NVMe-oF - What's new and what's next

Aviv Caro

Agenda

- Recap of what is already there and was previously presented in more details

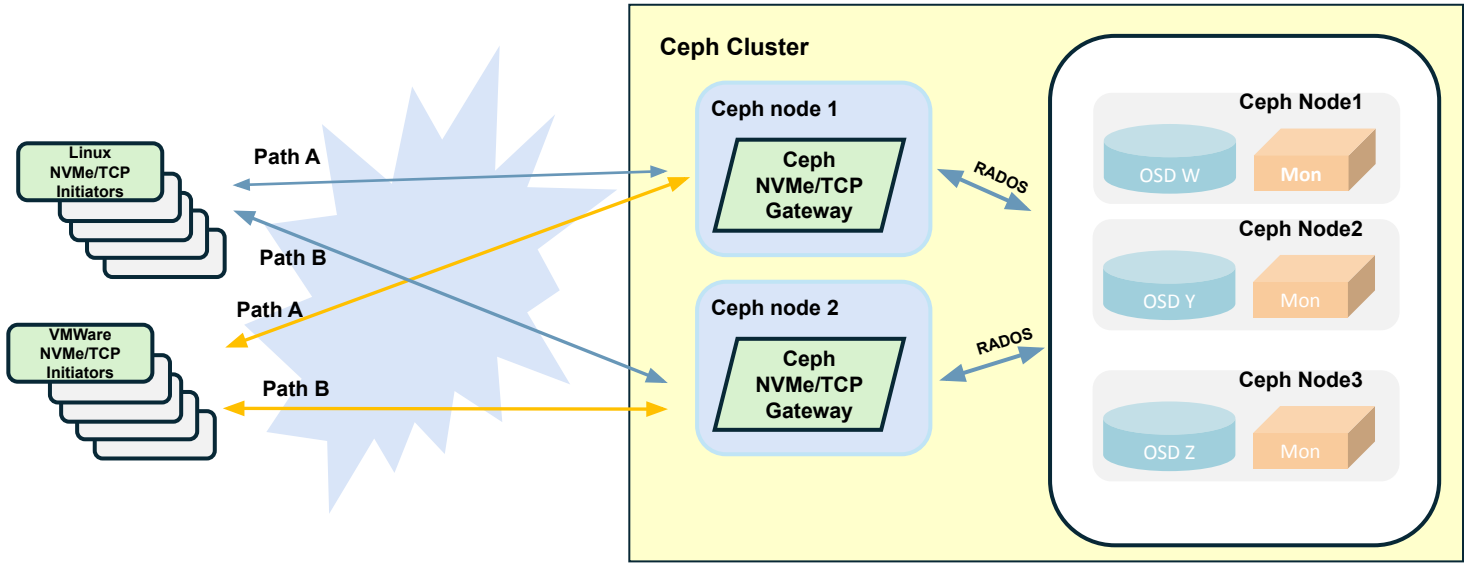
- New features:
  - Cluster Context allocation
  - Automatic namespaces load balancing
  - Security
    - Namespaces masking
    - In-band authentication
  - Dashboard
  - Alerts

- Upcoming features:
  - Reservation command support
  - Cancel command support
  - Native CLI
  - Events
  - HW accel.

- Also working on
  - CSI driver
  - Load balancing by load
  - Better integration with lib rbd

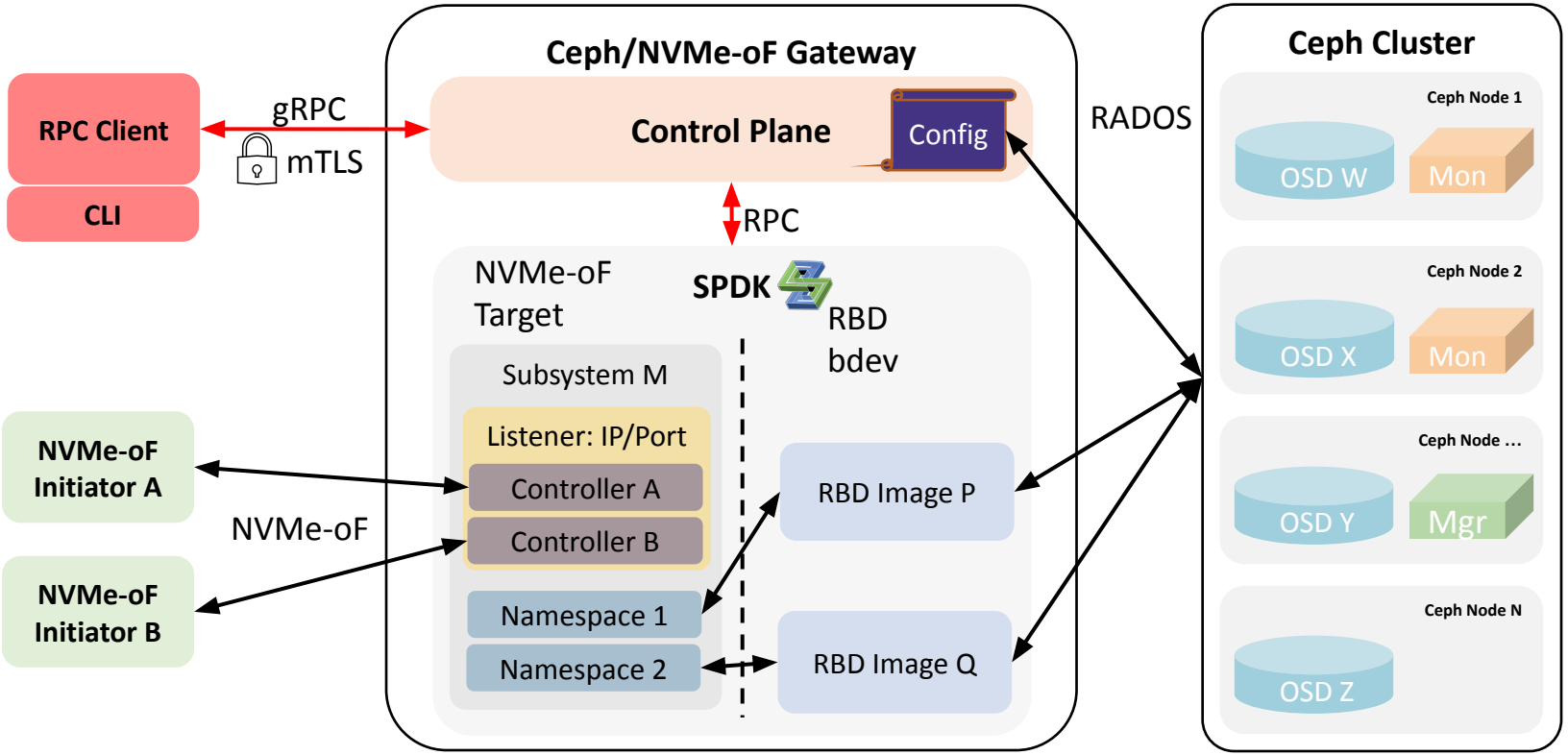
Ceph NVMe/TCP Gateway

- Multiple GWs can be deployed on the same Ceph cluster to provide HA, and Load balancing
- Multiple NVMe Subsystems to allow Access control
- Both Linux and ESX initiators



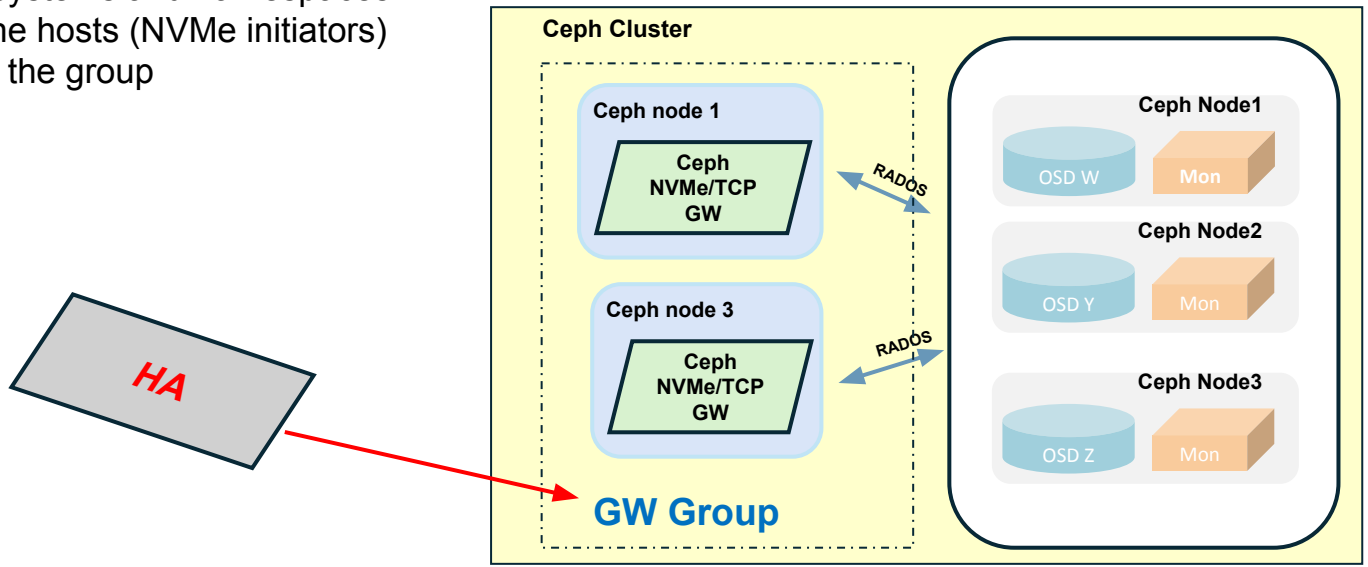
3

Ceph NVMe-oF Gateway



High Availability

- GWs are deployed as a part of a “GW group”
- HA is within the context of a “GW group”
- HA requires at least 2 Gateways in the group.
- All Gateways within the same GW group share the same configuration
  - All the NVMe Subsystems and namespaces are presented to the hosts (NVMe initiators) by all Gateways in the group



