CS2104 Problem Solving in Computer Science

Margaret Ellis,
Naren Ramakrishnan,
Sehrish Basir Nizamani

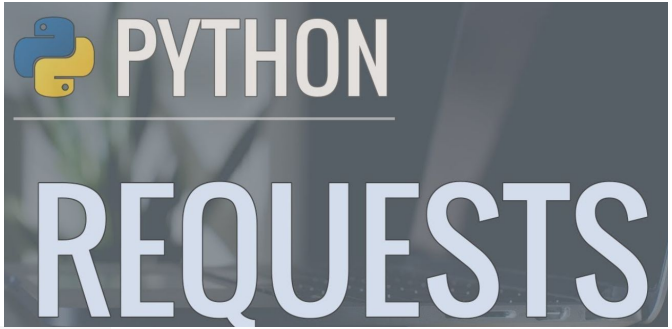
VIRGINIA TECH™

Web Scraping

What is Web Scraping

- An automatic way to retrieve unstructured data from a website and store them in a structured format.

tools and library for web scraping



Scraping all up-front websites

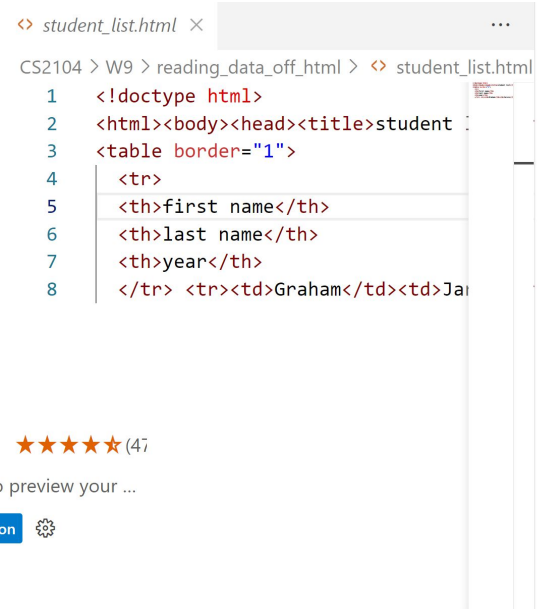
1. Inspect the website HTML that you want to crawl
2. Access URL of the website using code and download all the HTML contents on the page
3. Format the downloaded content into a readable format
4. Extract out useful information and save it into a structured format
5. For information displayed on multiple pages of the website, you may need to repeat steps 2–4 to have the complete information.

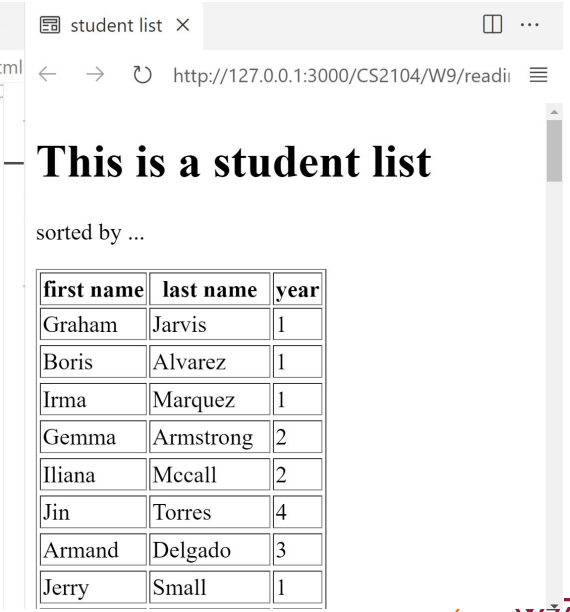
Scraping websites with API

1. Inspect the XHR network section of the URL that you want to crawl
2. Find out the request-response that gives you the data that you want
3. Depending on the type of request(post or get) and also the request header & payload, simulate the request in your code and retrieve the data from API. Usually, the data got from API is in a pretty neat format.
4. Extract out useful information that you need
5. For API with a limit on query size, you will need to use ‘for loop’ to repeatedly retrieve all the data

