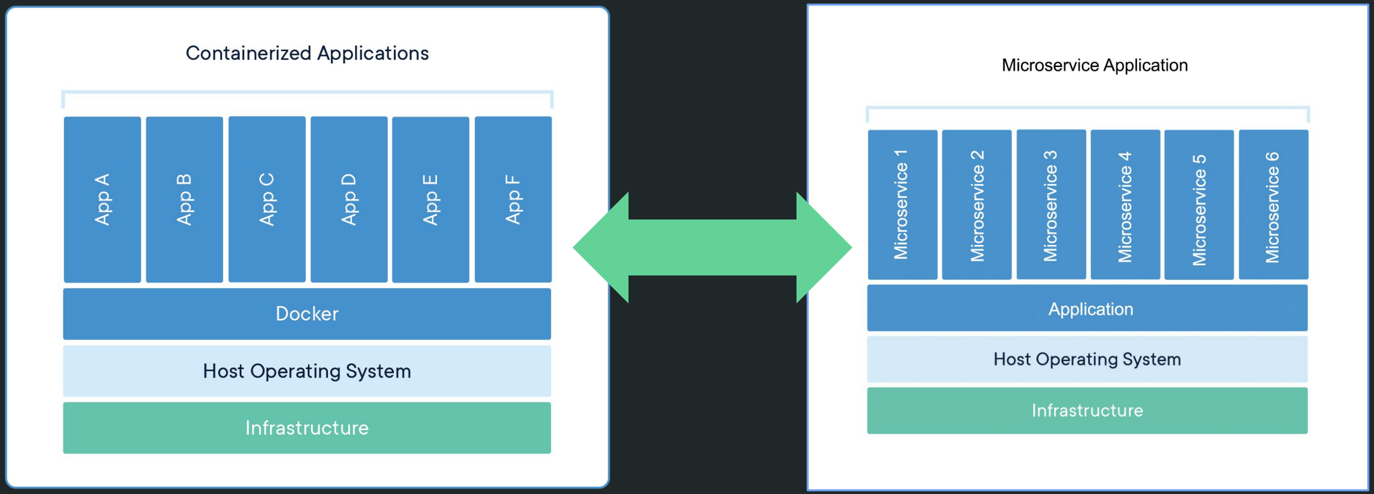
Container-Based Distributed System.

Team: GKE

Rajan Patel

Docker

- Containers
  - OS level virtualization
  - Lightweight as no whole OS image
  - Less memory, network bandwidth requirement, also has version control
  - Process level isolation



Docker

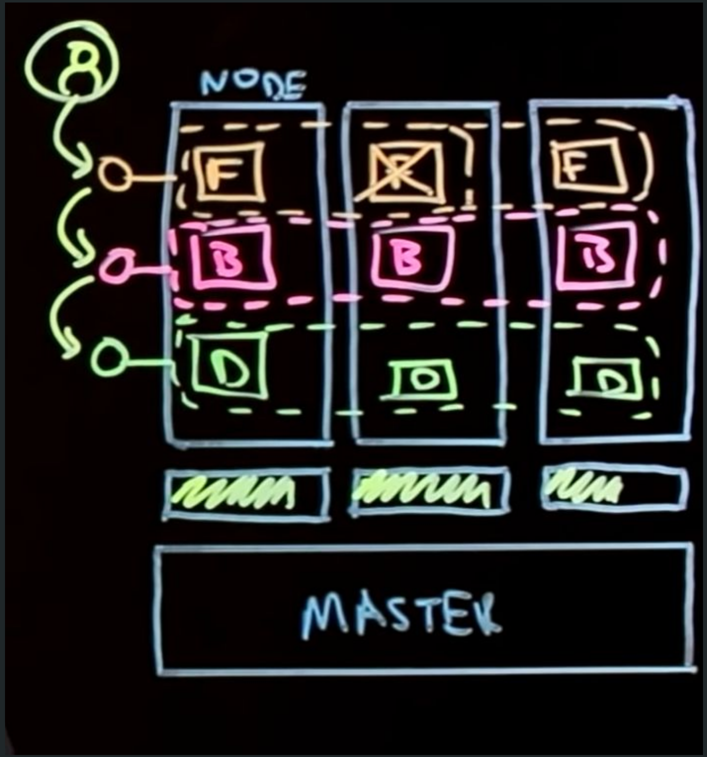
- Docker (allows you to easily deploy and run applications in container)
    - Provides version control, just push the changes
    - Continuous deployment and testing
    - Portability
    - Isolation (resources and configuration)
    - Security (own set of resources, read only mount points)

