Docker

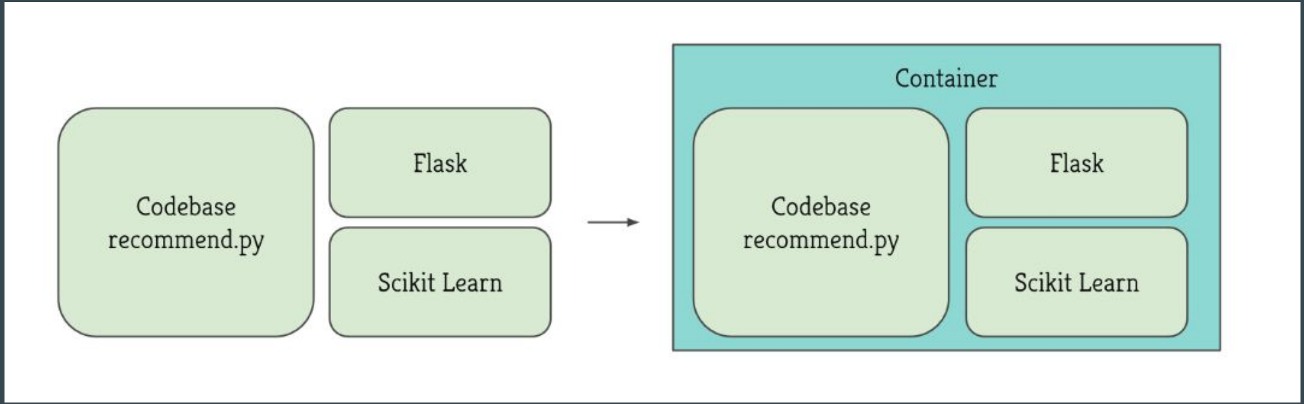
AI Engineering - Recitation 7

Overview

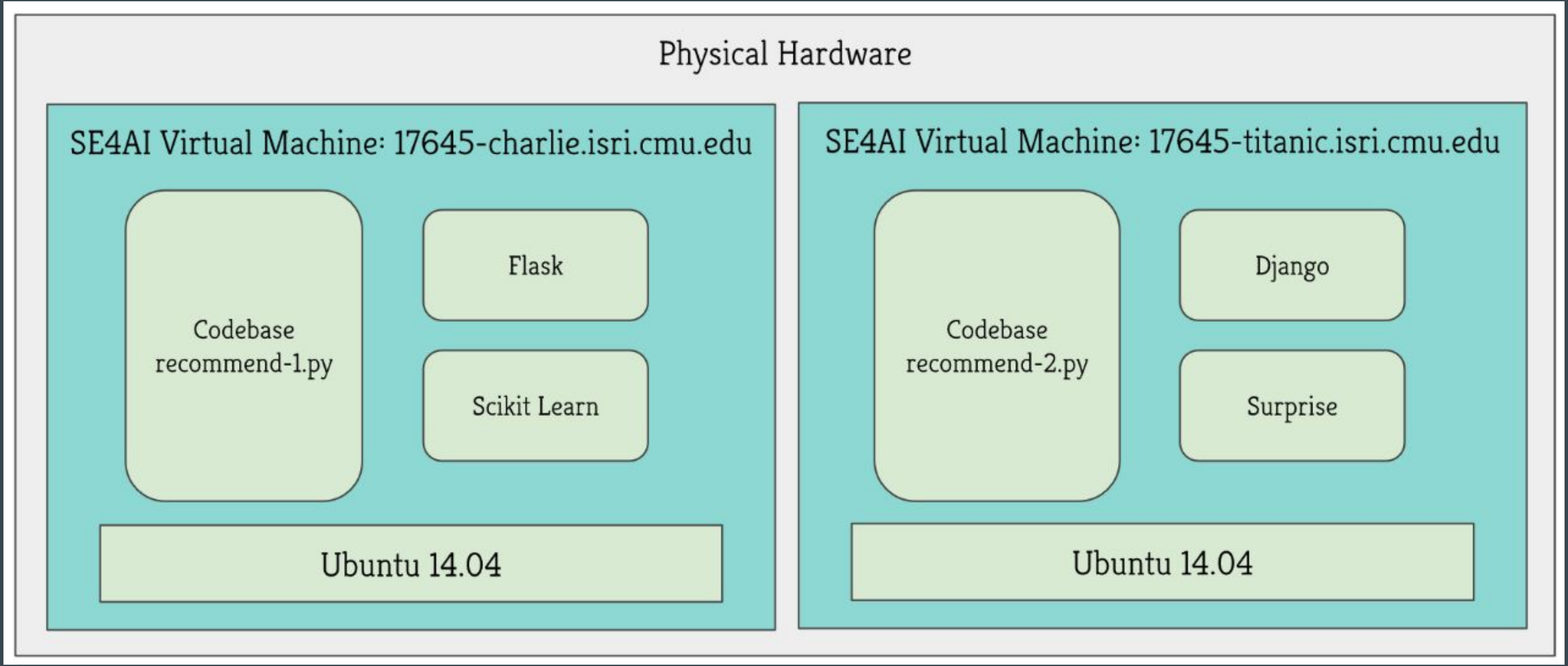
- Containerization
- Virtual Machines vs Containers
- Docker
- Demo - Docker Images and Containers
- A/B Testing using Containers
- Demo - Load Balancers

