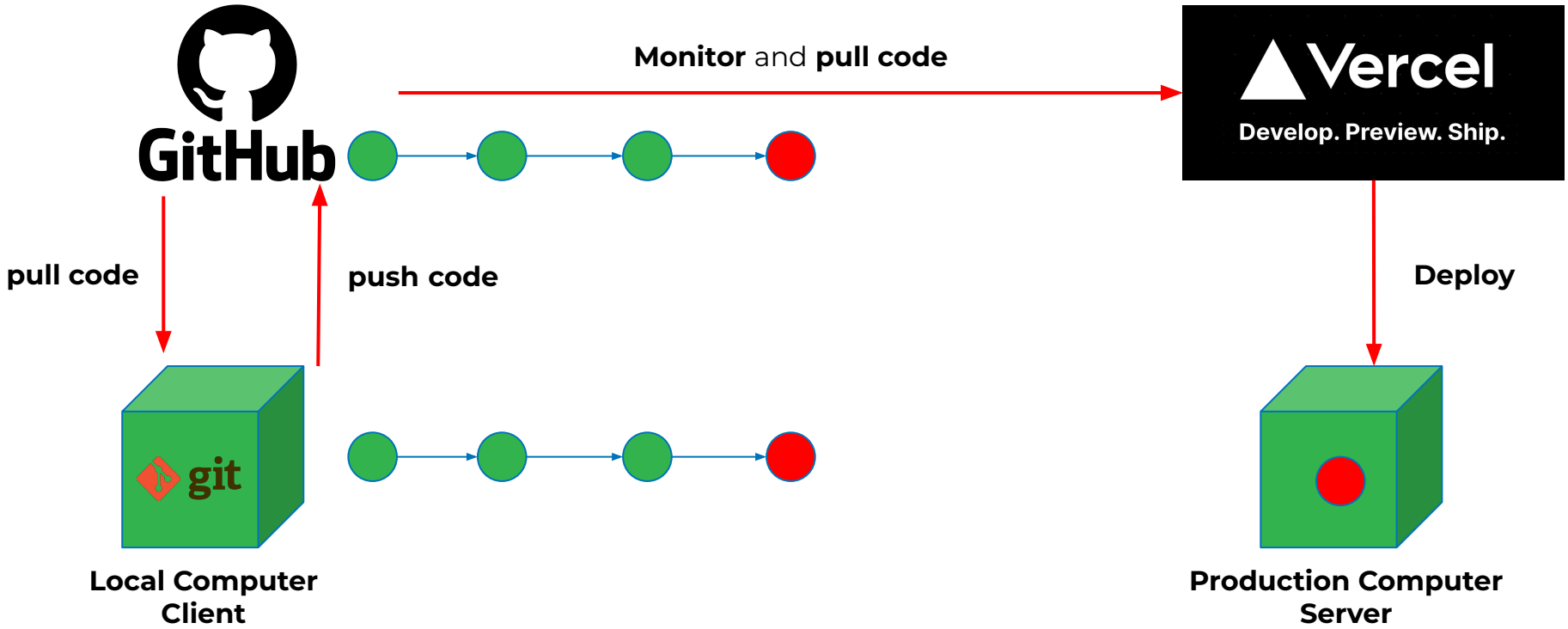
- Explain **source control** and why it is important
- Explain and use **Git**
- Use **Github** to create a personal account for a portfolio
- Explain and use the main **Git commands**:
  - git *clone*
  - git *status*
  - git *add*
  - git *commit*
  - git *pull*
  - git *push*
  - git *branch*
  - git *checkout*
  - git *merge*
  - git *fork*
  - git *reset*
- Use **feature branches** to implement code changes