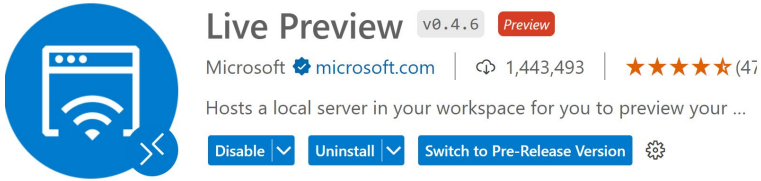
Classwork : reading data off html

Show preview
Side by side view





first name	last name	year
Graham	Jarvis	1
Boris	Alvarez	1
Irma	Marquez	1
Gemma	Armstrong	2
Iliana	Mccall	2
Jin	Torres	4
Armand	Delgado	3
Jerry	Small	1

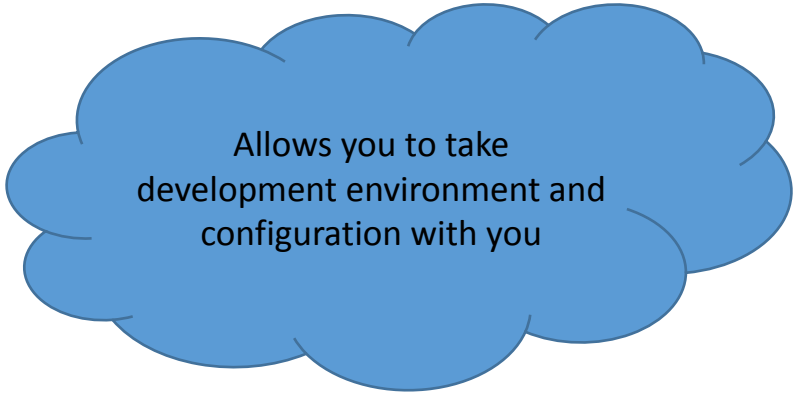


Sample Code:

```
import re
# use regex to parse the html string and extract the target data

in_filename = 'student_list.html'
out_filename = 'student_list.txt'
# defines the name of the file that you're going to read and write into

content = open(in_filename,"r")
output = open(out_filename,"a")
# content and output are file stream object, you could use 'r'(read), 'w'(write) or 'a'(append)

str = content.read()
# read() method will turn a file stream into a regular string

# use matches = re.findall(pattern,str), you need to figure out what pattern to use
# pattern = r'(\w*)<\/td><td>(\w*)<\/td><td>(\d)'
# once you get a match, you access different components from the match object using indices.
# use output.write() to write content to the output file
# for output formatting, you can try rjust()

output.close()
content.close()
#close the stream
```

Introduction to Docker

What are containers?



- **Containers** are a method of operating system virtualization that allow you to run an application and its dependencies in resource-isolated processes.
- “A standardized unit of software”
- For example: https://www.docker.com/why-docker
- Amazon Web Services, Google Cloud Platform provide this

Why use containers?

