

# Container Orchestration

Ryan Tolboom

New Jersey Institute of Technology



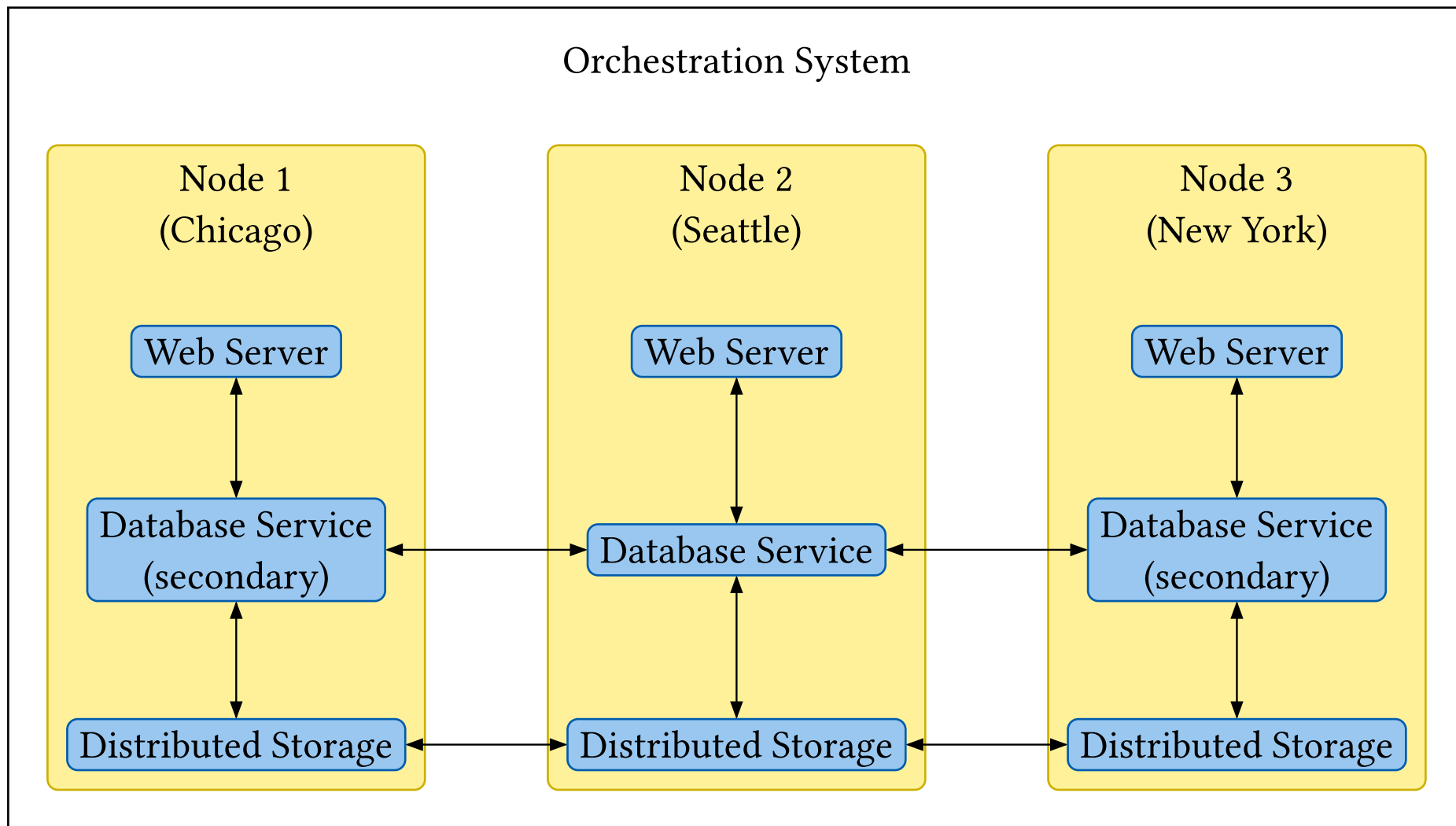
# Purpose



[“Conductor, balground”](#) by [Alex Azman](#) is licensed under [CC BY-SA 4.0](#)

