dCache deployment in kubernetes

Tigran Mkrtchyan for dCache team

DESY. Fermilab neic NORDIC E-INFRASTRUCTURE COLLABORATION

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

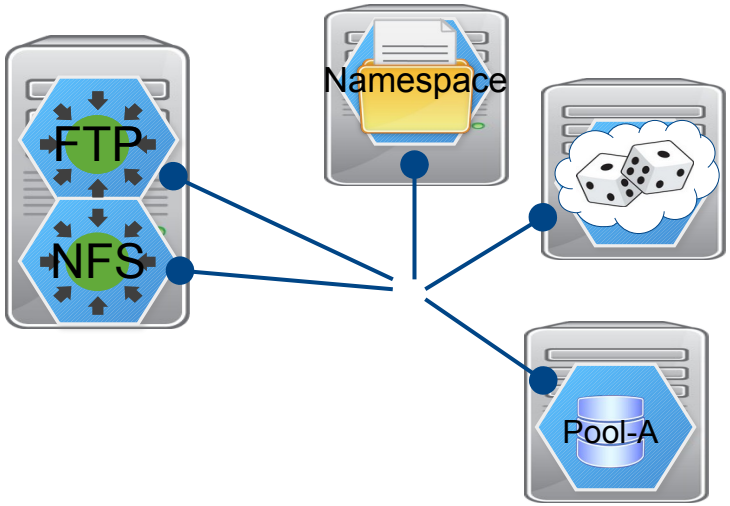
The MISSION

“... to provide a system for storing and retrieving huge amounts of data, distributed among a large number of heterogeneous server nodes, under a single virtual filesystem tree with a variety of standard access methods.”