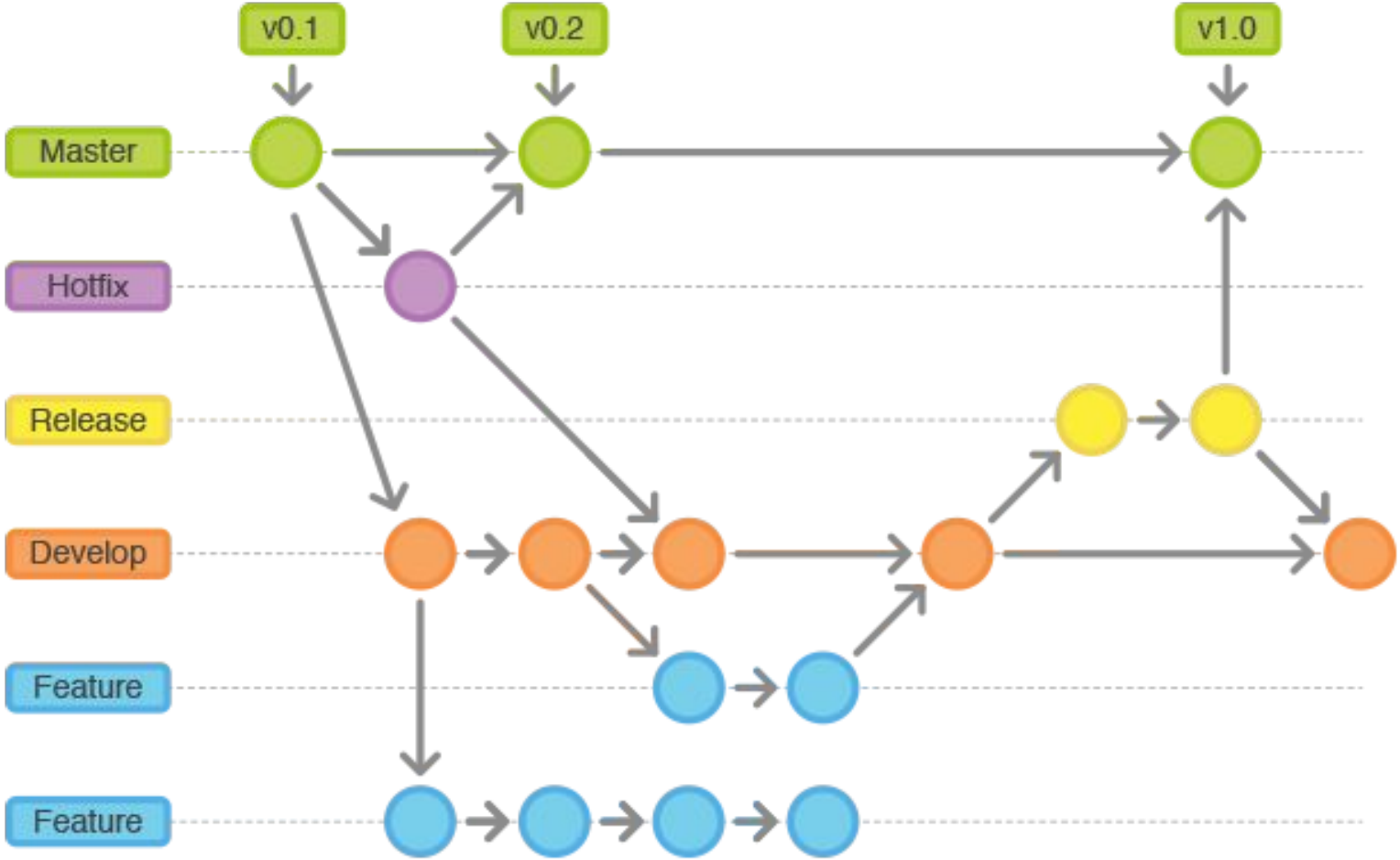

git


GitHub

	
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features


GitHub
Monitor and pull code
Vercel
Develop. Preview. Ship.
Push code
Deploy
git
Local Computer
Production Computer


GitHub
Monitor and pull code
Vercel
Develop. Preview. Ship.
pull code
push code
Deploy
git
Local Computer
Client
Production Computer
Server


v0.1
v0.2
v1.0
Master
Hotfix
Release
Develop
Feature
Feature

Know your tools:

- **GIT** → version control of source code
- **GitHub** → remote repository
- **Vercel** → deployment

Why GIT?

- Code (*html*, *css*, *js*)
- Folder = source code
- Folder > version control
- Backtracking

Why GitHub?

- Track online
- Share
- Contribute

Why Vercel?