High Availability Connectivity

- Hosts must be connected to all of the Gateways in the GW group
- Each connected Gateway provides a different path to the Subsystem and Namespaces
- Only one active path to each namespace at a given time
- All other paths are in Stand-by and can become Active during a Failover

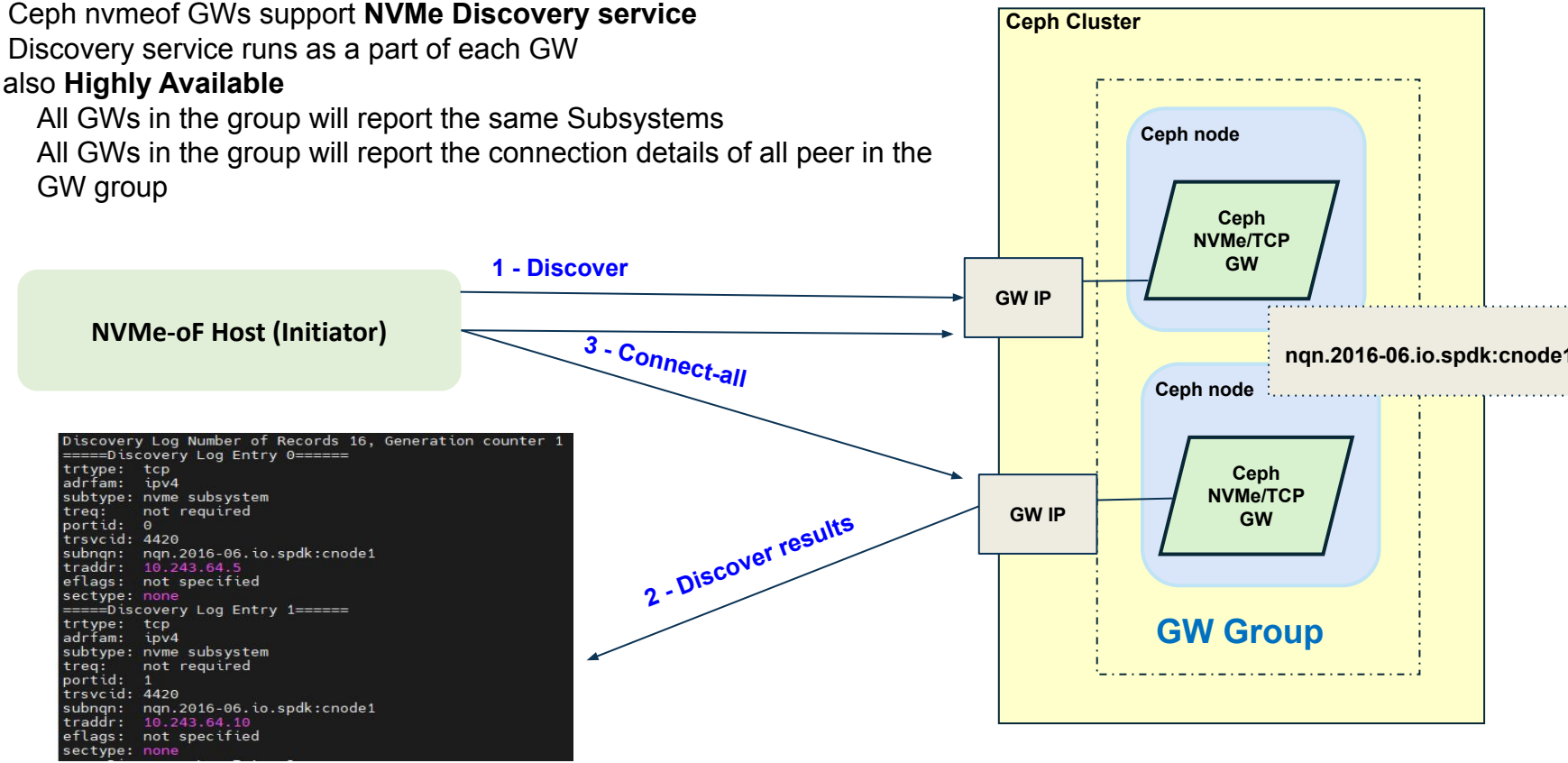
```
nvme-subsys5 - NQN=nqn.2016-06.io.spdk:cnode1
\
 +- nvme5 tcp traddr=10.243.64.5,trsvcid=4420 live
 +- nvme6 tcp traddr=10.243.64.10,trsvcid=4420 live
 +- nvme7 tcp traddr=10.243.64.11,trsvcid=4420 live
 +- nvme8 tcp traddr=10.243.64.12,trsvcid=4420 live
```

```
[root@init-nvme-vm5 ~]# nvme list-subsys /dev/nvme1n12
nvme-subsys1 - NQN=nqn.2016-06.io.spdk:cnode10
\
 +- nvme1 tcp traddr=10.243.64.5,trsvcid=4420 live inaccessible
 +- nvme2 tcp traddr=10.243.64.10,trsvcid=4420 live optimized
 +- nvme3 tcp traddr=10.243.64.11,trsvcid=4420 live inaccessible
 +- nvme4 tcp traddr=10.243.64.12,trsvcid=4420 live inaccessible
```

```
[root@ceph-nvme-vm4 ~]# ceph nvme-gw show mypool ''
{
    "epoch": 725,
    "pool": "mypool",
    "group": "",
    "num gws": 4,
    "Anagrp list": "[ 4 3 2 1 ]"
}
{
    "gw-id": "client.nvmeof.mypool.ceph-nvme-vm10.zaiihd",
    "anagrp-id": 4,
    "performed-full-startup": 1,
    "Availability": "AVAILABLE",
    "ana states": " 4: ACTIVE , 3: STANDBY , 2: STANDBY , 1: STANDBY ,"
}
{
    "gw-id": "client.nvmeof.mypool.ceph-nvme-vm2.fwdklo",
    "anagrp-id": 3,
    "performed-full-startup": 1,
    "Availability": "AVAILABLE",
    "ana states": " 4: STANDBY , 3: ACTIVE , 2: STANDBY , 1: STANDBY ,"
}
{
    "gw-id": "client.nvmeof.mypool.ceph-nvme-vm3.unrawc",
    "anagrp-id": 2,
    "performed-full-startup": 1,
    "Availability": "AVAILABLE",
    "ana states": " 4: STANDBY , 3: STANDBY , 2: ACTIVE , 1: STANDBY ,"
}
{
    "gw-id": "client.nvmeof.mypool.ceph-nvme-vm4.wyests",
    "anagrp-id": 1,
    "performed-full-startup": 1,
    "Availability": "AVAILABLE",
    "ana states": " 4: STANDBY , 3: STANDBY , 2: STANDBY , 1: ACTIVE ,"
}
```

Discovery

- The Ceph nvmeof GWs support **NVMe Discovery service**
- The Discovery service runs as a part of each GW
- It is also **Highly Available**
  - All GWs in the group will report the same Subsystems
  - All GWs in the group will report the connection details of all peer in the GW group



Security Features: TLS PSK

- Encrypt in transit data
- Lack of support on most NVMe-oF initiators (Linux RHEL 9, and ESX)
- Supported by the Ceph NVMe-oF GW and can be tested with the SPDK initiator (bdefperf)

1) Generate tls-key on the host

```
[root@init-nvme-vm5 ~]# nvme gen-tls-key
NVMeTLSkey-1:01:m13DYft49YC6pC9h/lr70280AnotttUjjyezSTcFNVJYSbZq:
```