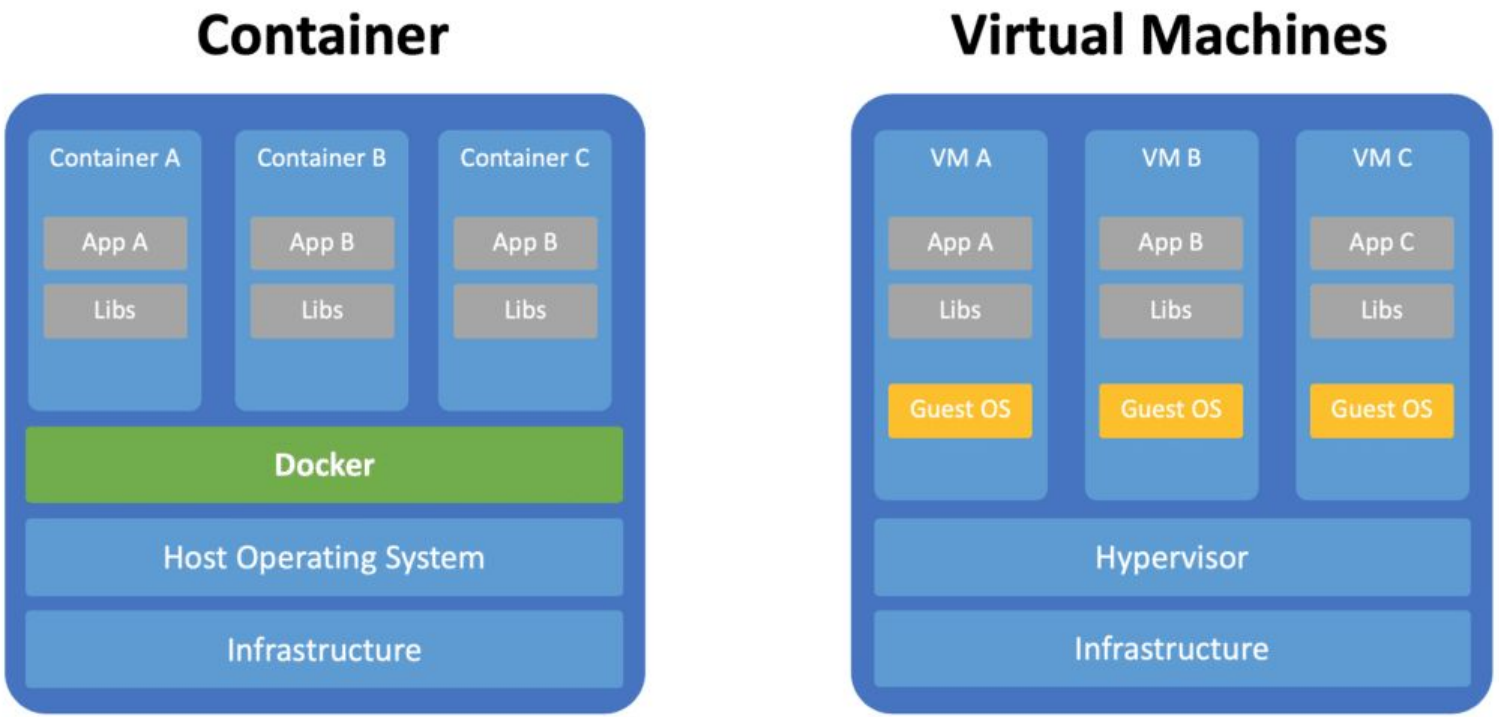
Trouble:

- Path dependency
- Software version
- Environment variables

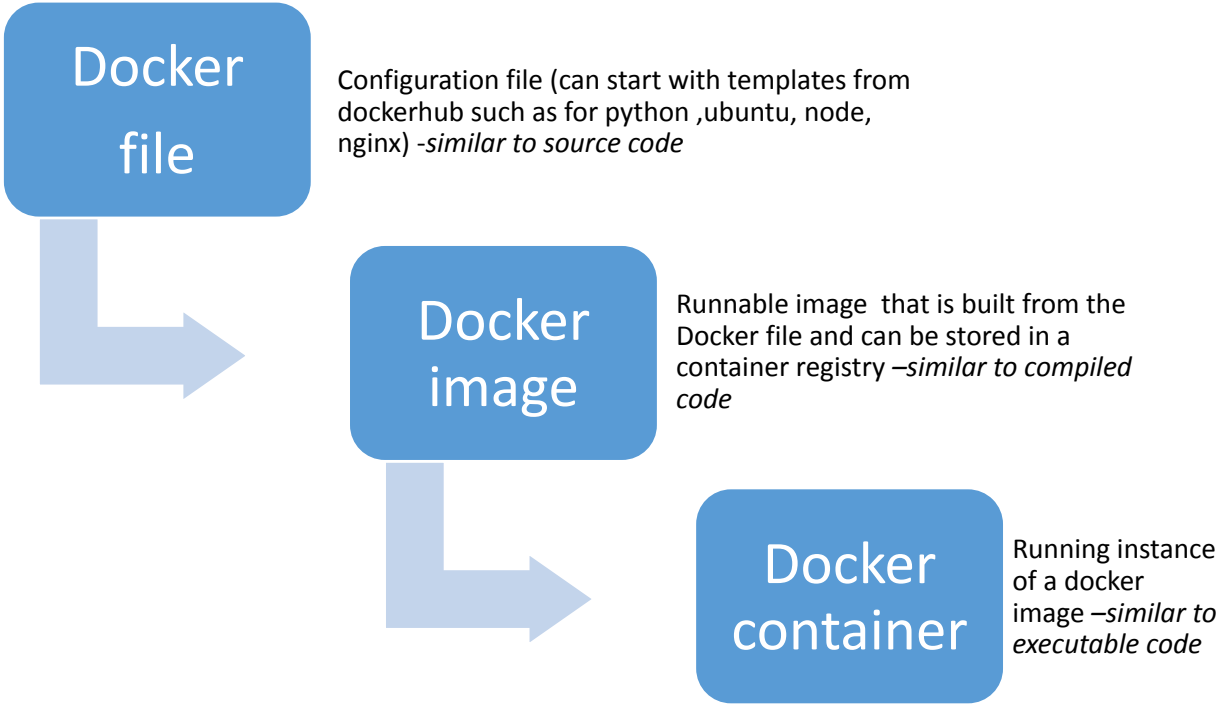
Benefits:

- Encapsulated-portable
- Lightweight-fast
- Demands less resource

Container vs VM



Containers
lightweight compared to virtual machines



Classwork: Dockerize a website and deploy it on the cloud

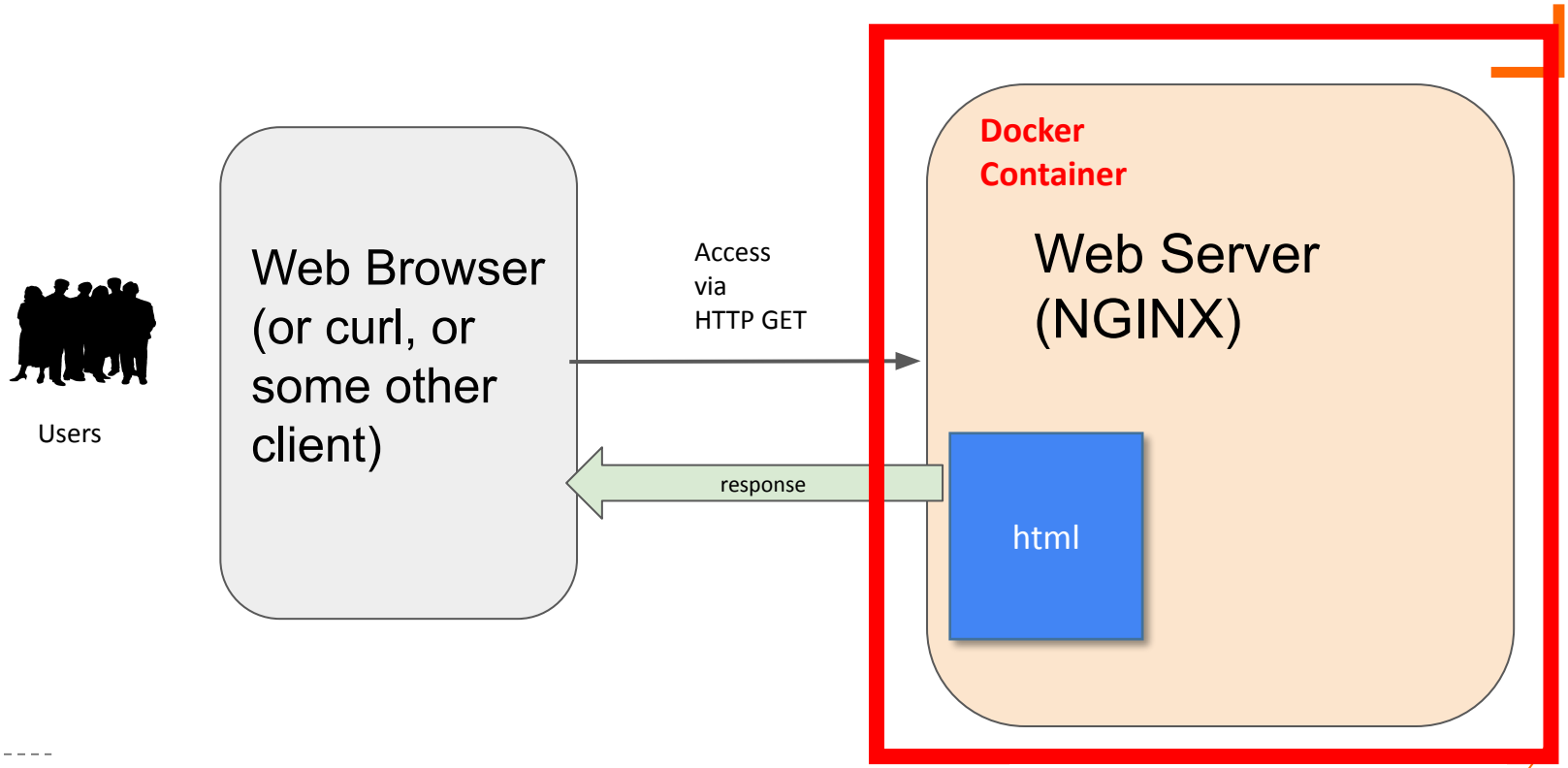
Fork the project on Gitlab

README | Add LICENSE | Add CHANGELOG | Add CONTRIBUTING | Add Kubernetes cluster | Set up CI/CD

Configure Integrations

Name	Last commit	Last update
images	Base code	1 week ago
site-content	Base code	1 week ago
Dockerfile	Base code	1 week ago
README.md	Update instructions to be more specific	1 day ago

Sample Static Web App Architecture