- Publish
- Deploy

Scenario 1

As a software developer, I want to **track the versions of my code** and be able to **revert back to any versions** whenever I need to.

GIT is the tool I will use to achieve this.

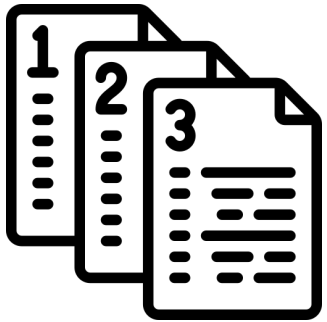
As a software developer, I want to **track the versions of my code** and be able to **revert back to any versions** whenever I need to.

Pattern 1 - Setting, overview and branching	**Pattern 2 - Save and track a new version of my source code.**
git init git branch git branch -M main git status git add . git commit -m “initial commit” git status git add . git commit -m “your commit short description” git status git log	git status git add . git commit -m “your commit short description”

Scenario 2

As a software developer, I also want to **keep the version control record online** not just in my computer so that I could share my code with other developers.

GitHub is the tool I will use to achieve this.



Scenario 2

As a software developer, I also want to **keep the version control record online** not just in my computer so that I could share my code with other developers around the world.

Pattern 1 - Create remote repository and connect	Pattern 2 - Save and track a new version of my source code on the remote repository
git remote -u origin “url”	git status git add . git commit -m “your commit short description” git push

Scenario 3

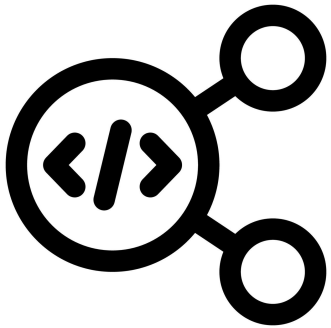
As a software developer, I need a way to **publish my website/app to the world**.

Vercel is the tool I will use to achieve this.

Scenario 4

As a software developer working with other developers, I need a way to **contribute my code and get up to date code contributed by other developers**.

GitHub is the tool I will use to achieve this.



Scenario 4

As a software developer working with other developers, I need a way to **contribute my code and get up to date code contributed by other developers**.

Pattern 1 - Create remote repository and connect	Pattern 2 - Save and track a new version of my source code on the remote repository.
git pull git status git add . git commit -m “your commit short description” git push	git status git add . git commit -m “your commit short description” git push

Local Repository

Save and track versions of my code on my computer.

