IDUG

IDUG
2022 NA **Db2** Tech Conference

Running Db2 pureScale in a public cloud

Dale McInnis, IBM Canada Ltd.

Session Code: C7

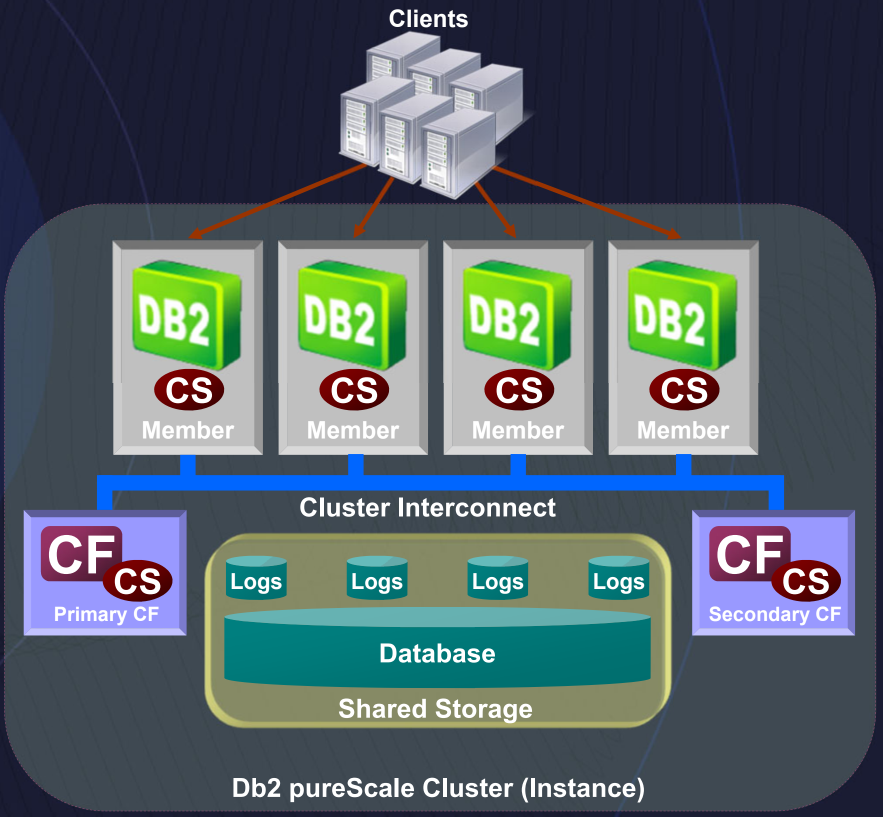
Boston, MA

Agenda

- What is pureScale
- Meeting pureScale requirements
- Allocating the environment
- Setting up pureScale

Db2 pureScale Architecture

Leveraging IBM's System z Sysplex Experience and Know-How



- Multiple Db2 members for scalable and available database environment
- Client application connects into any Db2 member to execute transactions
  - Automatic workload balancing
- Shared storage for database data and transaction logs
- Cluster caching facilities (CF) provide centralized global locking and page
  cache management for highest levels of availability and scalability
  - Duplexed, for no single point of failure
- High speed, low latency interconnect for efficient and scalable communication between members and CFs
- Db2 Cluster Services provides integrated failure detection, recovery automation and the clustered file system

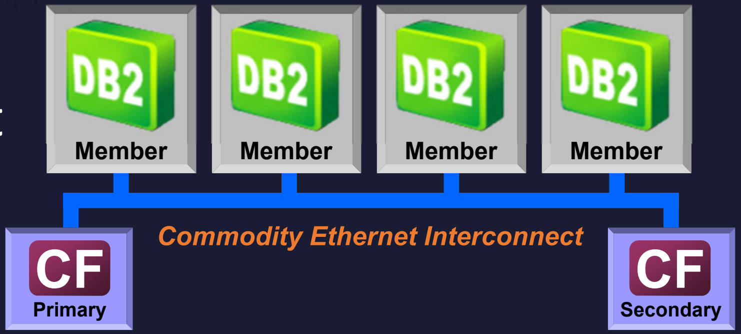
Db2 pureScale with TCP/IP Interconnect

- TCP/IP (sockets) interconnect for faster cluster setup and lower cost deployments using commodity network hardware
  - RDMA adapters no longer required (but still recommended for workloads with strong performance and scalability requirements)

- Provides exactly the same level of high availability as RDMA-based pureScale environments

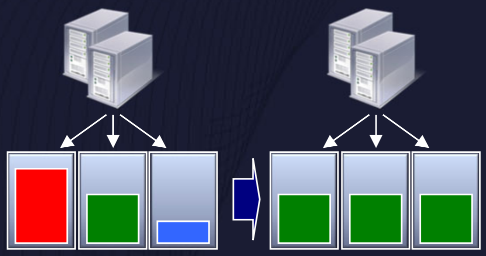
- Appropriate for small clusters with moderate workloads where availability is the primary motivator for pureScale

- Minimum 10 Gigabit Ethernet strongly recommended for production installations

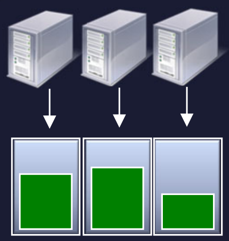


pureScale Client Configuration

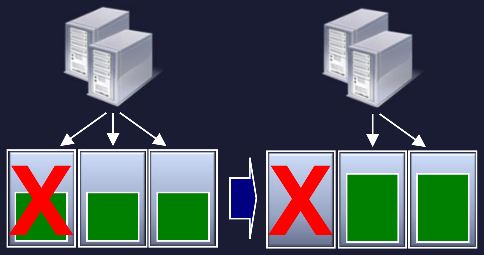
- Workload Balancing (WLB)
  - Application requests balanced across all members or subsets of members
  - Takes server load of members into consideration
  - Connection-level or transaction-level balancing



- Client Affinity
  - Direct different groups of clients or workloads to specific members in the cluster
  - Consolidate separate workloads/applications on same database infrastructure
  - Define list of members for failover purposes



