Simplifying Kubernetes

Simplifying Kubernetes

- K8s specs are complicated
- K8s specs for an app need to be consistent
- multiple solutions
  - Ansible - general software installation and configuration
  - Helm - configure and deploy K8s applications
  - Kubeflow - AI/ML framework and GUI on top of K8s

Want to stick close to plain K8s for control over performance, easy deployment.

Templating

- put the boilerplate text into templates (Jinja2)
- generate actual YAML files by running a Jinja preprocessor
- `kubetpl` is a small Jinja processor with useful K8s templates

Simple Templating with Shell Scripts

We can get simple templates and shared parameters with shell scripts.

In [ ]:

```
cat > variables <<'EOF'
app=bigdata19
subdomain=bigdata19
image=gcr.io/research-191823/bigdata19
EOF

source variables

cmd=uptime
kubectl apply -f - <<"EOF"
apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: $app
spec:
  containers:
  - name: mypod
    image: $image
    command: ["$shell", "-c", "$cmd"]
  restartPolicy: Never
EOF
```

In [ ]:

```
sleep 15
```

In [ ]:

```
kubectl delete pods --all
```

Running a Job (the simple way)

In [74]:

```
kubectl delete job.batch/mytask || true
kubetpl job -c uptime | kubectl apply -f -
```

```
job.batch "mytask" deleted
job.batch/mytask created
```

In [ ]:

```
sleep 15
```

In [75]:

```
kubectl get jobs
```

```
NAME     COMPLETIONS   DURATION   AGE
mytask   1/1           2s         4s
```

In [76]:

```
kubectl logs job/mytask
```

```
 17:00:22 up 11:59,  0 users,  load average: 0.01, 0.05, 0.24
```

In [95]:

```
kubectl delete job/mytask
```

```
job.batch "mytask" deleted
```

Template Generation

In [58]:

```
kubetpl pod -G 1 -c nvidia-smi
```

```
---
apiVersion: v1
kind: Pod
metadata:
  name: mytask
  labels:
    app: bragi-tmb-bigdata19
spec:
  containers:
  - name: mytask
    image: ubuntu:18.04
    resources:
      limits:
        cpu: 1.5
        memory: 1G
        nvidia.com/gpu: "1"
      requests:
        cpu: 1.5
        memory: 1G
        nvidia.com/gpu: "1"
    command:
      - "/bin/bash"
      - "-c"
      - |
        nvidia-smi
    stdin: true
    tty: true
  hostname: mytask
  restartPolicy: Never
```

Shared Parameters

Often, we start related jobs that need to share parameters. The `kubetpl.yaml` file contains these.

In [122]:

```
cat > kubetpl.yaml <<'EOF'
image: gcr.io/research-191823/bigdata19
memory: 4G
cpu: 1
app: bigdata19
subdomain: bigdata19
port:
  - 7880
config_map: files
env:
  - MASTER_ADDR=master.bigdata19
  - MASTER_PORT=7880
EOF
```

In [128]:

```
kubectl delete service/bigdata19 || true
kubetpl service
kubetpl service | kubectl apply -f -
```

```
service "bigdata19" deleted
apiVersion: v1
kind: Service
metadata:
  name: bigdata19
spec:
  clusterIP: None
  ports:
    - port: 7880
      targetPort: 7880
  selector:
    app: bigdata19
service/bigdata19 created
```

Configmap Script

There is also a small script that simplifies creating configmaps.

In [129]:

```
kubefcm files *.py
```

```
-- --from-file=helpers.py=helpers.py
-- --from-file=training.py=training.py
configmap "files" deleted
configmap/files created
```

Server with Templates

In [130]:

```
kubectl delete pod/shards || true
kubetpl pod -n shards -c 'serve-imagenet-shards -b 96 zpub://0.0.0.0:7880' | kubectl apply -f -
```

```
Error from server (NotFound): pods "shards" not found
pod/shards created
```

In [131]:

```
sleep 15
```

In [132]:

```
kubectl get pods
```

```
NAME     READY   STATUS    RESTARTS   AGE
shards   1/1     Running   0          19s
```