1 Edit the Dockerfile. Dockerfiles can use another image as a base and then build on top of it, on **rlogin** create a Dockerfile based on nginx:latest

```
FROM nginx:latest
COPY …
```

2 Build the docker image. This takes Dockerfile source code and turns it into a Docker image, which is an executable

3 Push image to **gitlab container registry**. This makes it so your image can be used elsewhere without having to distribute source and rebuild.

Classwork will guide you through these steps!

container.cs.vt.edu container

git.cs.vt.edu repository

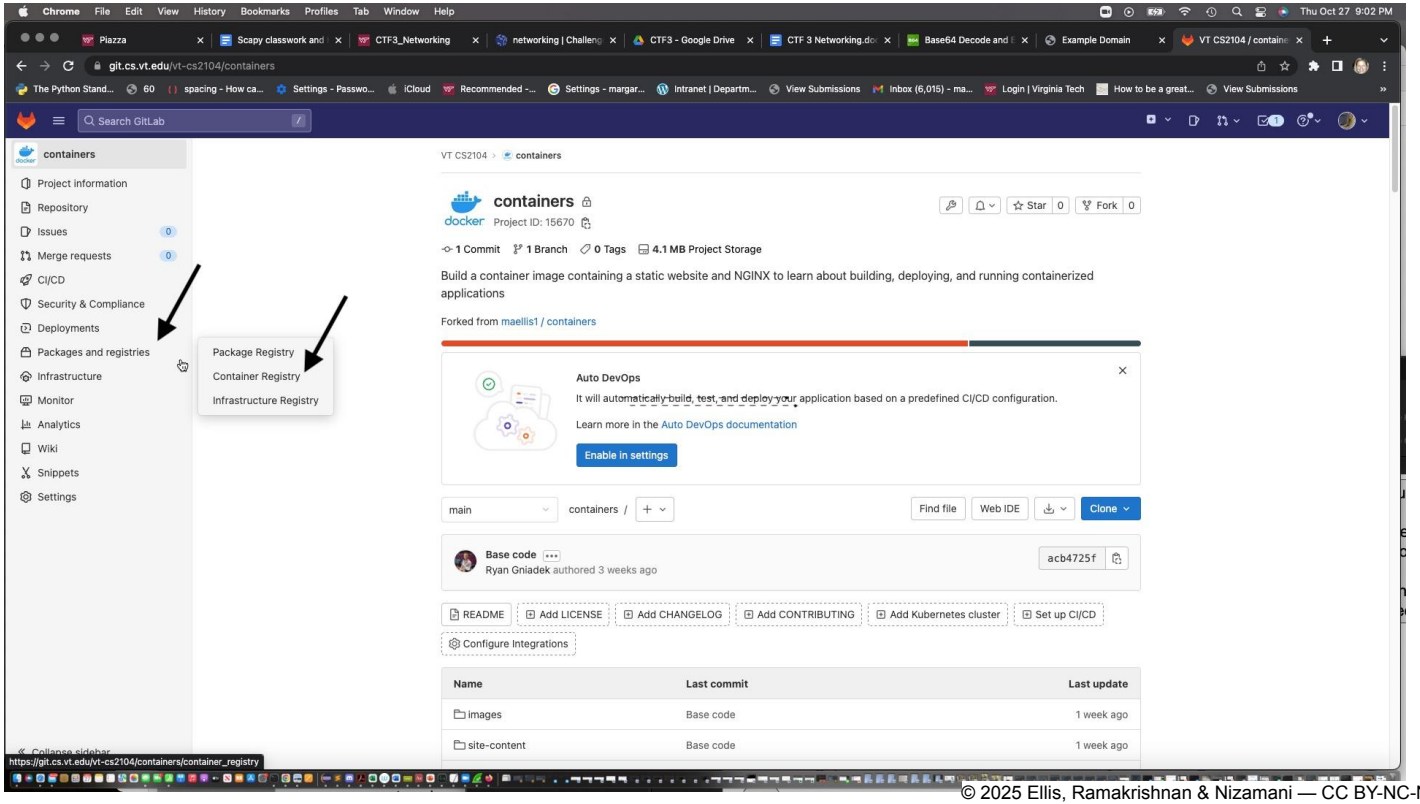
4 Pull docker image to **Azure cloud** hosting.