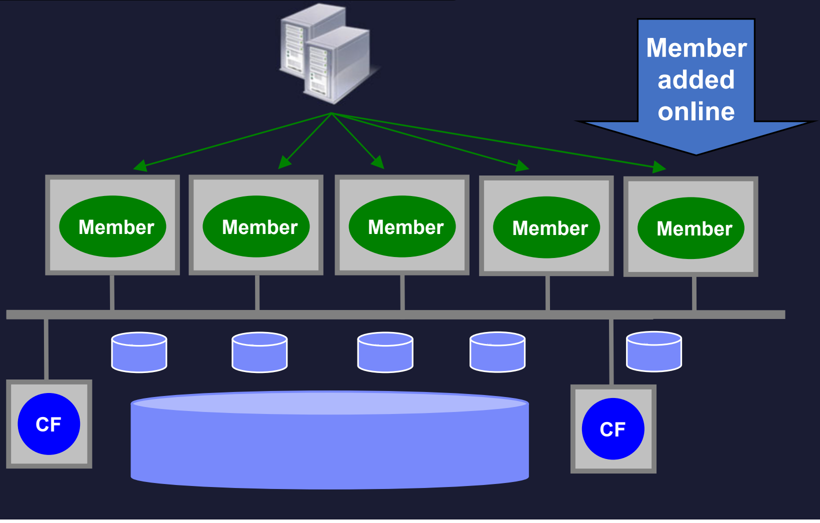
- Automatic Client Reroute (ACR)
  - Client automatically connected to healthy member in case of member failure
  - May be seamless in that no error messages returned to client
  - Application may have to re-execute the transaction



Online Add Member

- New members can be added to an instance while it is online
  - No impact to workloads running on existing members
  - Previously, required an outage of the entire instance to add a new member
- No change in add member command

```
db2iupdt -add -m <newHost> -mnet <networkName> <instance>
```

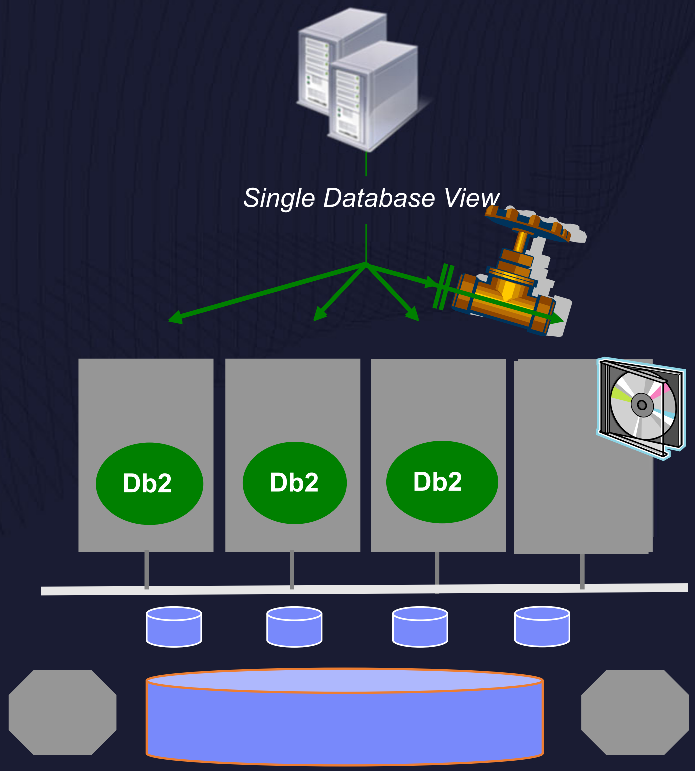


Stealth System Maintenance

- **Allows DBAs to apply system maintenance without negotiating an outage window**

- **Example: Upgrade the OS in a rolling fashion across the cluster**

- **Procedure:**
  1. Drain (a.k.a. Quiesce)
     - Wait for transactions to end there life naturally; new transactions routed to other members
  2. Remove & maintain
  3. Reintegrate into cluster
     - Workload balancing starts sending it work as a least loaded machine
  4. Repeat until done

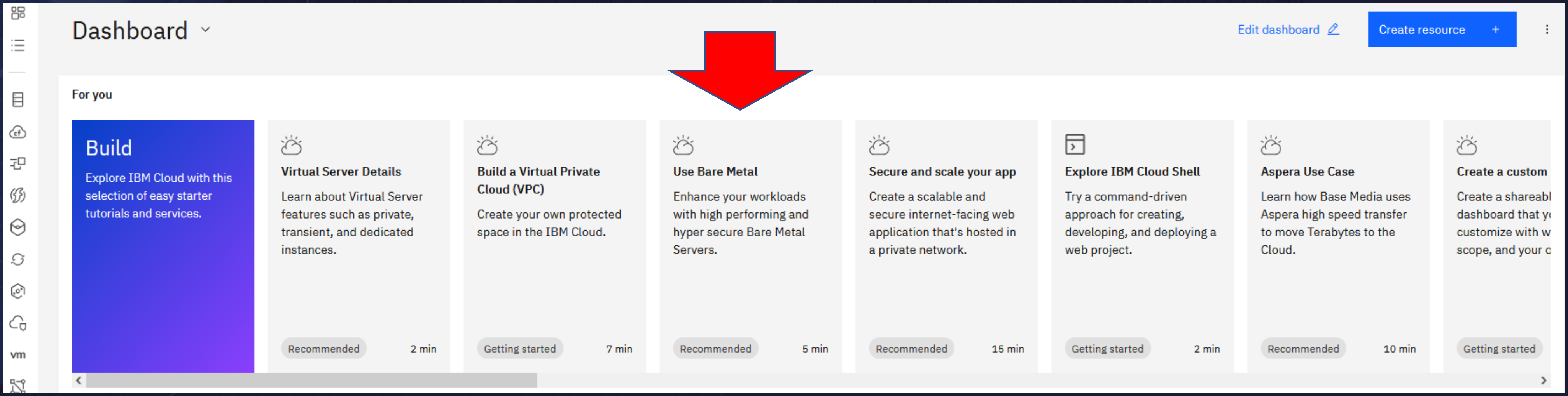


Agenda

- What is pureScale
- Allocating the environment
- Setting up pureScale
- Connecting to the pureScale cluster

Request the hardware from IBM cloud

- I selected “Use Bare Metal” from Dashboard



For you

Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Virtual Server Details

Learn about Virtual Server features such as private, transient, and dedicated instances.