Running multiple containers

- Example of running 3 services - Frontend, Backend and Database service
- Company wants to
  - Run these services for high availability,
  - Easy deployment and
  - Scale these services to a distributed environment for multiple regions
- Require tool for management, should provide
  - Provide easy deployment,
  - A communication mechanism between services,
  - Scale-ability,
  - Fault tolerance for the system...

4 key advantages of orchestration platform

- Deployment
- Scaling
- Network
- Insight



Container orchestration

- Features to look for
  - Installation and Cluster configuration,
  - Scalability,
  - GUI,
  - Auto- Scaling,
  - Updates and rollbacks,
  - Data Volumes,
  - Load-balancing,
  - Logging and Monitoring,
  - Downtime...

Tool we chose for research

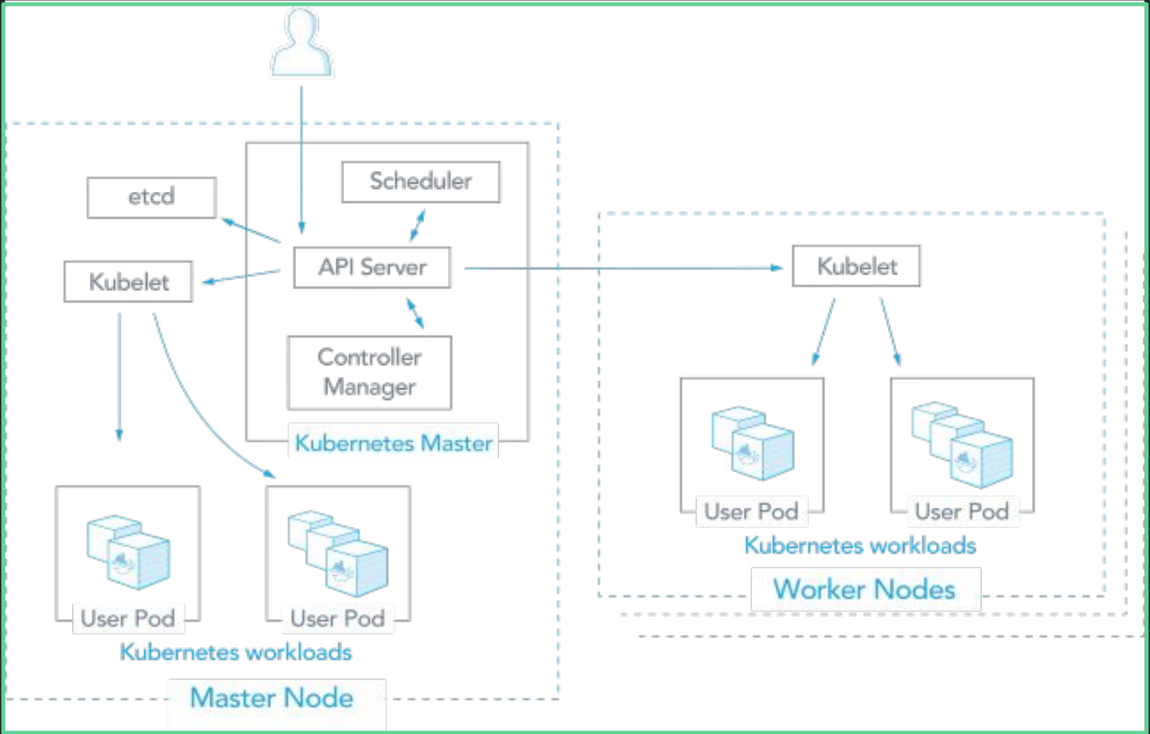
- As per the research papers docker swarm is good for handling 1000+ containers while kubernetes is made to handle more complex architecture with capability of handling 5000+ containers.
- Kubernetes provides a GUI and good cli for dashboard
- While both are great tools for managing containers we chose to move forward with GKE as it provides better features for distributed machine learning.