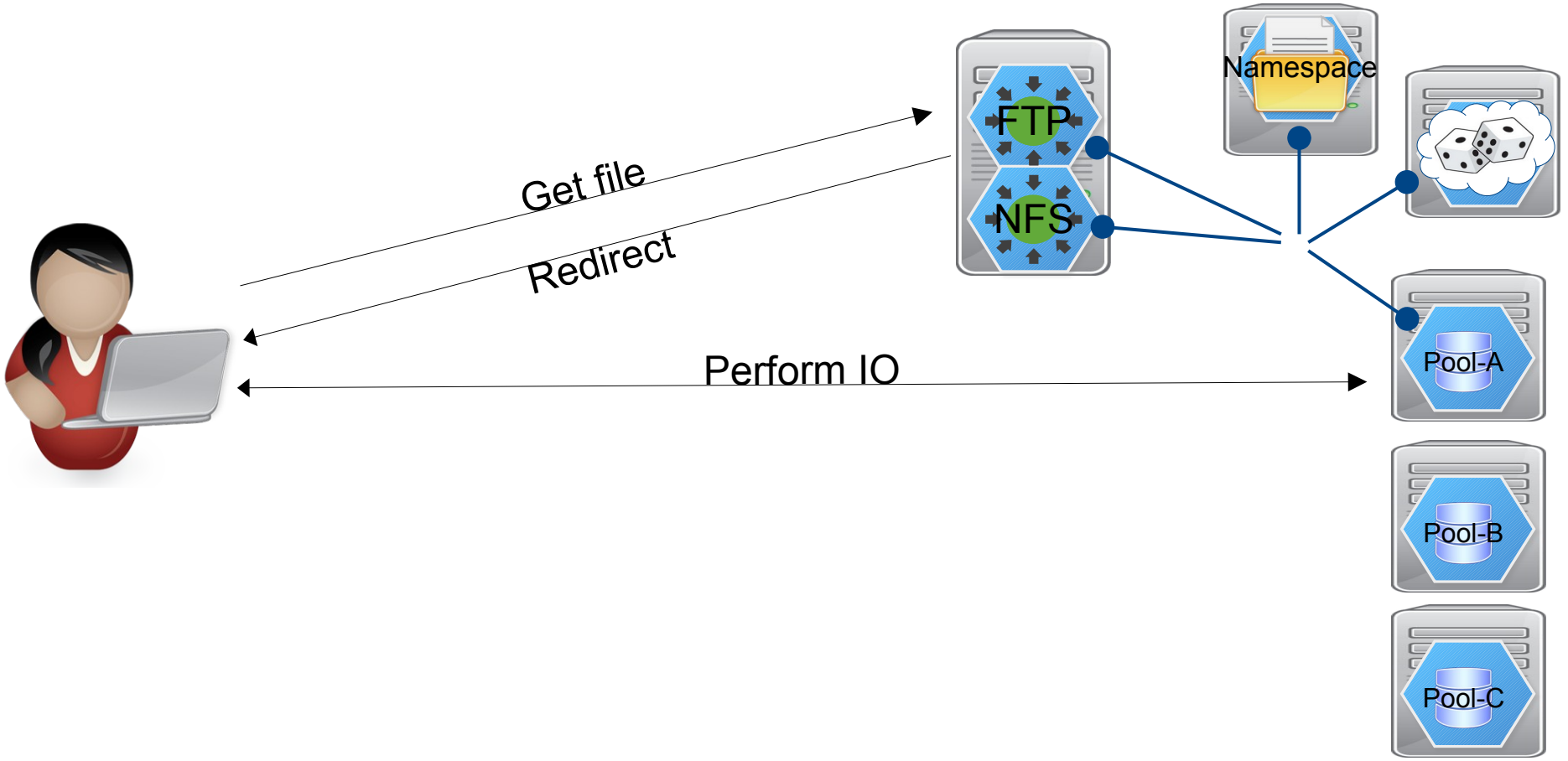
https://dcache.org/about/

Main Components

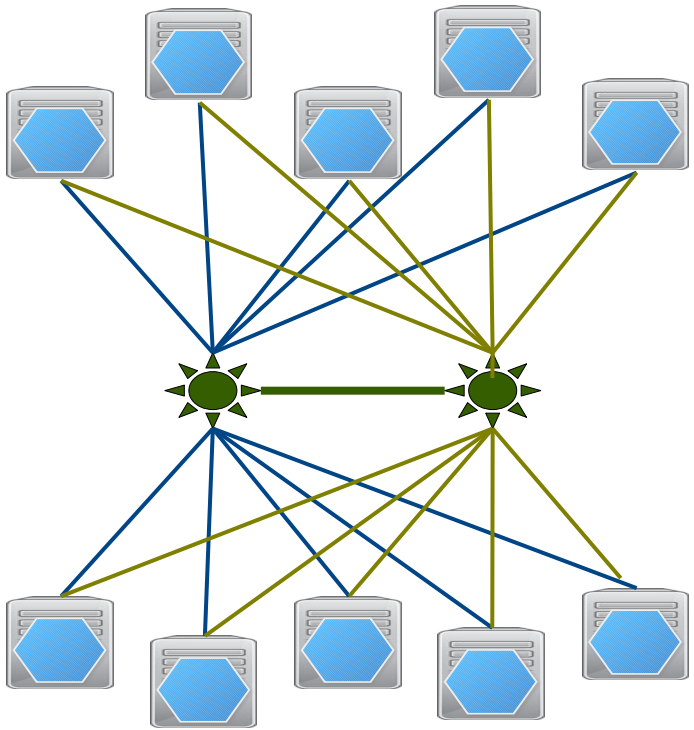
- Namespace
  - Inventory, POSIX view layer.
- Door
  - Protocol specific user entry point
    (FTP, HTTP, NFS …).
- Pool
  - Data storage node. Talk all protocols.
- PoolManager
  - Request distribution unit.



