

# COMP1531

## 9.2 - Deployment, Maintenance

# Preface

1. Quick UI help for ass3
2. Git hooks
3. Continuous integration in gitlab
4. Deploying on Heroku
5. Maintenance and other

# Funny



**Clifford** 2:04 PM

Exam idea. 20 percent of the exam marks come from explaining the change made by a specific commit. The commit message is pulled from their repo



All they get is the commit message

# Git hooks

## **Demonstration of:**

- pre-commit hook
- prepare-commit-message
- commit-message

[Read more about this here](#)

# Virtual Environments

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them.

You can read more about them [here](#) and [here](#).

You may be asked a question about them on the exam, but you will never be required to use them.

They are often required for use with CI/CD

# Virtual Environments

```
1 pip3 install virtualenv
2 python3 -m virtualenv venv/
3 source venv/bin/activate
4
5 # Do stuff
6
7 deactivate
```

# Continuous integration, gitlab

Gitlab, like many source control tools, has a way of doing continuous integration. An [overview is here](#) and a [start guide is here](#).

There is quite a lot of variance and depth to this, so we will not cover it in any detail besides high level

A simple example [can be found here](#).

# Deploying on Heroku

This is one of many simple guides to deployment. Note,  
"Procfile.txt" should be "Procfile"



# Deploying on Heroku

```
1 # Sign up to Heroku, then New => Create new app
2
3 # Create virtual env
4 pip3 install virtualenv
5 python3 -m venv venv
6 source venv/bin/activate
7
8 # Install relevant packages
9 pip3 install flask
10 pip3 install gunicorn
11 pip3 freeze > requirements.txt
12 echo "web: gunicorn app:app" > Procfile
13
14 # Deploy with git
15 git init .
16 echo "venv/*" > .gitignore
17 git add --all
18 git commit -m "First commit"
19 sudo snap install --classic heroku
20 git push heroku master
```

Note: you cannot install heroku on CSE machines

# A/B Testing

Is a randomised scientific experiment with multiple variants (typically two). It consists of one independent variable, with all other variables controlled.

Consists of having two "versions" randomly but equally distributed to end-users, and then monitoring the results. These versions can either be:

- Managed within the same instance
  - Sent to different instances via a load balancer
- 
- Required reading
  - Examples of AB testing

# Maintenance & Monitoring

**Maintenance:** After deployment, the use of analytics and monitoring tools to ensure that as the platform is used and remains in a healthy state.

**Monitoring often has two purposes:**

- Preserving user experience: Monitoring errors, warnings, and other issues that affect performance or uptime.
- Enhancing user experience: Using analytical tools to

# Maintenance

**Maintenance:** After deployment, the use of analytics and monitoring tools to ensure that as the platform is used and remains in a healthy state.

Health is defined by developers, but often consists of:

- Monitoring 4XX and 5XX errors
- Ensuring disk, memory, cpu, and network is not overloaded

Often these aren't actively monitored, but rather monitored with alerts and triggers

# Maintenance

Monitoring often has two purposes:

- 1.