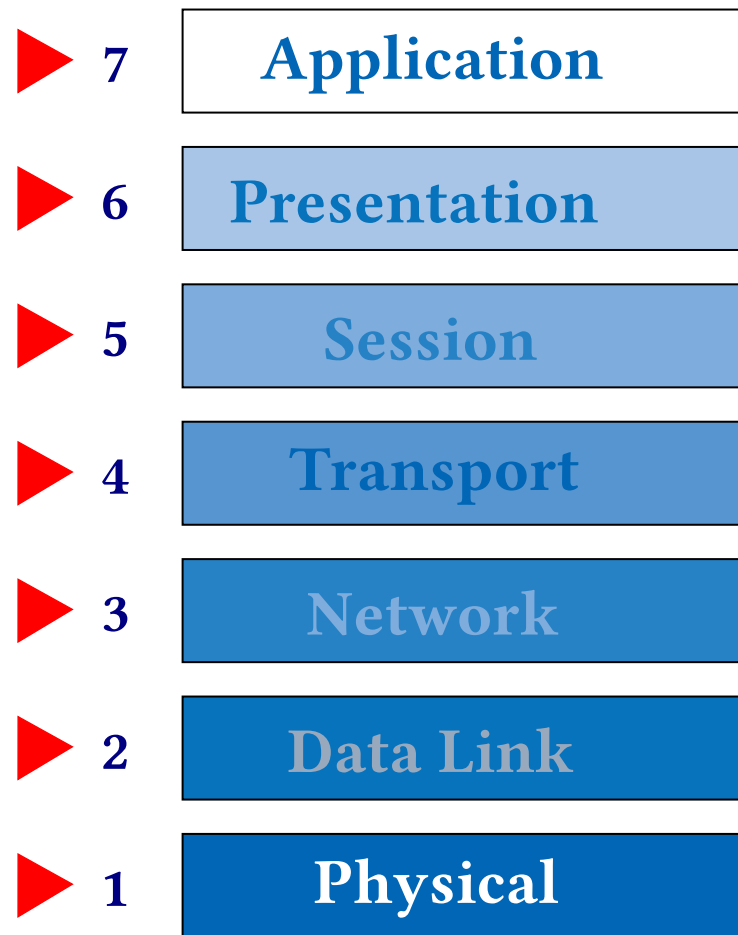
- It is relatively rare to run just one container
- Modern applications are made up of multiple containers working with each other
- Resilient applications are made up of multiple containers running on multiple nodes
- Scalable applications may require containers to be brought up or shut down in response to workload
- Handling monitoring, starting, stopping, and updating containers *is* orchestration.

