HTCondor in K8s and … K8s in HTCondor

Center for High Throughput Computing

In the Beginning…

In the Beginning…

docker®

Before the Beginning…

CHROOT

UNI
Engine
Grou
DOCTOR
WHO

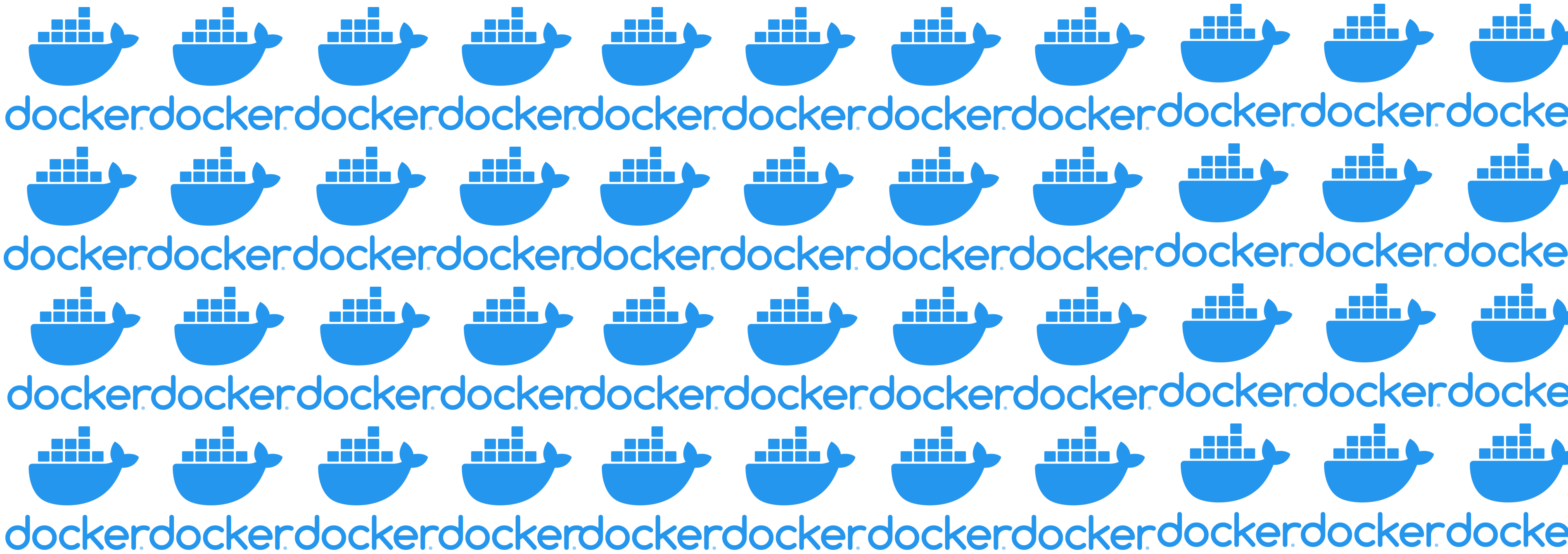
In the Beginning…

docker®

…and it was good

docker®

Until it was an unmanaged mess



Enter Kubernetes (k8s)

Tames the mess of containers on clusters

And their networks

And the storage

Kubernetes

Many users are here or working to getting here

K8s as distributed operating system
in an abstract way – like the schedd is db

Can k8s do some operating system things that HTCondor has do today, but we'd rather focus on HT scheduling?

Summary of Kubernetes

Container Orchestrator

Sets of containers as pods

Sets of pods as deployments, etc.

