redhat.

Openshift Persistent Storage
(NFS/Gluster/Ceph/EBS)

Elvir Kurić
Performance Engineer

Agenda

- Openshift overview
- Persistent storage types used by Openshift
- PV Persistent Volume and PVC Persistent Volume Claim
- This presentation will focus on
  - NFS/Gluster/Ceph/EBS persistent storage types

Agenda

- Will be demonstrated how to
  - Configure and use persistent storage with openshift - including demo examples
  - Openshift master / node configuration

About Elvir Kuric (aka ekuric / elko)

- Work for Red Hat as performance engineer
- Mostly docker/storage/kubernetes/openshift ... and again from start

Openshift PaaS

- PaaS platform
  - Fast and easy way to run apps in cloud
  - Application platform
- Openshift Enterprise / Openshift Origin

Openshift PaaS

- https://www.openshift.org/vm/ - openshift origin virtual machine
- Free account possible to get at https://www.openshift.com/
- https://github.com/openshift/openshift-ansible

Persistent Storage Types for Openshift

- NFS
- Gluster
- Ceph
- EBS
- GCE
- HOST
- Iscsi
- ....

Admins role divide

- Openshift administrator （ Oa ）
- Storage administrator （ Sa ）

Openshift PV and PVC

- PV - Persistent Volume
- PVC - Persistent Volume Claim

PV Persistent Volume - General Format

```
{
      "apiVersion": "v1",
      "kind": "PersistentVolume",
      "metadata": {
        "name": "PVNAME"
      },
      "spec": {
        "capacity": {
            "storage": "XGi"
            },
        "accessModes": [ "ReadWriteMany" ],
        "STORAGETYPE": {
            "Storage dependent : "/var/export/vol1",
            "Storage dependent: ""
        }
      }
    }
```

PV Persistent Volume - NFS

```
"nfs": {
     "path": "/export/path",
     "server": "nfs_ip"
     },
```

PV Persistent Volume - CEPH

```
"rbd": {
        "monitors": [
                "ceph_monitor_1_IP:6789",
                "ceph_monitor_2_IP:6789",
                "ceph_monitor_3_IP:6789"
        ],
        "pool": "ceph_pool_name",
        "image": "ceph_image",
        "user": "admin",
        "secretRef": {
                "name": "ceph-secret"
        },
        "fsType": "ext4",
```

PV Persistent Volume - Gluster

```
"glusterfs": {
          "endpoints": "glusterfs-endpoint”,
      "path": "gluster-volume",
```

PV Persistent Volume - EBS

```
"awsElasticBlockStore": {
          "fsType": "ext4|xfs",
          "volumeID": "vol-id - from EC2 side" - eg : vol-2aea309c
      },
```

Persistent Volume Claim - Format

```
{
    "apiVersion": "v1",
    "kind": "PersistentVolumeClaim",
    "metadata": {
    "name": "pvcname"
    },
    "spec": {
    "accessModes": [ "ReadWriteMany" ],
    "resources": {
        "requests": {
            "storage": "XGi"
        }
    }
    }
}
```

NFS

- On openshift nodes is necessary to have
  - # yum install nfs-utils
- No need for any special configurations on Openshift master / node configuration
- Sa (Storage administrator) will configure NFS and provide to Oa (Openshift Administrator)
  - NFS share to use
  - IP address of nfs server

NFS

- Storage Admin will also ensure that all ports and dns resolution works fine
- Do not forget on Selinux (# setsebool -P virt_use_nfs 1 and setsebool -P virt_sandbox_use_nfs 1) on Openshift nodes

pvnfs.json

```
{
  "apiVersion": "v1",
  "kind": "PersistentVolume",
  "metadata": {
     "name": "pvname"
  },
  "spec": {
     "capacity": {
     "storage": "XGi"
     },
     "accessModes": [ "ReadWriteOnce" ],
     "nfs": {
     "path": "/export/path",
     "server": "nfs_ip"
     },
     "persistentVolumeReclaimPolicy": "Recycle"
  }
}
```

nfs pod create demo

```
# python createpod.py --pvfile pv-nfs.json --pvcfile pvc.json --podfile
pod.json --nfsip 192.168.122.101 --nfsshare /mnt/nfs --pvsize 2 --pvcsize
2  --num 3 --storage nfs  --pvpvc nfs  --image fedssh


# oc exec podnfs0 -- mount | grep nfs
192.168.122.101:/mnt/nfs on /mnt/persistentvolume type nfs4 (rw,relatime,
seclabel,vers=4.2,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,
port=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.122.108,
local_lock=none,addr=192.168.122.101)
```

Gluster

- On openshift nodes gluster-fuse package has to be installed

  # yum install glusterfs-fuse

- By default installed if used openshift-ansible https://github.com/openshift/openshift-ansible

Gluster

- Create gluster service and gluster endpoints at Openshift side - done by Openshift Administrator
- Get IPs of gluster servers and gluster volume to use from Storage Administrator
- Important : DNS resolution between openshift nodes and gluster nodes *must* work properly (direct / reverse)
- Pay attention on Selinux on Openshift nodes in case “permission denied” errors

