

A primer on the UoE EIDF Cluster, including intros to Kubernetes and Docker

Antreas Antoniou



Agenda

1. The EIDF Informatics Community - Slack and Github
2. What is Docker? How does it relate to conda and pip?
3. Demonstrator of a docker container – intro to dockerfile, building, running and pushing images
4. What is Kubernetes? How does it relate to Slurm?
5. The EIDF Cluster - Overview
6. Creating an account for the EIDF cluster and applying for access
7. First log in, setting up ssh keys, second log in onwards.
8. How does one run Kubernetes Jobs using yaml manifests
9. Doing this via Python – kubejobs
10. Good Cluster Etiquette
11. Overview of current state of the cluster, the good and the bad
12. Current Problems with EIDF
13. Submitting Tickets

The EIDF Informatics Community - Slack and Github

If there is one thing that you take away from this whole tutorial – it should be this:

1. Community engagement and coordination are keys to ensuring that your experience using the cluster is the best and most productive it can be, as well as facilitates the continued improvement of the EIDF cluster.
2. Please join our Github and Slack
 - a. [Slack](#)
 - b. [Github](#)

The Links Slide

All the key resources that I will be using and referring to throughout this tutorial. Conveniently placed in a single slide that you can always come back to, ordered from what you are most likely to be using in a daily basis, to the least

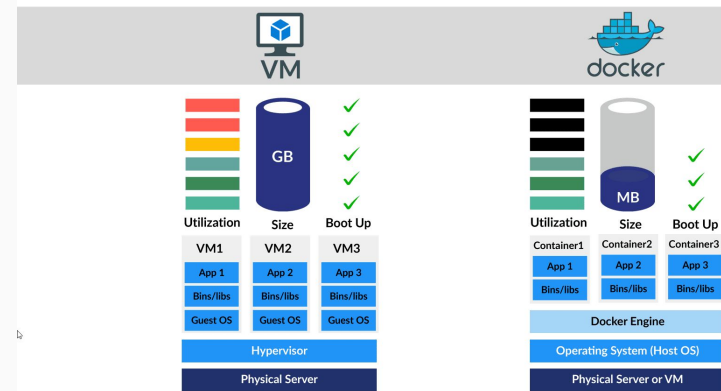
1. [Slack](#)
2. [Python tool for creating, running and managing Kubernetes jobs](#) - Made with EIDF cluster in mind
3. [EIDF Informatics Github](#)
4. [First community meet policies and voting results](#)
5. [EIDF Official Documentation](#)
6. [A Github repo with helpful kubernetes workflow nuggets](#)
7. [A recipe for achieving state-less, fault tolerant experimentation for smooth sailing](#)
8. [Docker build, push cheatsheet](#)
9. [EIDF Portal](#)
10. [Tutorial Repo: Includes all examples we will go through](#)
11. [Template: A more complex/realistic Dockerfile for ML workflows](#)
12. [List of community identified issues that have been submitted to EIDF](#)

Prerequisites and Audience

- Audience: Researchers and students already comfortable with using a Slurm workload manager.
- Aim: To understand and transition to the EIDF GPU cluster, using knowledge of the Slurm cluster as a reference point.
- We will demystify new concepts like Docker and Kubernetes and relate them to familiar terms like Conda + Pip and slurm

What is Docker? How does it relate to conda and pip?

- **Docker** uses containers to package applications with all dependencies, ensuring applications run seamlessly across environments.
- Allows **sudo-level** changes, giving the power to alter the operating system layer and everything above it, ensuring higher environment control and reproducibility.
- **conda** and **pip** are package managers in Python used to handle software packages and environments.
- - **conda** supports packages from any language and manages environments.
- - **pip** is python-specific.
- Docker, conda, and pip collectively ensure environment consistency.
- Docker encapsulates conda and pip environments in containers, ensuring consistent application behavior across systems.



Docker Hands On

Demo Mode

1. Simple Dockerfile
2. Build
3. Run
4. Setup github GHCR for pushing
5. Push

What is Kubernetes? How does it relate to Slurm?