Manages network and storage

Kubernetes architecture

- Central database holds all objects
  - Pods, services, storage, nodes
  - Note difference from condor – tightly coupled
- All objects described in yaml
- One command kubectl interacts with k8s

Deploying Kubernetes

- Local or in the cloud

- Most deployments are cloud based
  - could be universal cloud interface
  - ephemeral condor pools

- Local deployment is a lot of work
  - many distros

Cloud deployment easy…

```
$ eksctl create –name foo –region us-west-2

$ gcloud container cluster create –name foo
```

Two projects

- Deploying HTcondor in k8s
- k8s gahp

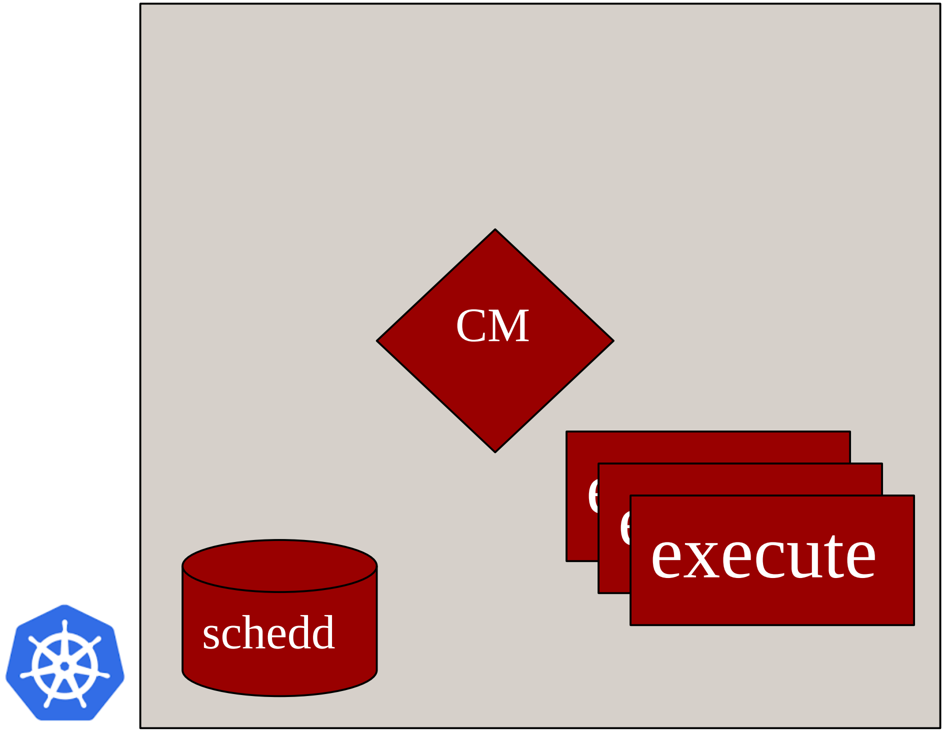
1st integration
Htcondor pool inside k8s

E.g.

Running a whole condor pool as service

Launching daemons as pods

One pod per service



This requires docker images

- So, we built some…
- `$ docker run htcondor/cm`
- `$ docker run htcondor/execute`
- `$ docker run htcondor/submit`