2) Add host as a TLS-PSK host

```
[root@init-nvme-vm5 ~]# nvmeof-cli --server-address 10.243.64.12 host add -n nqn.2016-06.io.spdk:cnode1.mygroup1 --psk NVMeTLSkey-1:01:m13DYft49
.nvmexpress:uuid:6b0fbb86-7853-460a-8332-336b42c51e4b
Adding host nqn.2014-08.org.nvmexpress:uuid:6b0fbb86-7853-460a-8332-336b42c51e4b to nqn.2016-06.io.spdk:cnode1.mygroup1: Successful
```

3) Connect using the TLS Key

```
[root@init-nvme-vm5 ~]# nvme connect-all --traddr=10.243.64.12 --transport=tcp -l 1800 --tls_key=
```

Security Features: mTLS

- Secured and encrypted communication for Nvme-oF management interface (gRPC API)
- Client/server certs/keys defined in the nvmeof service spec file
- Requires to apply the new spec and redeploying the service
- When calling the gRPC API need to add:
  - Path to local Server Cert
  - Path to local Client Cert and Key

```
service_id: mypool.mygroceph orch ls nvmeof --export
                    service_type: nvmeof
service_id: mypool.mygroup1
service_name: nvmeof.mypool.mygroup1
placement:
  hosts:
  - ceph-nvme-vm4
  - ceph-nvme-vm3
  - ceph-nvme-vm2
  - ceph-nvme-vm10
spec:
  allowed_consecutive_spdk_ping_failures: 1
  bdevs_per_cluster: 32
  client_cert: '-----BEGIN CERTIFICATE-----
    MIIFBTCCAu2gAwIBAgIUZ2LmIB85hU1EqFEMf87ndUCDhr4wDQYJKoZIhvcNAQEL
    BQAwEjEQMA4GA1UEAwwHY2xpZW50MTAeFw0yNDExMjYwODE2MzFaFw0zNDExMjQw
        ...
    -----END CERTIFICATE-----
    '
  client_key: '-----BEGIN PRIVATE KEY-----
    MIIJQgIBADANBgkqhkiG9w0BAQEFAASCCSwwggkoAgEAAoICAQCKNO/JHgwpyjjr
    PqYXZKCpfraiK5EaMsaP38skBi+zeE331t6SJ3kQBpV/PRPa4AO3ULjQBvfut86T
    ...
        -----END PRIVATE KEY-----
        '
  conn_retries: 10
  discovery_port: 8009
  enable_auth: true
  enable_monitor_client: true
  enable_prometheus_exporter: true
  group: mygroup1
  log_directory: /var/log/ceph/
  log_files_enabled: true
  log_files_rotation_enabled: true
  log_level: INFO
  max_log_directory_backups: 10
  max_log_file_size_in_mb: 10
  max_log_files_count: 20
  monitor_timeout: 1.0
  omap_file_lock_duration: 20
  omap_file_lock_retries: 30
  omap_file_lock_retry_sleep_interval: 1.0
  omap_file_update_reloads: 10
  pool: mypool
  port: 5500
  root_ca_cert: mountcert
  rpc_socket_dir: /var/tmp/
  rpc_socket_name: spdk.sock
  server_cert: '-----BEGIN CERTIFICATE-----
    MIIFLDCCAxSgAwIBAgIUffN6M0Z5r/eDFHl8bkt4RDcJTmYwDQYJKoZIhvcNAQEL
    BQAwFDESMBAGA1UEAwwJbXkuc2VydmVyMB4XDTI0MTEyNjA4MTYyOVoXDTM0MTEy
```

Quality of Service (QoS)

Volume QoS limit per Namespace

- I/Os per second
- MB/s
- Read MB/s
- Write MB/s
- Per GW - not global

```
[root@init-nvme-vm5 ~]# nvmeof-cli --server-address 10.243.64.12 namespace list -n nqn.2016-06.io.spdk:cnode2.mygroup1
Namespaces in subsystem nqn.2016-06.io.spdk:cnode2.mygroup1:
```

NSID	Bdev Name	RBD Image	Image Size	Block Size	UUID	Load Balancing Group	Visibility	R/W IOs per second	R/W MBs per second	Read MBs per second	Write MBs per second
1	bdev_1f5754e2-dc34-44d6-a034-a13d069a5431	mypool/myimage1	200 MiB	512 Bytes	1f5754e2-dc34-44d6-a034-a13d069a5431	1	All Hosts	unlimited	unlimited	20	100
2	bdev_16fbd8cf-e7f6-4962-96aa-120a4798604e	mypool/myimage2	200 MiB	512 Bytes	16fbd8cf-e7f6-4962-96aa-120a4798604e	2	All Hosts	1000	160	unlimited	unlimited
3	bdev_55bd3c6b-41da-437a-b3a1-6272a190bf5e	mypool/myimage3	200 MiB	512 Bytes	55bd3c6b-41da-437a-b3a1-6272a190bf5e	3	All Hosts	unlimited	unlimited	200	10
4	bdev_a6f1ab53-5c3d-4b8a-954e-220563983588	mypool/myimage4	200 MiB	512 Bytes	a6f1ab53-5c3d-4b8a-954e-220563983588	4	All Hosts	2000	320	unlimited	unlimited
5	bdev_2816314a-7160-4b8c-92b9-e4cc2fd6c41c	mypool/myimage5	200 MiB	512 Bytes	2816314a-7160-4b8c-92b9-e4cc2fd6c41c	1	All Hosts	unlimited	unlimited	26	8
6	bdev_d751c389-1a80-40fe-bfb6-927531e711af	mypool/myimage6	200 MiB	512 Bytes	d751c389-1a80-40fe-bfb6-927531e711af	2	All Hosts	unlimited	unlimited	unlimited	unlimited
7	bdev_701254a6-744d-46b8-8725-4d17c8b99c64	mypool/myimage7	200 MiB	512 Bytes	701254a6-744d-46b8-8725-4d17c8b99c64	3	All Hosts	unlimited	unlimited	unlimited	unlimited
8	bdev_16a8e4b2-9cec-4454-b43a-1161f5bb7229	mypool/myimage8	200 MiB	512 Bytes	16a8e4b2-9cec-4454-b43a-1161f5bb7229	4	All Hosts	unlimited	unlimited	unlimited	unlimited
9	bdev_af19df31-cd0b-44aa-b638-8bce4f6f7f73	mypool/myimage9	200 MiB	512 Bytes	af19df31-cd0b-44aa-b638-8bce4f6f7f73	1	All Hosts	unlimited	unlimited	unlimited	unlimited
10	bdev_2dfe403e-0f5d-	mypool/myimage10	200 MiB	512 Bytes	2dfe403e-0f5d-4ee5-	2	All Hosts	unlimited	unlimited	20	100

Scale

	Tentacle
Gateways in a GW group	8
GW groups in a cluster	4
Subsystems in a GW group	128
Namespaces for a GW/GW group	2048
Namespaces in a Cluster	8192
Hosts per Subsystems	Up to 128
Hosts per GW group	At least 512 hosts

New Features

Cluster contexts allocation policy

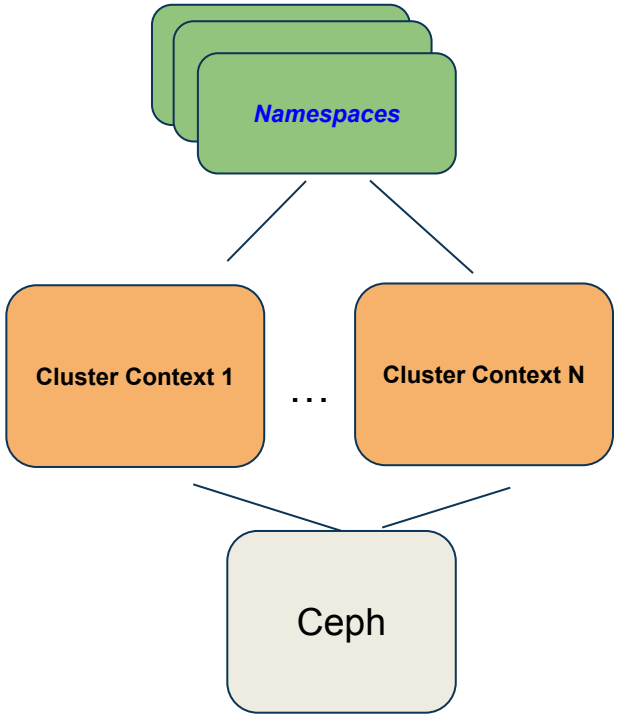
The NVMeoF GW is a client of Ceph. It requires RADOS Cluster Contexts to be able to send IOs to Ceph.