Recommended 2 min

Build a Virtual Private Cloud (VPC)

Create your own protected space in the IBM Cloud.

Getting started 7 min

Use Bare Metal

Enhance your workloads with high performing and hyper secure Bare Metal Servers.

Recommended 5 min

Secure and scale your app

Create a scalable and secure internet-facing web application that's hosted in a private network.

Recommended 15 min

Explore IBM Cloud Shell

Try a command-driven approach for creating, developing, and deploying a web project.

Getting started 2 min

Aspera Use Case

Learn how Base Media uses Aspera high speed transfer to move Terabytes to the Cloud.

Recommended 10 min

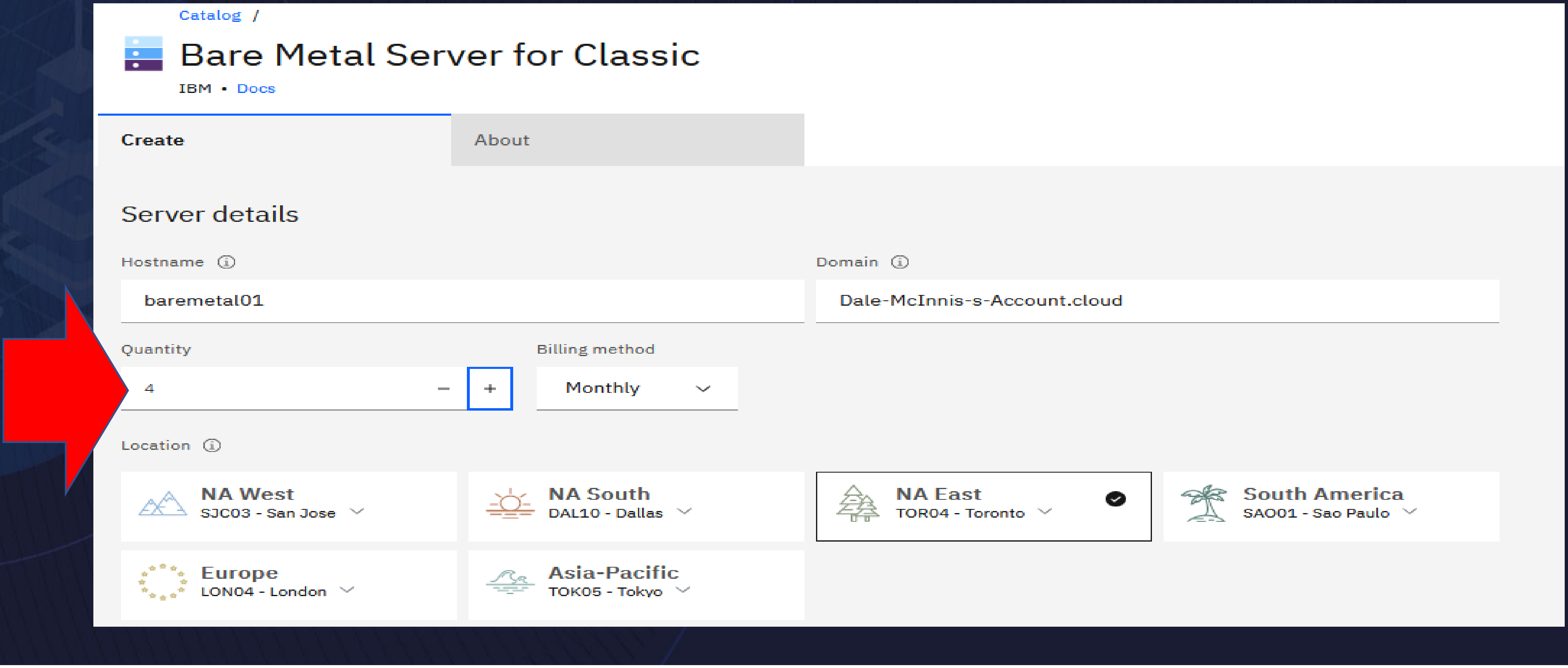
Create a custom

Create a shareabl dashboard that y customize with w scope, and your c

Getting started

Choose nearest data center for deployment

Order all servers at once so they will be allocated on the same IP subnet



Catalog /

Bare Metal Server for Classic