This requires docker images

```
$ docker run htcondor/cm:8.9.6-el7
```

Operating System

HTCondor version

And we'll be releasing with HTCondor release

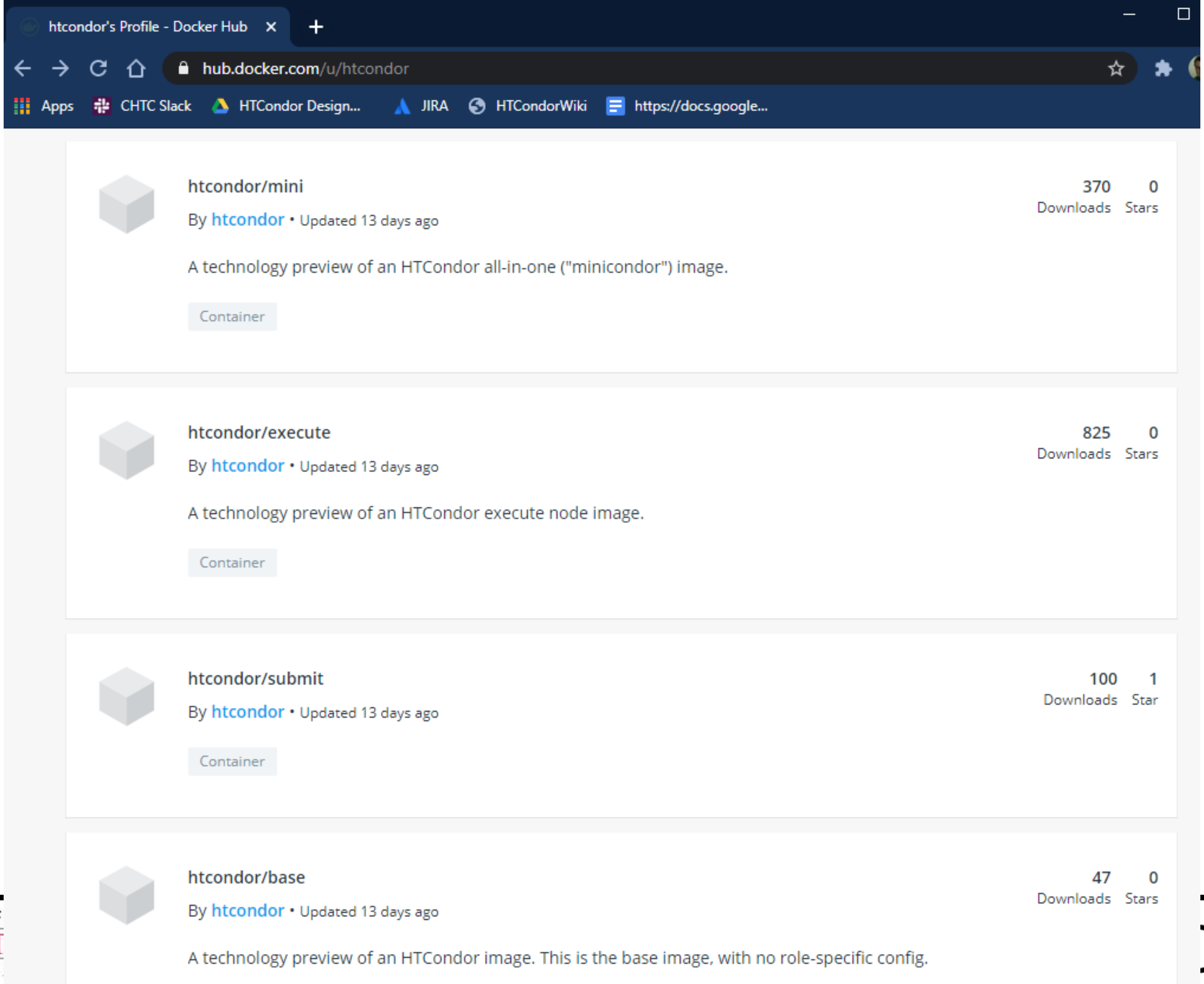
You can rebase on your own distro favs

Note that the OS is coupled with the HTCondor

And a minicondor

- `$ docker run -t -I htcondor/mini:8.9.8-el7`

- Very handy for testing and debugging


htcondor's Profile - Docker Hub

hub.docker.com/u/htcondor

Apps | CHTC Slack | HTCondor Design... | JIRA | HTCondorWiki | https://docs.google...

htcondor/mini

By htcondor • Updated 13 days ago

370 Downloads 0 Stars

A technology preview of an HTCondor all-in-one ("minicondor") image.

Container

htcondor/execute

By htcondor • Updated 13 days ago

825 Downloads 0 Stars

A technology preview of an HTCondor execute node image.