Kubernetes

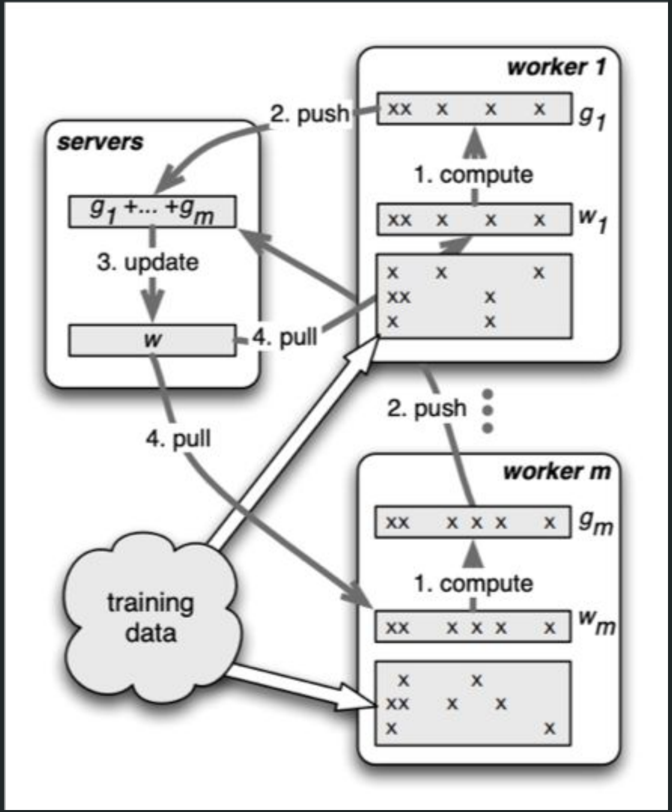
- Kubernetes is a large and complex system for deploying, automating, scaling and operating containers
- Comes with a master node which runs cluster services and several worker nodes which runs your pods (set of containers)
- We feed these cluster services with specific configuration and cluster services deal with running that particular configuration in the infrastructure.

Kubernetes (Architecture components)

- Pods
- Master node
- Replication Controllers
- Schedulers
- Label (artifact)
- YAML (deployment)
  - Server info
  - Port on which pods running
  - Initial state
  - Number of pods
- Worker Nodes
  - Kubelet



Experiment - Distributed Machine Learning.



Experiment - Distributed Machine Learning.

- MNIST dataset.
  - 60,000 - Training set images
  - 10,000 - Test set images.
  - 28x28 - Resolution.
- DNNClassifier
- TensorFlow Estimator: to perform distributed training.

Packaging up model in a Docker containers

Code: Model
Execution: path to run code, Read parameter on network port.
Dependency (23):
Absl-py v0.2.2, astor v0.6.2,, html5lib v0.999, Markdown v2.6.11, mock v2.0.0, numpy v1.14.5, olefile v0.45.1, pbr v4.0.4, Pillow v4.0.0, protobuf v3.6.0, scipy v0.18.1, six v1.11.0, tensorboard v1.8.0, tensorflow v1.8.0, tensorflow-serving-api v1.5.0, termcolor v1.1.0, virtualenv v16.0.0, Werkzeug v0.14.1 ...
Network Connection: NFS and PS

Docker Container

→

Docker Image

learnk8s/mnist:1.0.0

Creating and configuring a Google Kubernetes Engine (GKE) Cluster

1. Create cluster.

```
gcloud container clusters create distributed-tf --machine-type=n1-standard-8 --num-nodes=3
```

3 x Machine:

Standard machine type with **8 vCPUs** and **30 GB of memory**.