# It Gets Complicated



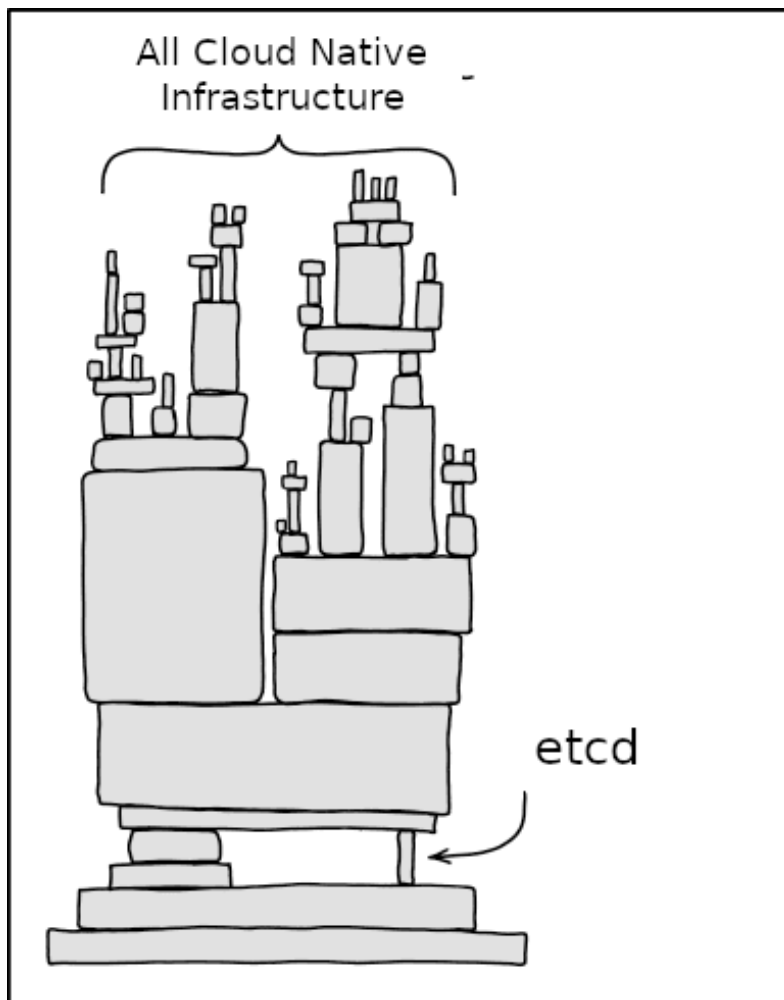
# Networking

- Containers need to be able to talk to each other on a *virtual* network
- This network may or may not traverse *real* networks
- This network may or may not traverse nodes
- [Docker supports many network types](#) but it's still not typically used in production.
- Isolation is a security feature, but use common sense



[osi model](#) is in the public domain

# Service Discovery

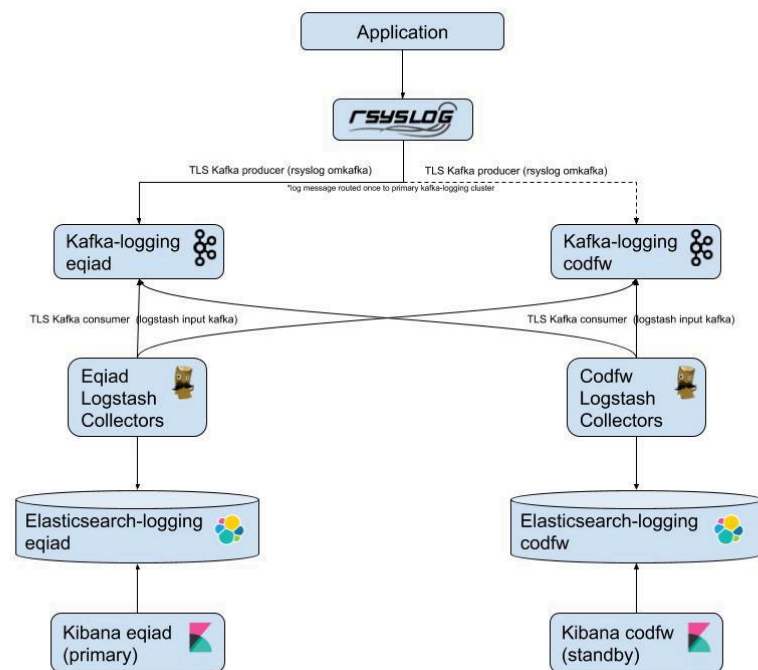


“All Cloud Native Infrastructure” by Josh Berkus is licensed under [CC BY-NC 2.5](#)

- Containers need to be able to find each other
- This can be handled with services: [etcd](#) , [consul](#) , [k8s](#)
- This can be handled through DNS: [Docker Compose](#)

# Logging

- Logs from all of the containers need to be aggregated
- This is trending towards being one of the jobs of the orchestration system: [k8s](#) , [Docker \(Compose / Swarm\)](#)
- You can (and maybe should) still do this externally! Things like rsyslog and logstash still work.



“Logging Pipeline Arch Diag” by [Keith Herron](#) is licensed under [CC BY-SA 4.0](#)

What options are out there?

# Docker Compose

- Uses a `docker-compose.yml` file to describe multiple services
- Allows for many of the Docker options you are familiar with

# Docker Swarm

- Extends the Docker Compose syntax to allow replication and running on multiple nodes

# Kubernetes (k8s)

- *Very popular*
- Uses its own syntax for container orchestration
- Can use multiple container runtimes (docker, containerd, etc.)

