

Compose and Swarm

Virtualisation

Otago Polytechnic
Dunedin, New Zealand

TO SOME EXTENT, IT JUST WORKS

Make a `docker-compose.yml` file, then run

```
$ docker-compose up
```

There are some issues, however.

FIRST ISSUE, BUILDING

- ▶ We saw that you can specify that Docker build an image in a Compose file.
- ▶ But we also saw that Swarm needs images to be accessible in a registry.
- ▶ So, building in a compose file used for Swarm deployment is right out.
- ▶ Really, when you're deploying to a production setting, you shouldn't be building on the fly anyway.

NEXT ISSUE, VOLUMES

- ▶ We use volumes to let containers work directly with the host file system.
- ▶ But we don't always know what host our containers will run upon.
- ▶ If our container creates volumes, use named ones.
- ▶ If our containers read from volumes, have other containers that populate them, then use a `volumes_from` directive.

ISSUE THREE: DEPENDENCIES

- ▶ We can create dependencies between containers
 - ▶ Explicitly: `depends_on`
 - ▶ Implicitly: `volumes_from`
- ▶ In either case, Docker interprets this to mean that a container must be deployed on the same node as its dependencies.
- ▶ In simple cases this works fine.
- ▶ We need to be aware of this when scaling services. When scaling a service, we may need to scale its dependencies.

THE TRICKY CASE: MULTIPLE DEPENDENCIES

```
version: '2'
services:
  svc_a:
    image: image_a
    depends_on:
      - svc_b
      - svc_c
  svc_b:
    image: image_b
  svc_c:
    image: image_c
```

FINAL ISSUE: NETWORKS

- ▶ We've seen that Docker can set up internal networks that allow containers to communicate and find each other by name.
- ▶ What happens when the containers are on separate hosts?
- ▶ We can get the same result by creating *overlay networks*.

IN COMPOSE

```
networks:  
  foo:  
    driver: overlay
```


FINAL CAVEATS

- ▶ This stuff is highly version-dependent.
- ▶ Documentation is sometimes lacking and what's there can be confusing.
- ▶ Stack Overflow does not care about you.