git init

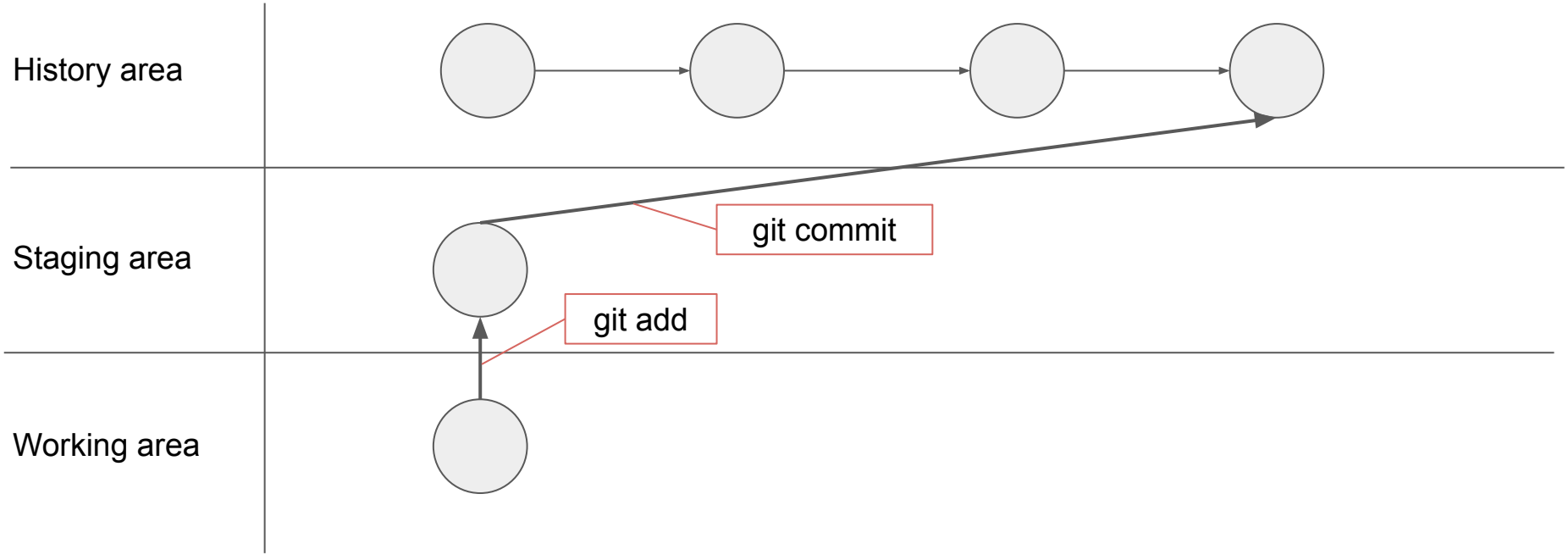
git branch

git checkout

git status

git log

GIT



git add

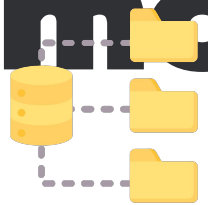
git commit

Let’s build!

Exercise 1

“Safe and sound with git init and Github repo.”

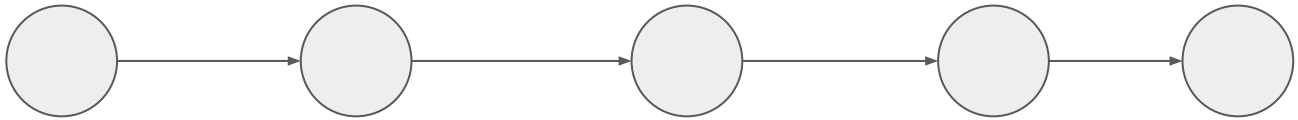
Remote Repository



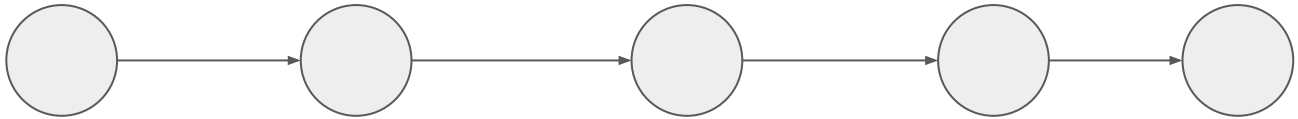
Save and track versions of my code online.