2. Create NFS

```
gcloud compute disks create --size=10GB gce-nfs-disk
```

Share **Network File System:**

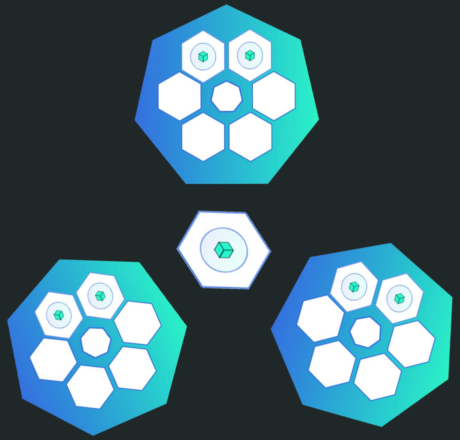
Size: **10 GB**

3. Configure Kubernetes on Cluster

```
ks generate core kubeflow-core --name=kubeflow-core
```

Automate:

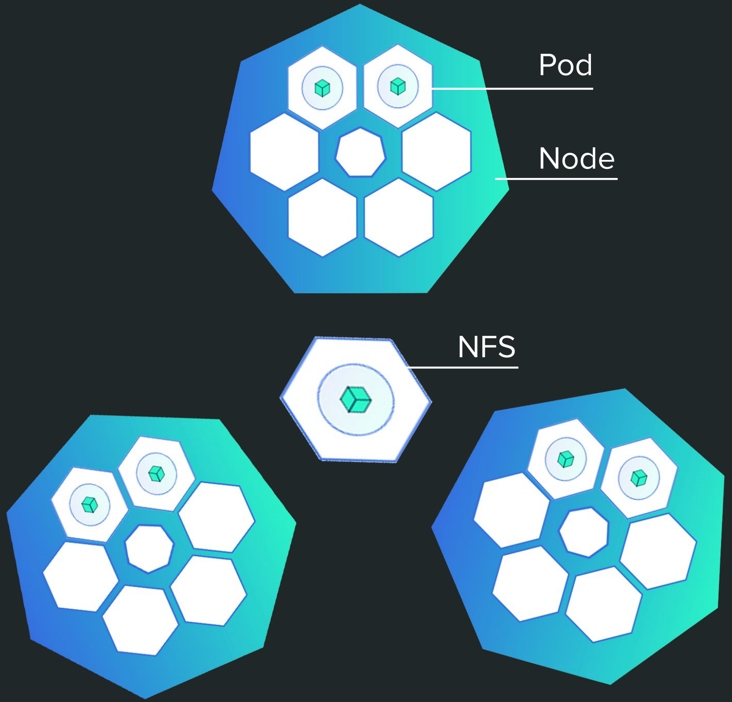
Container deployment, scaling, and management



Schedule job on GKE cluster


```
apiVersion: kubeflow.org/v1alpha1
kind: TFJob
Master: 1
   volumes:
     - name: nfs-volume
   ...
Worker: 5
   volumes:
     - name: nfs-volume
     image : learnk8s/mnist:1.0.0
     ..
PS: 1
   image: learnk8s/mnist:1.0.0
   imagePullPolicy: IfNotPresent
   ..
```


TFJob.yaml



GKE Cluster

Demo

Schedule single job to cluster.

Nodes: 3
Master: 1
Worker: 5
PS: 1

```
schedule_job( TFJob( hidden_layer = 3, learning_rate = 0.01 ) )
```