Gluster

- Important : DNS resolution between openshift nodes and gluster nodes *must* work properly (direct / reverse)
- Pay attention on Selinux on Openshift nodes in case “permission denied” errors
- Min 10 Gb network between openshift nodes and gluster nodes

Gluster endpoint

```
"apiVersion": "v1",
     "kind": "Endpoints",
     "metadata": {
     "name": "glusterfs-cluster"},
     "subsets": [{
          "addresses": [{"ip": "192.168.122.158"}],
          "ports": [{"port": 1}]}]}


# oc create -f glusterfs-endpoint.json
```

gluster endpoint

```
# oc get ep | grep gluster
glusterfs-cluster   192.168.122.158:1,192.168.122.159:1,192.168.122.160:1
```

Gluster pv.json

```
{
    "apiVersion": "v1",
    "kind": "PersistentVolume",
    "metadata": {
    "name": "pv1"},
    "spec": {"accessModes": ["ReadWriteMany"],
    "glusterfs": {
        "endpoints": "glusterfs-cluster",
    "path": "osevolume",
    "readOnly": false
    },
    "capacity": {
        "storage": "1Gi"
    },
    "persistentVolumeReclaimPolicy": "Recycle"
    }
}
```

gluster pod create demo

```
# python createpod.py --pvfile pv-gluster.json --pvcfile pvc.json --
podfile pod.json --glusterfsep glusterfs-cluster --pvsize 2 --pvcsize 2
--num 3 --storage gluster --glustervolume osevolume --pvpvc gluster  --
image fedssh

# oc exec podgluster0 -- mount | grep gluster
192.168.122.158:osevolume on /mnt/persistentvolume type fuse.glusterfs
(rw,relatime,user_id=0,group_id=0,default_permissions,allow_other,
max_read=131072)
```

CEPH

- Openshift node is CEPH client
- Create ceph secret on openshift master
- Get from ceph side /etc/ceph/ceph.conf and ceph.client.admin.keyring

CEPH

- ceph.client.admin.keyring is for admin - any other user with proper rights with be fine - check CEPHX documentation http://docs.ceph.com/docs/master/rados/configuration/auth-config-ref/
- # yum install ceph-common - on openshift nodes (ansible installation gets ceph-common by default)
- CEPH RBD supported, CEPH FS landed in latest ceph version Jewel - did not tried it with openshift
- Openshift is ceph pool agnostic - it does not care about pool type (replicated / EC)

CEPH

- # yum install ceph-common - on openshift nodes (ansible installation gets ceph-common by default)
- CEPH RBD supported, CEPH FS landed in latest ceph version Jewel - did not tried it with openshift
- Openshift is ceph pool agnostic - it does not care about pool type (replicated / EC)
- Min 10 Gb between CEPH OSDs and Openshift nodes

Ceph secret

```
apiVersion: v1
kind: Secret
metadata:
name: ceph-secret
data:
key: < here key : get it with :  grep key /etc/ceph/ceph.client.admin.
keyring |awk '{printf "%s", $3}'|base64 >


# oc create -f cephsec.yaml
# oc get secret | grep ceph
ceph-secret            Opaque                                1         96d
```

Ceph secret

```
# oc create -f cephsec.yaml
# oc get secret | grep ceph
ceph-secret            Opaque                                1         96d
```

Ceph secrets are not shared between openshift projects

Pvceph.json

```
{
    "apiVersion": "v1",
    "kind": "PersistentVolume",
    "metadata": { "name": "pvname"},"spec": { "capacity": {"storage":
"XGi"},"accessModes": ["ReadWriteMany"],
    "rbd": {
        "monitors": [
            "ceph_monitor_1_IP:6789",
            "ceph_monitor_2_IP:6789",
            "ceph_monitor_3_IP:6789"
        ],
        "pool": "ceph_pool_name","image": "ceph_image",
        "user": "admin","secretRef": {"name": "ceph-secret" },"fsType":
ext4",
        "readOnly": false
    }
    }
}
```

Ceph demo

```
# python createpod.py --pvfile pv-ceph.json --pvcfile pvc.json --podfile
pod.json --cephsecret ceph-secret --cephmonitors 192.168.122.101:
6789,192.168.122.102:6789,192.168.122.103 --pvsize 2 --pvcsize 2  --num 3
--storage ceph --cephimagename cephvienna --cephimagesize 2  --pvpvc ceph
--image fedssh --cephpool perfteam --fstype ext4
```

Ceph demo

```
# rbd showmapped
id pool   image     snap device
0  perfteam cephvienna0 -    /dev/rbd0
1  perfteam cephvienna2 -    /dev/rbd1
2  perfteam cephvienna1 -    /dev/rbd2

# oc exec -- podceph0 mount | grep rbd
/dev/rbd0 on /mnt/persistentvolume type ext4 (rw,relatime,seclabel,
stripe=1024,data=ordered)

Only ext4 supported at time - need to find BZ / issue for this statement
```

EC2 EBS

- It requires small changes in openshift master / node configuration
- Edit
  - /etc/origin/master/master-conf.yaml
  - /etc/origin/node/node-config.yaml
  - /etc/sysconfig/atomic-openshift-master
  - /etc/sysconfig/atomic-openshift-node