GIT

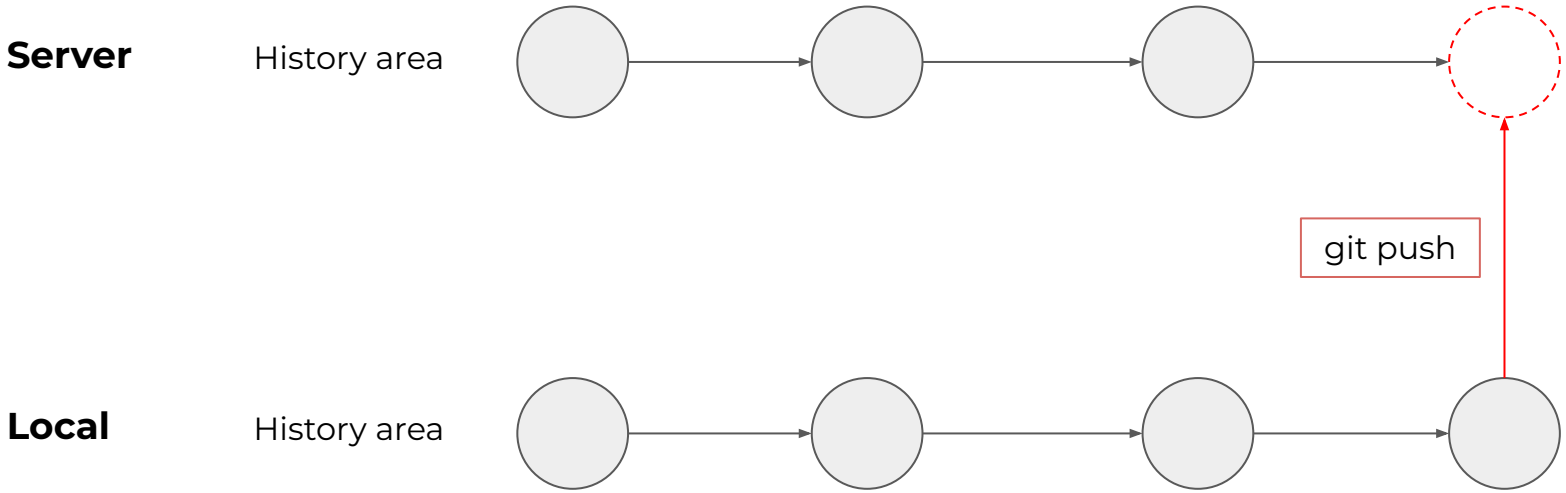
Server History area



Local History area



GIT



Create github and **git repository**

1) Go to **github**: https://github.com/
2) (*if any*) register **new user**.
3) Go to **repository**.
4) Create **new repository**.
5) Copy **url of new repository**.

git remote add [**name of server**] [url]

default is “origin”

git push

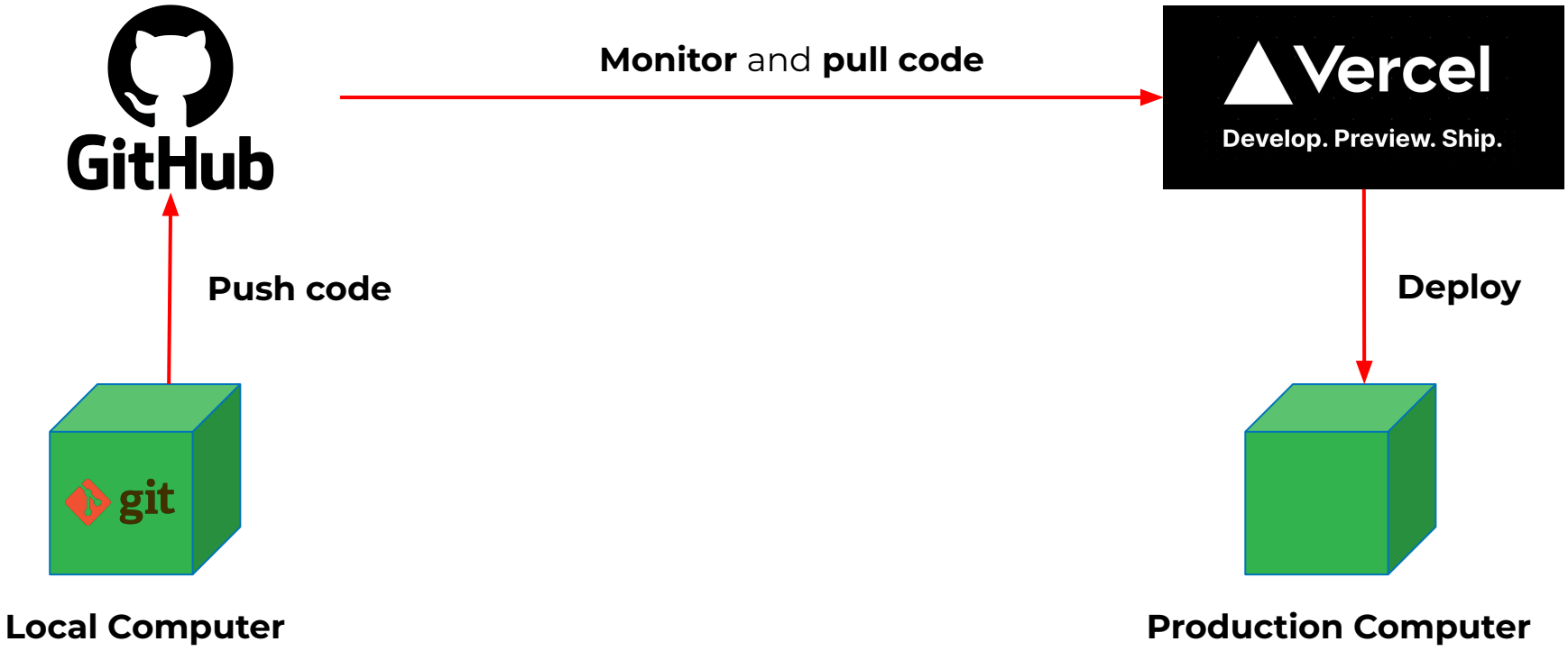
Let’s build!