IBM • Docs

Server details

Hostname: baremetal01

Domain: Dale-McInnis-s-Account.cloud

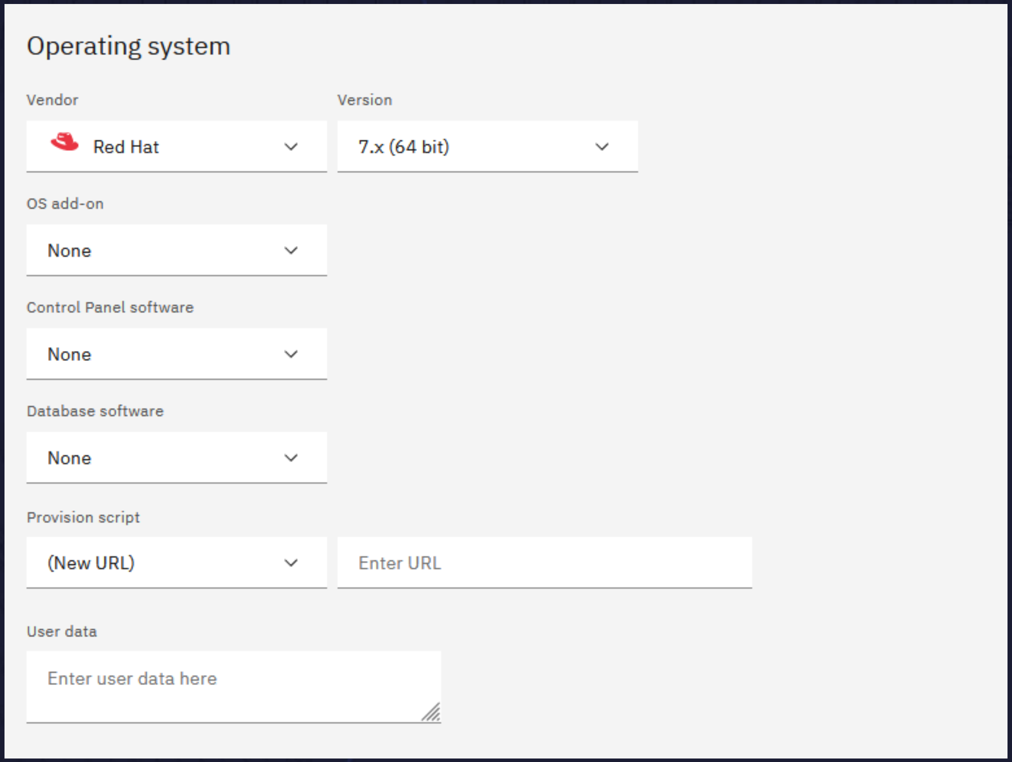
Quantity: 4

Billing method: Monthly

Location:

- NA West — SJC03 - San Jose
- NA South — DAL10 - Dallas
- NA East — TOR04 - Toronto ✓
- South America — SAO01 - Sao Paulo
- Europe — LON04 - London
- Asia-Pacific — TOK05 - Tokyo

Choose RHEL OS



Operating system

Vendor: Red Hat

Version: 7.x (64 bit)

OS add-on: None

Control Panel software: None

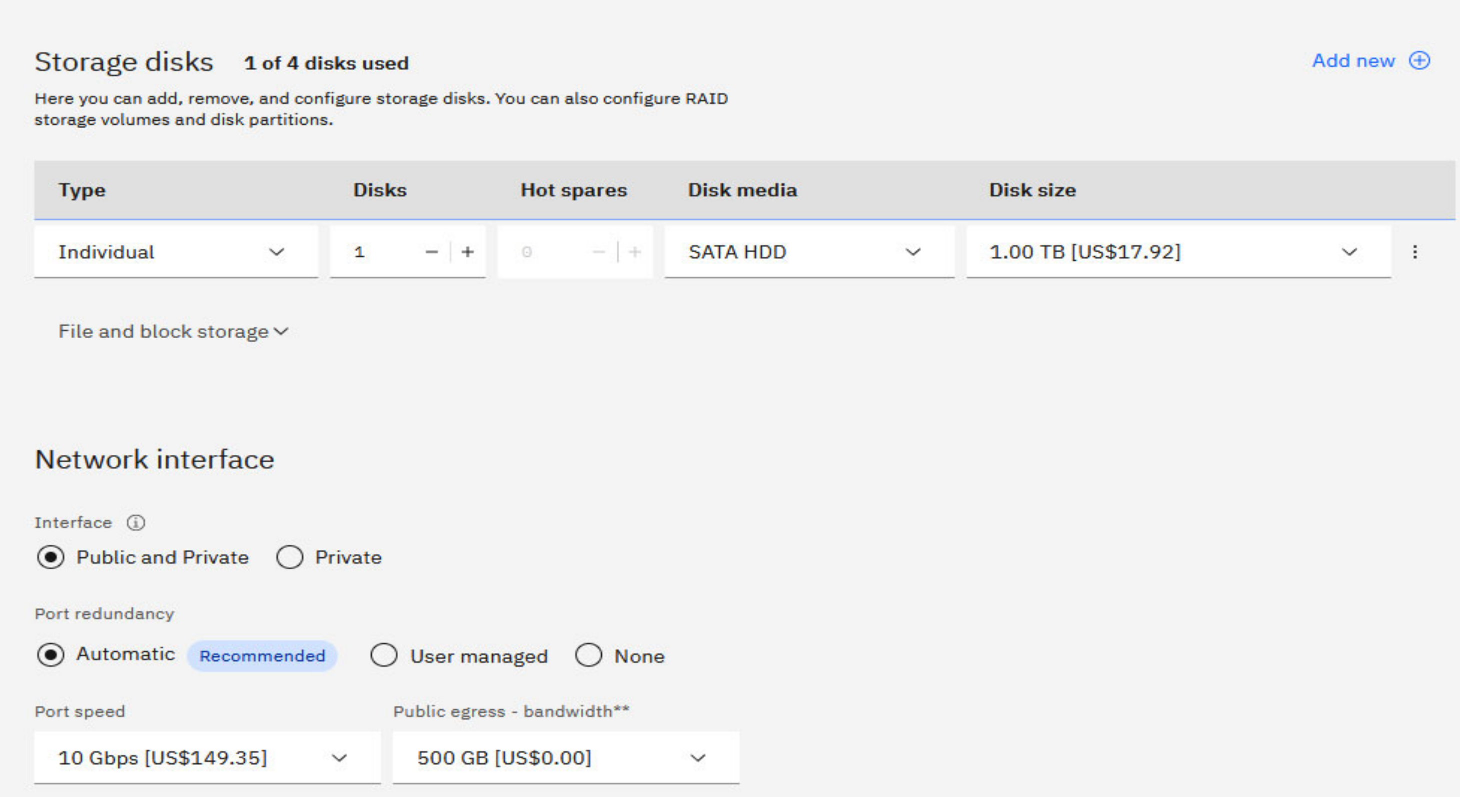
Database software: None

Provision script: (New URL) Enter URL

User data: Enter user data here

Could have used RHEL 8.x since I am only using TCP sockets for the interconnect between the members and the CFs

Select fastest network available



Storage disks **1 of 4 disks used**

Here you can add, remove, and configure storage disks. You can also configure RAID storage volumes and disk partitions.