In the older implementation:

- The number of Cluster Contexts depends on the number of Namespaces
- Drawbacks:
  - Small number of ns => GW will utilize few Cluster contexts
  - Big number of ns => GW will utilize too many Cluster contexts
  - User needs to tune the number of Namespaces per Cluster Context

In the new implementation:

- Max total number of Cluster Contexts is defined
- The allocation algorithm will equally divide the number of Namespaces on the Cluster Contexts.
- Advantages:
  - Load balancing is achieved also with minimal number of Namespaces
  - No user intervention is required
  - Max Cluster Contexts is known

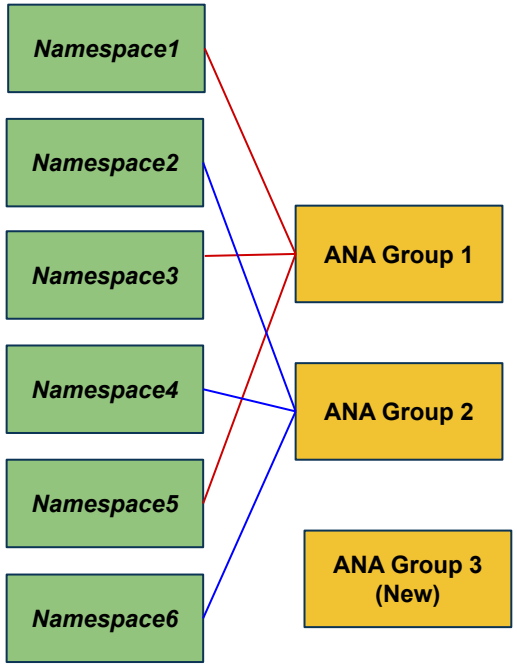


Auto namespaces load balancing

- Why is it required
  - Each GW in the group is Optimized on 1 ANA group
  - Each GW is running an instance of SPDK, which is using 1..N cores
  - The Namespaces are equally distributed across the ANA groups
  - When adding/removing a GW to the group, a new ANA group is added/removed
  - It is required that the number of Namespaces will be equally distributed

- In a Scale Up scenario - there is a new GW available to take some load, but the Namespaces are assigned to N-1 GWs.

- In a Scale Down scenario - there is one less GW available to take the load, and some Namespaces are assigned to a non-existent ANA group.



Auto namespaces load balancing

- The solution is slightly different between Scale Up and Scale Down.

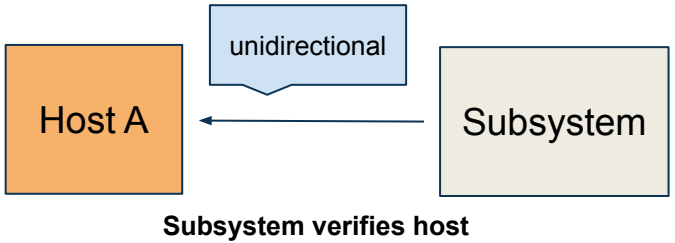
- In Scale Down
  - Failover the Orphan ANA Group => no DU situation.

- In Scale Down/Scale Up
  - A background process, reassigning new ANA groups to the Namespaces such that eventually, all of the Namespaces, Per Subsystem, are equally divided between the ANA groups.
  - The Reassignment of the Namespace to a different ANA group is non disruptive.
  - If a Namespace is owned by any of the GWs, only this GW is allow to give up and move it on.

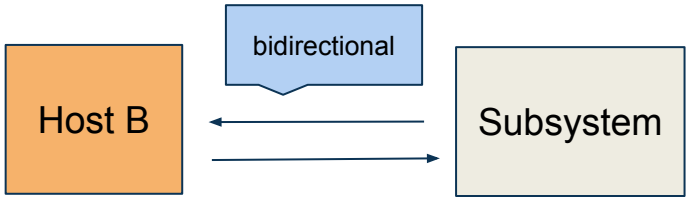
- How does it work
  - Controlled by the nvmeof monitor
  - Every GW in turn gets few seconds to move some namespaces
  - This process will continue until a perfect balance is reached

Security - inband auth.

- For any Subsystem and host pairs
  - It is possible to define unidirectional or bidirectional authentication
  - Unidirectional means that the Subsystem will verify the host
  - Bidirectional means that the Subsystem will verify the host, and the host will verify the Subsystem
  -
- User specifies if a Subsystem requires Authentication during Subsystem creation
  - The Key is saved encrypted per Subsystem in the GW group state OMAP file
  - In Subsystem list command, the Subsystem auth status is shown
- User use the add host command and provides the host key
  - The host Key is mandatory in case that the Subsystem is defined with a Key (bidirectional authentication) .
  - The host Key is optional in case the subsystem is not defined with a Key
    - If provided that means unidirectional authentication.
- In nvme connect command, user needs to provide the host and subsystem keys to be able to connect.



Subsystem verifies host



Subsystem verifies host
Host verifies Subsystem

Security - Namespace masking

- By default, host will access all of the namespaces in a Subsystem it is connected to (if Authorized)
- In some cases it is required to restrict the access to specific Namespaces

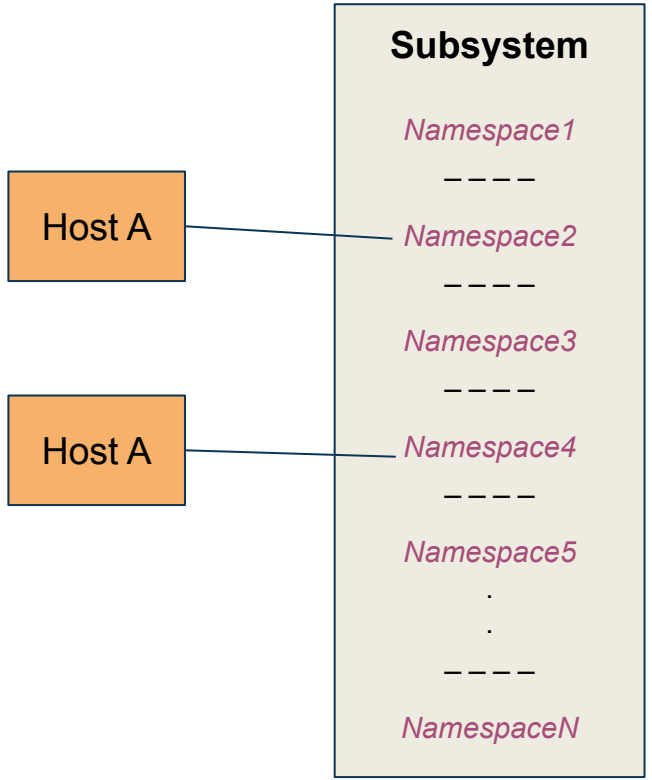
Implementation

- By default all of the namespaces in the subsystem remain visible, but if a namespace is created with the flag of “Visibility=Selective”, then only selected hosts will be able to access the namespace.

- To define that selected host can access a namespace, the user needs to use a new cli - “namespace add_host” command.

Restrictions

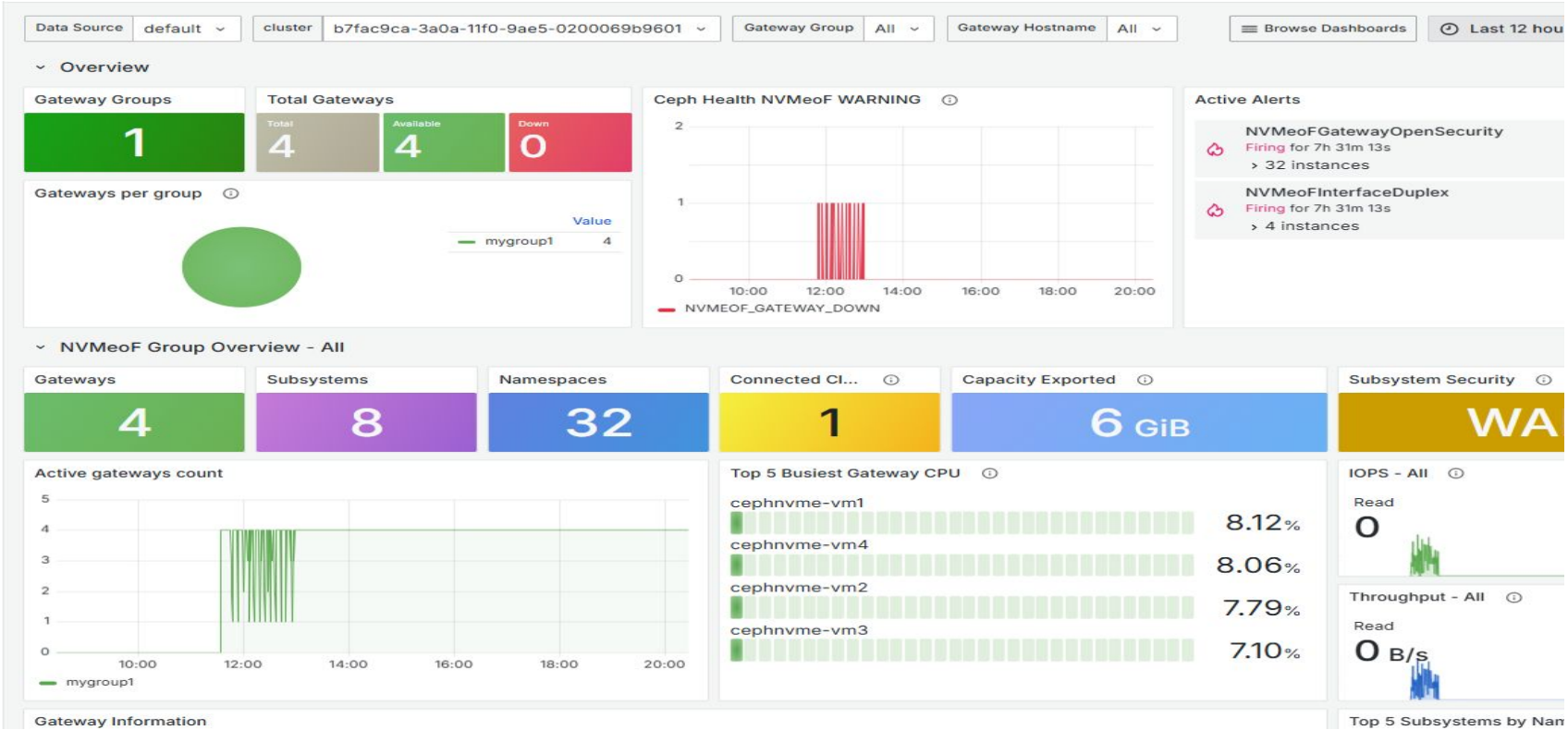
- There is a limitation on the total number of namespaces that can be “selective”
- There is a limitation to 8 hosts that can be selective per namespace



