



XCache deployment

experience



What is XCache?

Basically an xrootd proxy server that also stores data passing through it. On next access it delivers data from disk.

It needs:

- a) “Dedicated” node
- b) Local storage
- c) IP
- d) Secrets (to authenticate against origin servers)
- e) Integration with ATLAS workflows (RUCIO, AGIS, monitoring)

First big choice

Xcache can be setup as a standalone or as a cluster.

I chose standalone:

- Simpler deployment (only xrootd service, no cmsd needed)
- Reliability
- External control of individual nodes
- Cluster anyhow does not rebalances disk usage
- We are still far from utilizing single node instances fully and efficiently

Docker container

Everything in a [github repo](#) and docker image built automatically in [dockerhub, documentation](#) in github too.

The image is rather basic:

- Based on centos
- Xrootd-server, xrootd-client, vomsxrd, fetch-crl, python,...
- xrootd user has fixed GID and UID
- Creates all directories needed, makes them owned by xrootd (but only if needed!)

Containers

3 containers run in each pod:

- xcache - server itself
- x509 - renews proxy
- reporter - collects info on cached files and sends to logstash

All server configuration done through environment variables.

XCache:

- Sets few default environment variables if not already defined.
- Sleeps 2 min for x509 container to finish first update of CA
- Starts server
- Activates itself in AGIS using REST API
- Sleeps indefinitely

X509:

- Updated x509 proxy
- Fetches crls
- sleeps 6 h

Reporter:

- Collects info from .cinfo files
- Reports to ES
- Sleeps 1h

Server - K8s deployment

Secrets: service certificate (2 files)

As k8s deployment (not a simple pod)

Since it requires special node it uses nodeSelector

You don't want anything else using this node so *

Volume to be used for caching is a hostPath

Liveness probe on server container

All configs done through environment variables. In hindsight it would be nicer to use ConfigMaps.

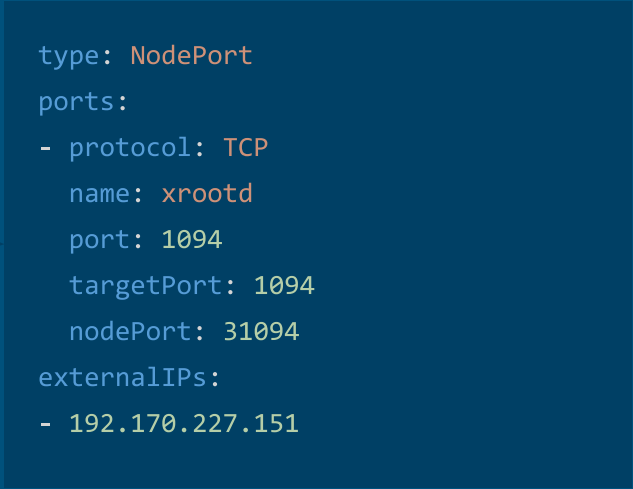
```
tolerations:  
- key: "special"  
  operator: "Exists"  
  value: true  
  effect: PreferNoSchedule
```

```
- name: xcache-data  
  hostPath:  
    path: /scratch
```

```
livenessProbe:  
  tcpSocket:  
    port: 1094  
  initialDelaySeconds: 180  
  periodSeconds: 60
```

* `kubectl taint nodes "xcache nodename" special=true:PreferNoSchedule`

Service is a NodePort. IP is fixed.



```
type: NodePort
ports:
- protocol: TCP
  name: xrootd
  port: 1094
  targetPort: 1094
  nodePort: 31094
externalIPs:
- 192.170.227.151
```

Stress test - k8s deployment

Used to stress test any xcache instance and report about results.

Uses the same image, same secrets, just runs different code.

Helm chart

Maybe an overkill for app this simple, but required by slate and makes config more readable. Basically replaced values with placeholders like this:

```
containers:
  - name: {{ .Chart.Name }}
    image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
    imagePullPolicy: {{ .Values.image.pullPolicy }}
    env:
      - name: XC_SPACE_HIGH_WM
        value: "{{ .Values.XCacheConfig.HighWaterMark }}"
      - name: XC_SPACE_LOW_WM
        value: "{{ .Values.XCacheConfig.LowWaterMark }}"
```


Helm values

Clean and with a lot of comments (not shown here).

```
Instance: global
Service:
  Port: 1094
  ExternalIP: 192.170.227.151
SiteConfig:
  Name: MWT2
  AGISprotocolID: 433
Monitoring:
  Collector: http://uct2-collectd.mwt2.org:8080
XCacheConfig:
  CacheDirectory: /scratch
  HighWaterMark: 0.95
  LowWaterMark: 0.90
  RamSize: 16g
  BlockSize: 1M
  Prefetch: 0
  CertSecret: xcache-cert-secret
image:
  repository: slateci/xcache
  tag: latest
  pullPolicy: IfNotPresent
```