Type	Disks	Hot spares	Disk media	Disk size
Individual	1	0	SATA HDD	1.00 TB [US$17.92]

Network interface

Interface

● Public and Private ○ Private

Port redundancy

● Automatic Recommended ○ User managed ○ None

Port speed: 10 Gbps [US$149.35]

Public egress - bandwidth**: 500 GB [US$0.00]

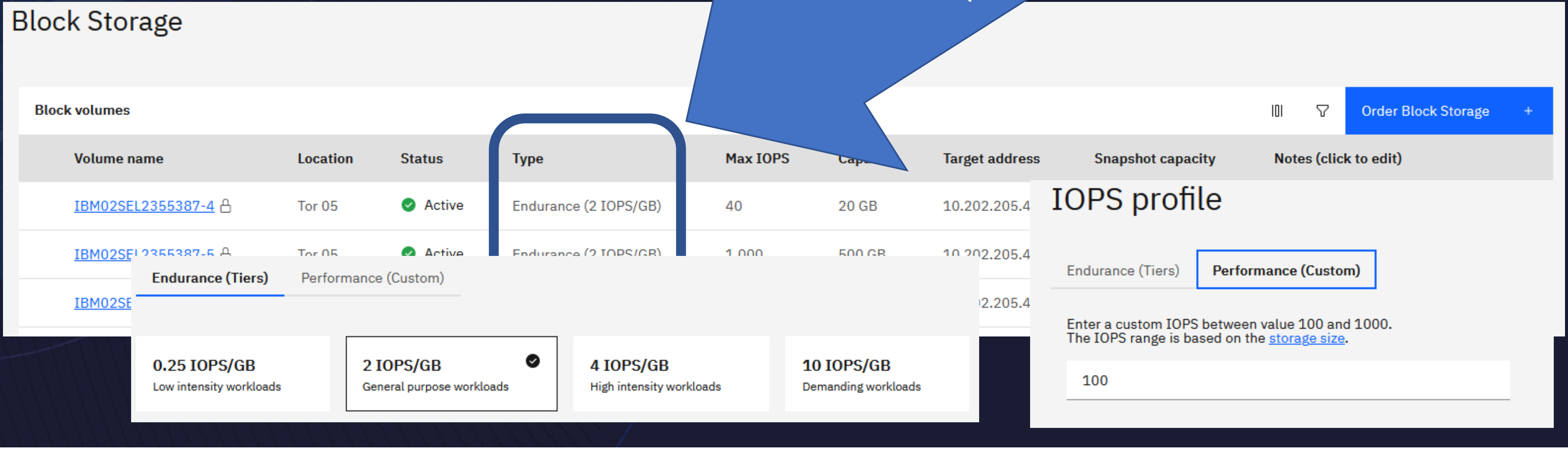
I choose an internal disk to host the install binaries, not the db2 instance

Pick the fastest network possible.

Storage for Spectrum Scale use

- Order block storage that will be mounted on all ser
  - Minimum of 1 volume for the database
  - 1 volume for the tiebreaker
  - 1 volume for shared sqllib

I choose slow disk because of the price



Block Storage

Block volumes

Order Block Storage +

Volume name	Location	Status	Type	Max IOPS	Cap	Target address	Snapshot capacity	Notes (click to edit)
IBM02SEL2355387-4	Tor 05	Active	Endurance (2 IOPS/GB)	40	20 GB	10.202.205.4		
IBM02SEL2355387-5	Tor 05	Active	Endurance (2 IOPS/GB)	1 000	500 GB	10.202.205.4		
IBM02SE						2.205.4		

Endurance (Tiers) | Performance (Custom)

0.25 IOPS/GB	2 IOPS/GB ✓	4 IOPS/GB	10 IOPS/GB
Low intensity workloads	General purpose workloads	High intensity workloads	Demanding workloads

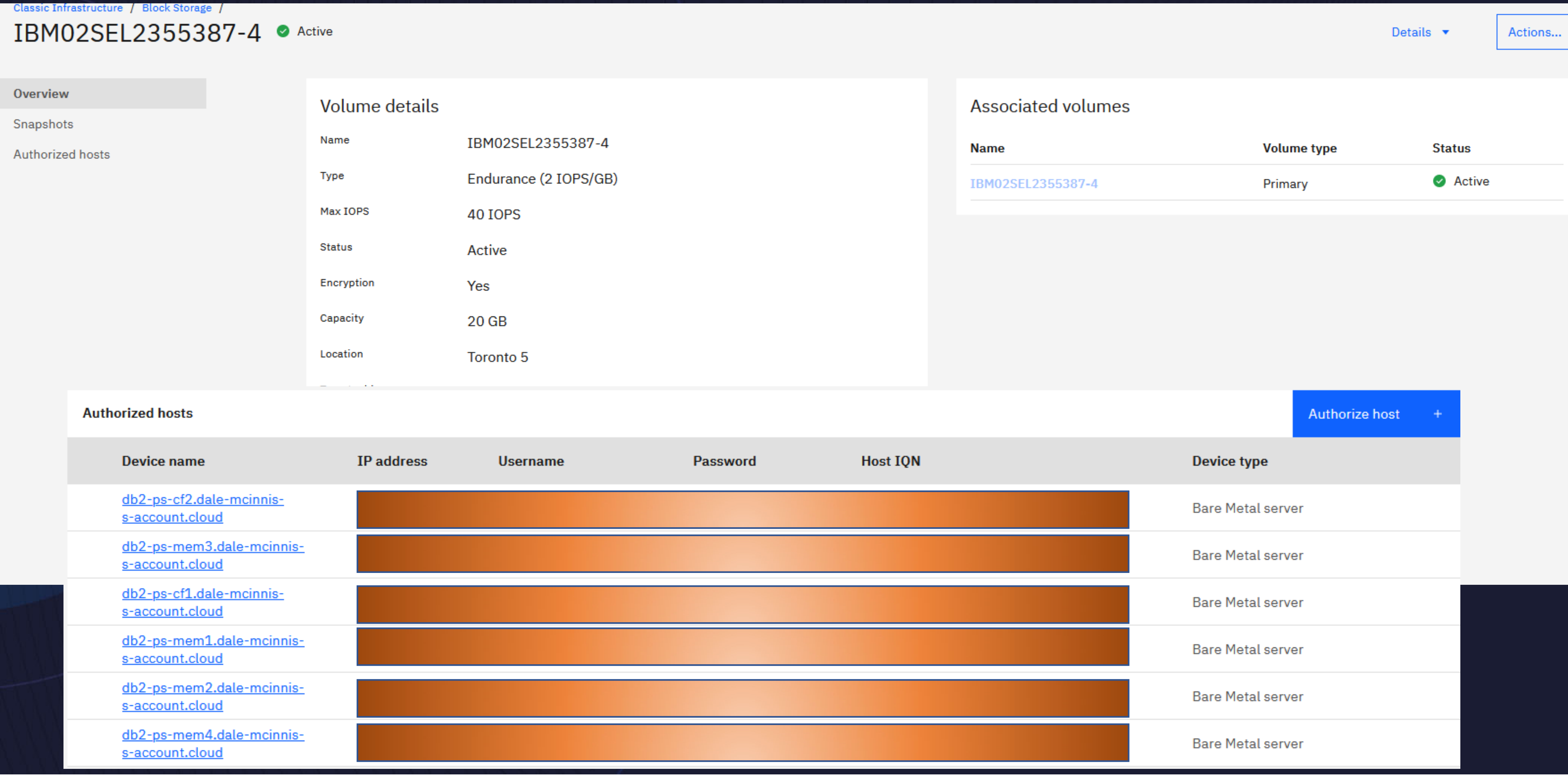
IOPS profile

Endurance (Tiers) | **Performance (Custom)**

Enter a custom IOPS between value 100 and 1000.
The IOPS range is based on the storage size.

100

For each block storage device authorize all hosts



Classic Infrastructure / Block Storage /

IBM02SEL2355387-4 Active

Volume details

Name	IBM02SEL2355387-4
Type	Endurance (2 IOPS/GB)
Max IOPS	40 IOPS
Status	Active
Encryption	Yes
Capacity	20 GB
Location	Toronto 5

Associated volumes

Name	Volume type	Status
IBM02SEL2355387-4	Primary	Active

Authorized hosts

Device name	IP address	Username	Password	Host IQN	Device type
db2-ps-cf2.dale-mcinnis-s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
db2-ps-mem3.dale-mcinnis-s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
db2-ps-cf1.dale-mcinnis-s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
db2-ps-mem1.dale-mcinnis-s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
db2-ps-mem2.dale-mcinnis-s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
db2-ps-mem4.dale-mcinnis-s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server

Why did I choose Bare Metal?

- Originally, I choose “Virtual server for Classic (VMs)”
- However, I ran into several issues while attempting to setup pureScale
  - Host name would change after reboot (mixed case would be converted to lower case)
  - After installing my prereq packages and rebooting some of the packages would no longer be installed
    - I suspect the Image was being restarted on another host and was an older copy of the image
  - My ssh keys would change after reboot
    - Time and time again I had to reset db2locssh and ensure the keys were stored locally
- Once I moved to bare metal servers all of these issues went away
- Other team members have had success with PowerVM on P-Series

Agenda

- What is pureScale
- Allocating the environment
- Setting up pureScale
- Connecting to the pureScale cluster

Setting up pureScale in the IBM Cloud

- **Change /etc/hostname to the fully qualified host name**
- **Setup /etc/hosts**
  - Ensure you use ONLY lower case for the host names
  - Ensure you place both the public and private names (short and long)

```
163.75.74.58     db2-ps-mem1.dale-mcinnis-s-account.cloud db2-ps-mem1
163.75.74.59     db2-ps-mem2.dale-mcinnis-s-account.cloud db2-ps-mem2
163.75.74.61     db2-ps-mem3.dale-mcinnis-s-account.cloud db2-ps-mem3
163.75.74.55     db2-ps-mem4.dale-mcinnis-s-account.cloud db2-ps-mem4
163.75.74.50     db2-ps-cf1.dale-mcinnis-s-account.cloud db2-ps-cf1
163.75.74.53     db2-ps-cf2.dale-mcinnis-s-account.cloud db2-ps-cf2


10.243.71.32    db2-ps-mem1-priv.dale-mcinnis-s-account.cloud db2-ps-mem1-priv
10.243.71.25    db2-ps-mem2-priv.dale-mcinnis-s-account.cloud db2-ps-mem2-priv
10.243.71.27    db2-ps-mem3-priv.dale-mcinnis-s-account.cloud   db2-ps-mem3-priv
10.243.71.14    db2-ps-mem4-priv.dale-mcinnis-s-account.cloud  db2-ps-mem4-priv
10.243.71.10    db2-ps-cf1-priv.dale-mcinnis-s-account.cloud  db2-ps-cf1-priv
10.243.71.54    db2-ps-cf2-priv.dale-mcinnis-s-account.cloud  db2-ps-cf2-priv
```

Setting up pureScale in the IBM Cloud

- **Disable SELINUX**
  - Modify /etc/selinux/conf
  - Requires a reboot

- **Run db2prereqcheck on each host**
  - I copied the install image to each host to get access to the prereq executable

- **Ensure all of the required packages are install**
  - yum install kernel cpp gcc-c++ m4 kernel-devel kernel-headers perl patch ksh libstdc++.so.6 perl-Sys-Syslog yum-utils libaio pam.i686 ntp ntpd kernel-tools kernel-tools-libs vsftpd iscsi-initiator-utils device-mapper-multipath net-tools

Setting up pureScale in the IBM Cloud

Update grub

- Add ipcmni_extend to the /etc/default/grub file on the GRUB_CMDLINE_LIXUX_DEFAULT line
- Type the following command as root
  - sudo grub2-mkconfig –o /boot/grub2/grub.cfg