5 Run docker image on Azure to host website.

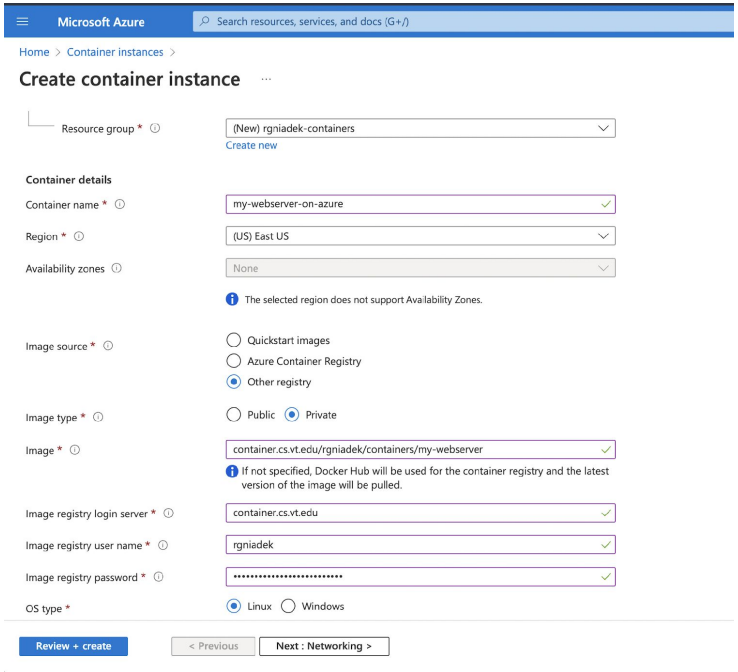
6 Visit your site!

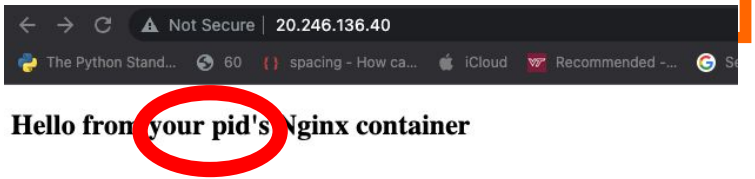
Dockerhub - open source containers
Nginx - web server
Docker image - usable executable
Docker file - source code for Docker image
Docker container - running instance of the Docker image executable

Container registry example in GitLab



Pull container to Azure and host website!





Edit this in your index.html using vi on rlogin