Slurm: Pros and Cons in ML Research Context

Pros:

1. Efficient management of static, long-running jobs: Ideal for long ML training jobs on dedicated resources.
2. Simplicity: Easier for users familiar with Linux and traditional HPC cluster computing.
3. Batch Processing: Helpful in the research environment, where tasks can be queued and executed in batches.

Cons:

1. Environment Setup: Environments must be pre-configured on the headnode/filesystem and given to batch scripts. This doesn't allow for building environments that require sudo access.
2. Limited support for dynamic scalability: Less flexible for on-demand resource provisioning
3. No built-in support for containerized environments: Ensuring software consistency and reproducibility could require additional management efforts
4. Lower synergy with cloud-native ML tools compared to K8s

What is Kubernetes? How does it relate to Slurm?

Kubernetes: Pros and Cons in ML Research Context

Pros:

- Containerization: Ideal for encapsulating the entire research environments, ensuring reproducibility across different systems.
- Dynamic scalability: Provides on demand resource allocation, optimal for variable ML workload.
- Ecosystem: Large open sourced tools and libraries tailored for ML/AI.
- Training stateless models: K8s inherent nature is well-suited for stateless tasks, typical with many ML models.

Cons:

- Complexity: Steeper learning curve and maintenance, may pose a challenge for students or researchers not proficient in DevOps.
- Limited resource efficiency for static, long-running tasks: Often common in ML research.

The EIDF Cluster - Overview

<https://epcced.github.io/eidf-docs/services/gpuservice/>

Creating an account for the EIDF cluster and applying for access

<https://github.com/uo-eidf-cluster-users/eidf-epcc-cluster>

First log in, setting up ssh keys, second log in onwards.

1. Show how to do first log-in, and change password
2. Setup ssh keys
3. Ssh via local terminal

How does one run Kubernetes Jobs using yaml manifests

Demo Mode:

1. Showcase simple yaml for kubernetes
2. Show how it connects to a python script
3. Run the job
4. Observe the logs
5. Recognize problems with this approach:
 - a. YAMLS do not provide the programmer with all valid fields and inputs for those fields
 - b. YAMLS do not allow much logic to be incorporated into them.
6. Solution: Wrap in python package

```
apiVersion: batch/v1
kind: Job
metadata:
  name: pytorch-job
spec:
  template:
    spec:
      containers:
      - name: pytorch
        image: pytorch/pytorch:latest
        command: ["python", "/scripts/script.py"]
        volumeMounts:
        - name: script-volume
          mountPath: /scripts
      volumes:
      - name: script-volume
        configMap:
          name: pytorch-script
          items:
          - key: script.py
            path: script.py
        restartPolicy: Never
      backoffLimit: 4
```

Doing this via Python – kubernetes

<https://github.com/AntreasAntoniou/kubejobs>

Demo Mode:

1. Show the kubernetes github
2. Show a simple script for each of these tasks:
 - a. Running pytorch script on 1 GPU
 - b. Start up a PVC - Explain what a PVC is
 - c. Running pytorch script with a PVC onto which a dataset was downloaded

Good Cluster Etiquette

1. Do not hog machines forever using the 'while true; sleep(60)' trick
2. Be mindful of other people on the cluster, both in terms of their pods and the actual humans
3. Always come to the slack group and ask when you are not sure if you should be doing something.
4. Run small pods that can execute quickly. The fastest and more portable a job is, the better it will be for scheduling and overall group productivity
5. Optional: Make sure your jobs are stateless, and fault tolerant, and can continue when the same command is rerun
6. Report issues on the slack and via tickets to eidf (ticket email and template)
7. Be good to each other

Current Problems with EIDF

1. <https://github.com/uo-eidf-cluster-users/eidf-epcc-cluster/discussions/19>
2. https://github.com/uo-eidf-cluster-users/eidf-epcc-cluster/tree/main/community_identified_problems

Submitting Tickets

<https://github.com/uo-eidf-cluster-users/eidf-epcc-cluster/blob/main/ticket-template.md>

Thank you for listening

For questions, contact me at:

- a.antoniou@ed.ac.uk
- @AntreasAntoniou on the EIDF slack group