Client Flow


Get file
Redirect
Perform IO
FTP
NFS
Namespace
Pool-A
Pool-B
Pool-C

Internal Messaging

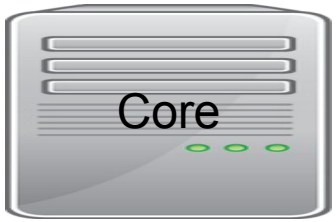


- Star-like topology
- Selected node configured as a hub called **CORE** domains
- Others called **SATELLITE**
- All communication goes through **CORE** domains
- Multiple **CORE** domains makes communication fault tolerant

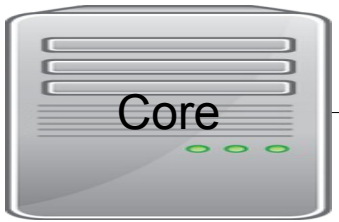
Zookeeper as Service Discovery

- A central registry of all CORE domains
  - Similar to DNS or routing table
- Leader election where actions must be performed by a single component
  - Staging, cleaning, pinning…

Service Discovery

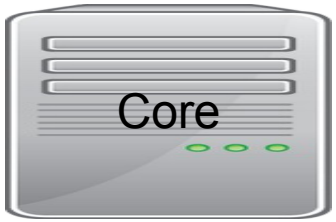


Service Discovery



I'm a core Domain, ip:a.b.c.d

Service Discovery

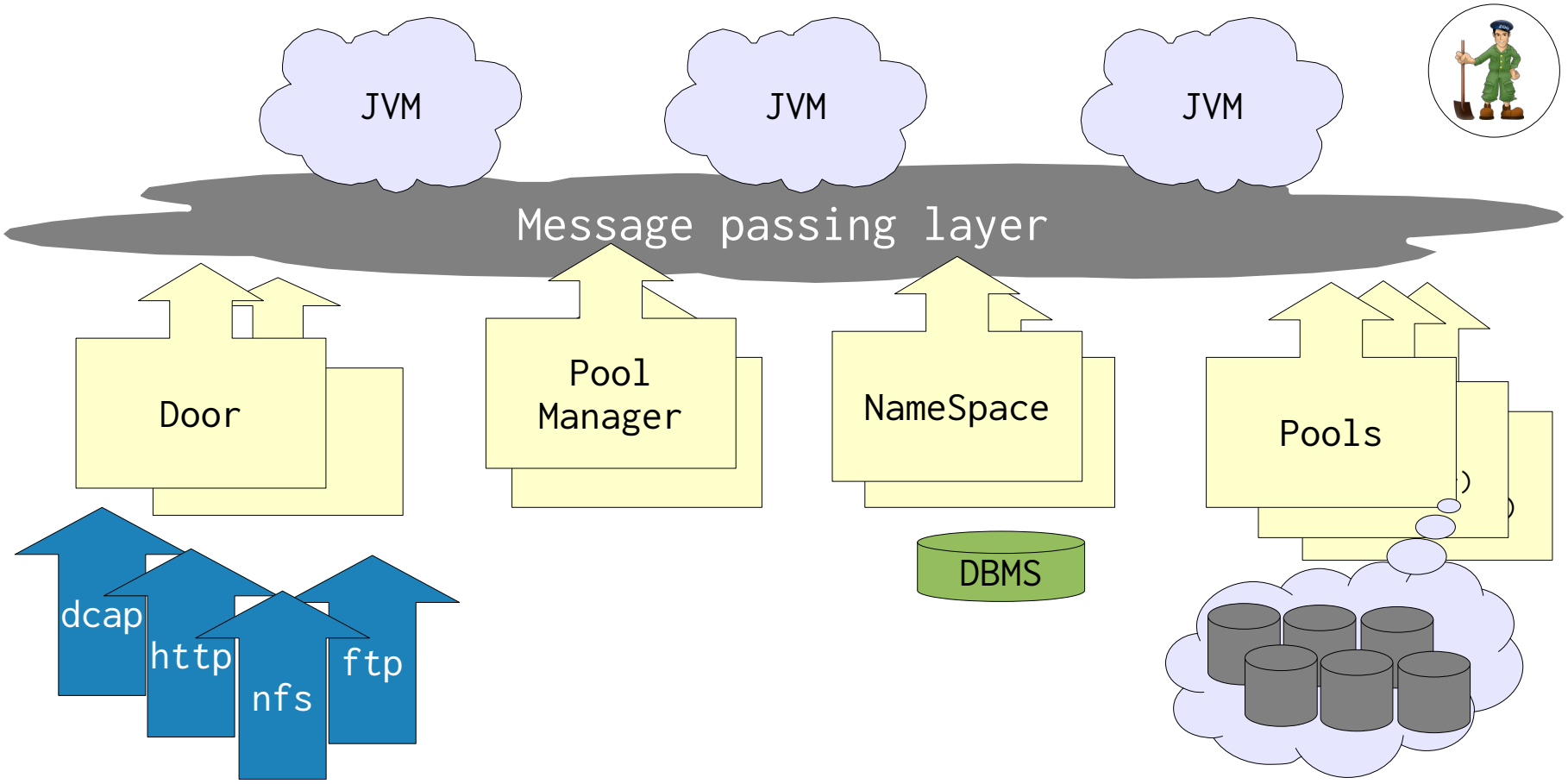


Where are a core Domains?