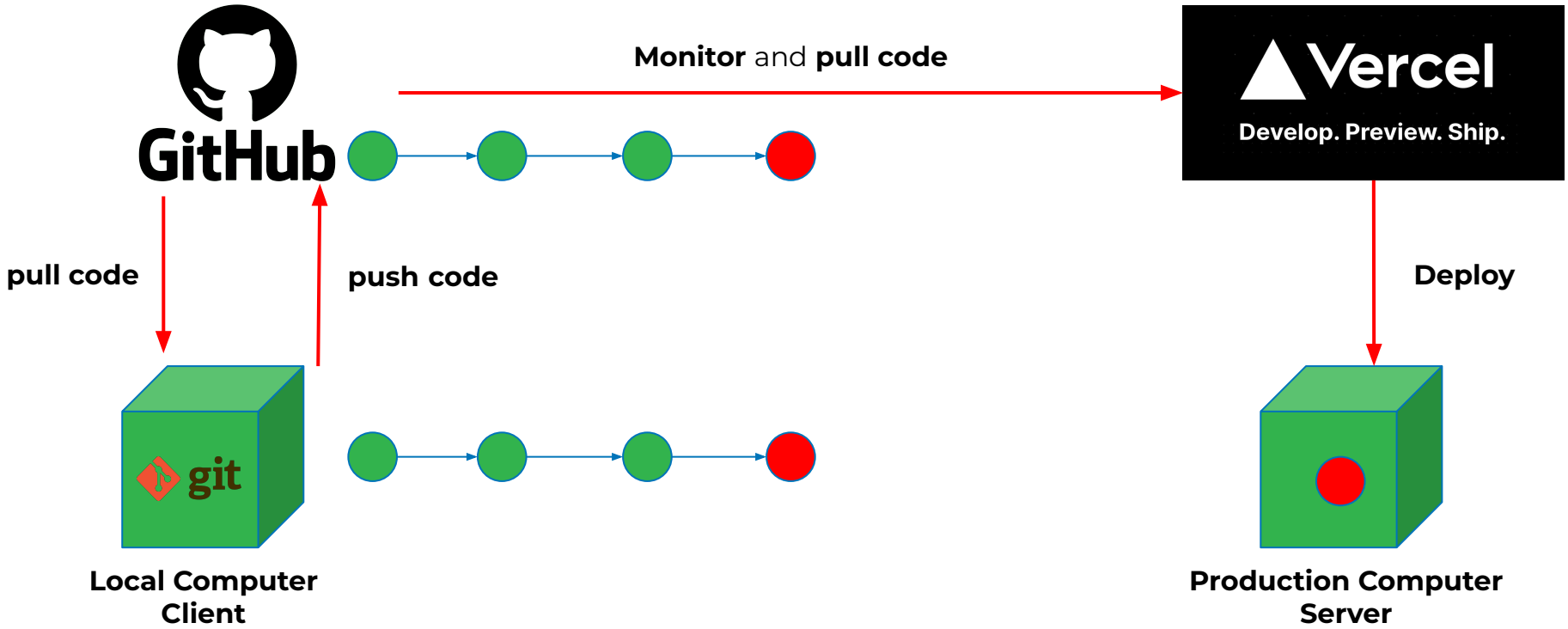
Exercise 2

“Safe and sound with git init and Github repo.”

Deploy

Publish my website/app for the world to see/use.


GitHub
Monitor and pull code
Vercel
Develop. Preview. Ship.
Push code
Deploy
git
Local Computer
Production Computer


GitHub
Monitor and pull code
Vercel
Develop. Preview. Ship.
pull code
push code
Deploy
git
Local Computer
Client
Production Computer
Server

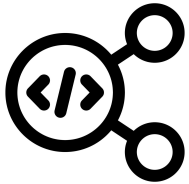
Let’s build!

