CSCI 3308
Software Dev Methodologies and Tools

Lab -1
January 15th, 2025

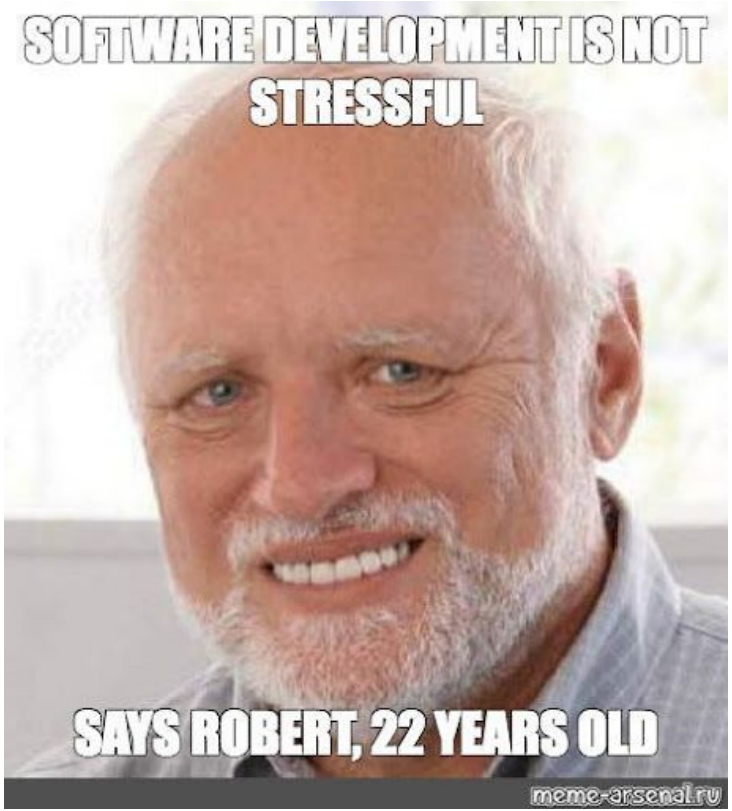
Key Focus

- To learn about technology tools and methods for software development.
- Strong emphasis on industry best practices and professional development.

The best way to learn is by doing it

Concepts we will be exploring

- Unix Shell Scripting
- Agile development methodologies
- Principles of Project Management
- Requirements Definition and Analysis
- Relational Database Design & Construction
- Pair Programming
- Git
- HTML & CSS (“Front-end”)
- NodeJS (“middle layer”)
- SQL Query Language (“Back-end”)
- Documentation of Code
- Web Services
- Cloud Computing
- Code Quality Assessment
- Testing Methods and Strategies
- Containerization, Docker
- Refactoring
- Code / Peer Review


SOFTWARE DEVELOPMENT IS NOT STRESSFUL
SAYS ROBERT, 22 YEARS OLD
meme-arsenal.ru

Course Structure

Lectures: 2 lectures per week

Lab Sessions: 1 per week

Group Project

Timing: Begins in Week 7 of the semester.

Team Composition: Groups of 4-6 students from the same lab section.

Objective: Develop a working software application using the tools and methods covered in lectures and labs.

Collaboration: Work together outside of class to apply the skills learned.

Recitation Format & Rules

- Must be **present** the entire time. We will take **attendance once at the end of class**. Students must give us **24 hours notice** prior to recitation to get an excused absence.
- Each recitation will start with a **short lecture** (~20 minutes), followed by a t**wo-part exercise**.
    - Part A must be completed before lab.
    - Must work on **part B during recitation**. We will check your work at the end. If you make little to no progress, you will not get attendance credit.
    - Must finish **B within one week (before the next lab)**.
    - All work is submitted to a **GitHub repository**.
- **Respect**




KEEP CALM AND
ALWAYS ASK QUESTIONS

Attendance

- If there is an **unavoidable reason** for which you will need to miss a recitation, you are expected to **reach out to your TA or the instructor** to inform them and get approval for your absence.
- If there is a **medical emergency**, you may **email the course staff** at the earliest possible (**no later than 7 days from the day of the lab**) to work out a schedule to complete missed work.
- To receive credit for attendance in case of an excused absence:
    - Complete tasks from Part A for the recitation
    - Show completed work to your TA at their office hour within 2 business(Mon-Fri) days of the missed recitation. If your TA doesn't have an office hour in that timeframe, you can visit another TA's office hours and have them evaluate your work.