In [133]:

```
kubectl logs shards | sed 10q
```

```
serving zpub://0.0.0.0:7880
0 rate 0.000000 msg/s throughput 0.00e+00 bytes/s
10 rate 5.500245 msg/s throughput 7.95e+07 bytes/s
20 rate 5.242535 msg/s throughput 7.58e+07 bytes/s
30 rate 5.165748 msg/s throughput 7.47e+07 bytes/s
40 rate 5.033483 msg/s throughput 7.27e+07 bytes/s
50 rate 4.983581 msg/s throughput 7.20e+07 bytes/s
60 rate 4.947316 msg/s throughput 7.15e+07 bytes/s
70 rate 4.837848 msg/s throughput 6.99e+07 bytes/s
80 rate 4.839921 msg/s throughput 6.99e+07 bytes/s
```

Client with Templates

In [134]:

```
kubectl delete pod/monitor || true
kubetpl pod -n monitor -c 'tensormon zsub://shards.bigdata19:7880' | kubectl apply -f -
```

```
Error from server (NotFound): pods "monitor" not found
pod/monitor created
```

In [135]:

```
sleep 15
```

In [136]:

```
kubectl get pods
```

```
NAME      READY   STATUS    RESTARTS   AGE
monitor   1/1     Running   0          17s
shards    1/1     Running   0          41s
```

In [137]:

```
kubectl logs monitor | sed 10q
```

```
input: ['zsub://shards.bigdata19:7880']
zsub://shards.bigdata19:7880
connected
                  10    5.455 batches/s  523.721 samples/s (batchsize: 96)
                  20    4.868 batches/s  467.284 samples/s (batchsize: 96)
                  30    4.829 batches/s  463.590 samples/s (batchsize: 96)
                  40    4.719 batches/s  453.022 samples/s (batchsize: 96)
                  50    4.899 batches/s  470.292 samples/s (batchsize: 96)
                  60    4.282 batches/s  411.041 samples/s (batchsize: 96)
                  70    4.706 batches/s  451.785 samples/s (batchsize: 96)
```

Training with Templates

In [138]:

```
kubectl delete job/training || true
kubetpl job -n training -G 1 -M 8G -c '
cp /files/*.py .
python3 training.py --tensorcom zsub://shards.bigdata19:7880
' | kubectl apply -f -
```

```
Error from server (NotFound): jobs.batch "training" not found
job.batch/training created
```

In [139]:

```
sleep 10
```

In [140]:

```
kubectl logs job/training
```

```
/opt/conda/lib/python3.6/site-packages/torchvision/io/_video_opt.py:17: UserWarning: video reader ba
sed on ffmpeg c++ ops not available
  warnings.warn("video reader based on ffmpeg c++ ops not available")
Mon Dec  9 17:40:25 UTC 2019; training; root; /workspace; GPU 0: Tesla T4 (UUID: GPU-fd29201b-d663-6
697-b413-a761dceb23c8);
creating resnet50
        0 bs    96 per sample loss 7.35e-02 loading 1.44e-03 training 2.09e-02
```

In [141]:

```
sleep 120
```

In [142]:

```
kubectl logs job/training
```

```
/opt/conda/lib/python3.6/site-packages/torchvision/io/_video_opt.py:17: UserWarning: video reader ba
sed on ffmpeg c++ ops not available
  warnings.warn("video reader based on ffmpeg c++ ops not available")
Mon Dec  9 17:40:25 UTC 2019; training; root; /workspace; GPU 0: Tesla T4 (UUID: GPU-fd29201b-d663-6
697-b413-a761dceb23c8);
creating resnet50
        0 bs    96 per sample loss 7.35e-02 loading 1.44e-03 training 2.09e-02
      960 bs    96 per sample loss 7.36e-02 loading 1.23e-03 training 1.00e-02
     1920 bs    96 per sample loss 7.34e-02 loading 1.13e-03 training 6.34e-03
     2880 bs    96 per sample loss 7.30e-02 loading 1.25e-03 training 5.00e-03
     3840 bs    96 per sample loss 7.32e-02 loading 1.53e-03 training 4.39e-03
     4800 bs    96 per sample loss 7.28e-02 loading 1.54e-03 training 4.18e-03
```

In [143]:

```
kubectl get jobs
```

```
NAME       COMPLETIONS   DURATION   AGE
training   0/1           72s        72s
```

In [144]:

```
kubectl delete jobs --all
kubectl delete pods --all
```

```
job.batch "training" deleted
pod "monitor" deleted
pod "shards" deleted
pod "training-jzzvk" deleted
```

Kubernetes with Templating

Makes using Kubernetes as simple as many job queuing systems:

- start service/server: `kubetpl pod -c ... | kubectl apply -f`
- submit job: `kubetpl job -c ... | kubectl apply -f`
- create service: `kubetpl service ... | kubectl apply -f`
- share files: `kubecfm name files...`