Container

htcondor/submit

By htcondor • Updated 13 days ago

100 Downloads 1 Star

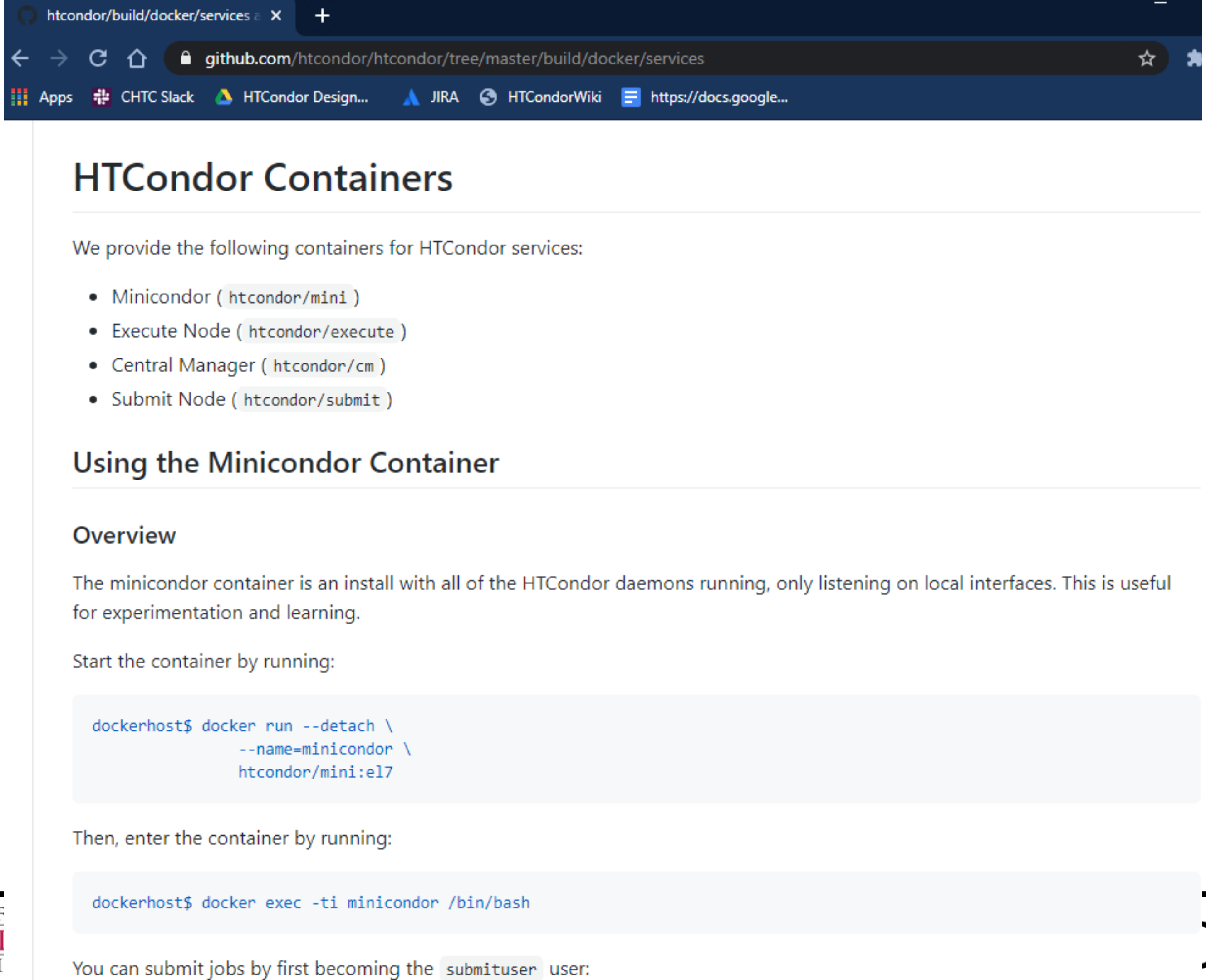
Container

htcondor/base

By htcondor • Updated 13 days ago

47 Downloads 0 Stars

A technology preview of an HTCondor image. This is the base image, with no role-specific config.


htcondor/build/docker/services
github.com/htcondor/htcondor/tree/master/build/docker/services
Apps
CHTC Slack
HTCondor Design...
JIRA
HTCondorWiki
https://docs.google...

HTCondor Containers

We provide the following containers for HTCondor services:

- Minicondor (`htcondor/mini`)
- Execute Node (`htcondor/execute`)
- Central Manager (`htcondor/cm`)
- Submit Node (`htcondor/submit`)

Using the Minicondor Container

Overview

The minicondor container is an install with all of the HTCondor daemons running, only listening on local interfaces. This is useful for experimentation and learning.