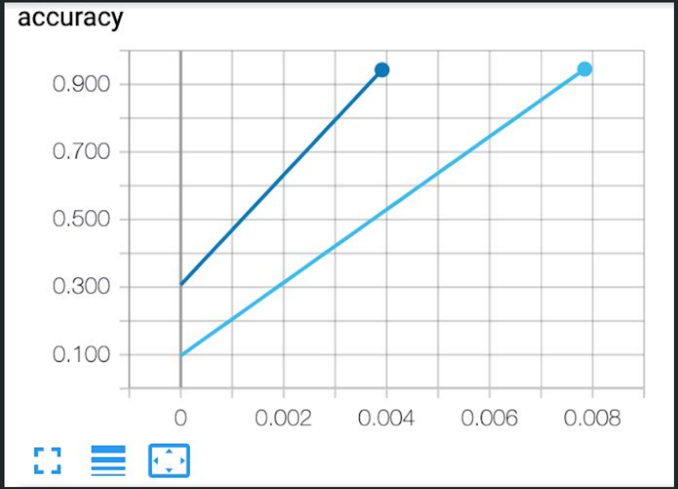
First result.

- Nodes: 3

  Master: 1
  Worker: 1
  PS: 1

- Nodes: 3

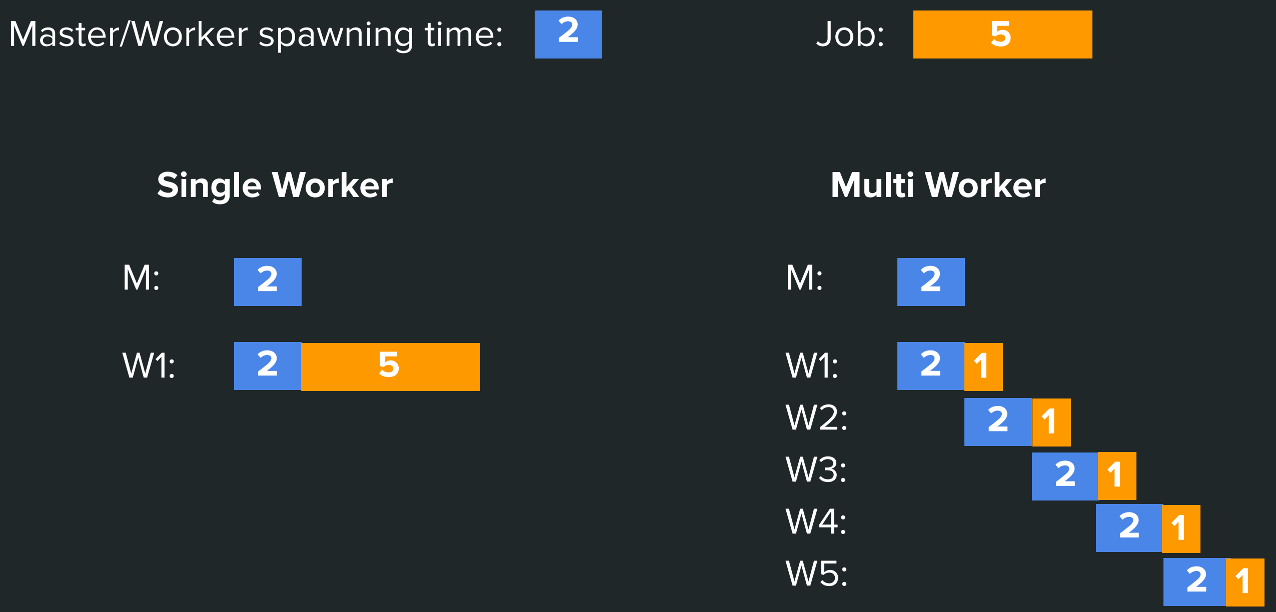
  Master: 1
  Worker: 5
  PS: 1



Name	Smoothed	Value	Step	Time	Relative
vars-1/eval	0.9429	0.9429	102.0	Thu Nov 14, 17:24:53	14s
vars-2/eval	0.9453	0.9453	100.0	Thu Nov 14, 17:30:21	28s

