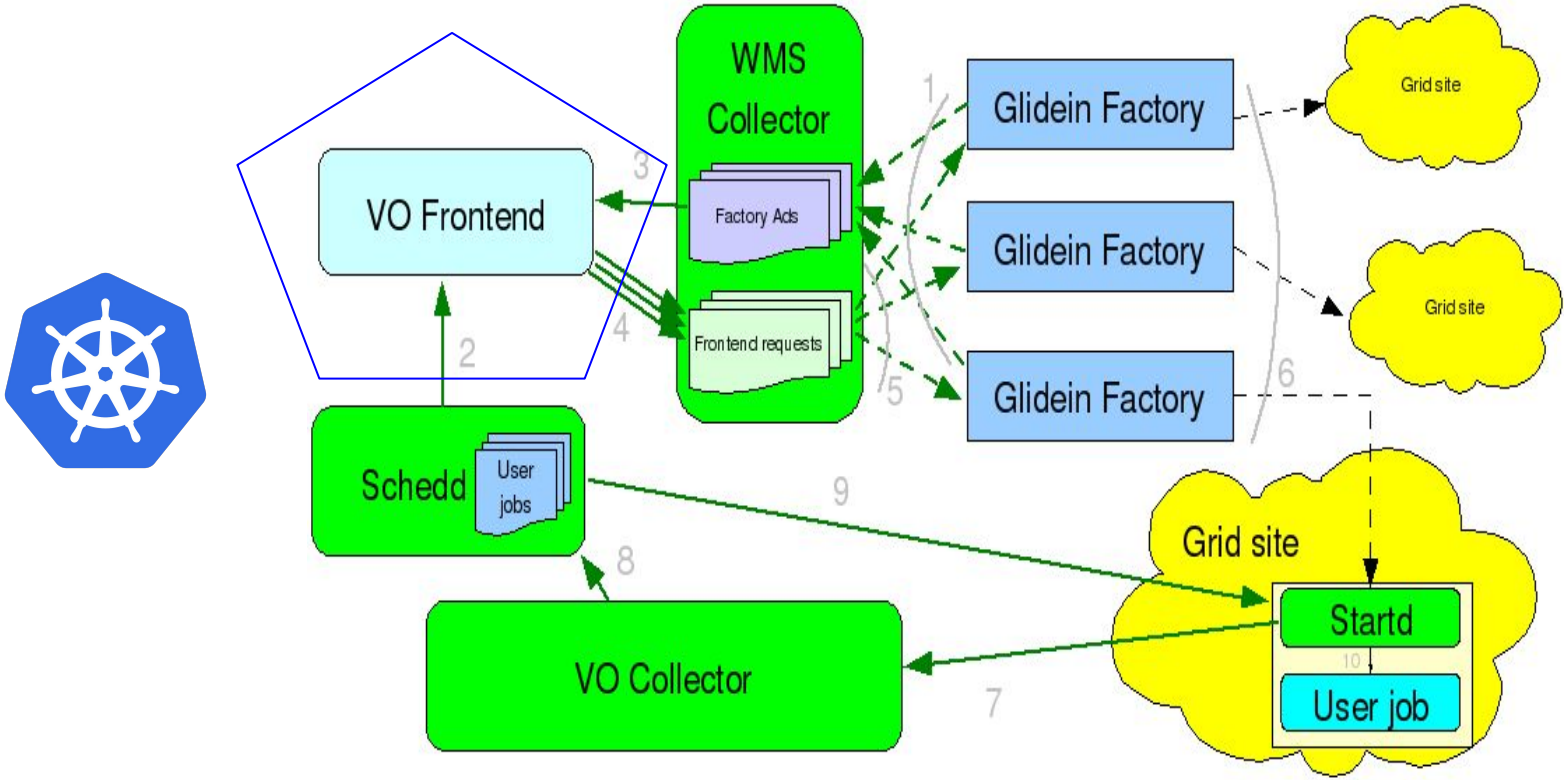
Kubernetizing GlideinWMS Frontend

Edgar Fajardo
Aashay Arora

GlideinWMS + Kuberneteized frontend



Motivation: Why going to containers

- Right now Edgar maintains the following Frontends:
  - BNL
  - IGWN
  - GLUEX
  - JLAB
  - UCLHC
  - ...
- Each one requires a virtual machine however all of them share a similar configuration and a same scripts: OSG Scripts.
- We cannot use frontend groups cause each group cannot talk to a separate VO Collector/Negotiator.

Containerizing Frontends

- A frontend configuration is made out of:
  - Frontend.xml file
  - A proxies.ini configuration file (to renew the proxies automatically)
  - Credentials to authenticate against sites (usually X509 proxy with VOMS attributes)
- State of a frontend is maintained in the frontend files that the pilot read:
  - /var/lib/gwms-frontend/vofrontend
- A frontend process consists of:
  - Python frontend process
  - Apache HTTP server

Task: translate the list above into kubernetes specifics

- A frontend configuration is made out of:
  - ~~Frontend.xml file~~
  - ~~A proxies.ini configuration file (to renew the proxies automatically)~~
  -
  - ~~State of a frontend is maintained in the frontend files that the pilot read:~~
  - ~~/var/lib/gwms-frontend/vofrontend~~

ConfigMaps

Secrets

Volume Claim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: jlab-storage
spec:
  storageClassName: rook-ceph-block
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Task: translate the list above into kubernetes specifics

- ~~A frontend process consists of:~~
  - ~~Python frontend process~~
  - ~~Apache HTTP server~~

SupervisorD

```
[root@gluex-frontend-68f688b57b-5xmsx /]# ps -xa -o pid,ruser=RealUser -o comm=Command
    PID RealUser Command
      1 root     supervisord
     83 root     httpd
     84 frontend python
     85 root     crond
 103375 apache   httpd
 111240 root     bash
 215340 apache   httpd
 218118 apache   httpd
 218145 apache   httpd
```

This is how the process looks like from inside the container

How do the frontends look like NOW

kubectl get pods -n osg-frontends -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
gluex-frontend-68f688b57b-5xmsx	1/1	Running	0	6d23h	10.244.168.141	k8s-gen4-08.calit2.optiputer.net	<none>	<none>
igwn-frontend-78ff5b949-8xfvm	1/1	Running	0	6d23h	10.244.164.72	k8s-gen4-06.calit2.optiputer.net	<none>	<none>
jlab-frontend-68898f6454-bkrxp	1/1	Running	0	29h	10.244.133.35	suncave-14	<none>	<none>

Questions?

Backup Slides

HTTP ingress

Pod Definition

```
spec:
  containers:
  - name: jlab-frontend
    image: aaarora/gwms-fe-docker:latest
    ports:
    - containerPort: 80
```

An ingress is created so HTTP(s) requests to a specific DNS on a port (80) are redirected to that container:

For example:

<stage base_dir="/var/lib/gwms-frontend/web-area/stage" use_symlink="True" web_base_url="**http://gluex-frontend.nautilus.optiputer.net/vofrontend/stage**"/>