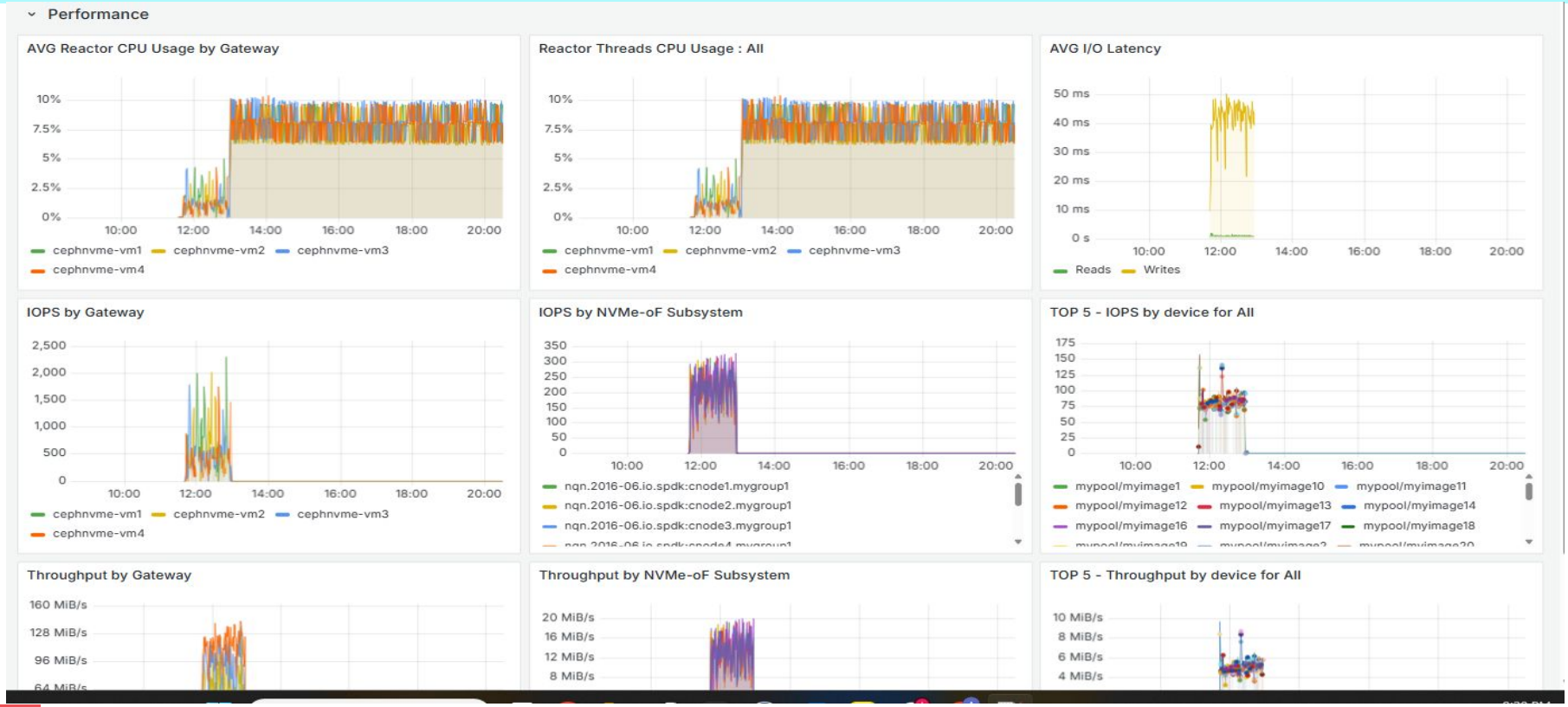
Grafana Dashboards

- Added 2 new Grafana Dashboards
  - One for a general overview of the Deployment/GWs/Subsystems/Namespaces/etc.
  - Another once for Performance analysis - throughput/latency/etc.

Dashboard - NVMe-oF Gateways - Overview

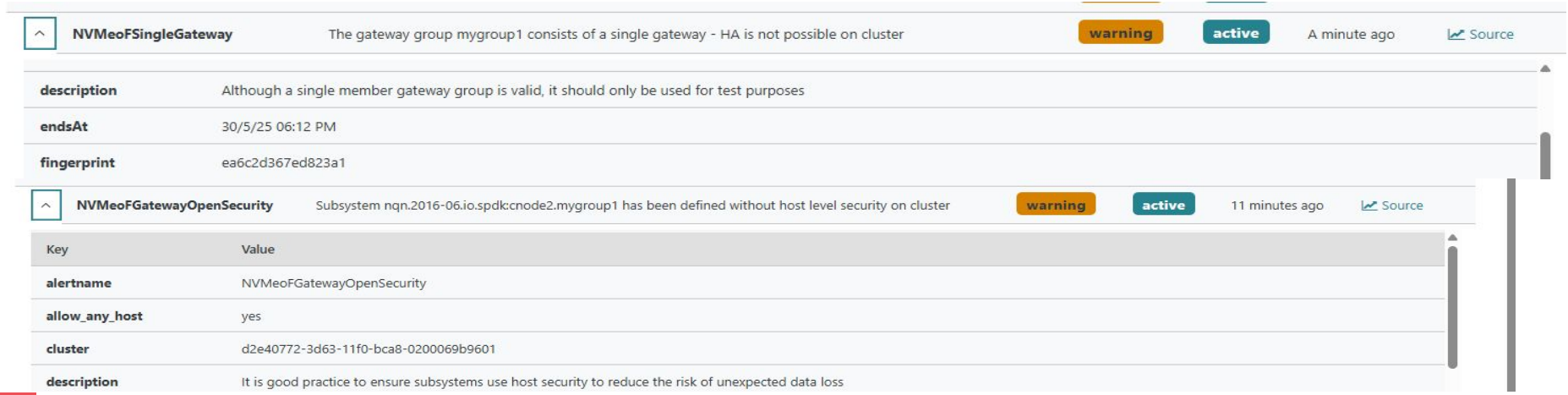


Dashboard -NVMe-oF Gateways - Performance



Dashboard Alerts/Warnings

- Added some new Alerts and Warnings
  - GW group with a single Gateway Node
  - Missing Listeners or Unbalanced number of listeners
  - Gateway in DELETING state
  - same pool/image is used for more than 1 namespace
  - gateway in UNAVAILABLE state