Reboot the server

- sudo reboot

ISCSI Setup

- **Attach ISCSI disk**
  - Authorize host from IBM Cloud Console

- **Add /etc/multipath.conf**

Specific to RHEL 7.x

```
defaults {
          user_friendly_names no
          max_fds max
          flush_on_last_del yes
          queue_without_daemon no
          dev_loss_tmo infinity
          fast_io_fail_tmo 5
}
# All data in the following section must be specific to your system.
blacklist {
          wwid "SAdaptec*"
          devnode "^hd[a-z]"
          devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]"
          devnode "^cciss."
}
devices {
          device {
                    vendor "NETAPP"
                    product "LUN"
                    path_grouping_policy group_by_prio
                    features "3 queue_if_no_path pg_init_retries 50"
                    prio "alua"
                    path_checker tur
                    failback immediate
                    path_selector "round-robin 0"
                    hardware_handler "1 alua"
                    rr_weight uniform
                    rr_min_io 128
          }
}
```

ISCSI setup con’t

Execute:

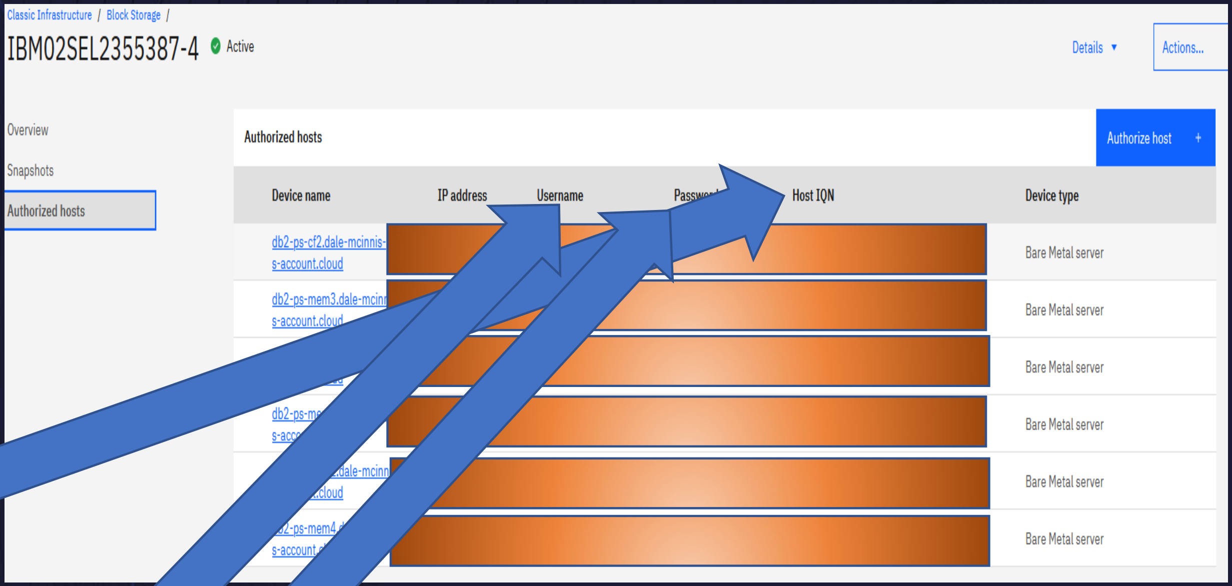
- modprobe dm-multipath
- service multipathd start
- chkconfig multipathd on
- multipath -l (should return nothing)

vi /etc/iscsi/initiatorname.iscsi

```
InitiatorName=<value-from-the-Portal>
```

vi /etc/iscsi/iscsid.conf

```
node.session.auth.authmethod = CHAP
node.session.auth.username = <Username-value-from-Portal>
node.session.auth.password = <Password-value-from-Portal>
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = <Username-value-from-Portal>
discovery.sendtargets.auth.password = <Password-value-from-Portal>
```



Classic Infrastructure / Block Storage /

IBM02SEL2355387-4 Active

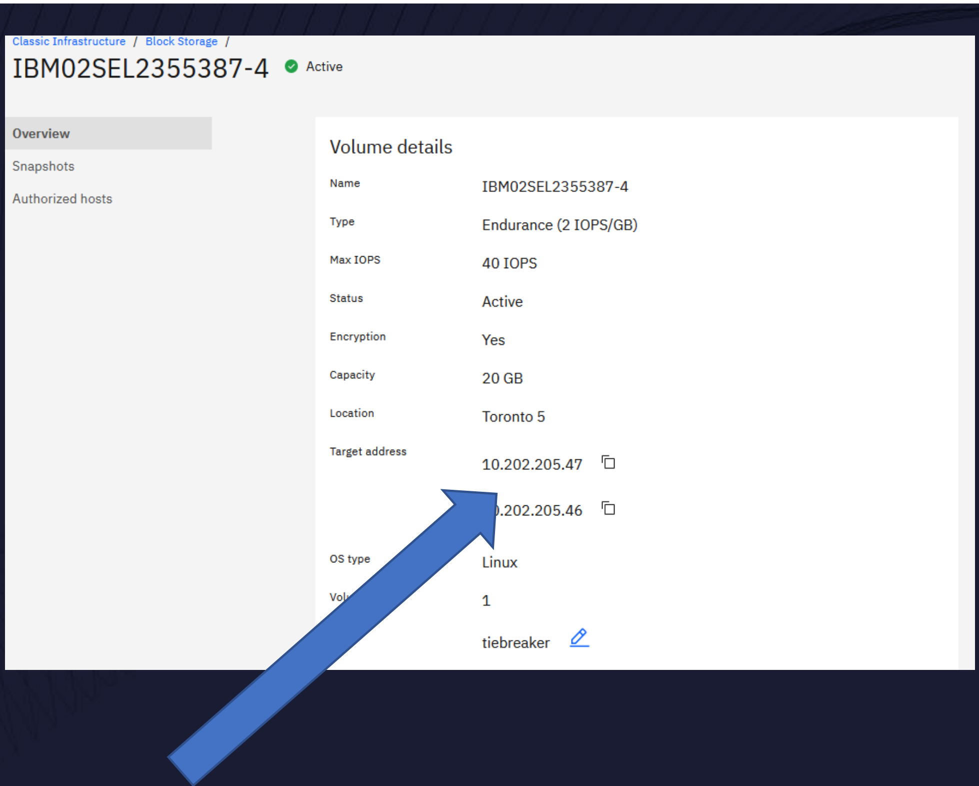
Authorized hosts

Device name	IP address	Username	Password	Host IQN	Device type
db2-ps-cf2.dale-mcinnis- s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
db2-ps-mem3.dale-mcinn s-account.cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
db2-ps-m s-acc	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
dale-mcinn .cloud	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server
2-ps-mem4 s-account	[redacted]	[redacted]	[redacted]	[redacted]	Bare Metal server