Exercise 3

“Deploy on Vercel”

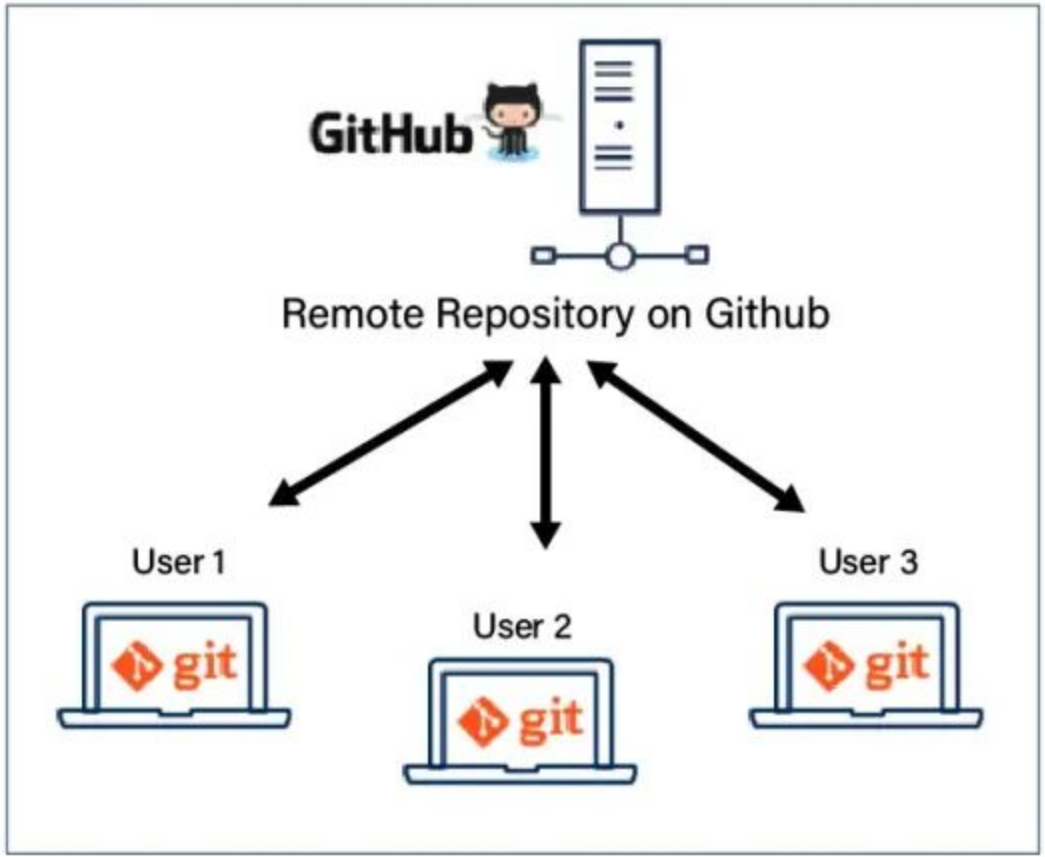
Exercise 3 Instruction

1. Go to Vercel: https://vercel.com/
2. (if any) register new user.
3. Copy **git** repository url.



Contribute Your Code

Build projects with other developers.


GitHub
Remote Repository on Github
User 1
User 2
User 3
git
git
git

git fetch

git pull

git clone

git merge

Forking

Pull Request

Let’s build!