Work in progress

Ceph Native CLI

- Currently the NVMe-oF GW CLI is in a separate container
- The Native CLI will be a part of the Ceph command line
- How its implemented:
  - Ceph API already includes the call to the GW gRPC interface
  - Added a CLI decorator to register the GWs API also as Ceph CLI
  -

```
[root@cephnvme-vm29 ~]# ceph nvmeof gw info
+-----------+-------------+-------+-----------------------------------------------------+--------+------------+
|Bool Status|Hostname     |Version|Name                                                 |Group   |Addr        |
+-----------+-------------+-------+-----------------------------------------------------+--------+------------+
|True       |cephnvme-vm29|1.5.3  |client.nvmeof.mypool.mygroup1.cephnvme-vm29.itfihp|mygroup1|10.242.64.51|
+-----------+-------------+-------+-----------------------------------------------------+--------+------------+
```

```
[root@cephnvme-vm29 ~]# ceph nvmeof subsystem list
+--------------------------------+---------------+-------------------+--------------+-------+--------------+--------------+--------------+---------------
|Nqn                             |Serial Number  |Model Number       |Namespace Count|Subtype|Max Namespaces|Has Dhchap Key|Allow Any Host|Created Without
+--------------------------------+---------------+-------------------+--------------+-------+--------------+--------------+--------------+---------------
|nqn.2016-06.io.spdk:cnode1.mygroup1|Ceph89703713501961|Ceph bdev Controller|4             |NVMe   |1024          |False         |True          |False
|nqn.2016-06.io.spdk:cnode2.mygroup1|Ceph33653287294662|Ceph bdev Controller|4             |NVMe   |1024          |False         |True          |False
+--------------------------------+---------------+-------------------+--------------+-------+--------------+--------------+--------------+---------------
```

NVMe Reservation

- NVMe reservation commands are primarily used by hosts that need to coordinate access to a shared NVMe namespace
- For example, Windows Server Failover Clustering (WSFC) with virtual machines
- SPDK already supports nvmeof reservation but it requires some customization to support it for a group of GWs
- The way it is done in the NVMe-oF GW:
  - Keep the reservation meta-data in the rbd image metadata section (the same image that backs up the namespace)
  - The metadata is saved as Key/Value. The value is a json that includes the reservation information
  - Take advantage of the SPDK/RBD Image watch mechanism to Load the updated reservation information on all GWs

NVMe Cancel

- Implement Cancel command in SPDK as defined in TP 4097
- This Cancel command is a better way to Abort commands
  - The Abort command has some issues, mainly because it is done via the Controller Admin queue and the number of commands that can be handled simultaneously is low (~32).
  - The Cancel command is done on the IO queues and there is no limit to the number of commands that can be handled concurrently
- The Cancel is a best effort command
- Commands will be Cancelled if they're still in the SPDK queues
- The command will be sent as all other IO commands - on the Optimized path
- Command can be set to
  - A specific Namespace commands exist in the Queue
  - All namespaces commands exist in the Queue
- The action can be to
  - Cancel a specific IO command
  - Cancel all of the commands

More features WIP

SPDK Events

- SPDK events are currently only logged into the log files
- Events such as host keep alive timeout, or not enough memory to establish more QPs, and more
- The idea is to catch these events in the SPDK code and raise it as Alerts in the Ceph dashboard

HW Acceleration

- Introduce a support for Intel's Data Streaming Accelerator (DSA) in the GW
- Requires some build steps and configuration logic to allow SPDK to discover, configure and utilize the DSA devices
- The plan is to offload the CRC calculations

Failover time

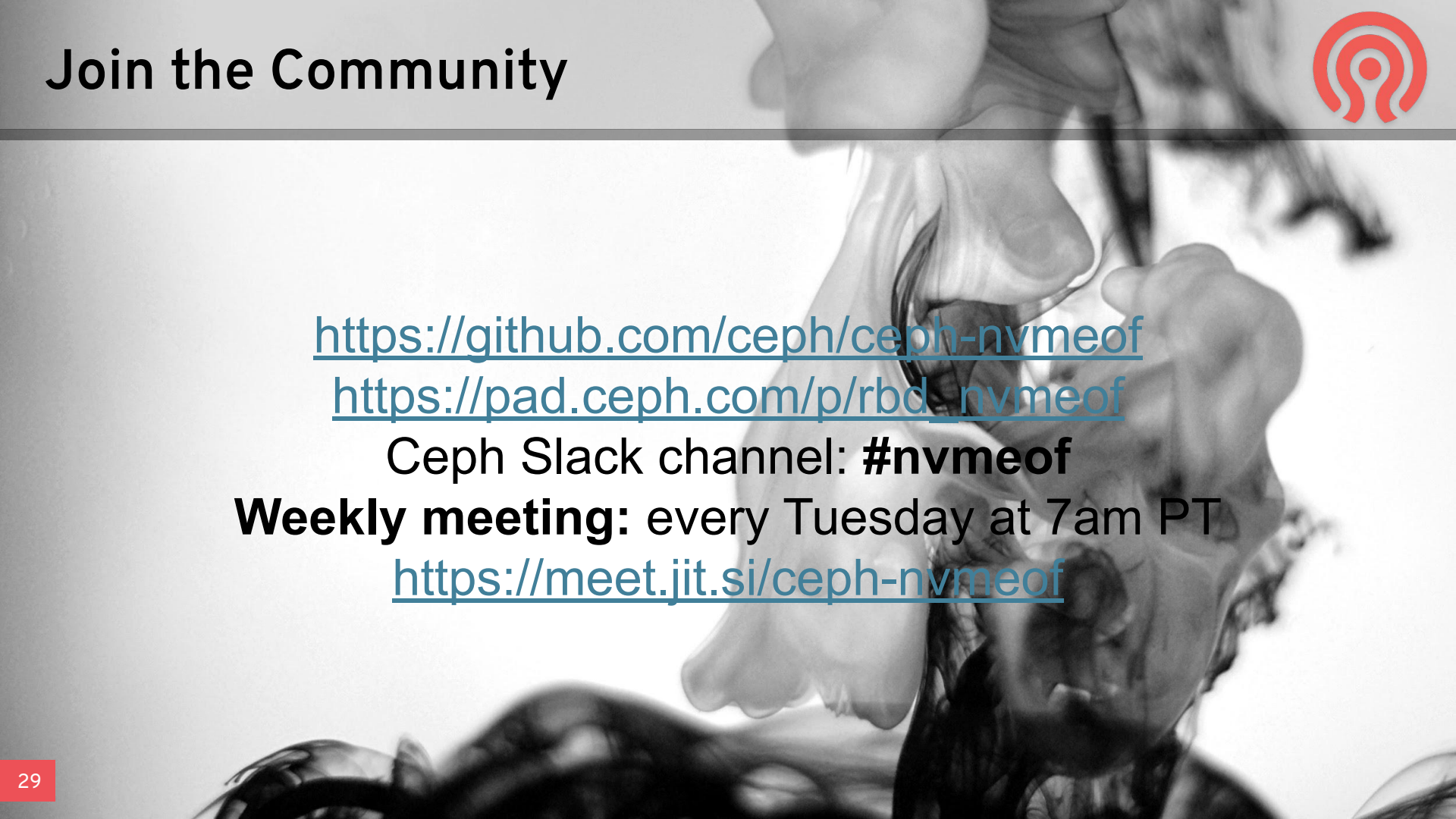
- Shortening Failover time from ~14 seconds down to ~6-8
- Mainly a change in the monitor behaviour.
- Tune the beacons timeout, possibly send more beacons in a shorter time

More things WIP

- CSI driver
- Stretch cluster optimization
- Usability - auto listeners, host groups
- Better integration with rbd (for performance)
- Load balancing based on real load
- …

Q&A

Join the Community

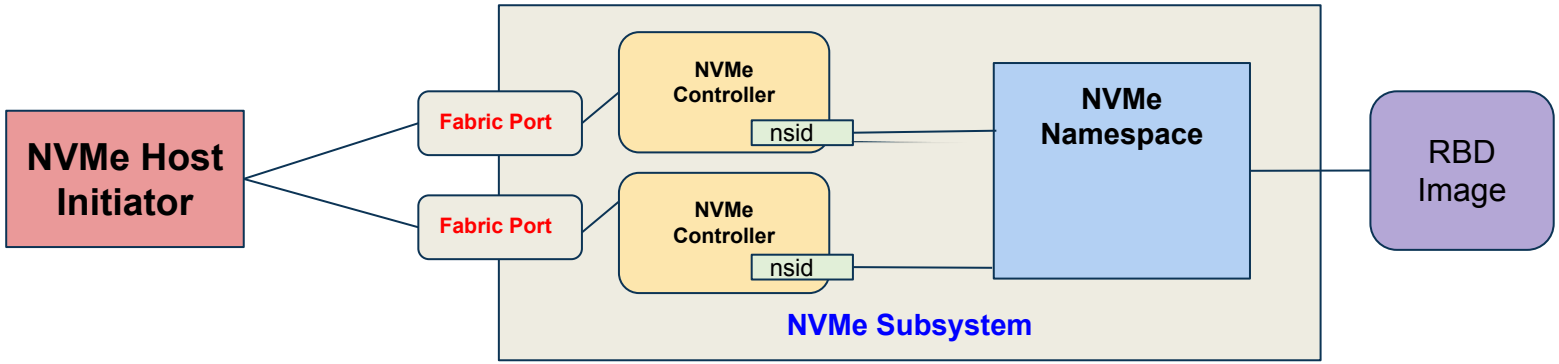


https://github.com/ceph/ceph-nvmeof
https://pad.ceph.com/p/rbd_nvmeof
Ceph Slack channel: **#nvmeof**
Weekly meeting: every Tuesday at 7am PT
https://meet.jit.si/ceph-nvmeof