What's wrong? 🤔

Problem cause



Total Time: 7 sec < **Total Time: 11 sec**

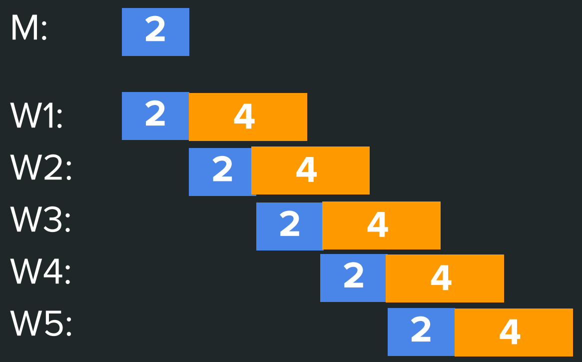
Problem solve

Master/Worker spawning time: 2

Job: 20

Single Worker

Multi Worker



Total Time: 22 sec > **Total Time: 14 sec**

Demo

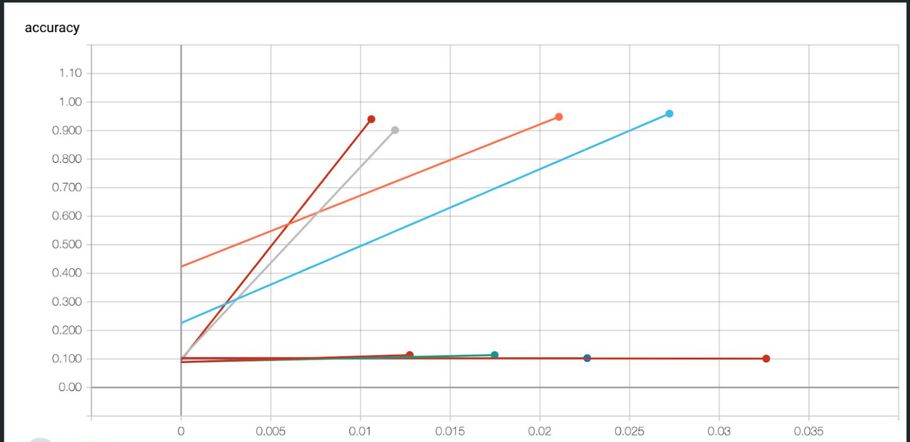
Schedule multiple jobs to cluster.

Nodes: 3
Master: 1
Worker: 5
PS: 1

```
for( hidden_layer : [1,2,3] ) {
    for( learning_rate : [0.1, 0.01, 0.001] ) {
        schedule_job(  TFJob( hidden_layer, learning_rate) )
    }
}
```

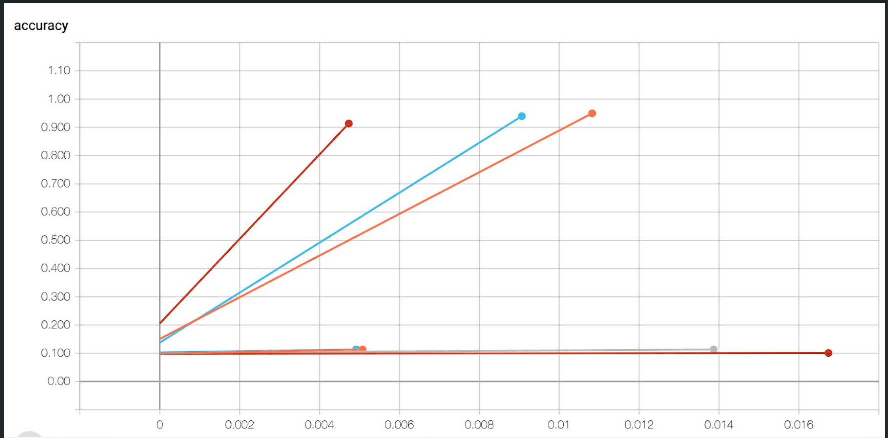
Final results

Nodes: 3, Master: 1, **Worker: 2**



Name	Smoothed	Value	Step	Time	Relative
vars-rate0-layer0-3-2/eval	0.1135	0.1135	100.0	Wed Nov 13, 14:03:58	45s
vars-rate0-layer1-3-2/eval	0.1135	0.1135	103.0	Wed Nov 13, 14:04:25	1m 2s
vars-rate0-layer2-3-2/eval	0.1010	0.1010	100.0	Wed Nov 13, 14:05:43	1m 57s
vars-rate1-layer0-3-2/eval	0.9595	0.9595	100.0	Wed Nov 13, 14:05:29	1m 38s
vars-rate1-layer1-3-2/eval	0.9017	0.9017	104.0	Wed Nov 13, 14:04:58	42s
vars-rate1-layer2-3-2/eval	0.1028	0.1028	100.0	Wed Nov 13, 14:05:49	1m 21s
vars-rate2-layer0-3-2/eval	0.9482	0.9482	100.0	Wed Nov 13, 14:05:41	1m 15s

