Deployment, Hosting and Monitoring
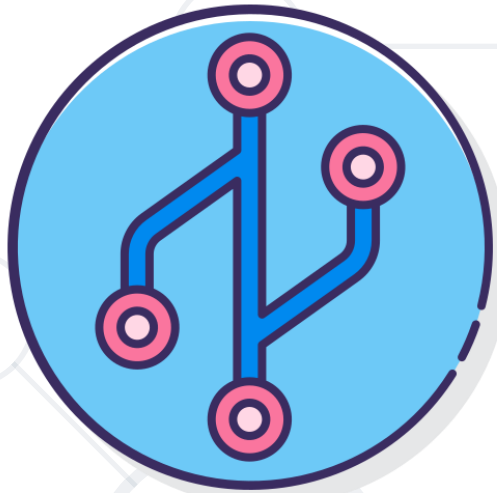
SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

Table of Contents

1. Git
2. Github
3. Deployment
4. Docker
5. Amazon Web Services

sli.do

#python-web

Git

Distributed Source-Control System

What is Git?

- Distributed **source-control system**
  - The most popular in the world
  - Free, open-source software
- Works with **local** and **remote** repositories
- **Git bash** - command line interface for Git
- Runs on Linux, macOS, and Windows (**msysGit**)
  - https://git-scm.com
  - https://www.atlassian.com/git/tutorials/setting-up-a-repository

Using Git

- Console-based Client

  - **git, GitBash**

- Windows/Mac GUI Client - SourceTree

  - **https://www.sourcetreeapp.com/**

- GitHub Desktop Client

  - **https://desktop.github.com**

Installing Git

- **msysGit** installation on Windows

  - Download Git for Windows: https://git-scm.com/downloads

  - Options to select (they should be selected by default)

    - "Use Git Bash Only"

    - "Checkout Windows-style, commit Unix-style endings"

- Git Installation on Linux

  - Using your distribution's package manager

Basic Commands (1)

- Cloning an existing Git repository

```
git clone [remote url]
```

- Fetch and merge the latest changes from the remote repository

```
git pull
```

- Check the status of your local repository (see the local changes)

```
git status
```

Basic Commands (2)

- Preparing (adding/selecting) files for a commit

```
git add [filename] ("git add ." adds everything)
```

- Adds everything in the current directory

```
git add .
```

- Adds all changes

```
git add -A
```

- Committing to the local repository

```
git commit –m "[your message here]"
```

Basic Commands (2)

- Creating a new local repository (in the current directory)

```
git init
```

- Creating a remote (assign a short name for remote Git URL)

```
git remote add [remote name] [remote url]
```

- Pushing to a remote (send changes to the remote repository)

```
git push [remote name] [local name]
```

GitHub

World's Largest Source Code Hosting Site

What is GitHub?

- **GitHub** is the world's #1 source code hosting site
  - Free for open-source projects
  - Paid plans for private repositories
- GitHub provides:
  - Git source code repository
  - Issue tracker (bug tracker)
  - Project board (Kanban style)
  - Wiki pages (documentation)

Deployment

What is Deployment?

- **Deployment** means to push changes or update from one environment to another

Local ⟹ Development ⟹ Staging ⟹ Live

Where to deploy a Python Project?

- We can deploy **one project** onto **multiple websites**

- Some of the deployment websites

  - Amazon Web Services (AWS)

  - Heroku

  - PythonAnywhere

Prepare for Deploy (1)

- Before deploying our project, we should add **3 important keys** to the project