Containerization

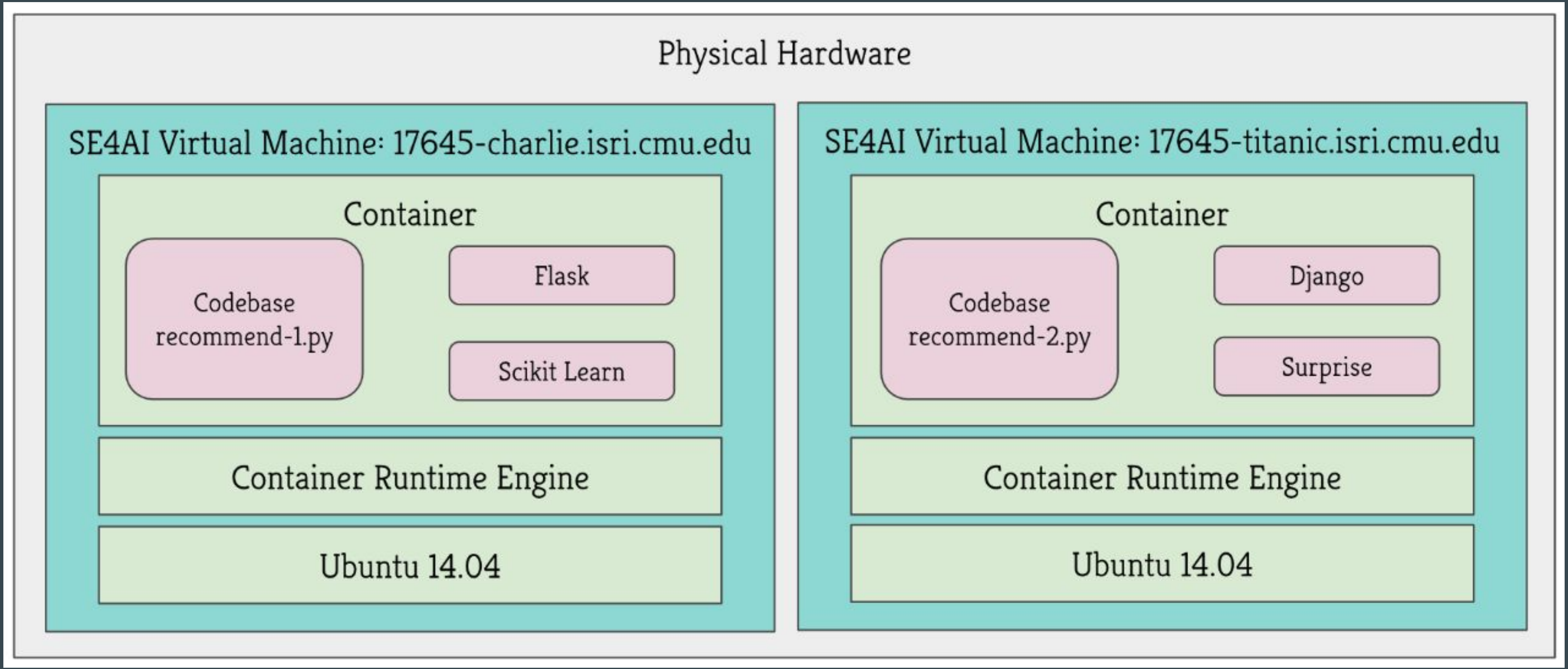
- A way to encapsulate or package an executable such that
  - It is isolated from other executables
  - It is easy to move along with its dependencies
  - It is a standalone executable that can be deployed independently
  - It is lightweight in terms of loading and transporting