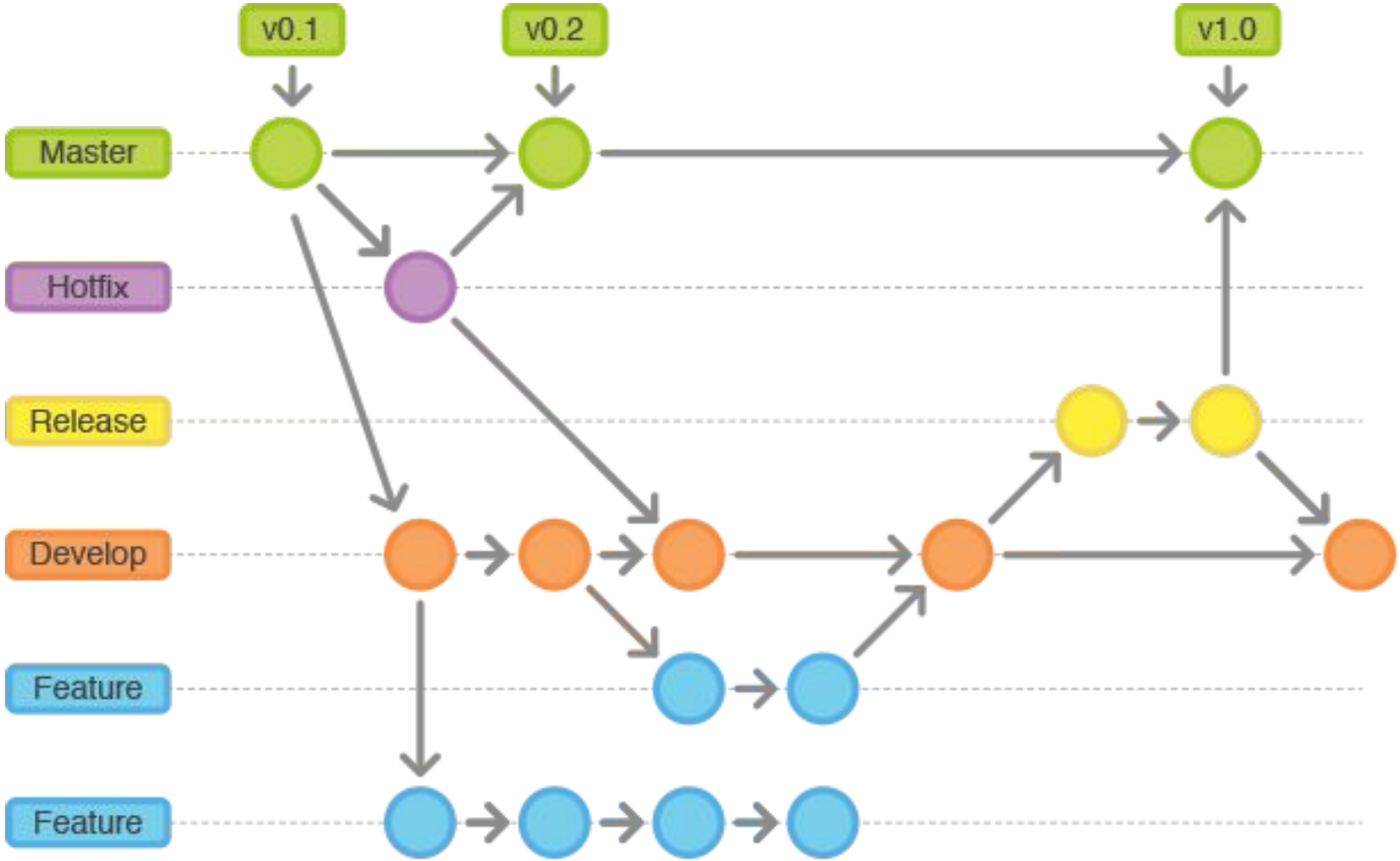
Exercise 4

“Contribute my code.”

Exercise 4 Instruction

1. **Fork a repo**.
2. **Contribute code** to help complete the website.
3. Create a **Pull Request**.
4. **Merge your pull request**.

Let's Recap!


v0.1
v0.2
v1.0
Master
Hotfix
Release
Develop
Feature
Feature

Session Cheat Sheet

OBJECTIVE

วัตถุประสงค์

- Explain **source control** and why it is important
- Explain and use **Git**
- Use **Github** to create a personal account for a portfolio
- Explain and use the main **Git commands**:
  - git *clone*
  - git *status*
  - git *add*
  - git *commit*
  - git *pull*
  - git *push*
  - git *branch*
  - git *checkout*
  - git *merge*
  - git *fork*
  - git *reset*
- Use **feature branches** to implement code changes

Thank you & Well done!

Happy Coding!