- First, add **"gunicorn"** to pip

- Next, create 2 new files in our **outer project folder**

  - `.env`

  - `requirements.txt`
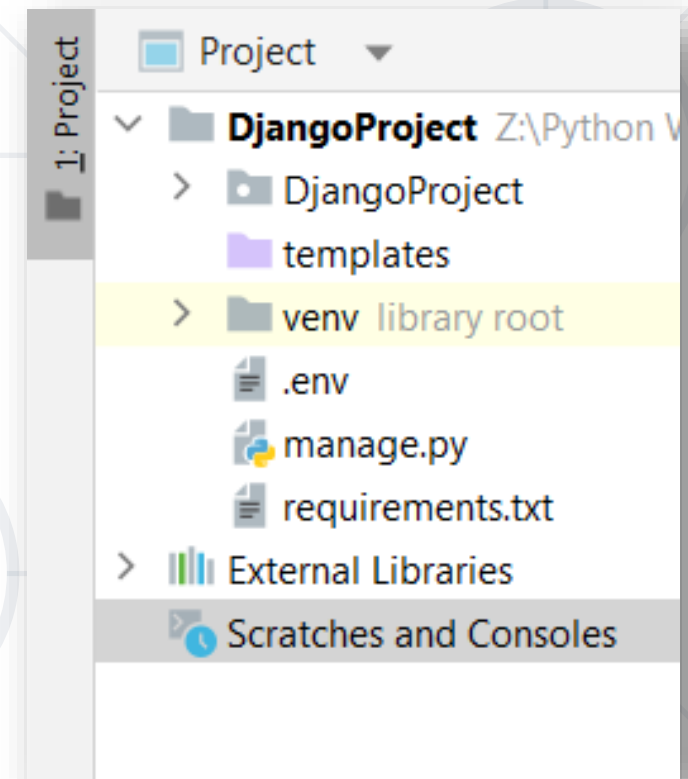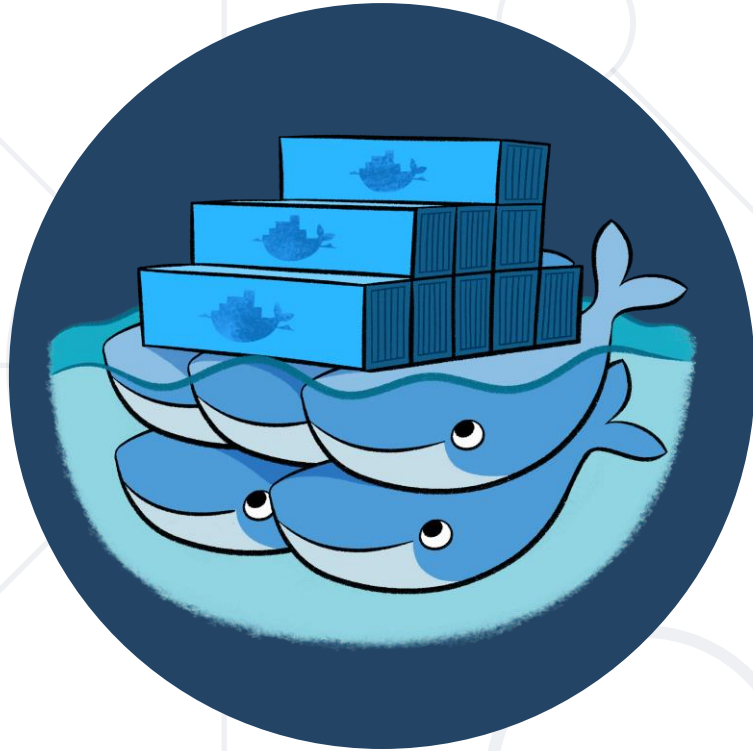
Prepare for Deploy (2)

- Requirements

  - Holds **all of your packages** and their versions

  - To add your packages

```
pip freeze > requirements.txt
```

- .env

  - Holds the **environmental variables** of the application

  - Should not be included in GitHub

Docker

Docker

- **Docker** is a software that performs operating-system-level **virtualization**

  - The process is also known as "**containerization**"

- Docker runs software packages called "**containers**"

  - **Containers** are isolated from each other

    - Each bundles its own application, tools, libraries, configuration files etc.