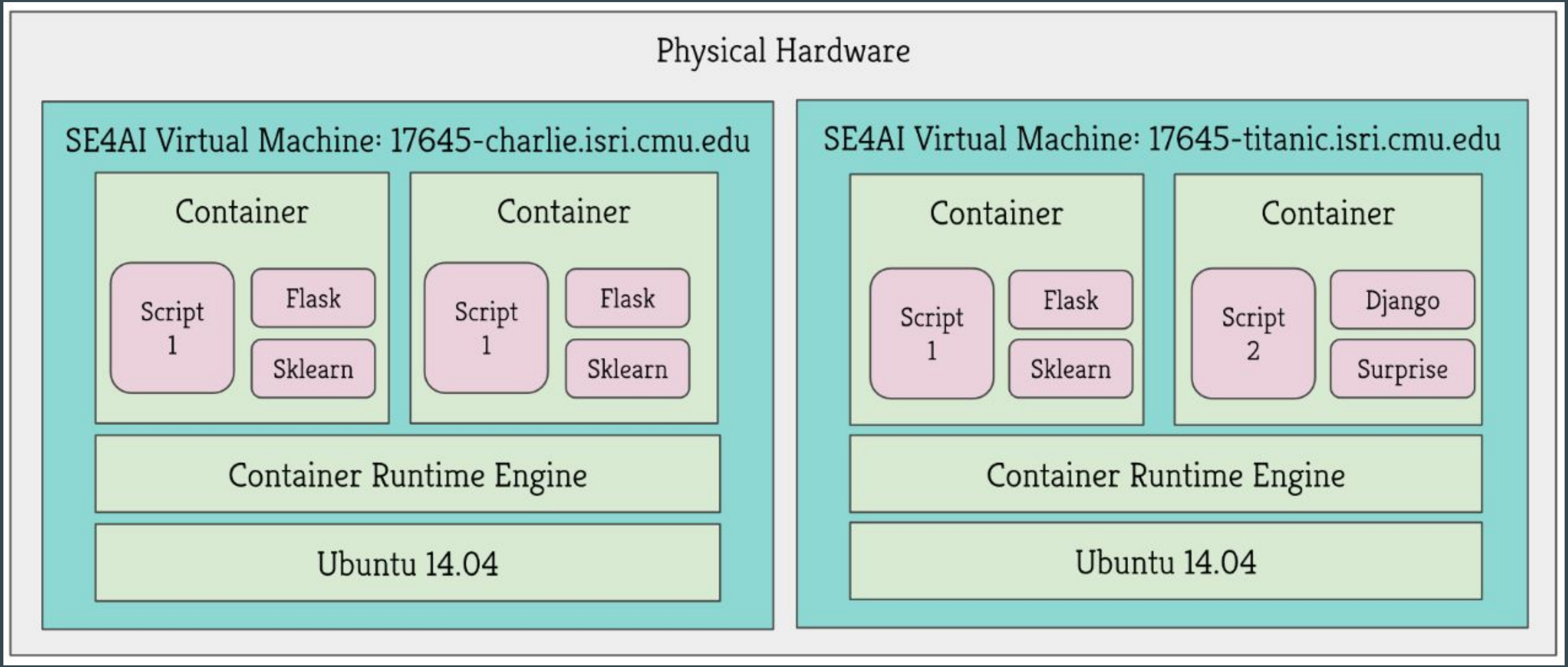
Containerization



Containerization



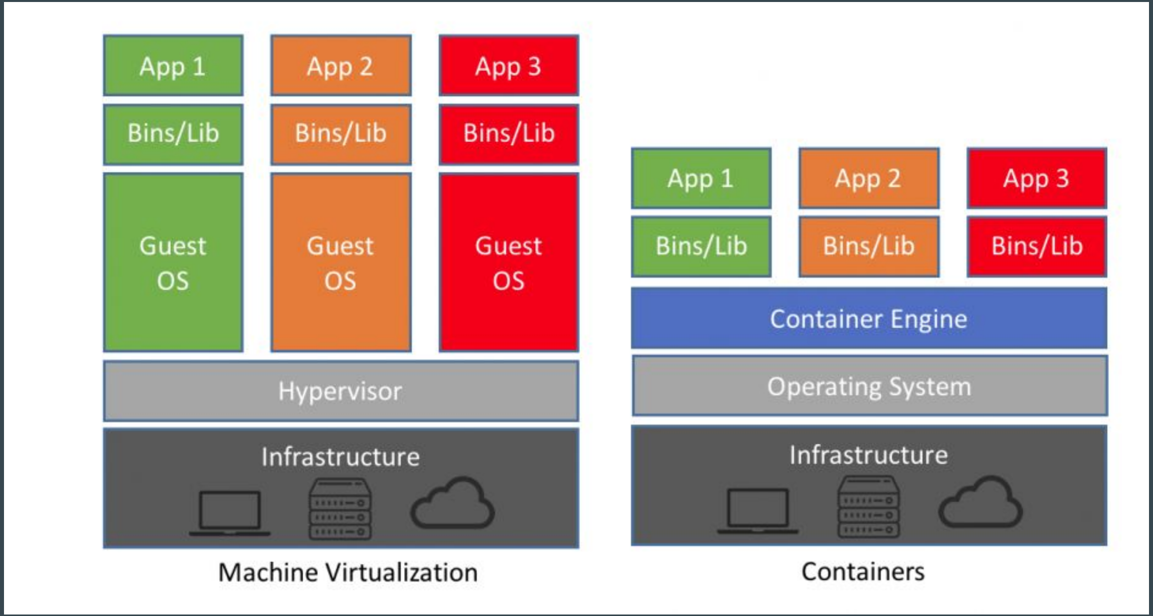
Containerization



Virtual Machines vs Containers

- Similarities
    - Both encapsulate your application
- Differences:
    - Size:
        - Containers are smaller in size as they do not contain the OS
    - Portability:
        - VMs are more portable (OS comes along with the VM)
        - Containers are portable as long as the container runtime supports the same format, ie. the same runtime engine has to be installed in the machine where it needs to run
    - OS:
        - Containers are constrained by the OS; VMs are not
        - Each VM has its own OS; Containers share OS of the host machine
    - State:
        - Containers are stateless by default (can be made stateful, although not recommended)

Virtual Machines vs Containers



Source: https://blog.netapp.com/blogs/containers-vs-vms/

Virtual Machines vs Containers

- Use containers when:
  - You care about the start times of your application (Containers are fast, VMs are slow)
  - Efficiency of resource utilization is of priority (Containers consume less RAM and CPU)
  - You have budget constraints (Docker & Kubernetes are free and open-source)
  - You want to share container images widely (Docker images can be created and shared easily, whereas VM images can be challenging)

- Use VMs when:
  - You are highly concerned with security want to isolate your environment (VMs provide a fully isolated environment by default)