Backup

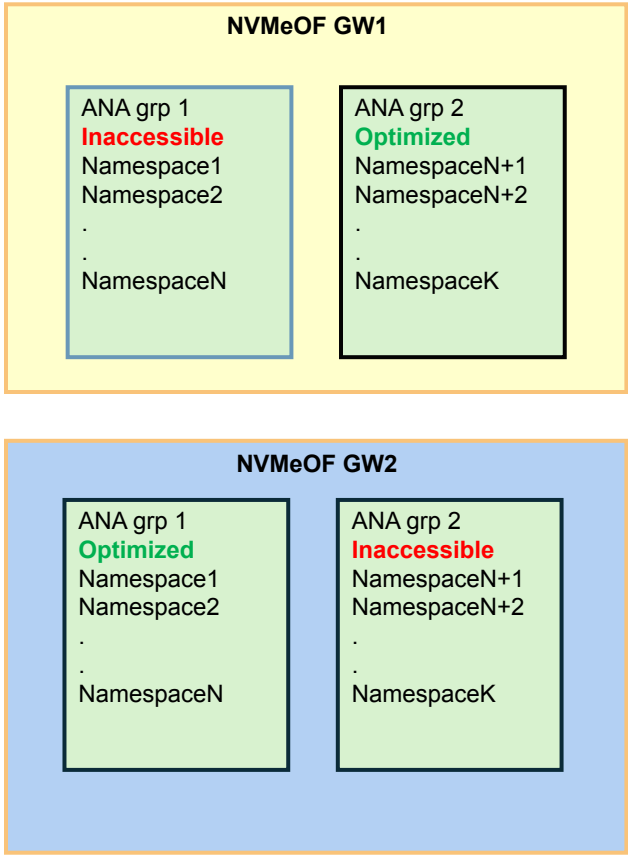
NVMe basic terminology

- **Namespace**
  - NVMe equivalent to FC and iSCSI LUNs (can be thought as a Volume)
  - Defined as a collection of LBAs (Logical block addresses)
  - In Ceph Nvme-oF GW a namespace is mapped to an RBD Image
- **NVMe Subsystem**
  - An entity that contains **Namespaces** and other NVMe elements such as **NVMe controllers**
  - Identified by an **NQN** (NVMe Qualified Name)
  - The Initiator connects to target IP/NVMe Subsystem
- **NVMe IO Controller**
  - Created for every connection between a the host and a target NVMe-oF fabric address per Subsystem.
  - Receives and processes NVMe commands sent over the network



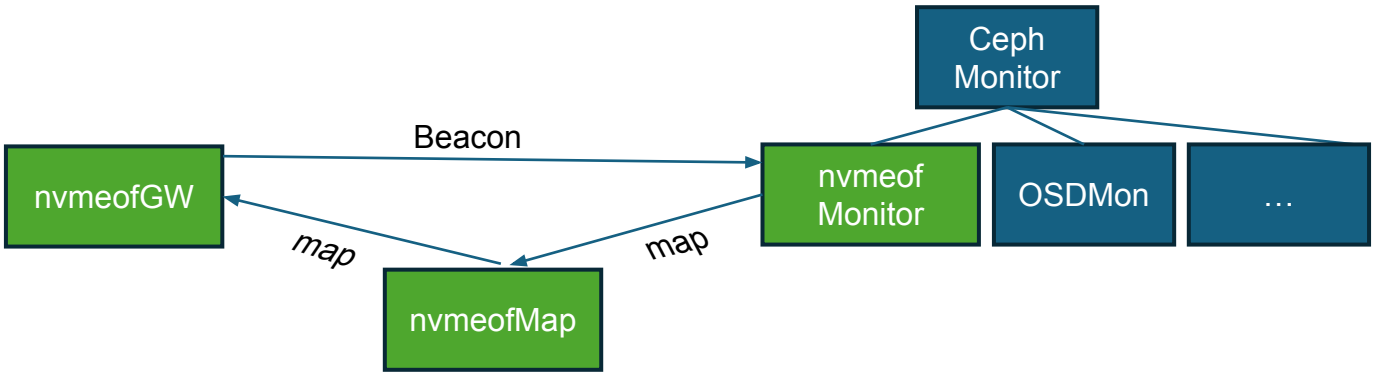
ANA (Asymmetric Namespace Access)

- Ceph NVMe/TCP HA is using NVMe ANA protocol to define the Optimized and and Inaccessible properties of the namespaces.
- Each live GW in the group owns one ANA group.
- The namespaces are divided between the ANA groups.
- In a Failover scenario one of the surviving GWs will own the failed GW ANA group (in addition to his).



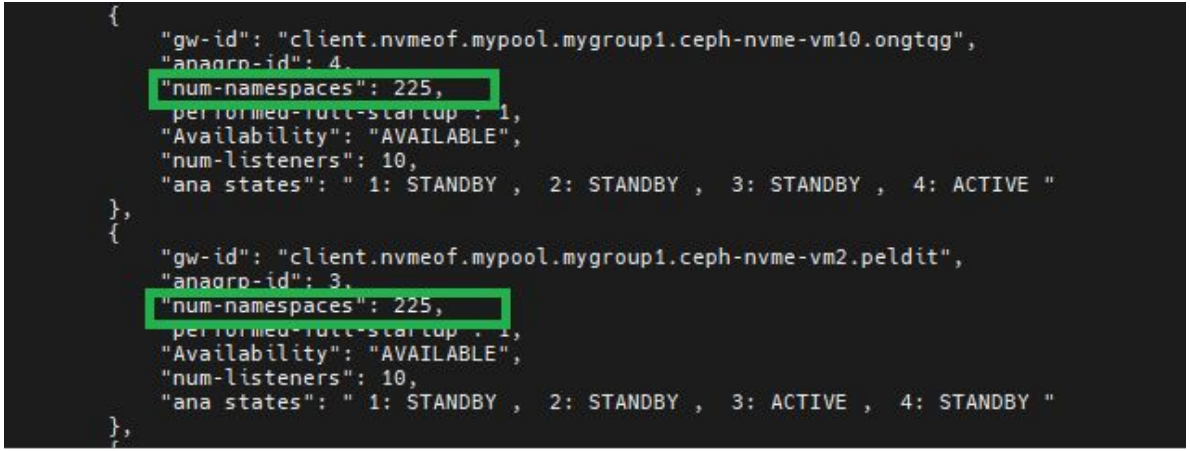
Ceph NVMe/TCP Monitor

- Assign an ANA group ID to each gateway
- Reassign as need in a Failover/Failback scenarios
- Monitor client process in each gateway sends Beacons to the nvmeof monitor
- Decide to perform Failover in case it is not getting Beacons from the client for a while
- Decide to perform Failback in case it started getting Beacons from a GW that was considered as Unavailable



Load Balancing

- Handle Gateway removal
  - Failover the ownership of the namespaces in the Gateway (ANA grp), to one of the remaining Gateways in the cluster
- Rebalance
  - Redistribute the namespaces between the Gateways and new Gateways
  - Currently in main, user need to do it manually
  - Tentacle: Automatic background process that will move the namespace between the Gateways



NVMe-oF CLI

Currently: A separate CLI in a separate container

```
#podman run -it cp.icr.io/cp/ibm-ceph/nvmeof-cli-rhel9:latest --server-address $ip_of_node --server-port 5500
create_subsystem --subnqn $nqn --max-namespaces 256
```

Future: NVMe-oF CLI as a part of Ceph CLI

```
#ceph nvmeof --server $server --port $port create_subsystem --subnqn $nqn --max-namespaces 256
```

Security Features: SubSystem Masking

- Control which host can connect to a subsystem
- NQN based

```
[root@init-nvme-vm5 ~]# nvmeof-cli --server-address 10.243.64.12 host list -n nqn.2016-06.io.spdk:cnode1.mygroup1
Hosts allowed to access nqn.2016-06.io.spdk:cnode1.mygroup1:
```

Host NQN	Uses PSK
Any host	n/a

All hosts allowed to access this subsystem

Only host ‘nqn.2014-08.org.nvmexpress:uuid:6b0fbb86-...’ allowed to access this subsystem

```
[root@init-nvme-vm5 ~]# nvmeof-cli --server-address 10.243.64.12 host list -n nqn.2016-06.io.spdk:cnode1.mygroup1
Hosts allowed to access nqn.2016-06.io.spdk:cnode1.mygroup1:
```