UNINFORMED / UNEXCUSED ABSENCE IN A RECITATION WILL RESULT IN 0 FOR THAT RECITATION, IRRESPECTIVE OF WHETHER WORK WAS SUBMITTED FOR IT.

ANY WORK SUBMITTED PAST THE DEADLINE WILL NOT BE ACCEPTED. PLEASE DO NOT REQUEST FOR AN EXTENSION UNLESS THERE IS A MEDICAL OR FAMILY EMERGENCY.

Your TA

Name: Abijith Trichur Ramachandran

Degree: MS in Computer Science

Research: Security, Networks and AI

Work Experience:

Gravity Foundation

SOCIETE GENERALE

GE HealthCare

ARISTA

BOSCH
Invented for life

Office Hours:

- Wednesday: 1:00PM - 3:00PM (ECCS 114E - CSEL)
- Thursday: 8AM - 10AM (Zoom)

Email: abra1867@colorado.edu

ToDo Checklist For Lab Questions

1. Google First: Search for solutions & troubleshooting tips.

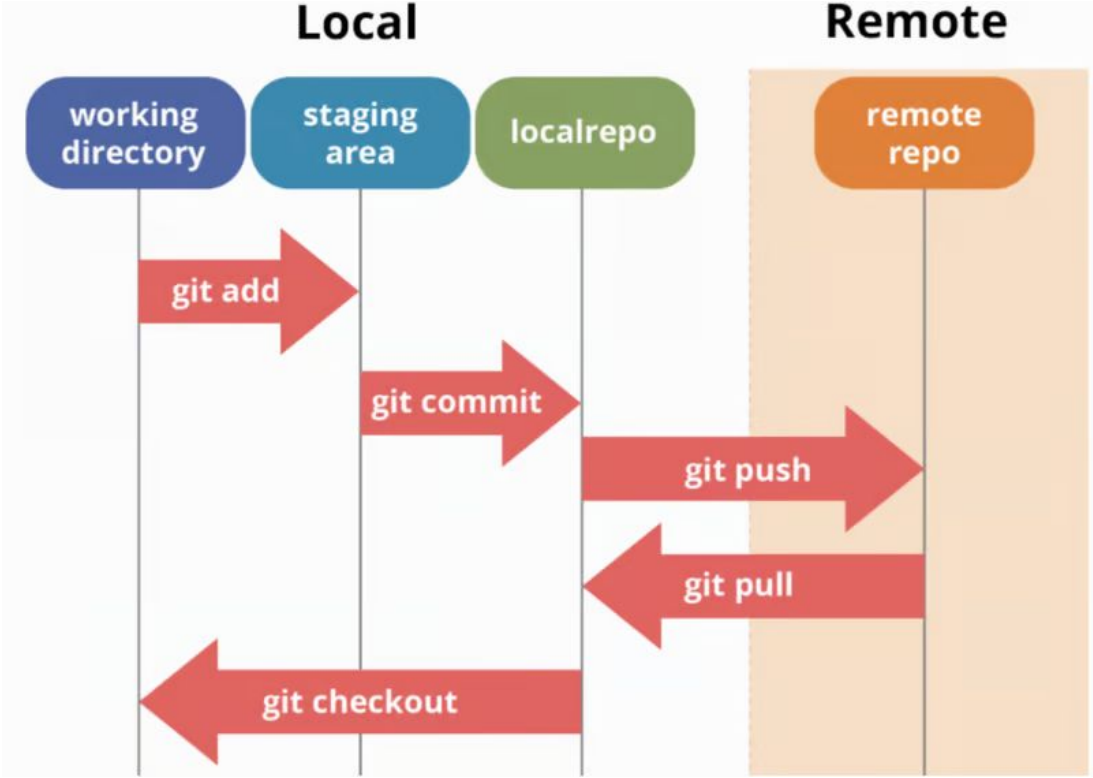
2. ChatGPT Next: Refine or clarify what you found. And focus on understanding what and why the solution or reasoning provided by the GPT is the way it is.