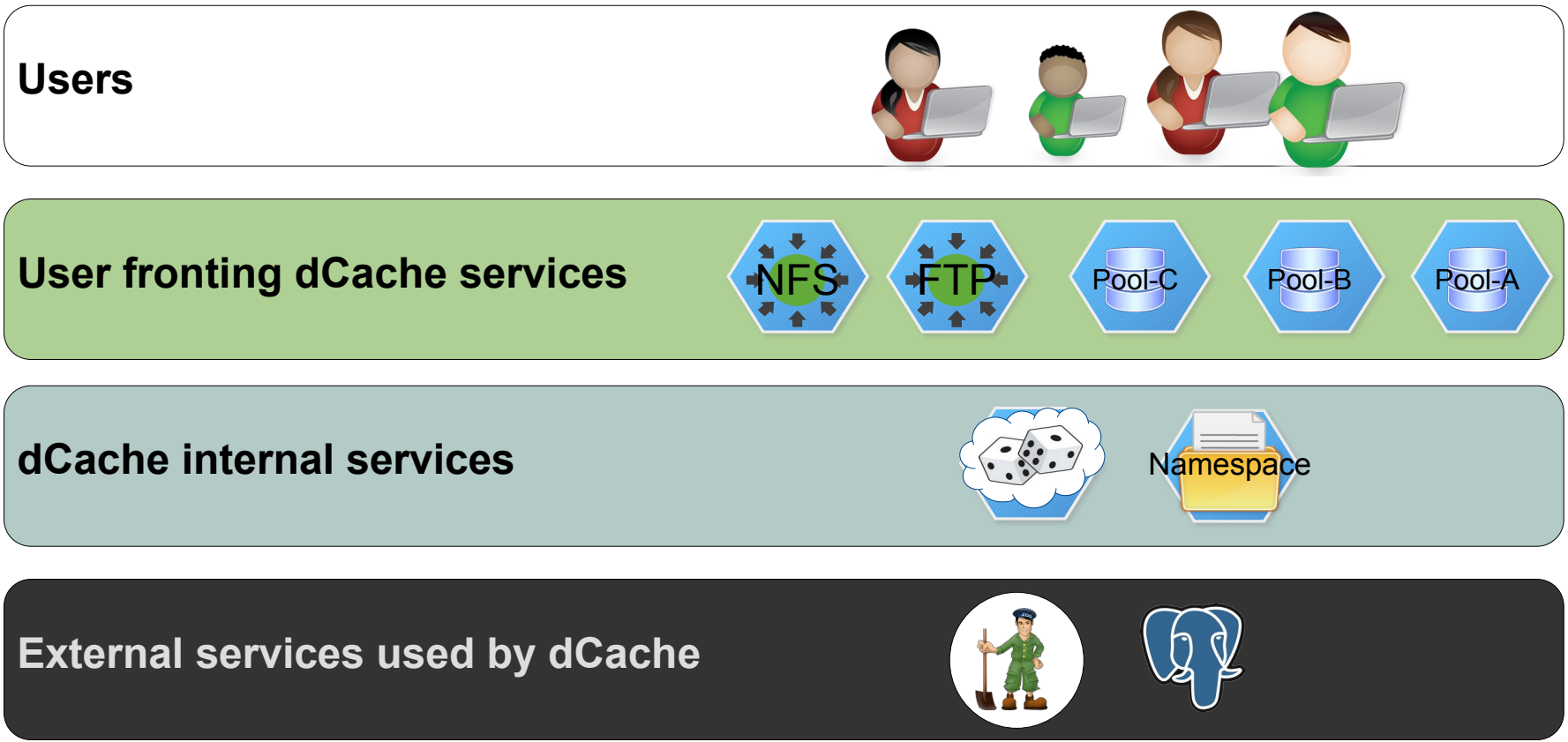
ip:a.b.c.d

Service Discovery

dCache on One Slide



Almost Full Picture



Testing Environment

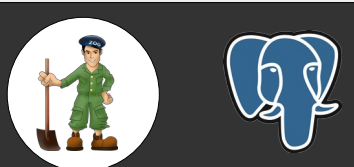
openstack®

CI & Agents

User fronting dCache services

dCache internal services

External services used by dCache



Testing Environment ++

openstack

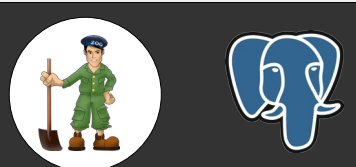
CI & Agents

User fronting dCache services

dCache internal services

podman