  - You want portability across operating systems (Windows VMs can be deployed on Linux hosts and vice versa; Docker is not as portable)
  - You want to have a rollback feature (VMs can easily go back to a previous snapshot)

Source: https://www.weave.works/blog/a-practical-guide-to-choosing-between-docker-containers-and-vms

Docker

- Platform-as-a-service product
- Uses OS-level virtualization
- Used to deliver software packages called “containers”
- Terminologies:

Image	- Everything that is need to configure a fully operational environment
Container	- A running instance of an image
Dockerfile	- Definition/Spec to create an image
Container Registry	- System to host and distribute images
Container Repository	- Specific physical locations to store related images

Demo - Docker Images and Containers

- Creating an image using a Dockerfile
- Creating a container using the image
- Running containers on different ports on the same machine
- Inspecting container logs

A/B Testing using Containers

- A/B testing is an experiment to compare two versions of a variable to find out which performs better in a controlled environment
- Compare performance of different models using this technique
- How to use containers for this?
  - Deploy different containers each having different models
  - Decide a strategy to route users to each model
  - Have a load balancer to execute this strategy
  - Collect results

Definition source: https://www.analyticsvidhya.com/blog/2020/10/ab-testing-data-science/

Demo - Load Balancers

Outcomes:

- Constructing an efficient load balancer
- Understanding a simple randomizer strategy to route traffic