3. Ask your TA or CA / CM

LAB - 1

LAB - 1

Git Workflow

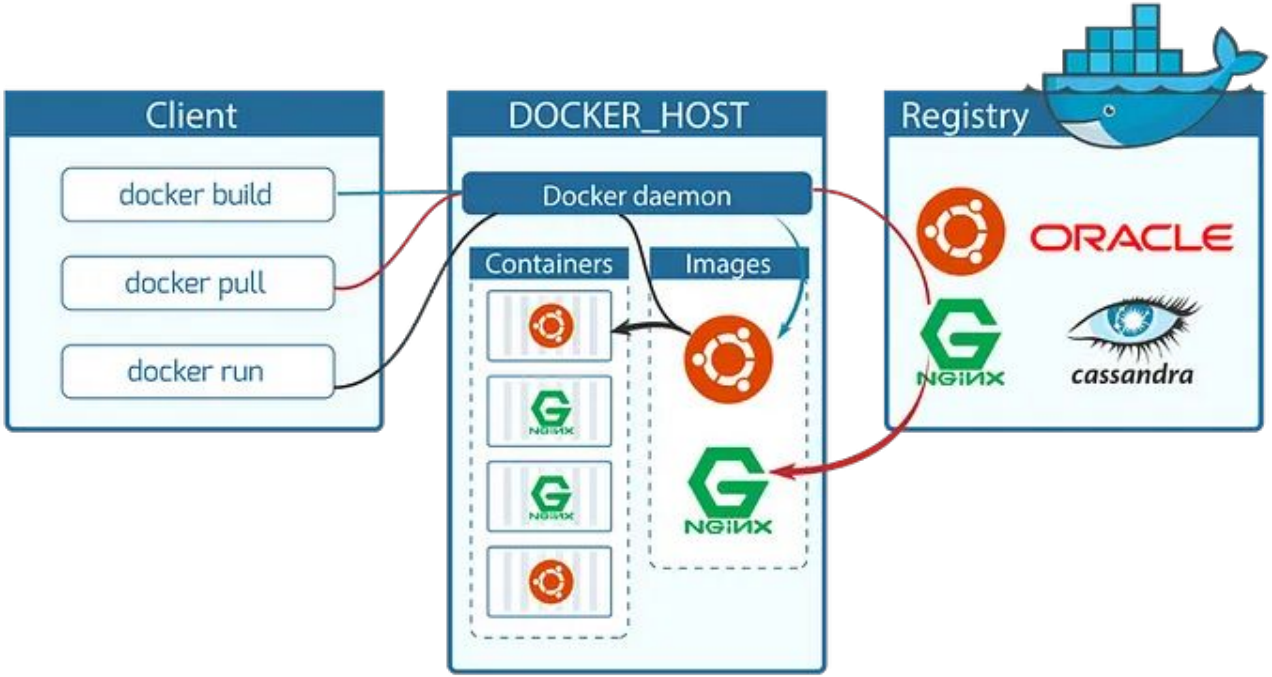


Docker

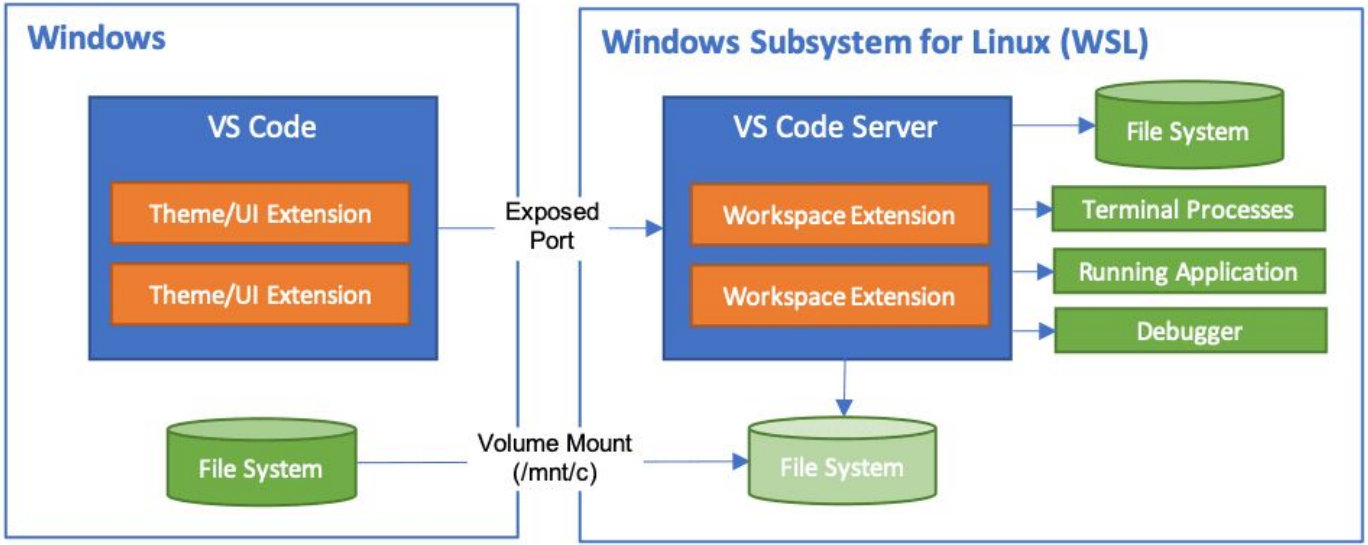
Docker is a *virtualization* software used to create isolated software environments (***containers***)

- Similar to a Virtual Machine (VM), but more lightweight
- Used everywhere in the software industry--from development, to testing, to deployment

DOCKER COMPONENTS



WSL - Windows Subsystem for Linux



Learning Objectives

- Get comfortable with installing software on your computer.
- Set up git and GitHub
- Learn to set up Docker containers
- Understand the configuration settings in a Docker-compose.yaml file
- Learn to run the Docker containers

Lets Get Started

https://cuboulder-csci3308.pages.dev/docs/labs/lab1/

Docker Compose

docker-compose.yaml

```yaml
services:
  web:
    image:
    working_dir:
    depends_on:
      -
    ports:
      -
    volumes:
      -
    command:
  db:
    image:
    env_file:
    expose:
      -
    volumes:
      -
volumes:
```

Connecting to Github

Step 1: Check for Existing SSH Keys

```
> ls -al ~/.ssh
```

Step 2: Generate a New SSH Key

```
> ssh-keygen -t ed25519 -C "<email_id>"
```

You'll be prompted to specify the file where the key will be saved. Press `Enter` to accept the default location (`/home/your_user/.ssh/id_ed25519` or `/home/your_user/.ssh/id_rsa`).

You'll then be asked to enter a passphrase. This adds an additional layer of security, but you can press `Enter` to skip this step.

Step 3: Add SSH Key to the SSH Agent (Optional: Its OK if it doesn't work)

```
> ssh-add ~/.ssh/id_ed25519
```

Step 4: Add Your SSH Key to Your GitHub Account

```
> cat ~/.ssh/id_ed25519.pub
```

Connecting to Github

Add the SSH key to GitHub:

- Log in to your GitHub account.
- Go to your profile settings.
- In the left sidebar, click **SSH and GPG keys**.
- Click **New SSH key**.
- In the "Title" field, add a descriptive label for the new key, like "My Laptop SSH Key".
- Paste your key into the "Key" field.
- Click **Add SSH key**.