  For details check openshift documentation

https://docs.openshift.com/enterprise/3.1/install_config/configuring_aws.html

EC2 EBS

- Use amazon SDK to create EBS volumes (eg python boto3)
- Tag EBS volumes (boto3)
-
- Limitation 39 EBS / ose node (at time)

https://github.com/kubernetes/kubernetes/blob/master/pkg/cloudprovider/providers/aws/aws.go#L86

- RH BZ #1322569 to change 39 EBS limit
- Amazon recommends 40 devices : http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/volume_limits.html

EC2 EBS

- EC2 has feature called IO credits
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

Performance will go down once IO credits exhausted

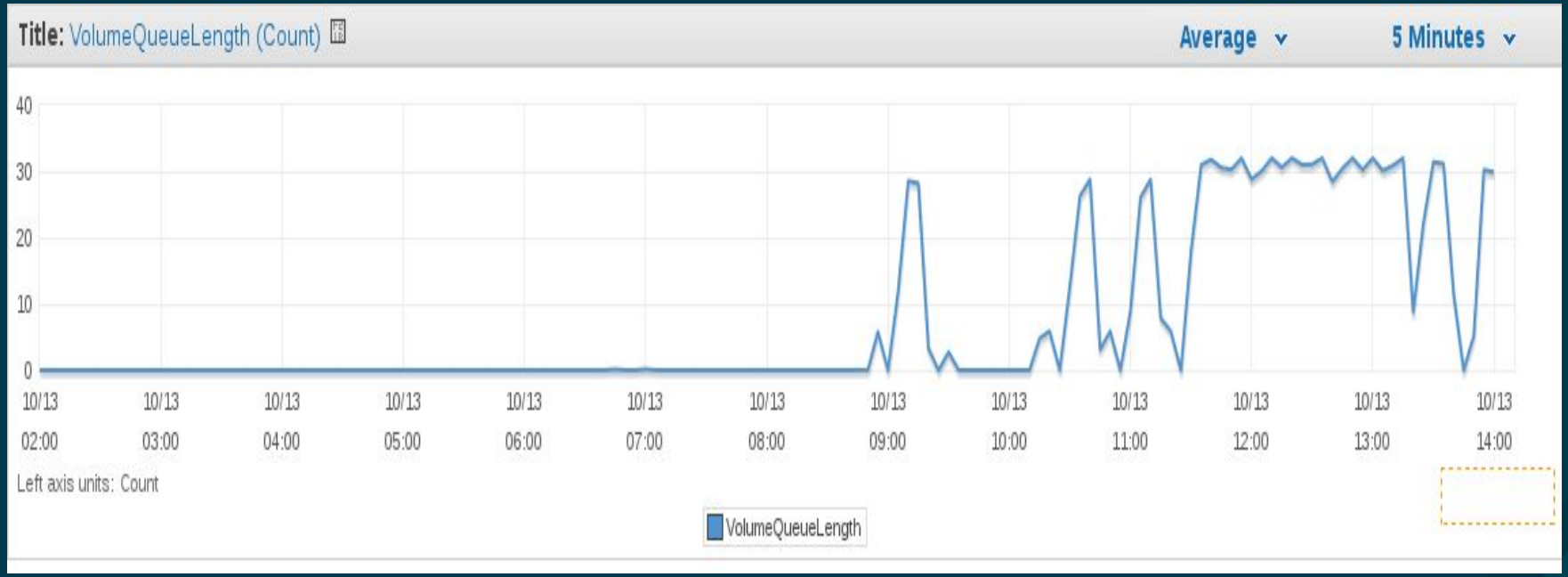
- 1 GB = 3 IOPS for GP2 EBS volume type
- EC2 has “api rate” - AWS SDK has support for it
- Dynamic EBS provisioning https://docs.openshift.com/enterprise/3.1/install_config/persistent_storage/dynamically_provisioning_pvs.html

EC2 EBS

Performance will go down once IO credits exhausted

- 1 GB = 3 IOPS for GP2 EBS volume type
- Solution for IO credits issue - use provisioned IOPS volume type ($$$)
- EC2 has “api rate” - AWS SDK has support for it

EC2 EBS IO burst / credits



EBS demo

```
# python createpod.py --pvfile pv.json --pvcfile pvc.json --podfile pod.
json --pvsize 2 --pvcsize 2  --num 3 --storage ebs --pvpvc ebs --image
fedssh --ebstagprefix=ekuric_test

#  oc exec podebs0 -- mount | grep pers
/dev/xvdba on /mnt/persistentvolume type ext4 (rw,relatime,seclabel,
data=ordered)
```

What to use?

- If requirement is to scale out without owning hardware -> EC2 / EBS
- Openshift cluster on EC2 - ebs/ceph/gluster/nfs
- If own hardware and use same storage solution for more systems - ceph/gluster

What to use?

- Have full control of data and tune performance to your needs -> ceph/gluster
- Security -- > on premise ceph/gluster
- createpod.py code https://github.com/ekuric/_talks/tree/master/wien

Questions

ekuric@redhat.com

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos