Composing and Deploying a Cluster of Docker Containers

Walter Moreira | wmoreira@tacc.utexas.edu | @walter_
Advanced Computing Interfaces

What is Docker?

Containers: Isolation from the OS kernel

Docker: A modern container engine

Very new technology  years

It fits people's brains 

Why it is useful?

Containers: modular, portable, reusable 

No installation: all dependencies packaged 

Docker Hub: a repository of images 

Being used in development, deployment, science, …

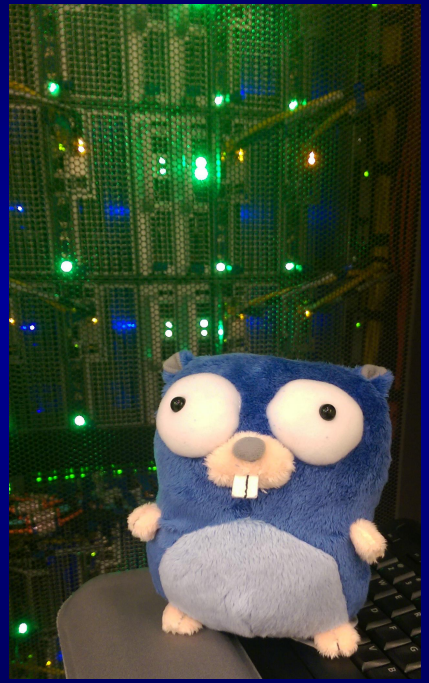
(with caution )

Why it is meaningful for TACC?

Uniform environment for users

Easy to deploy

Reproducible, Shareable

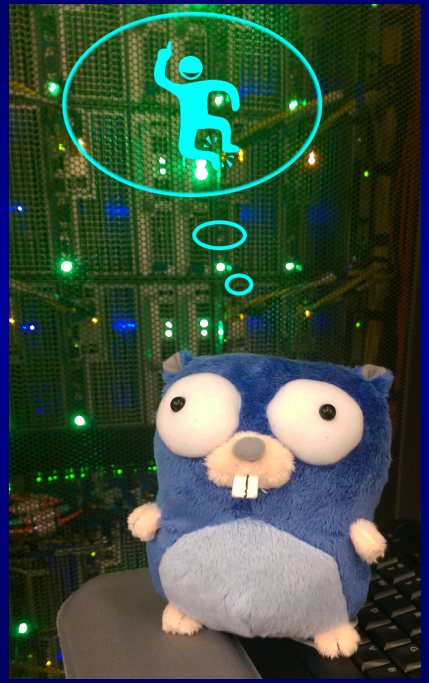


Why it is meaningful for TACC?

Uniform environment for users

Easy to deploy

Reproducible, Shareable



Why it is meaningful for TACC?

Uniform environment for users

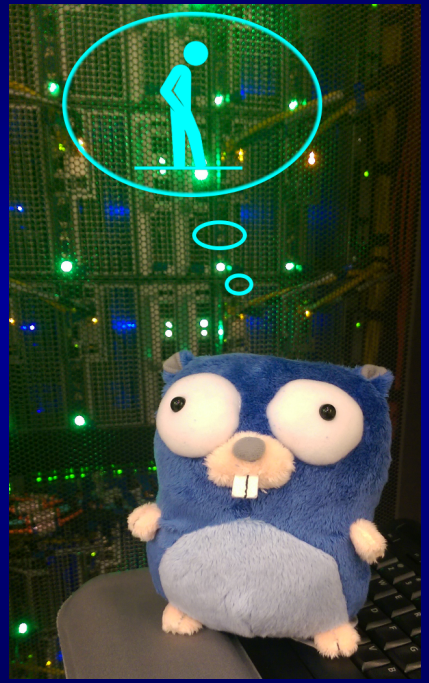
Easy to deploy

Reproducible, Shareable

Not there right now, but making progress

HPC: some difficulties

Services: much clear path



How do you use it?

Run a MySQL server


```
docker run --name foo -e MYSQL_ROOT_PASSWORD=mypasswd -d mysql
```

Run an Nginx connected to the SQL server


```
docker run --name bar -v /content:/usr/share/nginx/html:ro \
   --link foo:mysql -d nginx
```

How do you use it?

Run a MySQL server

```
docker run --name foo -e MYSQL_ROOT_PASSWORD=mypasswd -d mysql
```

Run an Nginx connected to the SQL server

```
docker run --name bar -v /content:/usr/share/nginx/html:ro \
   --link foo:mysql -d nginx
```

Run an IPython Notebook (go to https://localhost)

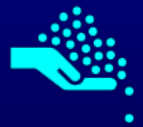
```
docker run -d -p 443:8888 -e PASSWORD=mypasswd ipython/notebook
```

But…, problems!

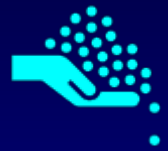
Composition: build images from images 

Service Discovery: many more containers than VM's 

Orchestration: schedule containers across hosts 

Networking, Data Volumes, Supervision, ... 

Many Solutions

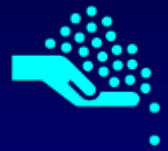


Mesos, Kubernetes, Consul, Brooklyn, Shipyard, Synapse, Maestro, Etcd, Confd, …, so many!

Plus Docker {compose, swarm, machine}

Very rapidly evolving ecosystem

Many Solutions



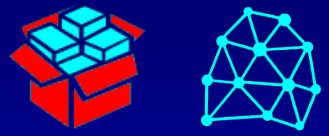
Mesos, Kubernetes, Consul, Brooklyn, Shipyard, Synapse, Maestro, Etcd, Confd, …, so many!

Plus Docker {compose, swarm, machine}

Very rapidly evolving ecosystem

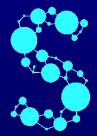
Our View: Serfnode

Fully decentralized, very lightweight,
wrapper for arbitrary images



Provides discovery, cluster membership, supervision

Powered by **Serf** (gossip, SWIM)
Favor Availability over Consistency



Building a Serfnode

- Grab (or build) an useful image (or more) `A`, `B`, `...`
- Write handlers (optional) for events
  [join, leave, fail, custom]
- Write a simple YAML file describing relations
- Build a **new** image `my_serfnode` 

Deploying a Serfnode

Run `docker run -v socket -e PEER=someone -e ROLE=foo my_serfnode`

Deploy by executing on one or more hosts

Containers find themselves via `/etc/hosts` or API

Ansible playbook to kickstart the cluster

Demo

The Future

People waiting to see who will prevail 

Hoping for **lightweight** and **interoperable** solution

Serfnode is used in Adama, Agave

Moving to integrate swarm, machine

Thanks!

Questions?

Links and attributions: github.com/waltermoreira/SEA15