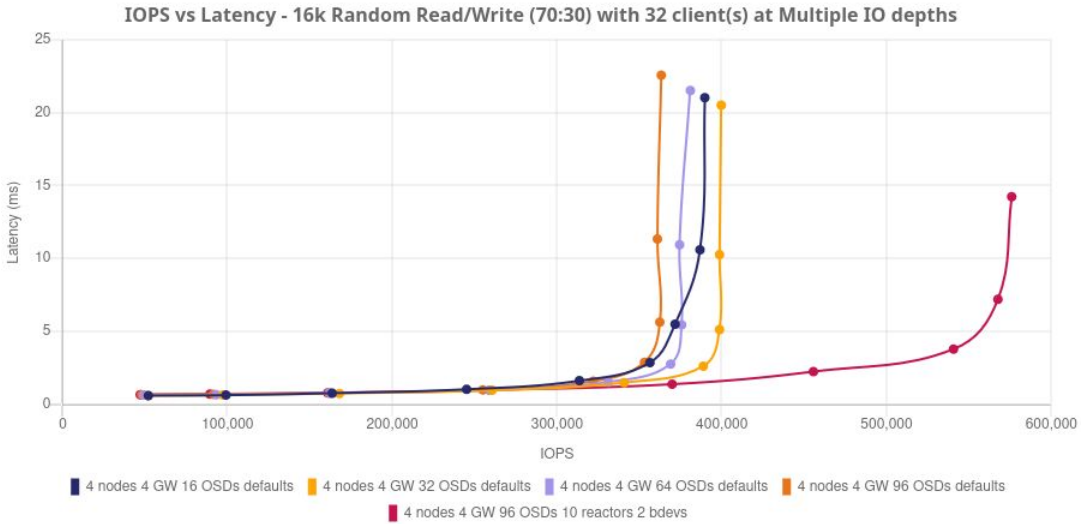
Host NQN	Uses PSK	Uses DHCHAP
nqn.2014-08.org.nvmexpress:uuid:6b0fbb86-7853-460a-8332-336b42c51e4d	No	No

Performance: Setup

- **Hardware:**
  - Ceph:
    - All-Flash IBM Storage Ready Nodes (Dell R760 X5D)
      - 2 x Intel(R) Xeon(R) Gold 6438N (32c/64T)
      - 512 GB
  - Client/Workload:
    - Dell R660
      - 2 x Intel(R) Xeon(R) Gold 5418Y (24c/48T)
      - 384 GB
- **Software:**
  - RHEL 9.4 (5.14.0-427.37.1.el9_4.x86_64)
  - Ceph 18.2.1-229.el9cp (IBM Ceph 7.1)

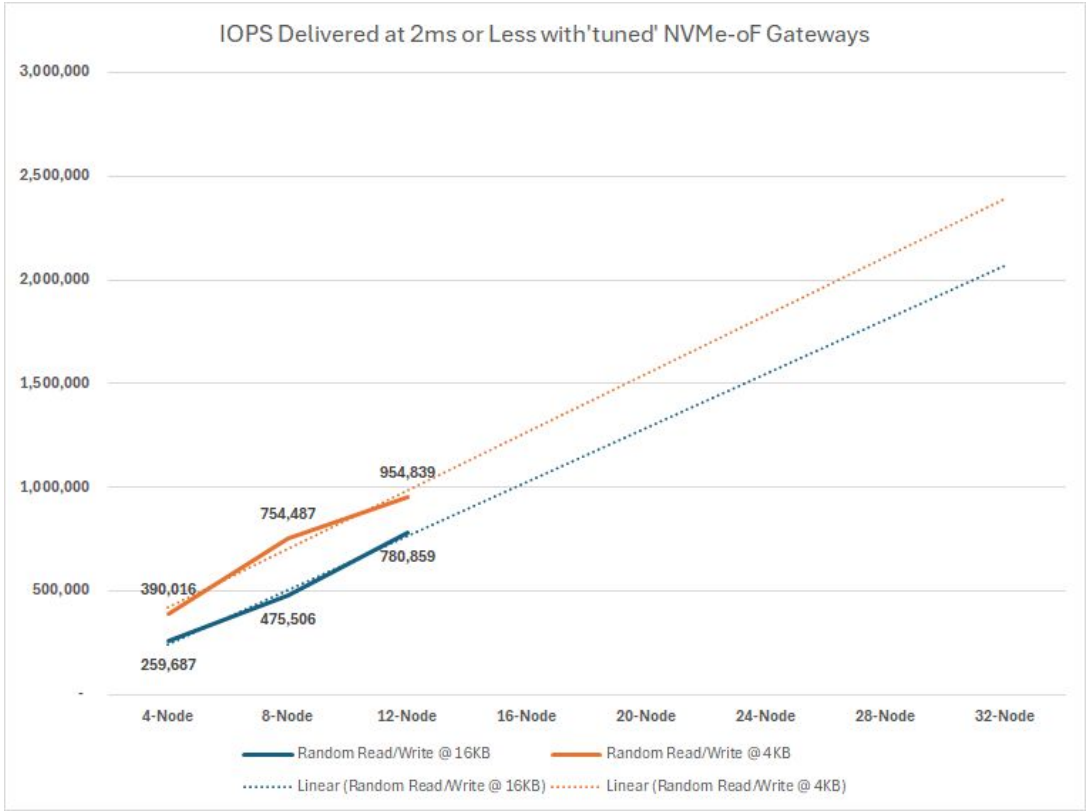
Performance Results

- Num of bdev = num of Ceph Context
  - 5 threads: 2 libRBD and 3 MSGR per context
- Defaults:
  - 4 reactors
  - *bdevs_per_cluster: 32*
- Performance profile:
  - 10 reactors
  - *bdevs_per_cluster:* 2



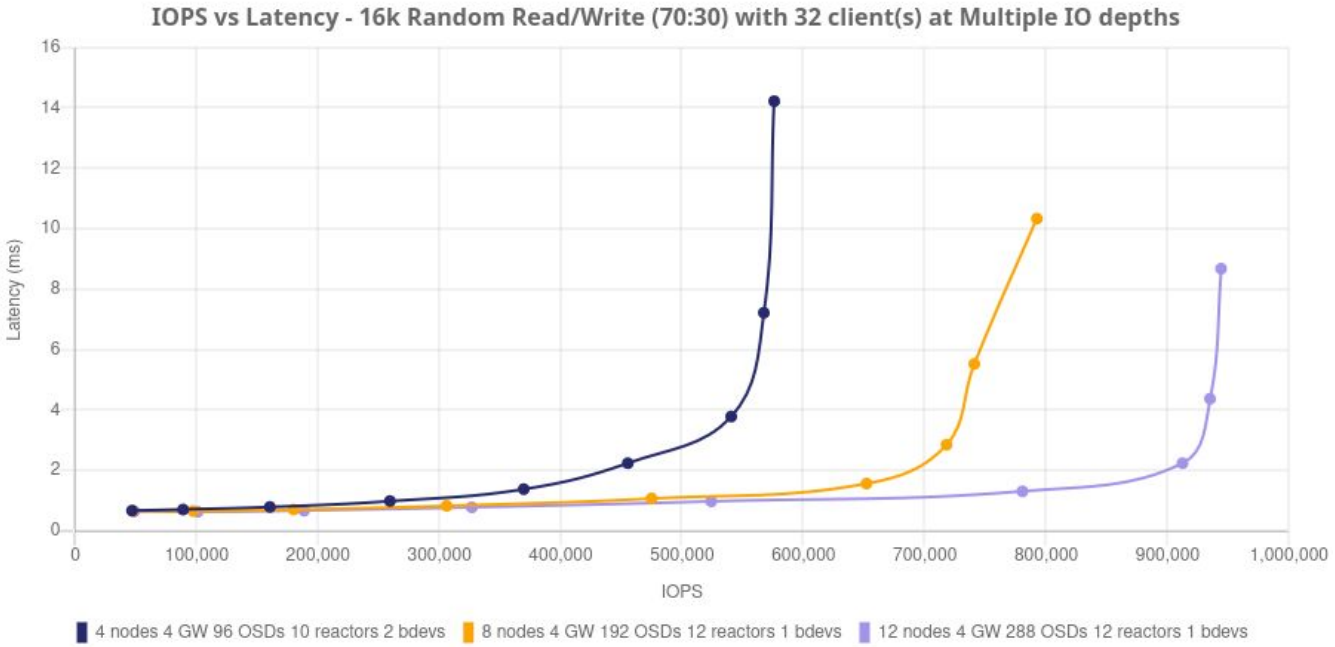
Performance Results

- IOPS scale linearly with the cluster size
- 16K block size provide better results



Performance Results

- Larger cluster provide more stable latency and throughput



Performance Enhancements

- Update the allocation algorithm of cluster contexts per namespaces
  - Currently per ANA group
- Reduce number of threads
  - ThreadPool
  - LibRBD reactor model to reuse SPDK reactors
- Network improvements (MSGR):
  - Allocate more threads for the MSGR
    - currently the default of 3 per ceph context
  - Threadpool
  - Reduce CPU the MSGR consumption
    - Reduce data copies
  - Zero Copy