External services used by dCache



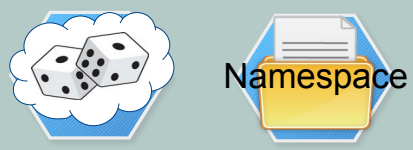
Testing Environment 2.0

kubernetes

CI & Agents

User fronting dCache services

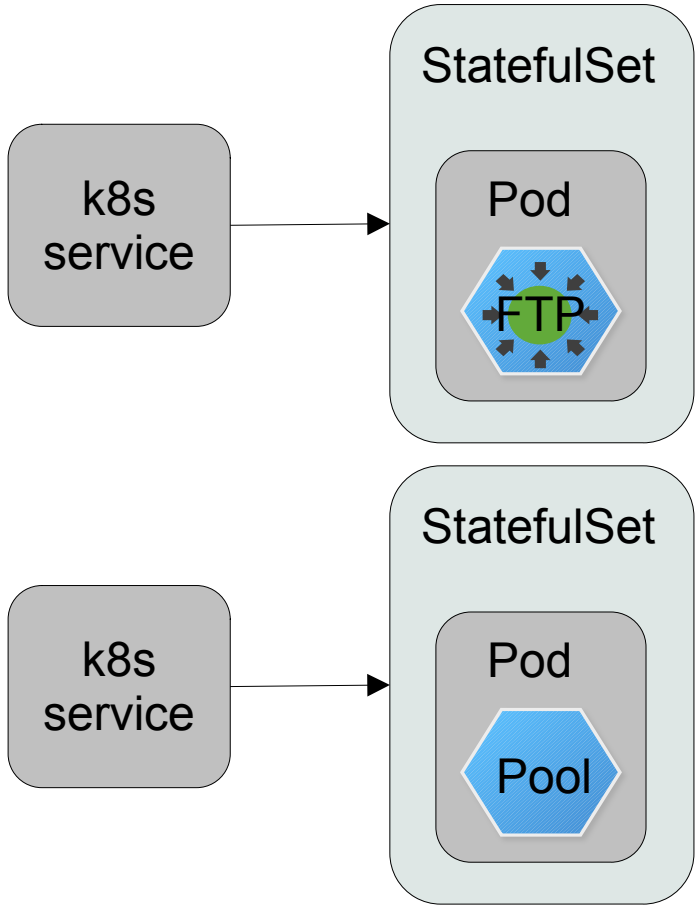
dCache internal services



External services used by dCache

dCache K8S services

- Each door exposed as a service
- Each protocol on pool exposed as a servers
- Port range exposed as a service per port
- Pool expose themselves by service name



Happy YAML Coding (door)!