# Nomad

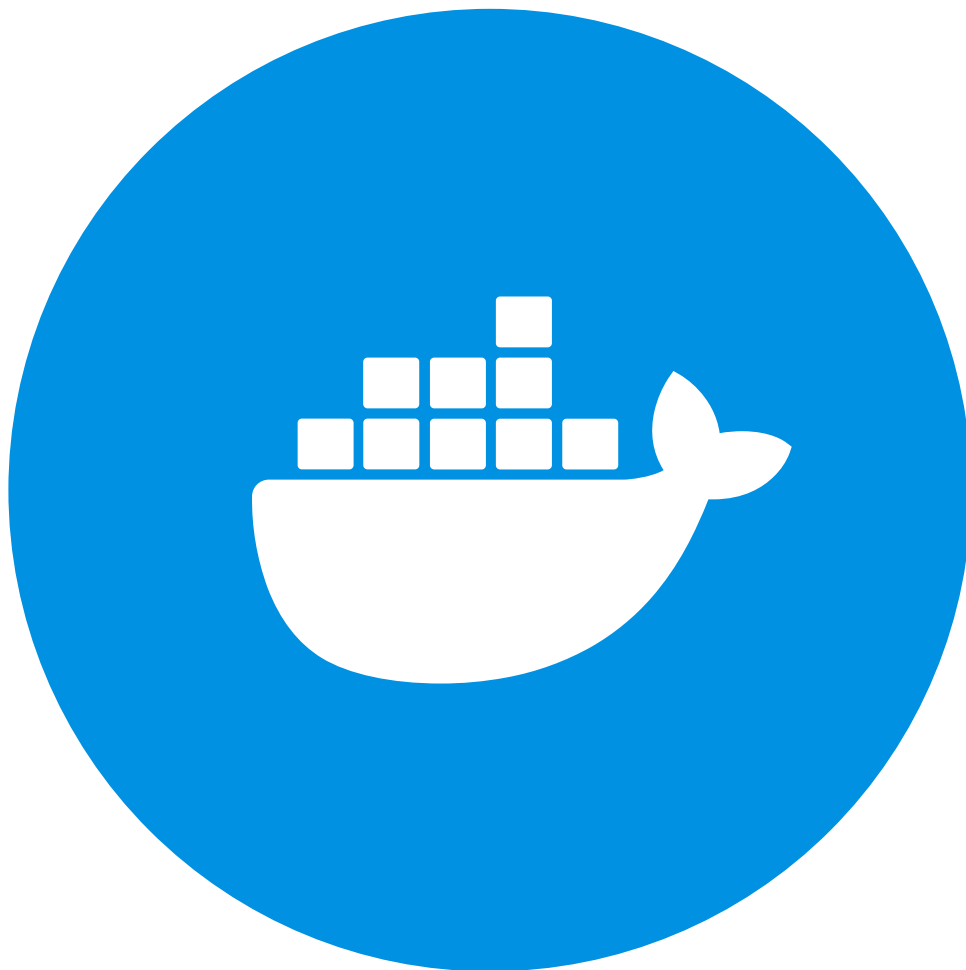
- Simpler than k8s
- Works with both VMs and containers



# Docker Compose Example

```
----  
version '3'  
  
volumes:  
  web_data: external: true  
  
services:  
  app:  
    image: nginx:alpine  
    ports:  
      - 80:80  
    volumes:  
      - web_data:/usr/share/nginx/html:ro  
  app2:  
    image: nginx:alpine  
    ports:  
      - 8080:80  
    volumes:  
      - web_data:/usr/share/nginx/html:ro  
  app3:  
    build: ./app3  
----
```

# Docker Compose Tips / Resources



[Docker Logo](#) is in the public domain under [CC0](#)

- It's YAML, watch your whitespace
- [Get started with Docker Compose](#)
- [Compose file version 3 reference](#)
- Docker Compose expects a `docker-compose.yml` in the working directory
- Commands start with `docker-compose` and are similar to `docker`: `build`, `run`, `exec`, `up`, `down`
- If you change a Dockerfile, don't forget to rebuild!
- Use multiple consoles to make seeing what's going on easier

# Nomad Example

```
job "pytechco-web" {
  type = "service"
  group "ptc-web" {
    count = 1
    network {
      port "web" {
        static = 5000
      }
    }
    service {
      name      = "ptc-web-svc"
      port      = "web"
      provider  = "nomad"
    }
    task "ptc-web-task" {
      driver = "docker"
      config {
        image = "ghcr.io/hashicorp-education/learn-nomad-getting-started/ptc-web:1.0"
        ports = ["web"]
      }
    }
  }
}
```

# Hashicorp Nomad

- Works with multiple container runtimes and VMs
- Uses Hashicorp Configuration Language (HCL) and domain specific language
- Simpler than k8s
- More configurable than Docker Swarm
- Free open source version with paid enterprise and managed service offerings



[Hashicorp Nomad Logo](#) is used under fair use