Working Remote or Remotely Working?

GAP Containerization solutions with Docker

Zach Newbery

What is Containerization?

- Packaging of applications and their dependencies into **containers**.

- Operating System-based virtualization:
  - Creates multiple containers (virtual units).
  - Share the same host kernel but are **isolated** from each other.

- **Images** can be used to create containers.
  - Containers are **runtime instances** of images

Advantages of Containerization

- Containers are able to run **virtually anywhere**, greatly easy development and deployment.

- Containers can run **uniformly** and **consistently** on **any** infrastructure
  - Useful for sharing/distributing programs

- Containers are **lightweight** – they do not require an operating system for each (as opposed to Virtual Machines).
  - Containers virtualise **just** the Operating System, whereas VMs virtualise the underlying hardware as well.

Introduction to Docker

- Founded in 2013.

- An open-source containerization platform for building, running, and managing containers on servers and the cloud.

- Containers are hosted through a software known as **Docker Engine**.

Using GAP’s Docker Containers

- **Pre-compiled** containers for a multitude of GAP use cases.
- Stored in GitHub Repositories in the **gap-system** workspace:
    - *gap-docker-master*
    - *gap-docker-stable*
    - *gap-docker-stable-4.11*
    - *gap-container*
    - *…*
- Can be downloaded and run using **Docker Engine:**

```
> echo <GITHUB PERSONAL ACCESS TOKEN> | docker login ghcr.io -u <GITHUB USERNAME> --password-stdin
> docker pull ghcr.io/gap-system/gap-docker-master:master
> docker run ghcr.io/gap-system/gap-docker-master:master
```

Aside: Docker Hub v GHCR.io

- All GAP Container Images used to be stored on Dockerhub.

- But, Dockerhub has switched their storage plans to be more costly

- GHCR.io offers a **free** alternative:
  - Container names are formatted as:
    - *ghcr.io/owner/image:branch*
  - Example: *ghcr.io/gap-system/gap-docker:main*

Creating your own Images

- Images require a **Dockerfile** so they can be built.
  - Can be done easily in a personal GitHub Repository
  - Main File Structure Components:
    - FROM
    - RUN (optional)
    - ENV (optional)
  - Can be built locally:

```
> docker build -f Dockerfile .
> docker run <image ID>
```

- Can use GAPs images as base image if need be.
- Deployment can be automized with the **Publish Docker Container** GitHub Action.

GAP CI with Docker Containers

- Containers can also possibly be used for CI tests for individual packages - more efficient than compiling GAP each time.
- Jobs can be run within containers:

```
jobs:
  my_job:
        container:
              image: ghcr.io/owner/image
              credentials:
                    username: ${{ github.actor }}
                    password: ${{ secrets.github_token }}
```

More At: https://docs.github.com/en/actions/using-jobs/running-jobs-in-a-container

Deepnote: Collaborative Notebooks

- Similar to Jupyter Notebooks
  - But **collaborative** like Google Docs

- Allows real-time interaction with team mates whilst maintaining an Jupyter-like environment.

- Lots of other useful features – input blocks, SQL blocks, integrated terminal etc.

GAP in Deepnote

- Currently, there is a pre-compiled container that allows GAP usage in Deepnote:
  - Hosted under: *gap-system/gap-docker-deepnote*

- Can be used by adding the image within Deepnote.

- For more information, you can refer to a post I wrote:
  - https://community.deepnote.com/c/custom-environments/using-gap-in-deepnote