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-tier-2-door-svc
spec:
  ports:
    - name: nfs-door
      port: 2049
      targetPort: 2049
    - name: xroot-door
      port: 1094
      targetPort: 1094
    - name: webdav-door
      port: 8080
      targetPort: 8080
```

```
[my-tier-2-door-svc]
localaddresses=my-tier-2-door-svc

[my-tier-2-door-svc/webdav]
webdav.cell.name=webdav-plain
webdav.net.port=8080


[my-tier-2-door-svc/webdav]
webdav.cell.name=webdav-tls
webdav.net.port=8083
webdav.authn.protocol=https
```

Happy YAML Coding (pool)!

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-tier-2-pool-a-svc
spec:
  ports:
    - name: nfs-mover
      port: 32049
      targetPort: 32049
    - name: xroot-mover
      port: 31094
      targetPort: 31094
    - name: http-mover
      port: 38080
      targetPort: 38080
```

```
[my-tier-2-pool-a-svc]

[my-tier-2-pool-a-svc/pool]
localaddresses=my-tier-2-pool-a-svc
pool.name=pool-a
pool.path=/pool
pool.mover.nfs.port.min=32049
pool.mover.nfs.port.max=32049
pool.mover.xrootd.port.min=31094
pool.mover.xrootd.port.max=31094
pool.mover.http.port.min=38080
pool.mover.http.port.max=38080
pool.mover.https.port.min=38083
pool.mover.https.port.max=38083
```

Happy YAML Coding (pool, door)!

```yaml
- name: wan-port-0
  port: 28000
  targetPort: 28000
- name: wan-port-1
  port: 28001
  targetPort: 28001
- name: wan-port-2
  port: 28002
  targetPort: 28002
- name: wan-port-3
  port: 28003
  targetPort: 28003
- name: wan-port-4
  port: 28004
  targetPort: 28004
```

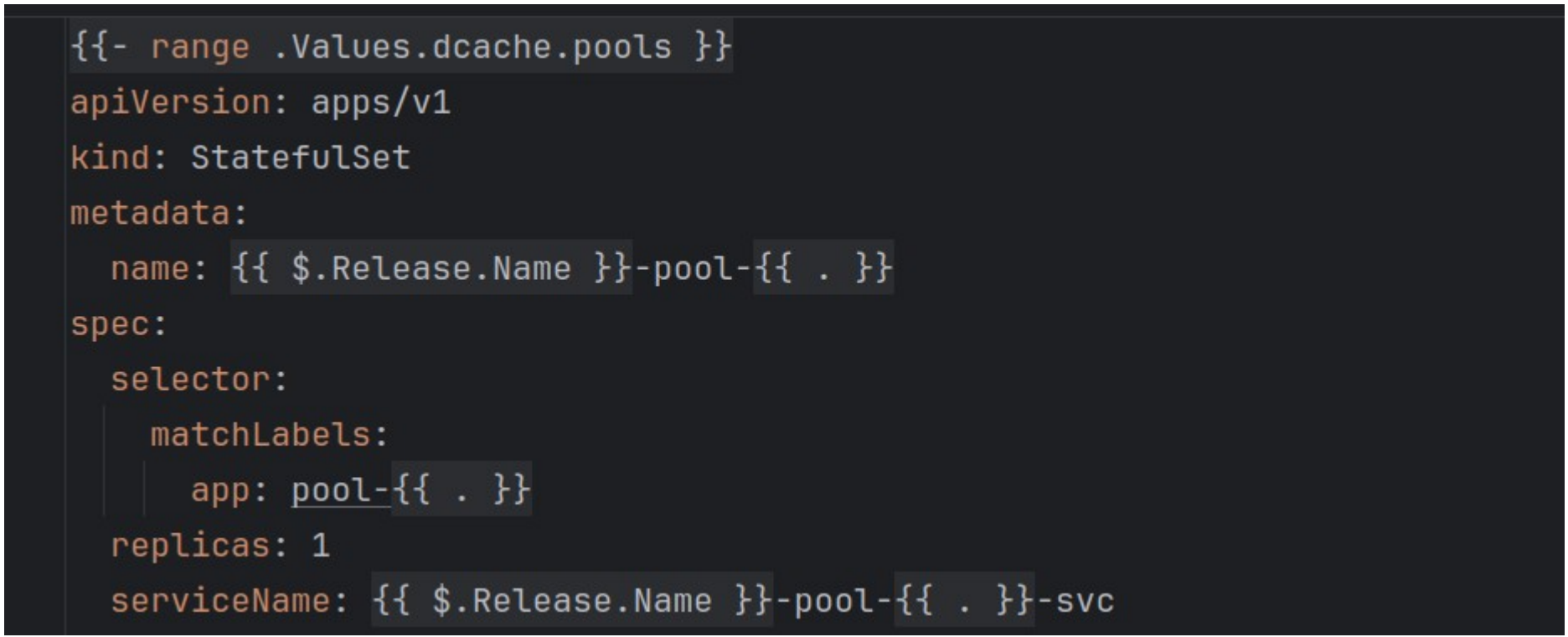
TCP pots for gridftp can’t be assigned dynamically, and require in advance mapping.