Start the container by running:

```
dockerhost$ docker run --detach \
                --name=minicondor \
                htcondor/mini:el7
```

Then, enter the container by running:

```
dockerhost$ docker exec -ti minicondor /bin/bash
```

You can submit jobs by first becoming the `submituser` user:

And some example yaml

```
# This is the service that names the ip address of the collector
# All pods get an environment variable with this ip in it
apiVersion: v1
kind: Service
metadata:
  name: condor
spec:
  selector:
    htcondor-role: cm
 ports:
   - protocol: TCP
     port: 9618
     targetPort: 9618
---
# This is pod yaml to describe the single htcondor central manager
apiVersion: v1
kind: Pod
metadata:
```

Demo time…

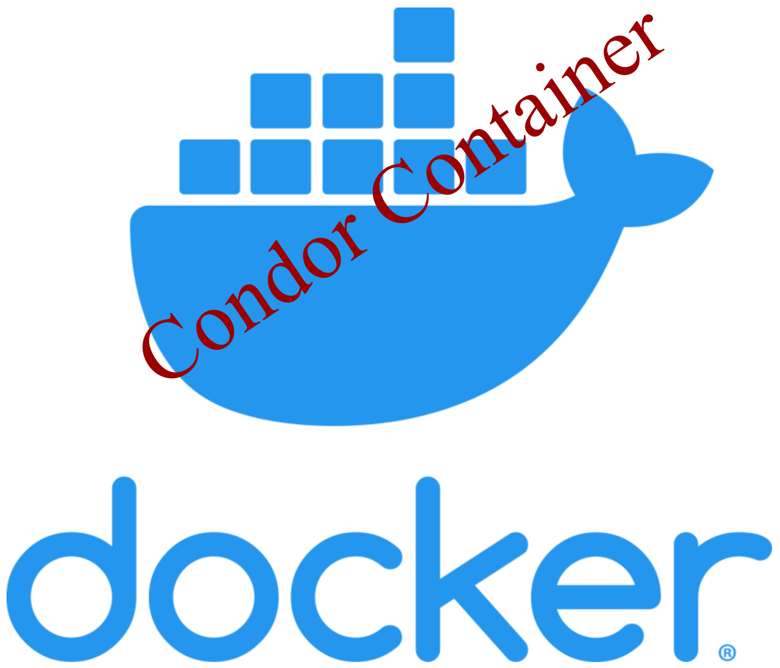
But there's a problem

Image contains HTCondor + OS

Fine for CM, probably OK for schedd

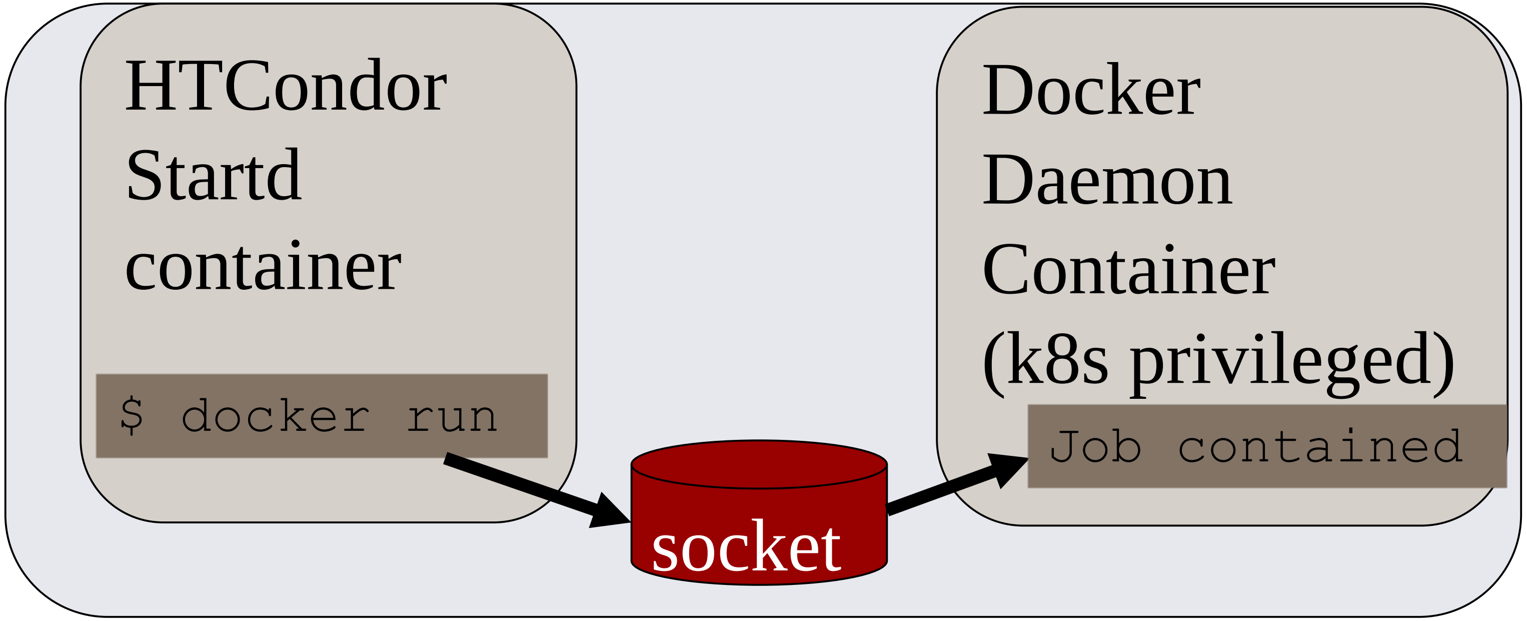
But what about the worker node?

"One Pod, Two containers"



Job Container

Worker node pod


HTCondor Startd container
$ docker run
socket
Docker Daemon Container (k8s privileged)
Job contained

Demo time…

2nd integration k8s grid type

E.g.

Submitting jobs as pods

K8s grid types warnings

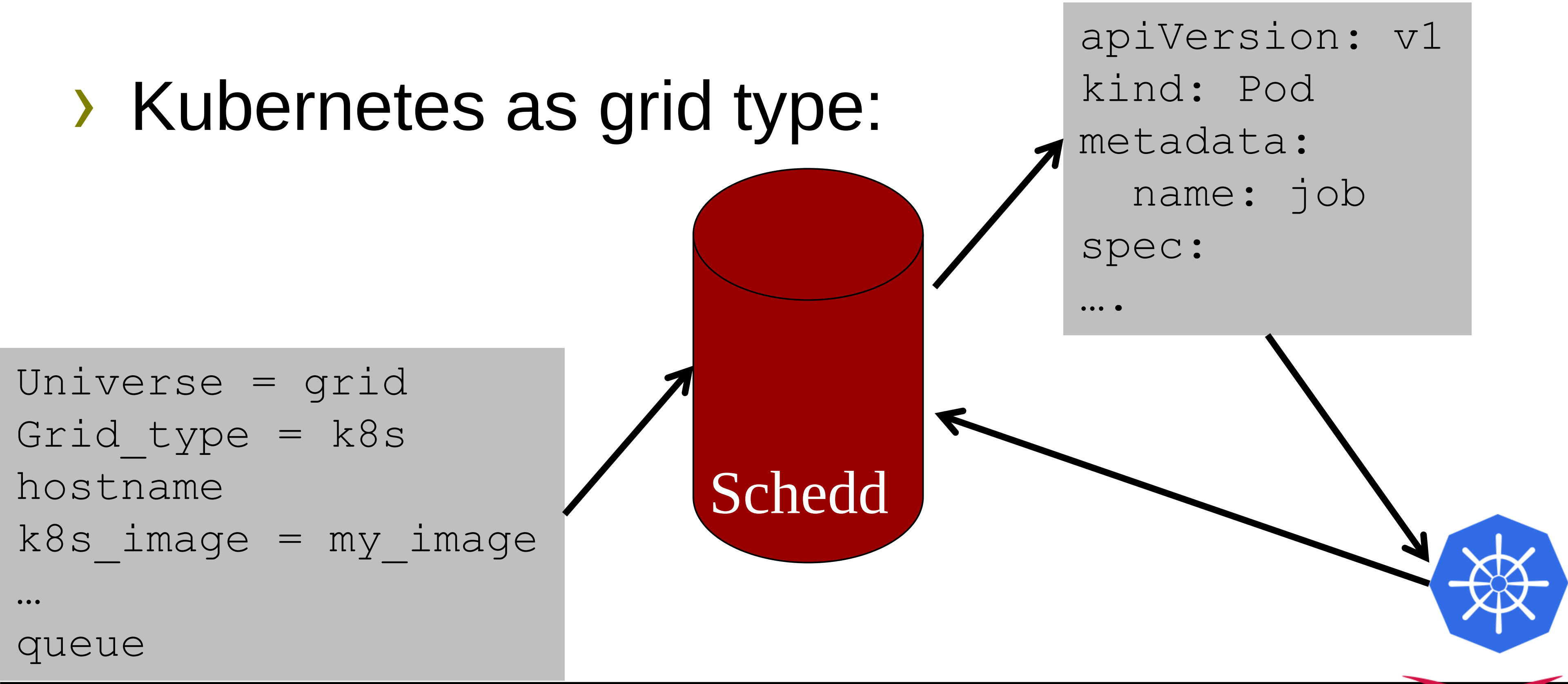
K8s is service centric,

no one-to-one mapping pods -> jobs

Best used for glideins or "livestock" jobs

K8s as HTCondor grid type

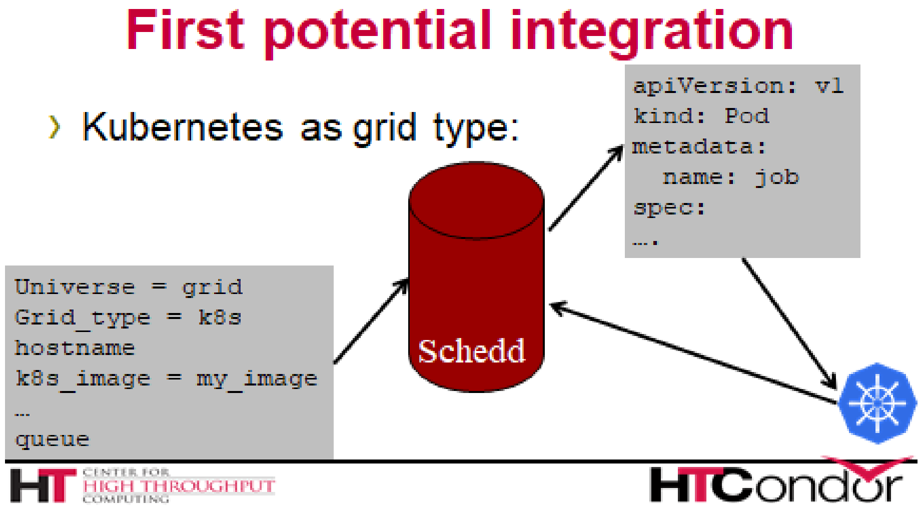
- Kubernetes as grid type:



```
Universe = grid
Grid_type = k8s
hostname
k8s_image = my_image
…
queue
```

```
apiVersion: v1
kind: Pod
metadata:
  name: job
spec:
….
```

- One scheduler
- K8s sees jobs



First potential integration

- Kubernetes as grid type:

```
Universe = grid
Grid_type = k8s
hostname
k8s_image = my_image
…
queue
```

```
apiVersion: v1
kind: Pod
metadata:
  name: job
spec:
….
```

Schedd
CENTER FOR HIGH THROUGHPUT COMPUTING
HTCondor

K8s add submit attrs

- k8s_image
- k8s_namespace
- k8s_priority
- …
- Idea is that the ce adds these to localize pilot
- Other attributes translated: RequestMemory

K8s gahp

- Shipping now
- Still early – missing file transfer

Demo time…

Thank you

Questions?

Please see YouTube Tutorials

Center for High Throughput Computing