  - They can **communicate** with each other through well-defined channels

  - All containers are run by a single **Operating System**

    - This makes them significantly more lightweight than virtual machines

  - **Containers** are created by **images** that specify their contents
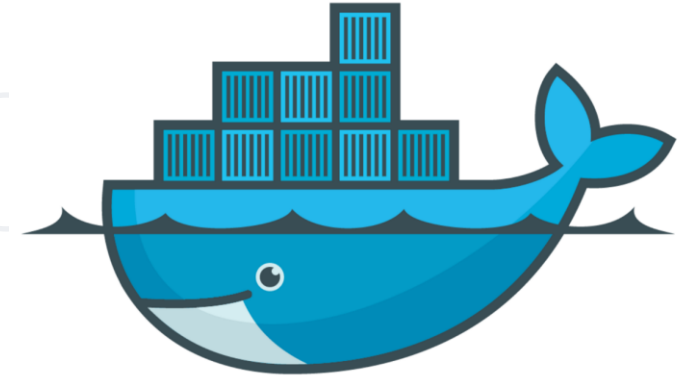
Dockerfile

- To create an image, we should **add a Dockerfile** to our project

- Docker **builds images** automatically by reading the instructions from the Dockerfile

```
FROM python:3
ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1
WORKDIR /app
COPY requirements.txt /app/
RUN pip install -r requirements.txt
COPY . /app/
```

Amazon Web Services

AWS

- Amazon Web Services is one of the world's **best cloud platforms**

- Offers a broad set of global **cloud-based products** that help organizations move faster

- Millions of customers are using AWS to lower costs, become more agile, and innovate faster

AWS Services

- AWS is separated into different **services**
  - Each can be configured in different ways based on the user's needs
- Users should be able to see **configuration options** and **individual server maps** for a service

Amazon EC2

- Amazon **Elastic Compute Cloud** is a web service that provides resizable compute capacity in the cloud

- It is designed to make web-scale computing easier for developers

Amazon S3

- Amazon **Simple Storage Service** is an object storage service

- Benefits
  - Data performance and durability
  - Flexible storage options
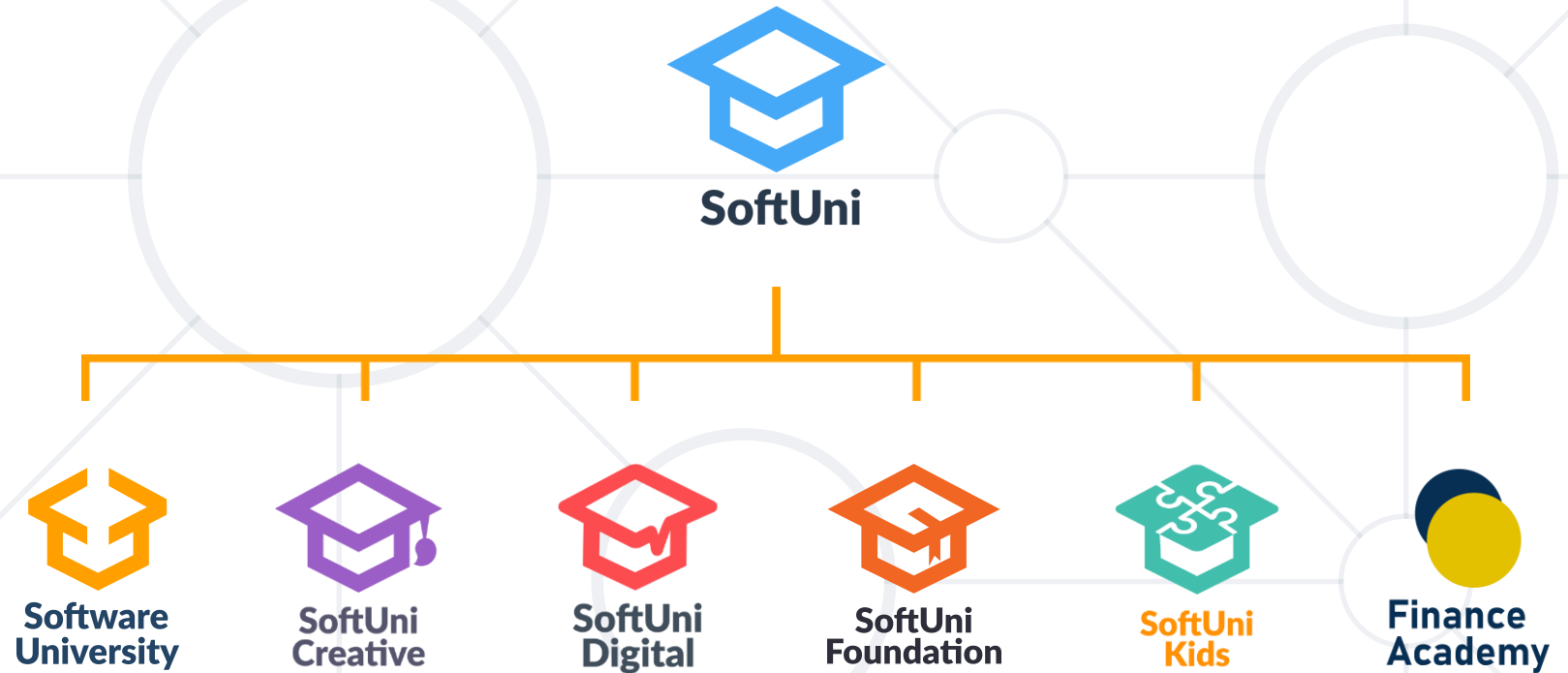 - Security, compliance, and auditing
  - Granular data control