Nodes: 3, Master: 1, **Worker: 5**



Name	Smoothed	Value	Step	Time	Relative
vars-rate0-layer1-3-8/eval	0.1135	0.1135	106.0	Wed Nov 13, 14:31:18	18s
vars-rate0-layer2-3-8/eval	0.1010	0.1010	100.0	Wed Nov 13, 14:32:07	1m 0s
vars-rate1-layer1-3-8/eval	0.1135	0.1135	102.0	Wed Nov 13, 14:31:25	17s
vars-rate1-layer2-3-8/eval	0.1135	0.1135	100.0	Wed Nov 13, 14:32:14	49s
vars-rate2-layer0-3-8/eval	0.9496	0.9496	100.0	Wed Nov 13, 14:32:07	38s
vars-rate2-layer1-3-8/eval	0.9135	0.9135	106.0	Wed Nov 13, 14:31:54	17s
vars-rate2-layer2-3-8/eval	0.9396	0.9396	100.0	Wed Nov 13, 14:32:34	32s

Final result

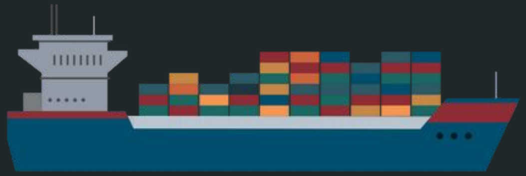
Nodes	Masters	PS	Worker	Max Time	Avg. Time
1	1	1	2	3m 15s	2m 45s
1	1	1	5	2m 05s	1m 45s
2	1	1	2	2m 15s	1m 54s
2	1	1	5	1m 54s	1m 7m
3	1	1	2	1m 56s	1m 32s
3	1	1	5	1m	45s

Advantages of Kubernetes

- Velocity
  - Update the application without a downtime as users expect a constant uptime
- Immutability
  - Artifact created, will not be changed upon user modifications.
- Declarative Configuration
  - Configuration enables the user to describe exactly what state the system should be
- Self healing
  - Continuously take actions to ensure that current state matches the desired state.
- Decoupled components
  - Components separated by api, services, load-balancer etc.

Conclusion

- Applications can be colocated → Fewer machines, resources, cost
- Abstraction of Infrastructure → Portability
- Building decoupled microservice architectures
    - Pods, or groups of containers can group together container images developed by different teams into a single deployable unit.
    - Services that provide load balancing, naming and discovery to isolate one microservice from another.
    - Namespaces provide isolation and access control so that each microservice can control the degree to which other services interact with it.
- Parallel model training and vertical scaling → Improved performance



Reference

- [1] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes,“Borg, omega, and kubernetes,” in ACM, 2016.
- [2] J. Cito, V. Ferme, and H. C. Gall, “Using docker containers to improve reproducibility in software and web engineering research,”in IEEE, 2016.
- [3] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, V. J. Amr Ahmed,J. Long†, E. J. Shekita, and B.-Y. Su, “Scaling distributed machine learning with the parameter serve,” in CMU, 2017.
- [4] B. B. Rad, H. J. Bhatti, and M. Ahmadi, “An introduction to docker and analysis of its performance,” in IEEE, 2017.
- [5] J. Shah and D. Dubaria, “Building modern clouds: Using docker,kubernetes google cloud platform,” in IEEE, 2019.
- GitHub: Distributed Tensorflow on Kubernetes by Eric Ho https://github.com/learnk8s/distributed-tensorflow-on-k8s

Thank you