Containerizing the Cloud
with Kubernetes and Docker

Google Cloud Platform Developer Roadshow - 2014
What is a container?

• Lightweight Linux environment
• Hermetically sealed, deployable application
• Introspectable, runnable artifact
• Recently popularized by Docker
Why do developers care?

Static application environment  
= reliable deployments

No stress deployment and update

Repeatable, runnable artifact  
= portability

Develop here, run there
Pick your cloud solely on its merits

Loosely coupled  
= easier to build and manage

Compose applications from micro-services
Mix in and extend third party services
Everything at Google runs in a container.

- Resource isolation
- Predictability
- Quality of service
- Efficient overcommit
- Resource accounting
We start over 2 billion containers per week.
Why do developers care?

Highly automatable = path to active management

- **Efficiency**: optimized packing, better scaling
- **Performance**: active environment tuning
- **Continuous integration**: easy and reliable
- **Robustness**: active monitoring, self healing
Google cluster management stack

Cluster Scheduler

Scheduled Containers

Node Container Manager

Managed Base OS
Node container management on the Google Cloud

Cluster Scheduler

Scheduled Containers

Node Container Manager

Managed Base OS

Container Optimized VM Image
Standardized Declarative Container Manifest
Container health monitoring and restart
Example with nginx

$ gcloud compute instances create my-nginx-container
   --metadata-from-file google-container-manifest=containers.yaml
   --zone us-central1-a
   --machine-type f1-micro
   --image projects/google-containers/global/images/container-vm-v20140522

version: v1beta1
containers:
  - name: www
    image: nginx
    ports:
      - name: http
        hostPort: 8080
        containerPort: 80
Example with nginx

version: v1beta1
containers:
  - name: www
    image: nginx
    ports:
      - name: http
        hostPort: 8080
        containerPort: 80
      - name: https
        hostPort: 443
        containerPort: 443
Example with data sharding

```yaml
version: v1beta1
containers:
  - name: www
    ...  
volumes:
  - name: dataShard
    path: /mnt/shard
    readOnly: true
  - name: dataLoader
    ...  
volumes:
  - name: dataShard
    path: /mnt/output
```

Diagram:
- **www**
- **dataLoader**
- **dataShard**
Host GCE VM

Google Provided
- init / systemd
- sshd
- logging agent
- monitoring agent
- OS

User Experience
- Container Env.
- User App
Cluster container scheduling on the Google Cloud

Cluster Scheduler

Scheduled Containers

Node Container Manager

Managed Base OS

Schedule containers across machines
Replication and resizing
Service naming and discovery
But it takes a community...

Containers are **portable**. The active management framework must be portable too:

- Run on your development machine.
- Run on your cloud.
- Run on Google Cloud Platform.
- Run a different provider or your own hardware.

And it must be enterprise grade.

The community is working to create a framework that runs well everywhere: **Kubernetes**

Microsoft, IBM, Red Hat, Docker, Mesosphere, SaltStack, and CoreOS, have joined the family.
Kubernetes

κυβερνήτης: Greek for “pilot” or “helmsman of a ship”
the open source cluster manager from Google
Kubernetes

- Kubernetes Master/Scheduler

- Container Agent
  - Machine Host

- Container Agent
  - Machine Host

- Container Agent
  - Machine Host

- Container Agent
  - Machine Host

- Container Agent
  - Machine Host

- Container Agent
  - Machine Host

- Container Agent
  - Machine Host

- Container Agent
  - Machine Host
Pods

- Web Server
- Log Roller

Kubernetes Master/Scheduler

Container Agent

Machine Host

Container Agent

Machine Host

Container Agent

Machine Host

Container Agent

Machine Host

Container Agent

Machine Host

Container Agent

Machine Host

Container Agent

Machine Host

Container Agent

Machine Host
Too Many Pods

Kubernetes - Master/Scheduler

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host
Labels

labels: role: frontend

Kubernetes - Master/Scheduler

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host
Labels

Kubernetes - Master/Scheduler

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Google Cloud Platform
Replica Controller

replicas: 4
template:
...  
labels:
role: frontend
stage: production

Kubernetes - Master/Scheduler
Replica Controller

replicas: 1
template:
...
labels:
  role: frontend
  stage: production

Kubernetes - Master/Scheduler

Container Agent
  Machine Host

Container Agent
  Machine Host

Container Agent
  Machine Host

Container Agent
  Machine Host

Container Agent
  Machine Host

Container Agent
  Machine Host

Container Agent
  Machine Host

Container Agent
  Machine Host
Replica Controller

replicas: 3

template:
...

labels:
role: frontend
stage: production

Kubernetes - Master/Scheduler

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host
Service

id: backend-service
port: 9000
labels:
  role: backend
  stage: production

Kubernetes - Master/Scheduler

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host

Container Agent
Machine Host
We’re just getting started

● Clone Kubernetes at: https://github.com/GoogleCloudPlatform/kubernetes

● Check out container VMs at: https://developers.google.com/compute/docs/docs/containers

● Join the discussion on freenode: http://webchat.freenode.net/?channels=google-containers
Summary

1. We’re taking lessons we’ve learned and open sourcing them.
2. Kubernetes is our evolving effort to make cluster management easy.
3. We’re eager to hear from you!