Live Demo

Deploying Django App on EC2

Summary

- Git is a **source-control system**

- GitHub is a **source code hosting site**

- Docker is **an open-source containerization platform**

- AWS is a **cloud computing platform**

Questions?

SoftUni Diamond Partners

Educational Partners

Trainings @ Software University (SoftUni)
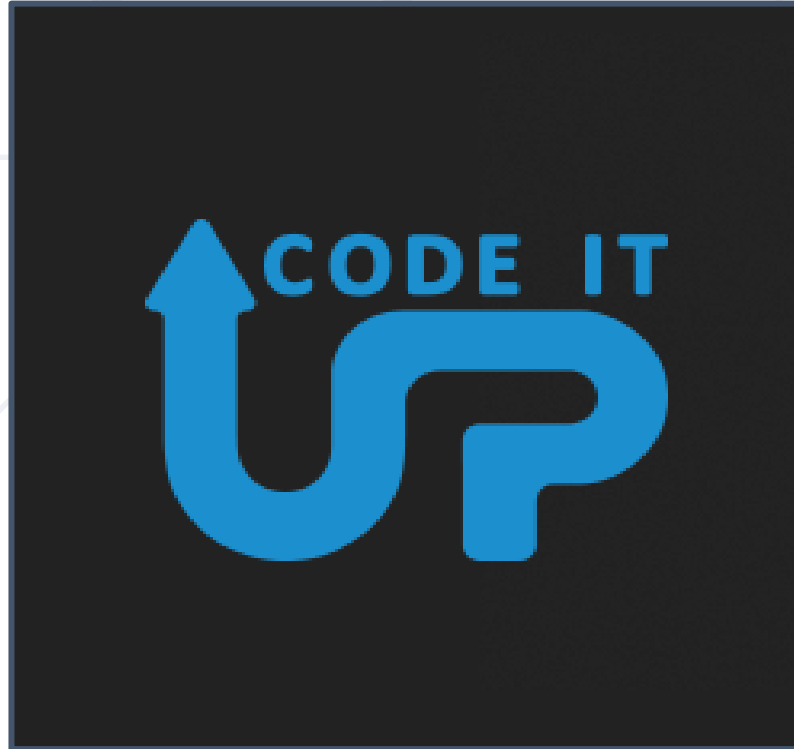
- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, softuni.org
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**

- Unauthorized copy, reproduction or use is illegal

- © SoftUni – https://softuni.org

- © Software University – https://softuni.bg