This must be done on ftp door and all pools to support various gridftp transfer modes.

Helm Charts (port range)


```
{{ $range_start := ( $.Values.mover.wan_range_min | int) }}
{{ $range_stop := ( $.Values.mover.wan_range_max | int) }}
{{- range $port_index, $port := untilStep $range_start $range_stop 1 }}
- name: wan-port-{{ $port_index }}
  port: {{ $port }}
  targetPort: {{ $port }}
{{- end }}
```

Helm Charts (pool)



```
{{- range .Values.dcache.pools }}
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: {{ $.Release.Name }}-pool-{{ . }}
spec:
  selector:
    matchLabels:
      app: pool-{{ . }}
  replicas: 1
  serviceName: {{ $.Release.Name }}-pool-{{ . }}-svc
```

```
$ helm install --set dcache.pools="{pool1, pool2, pool3}" ...
```

Build Infrastructure: GitLab + k8s

- Documented release/test process
- Shareable build pipelines
- Can be replicated at sites
- Transparent release process
- Code will stay on Github

K8S Based dCache Deployment

- dCache containers available at docker hub
- Helm carts to deploy dCache with three commands


```
$ helm install dcache-db bitnami/postgresql
$ helm install cells bitnami/zookeeper
$ helm install --set image.tag=9.2.0 my-tier-2 dcache/dcache
```

Missing Bits …

- Ingress is not possible
  - Currently dCache and worker nodes deployed in a single k8s namespace
- Helm chart comes with pre-defined dCache
  - Only number of pools can be specified
- Everything is *StatefulSet*
  - Stateless components can be defined as *Deployment*

Get involved

- Use our container in your testing
- Help to make helm charts production ready
- Help with documentation
- Share your experience and knowledge
- Share your needs

"Rosie the Riveter", National Museum of American History, Miller, J. Howard
https://americanhistory.si.edu/collections/search/object/nmah_538122

Conclusions & Outlook

- The dCache have demonstrated successful deployment of dCache in k8s
- Gitlab+k8s testing makes dCache release reproducible by sites (FAIR development ?!)
  - Can be used by other projects that need dCache or storage
- Starting 9.2 dCache ‘official’ containers are published at the docker hub
  - ! Not recommended for production use (yet)
- With help from the community we can turn site deployment into a single command
  - dCache developers are not experts in k8s or helm charts, any help is welcome!!!

Thank You!

More info:

https://dcache.org

To steal and contribute:

https://github.com/dCache/dcache

https://github.com/dCache/dcache-helm

Help and support:

support@dcache.org, user-forum@dcache.org