ISCSI setup cont

Execute the following sequence of commands

- chkconfig iscsi on
- chkconfig iscsid on
- service iscsi start
- service iscsid start
- iscsiadm -m discovery -t sendtargets -p ww.xx.yy.zz
- iscsiadm -m node -L automatic
- iscsiadm -m session



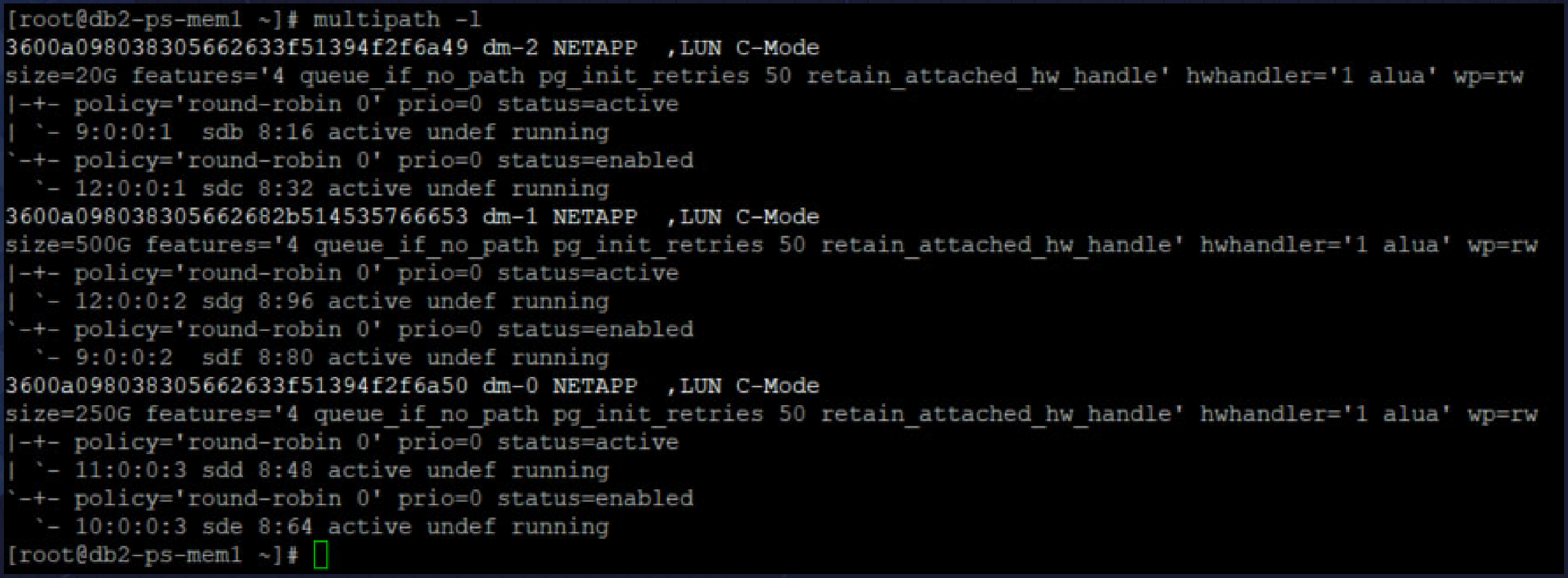
IBM02SEL2355387-4 Active

Volume details

Name	IBM02SEL2355387-4
Type	Endurance (2 IOPS/GB)
Max IOPS	40 IOPS
Status	Active
Encryption	Yes
Capacity	20 GB
Location	Toronto 5
Target address	10.202.205.47).202.205.46
OS type	Linux
Vol	1
	tiebreaker

ISCSI setup con’t

- Execute: multipath -l



```
[root@db2-ps-mem1 ~]# multipath -l
3600a09803830566263 3f51394f2f6a49 dm-2 NETAPP  ,LUN C-Mode
size=20G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=0 status=active
| `- 9:0:0:1  sdb 8:16 active undef running
`-+- policy='round-robin 0' prio=0 status=enabled
  `- 12:0:0:1 sdc 8:32 active undef running
3600a098038305662682b514535766653 dm-1 NETAPP  ,LUN C-Mode
size=500G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=0 status=active
| `- 12:0:0:2 sdg 8:96 active undef running
`-+- policy='round-robin 0' prio=0 status=enabled
  `- 9:0:0:2  sdf 8:80 active undef running
3600a098038305662633f51394f2f6a50 dm-0 NETAPP  ,LUN C-Mode
size=250G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='round-robin 0' prio=0 status=active
| `- 11:0:0:3 sdd 8:48 active undef running
`-+- policy='round-robin 0' prio=0 status=enabled
  `- 10:0:0:3 sde 8:64 active undef running
[root@db2-ps-mem1 ~]# 
```

Create users and groups

Create users (as root)

- groupadd --gid 980 db2iadm1
- groupadd --gid 981 db2fadm1
- useradd -u 1004 -g db2iadm1 -m -d /home/db2sdin1 db2sdin1
- useradd -u 1003 -g db2fadm1 -m -d /home/db2sdfe1 db2sdfe1
- passwd db2sdin1
- passwd db2sdfe1

Enable passwordlessh

Add to /etc/ssh/sshd_config

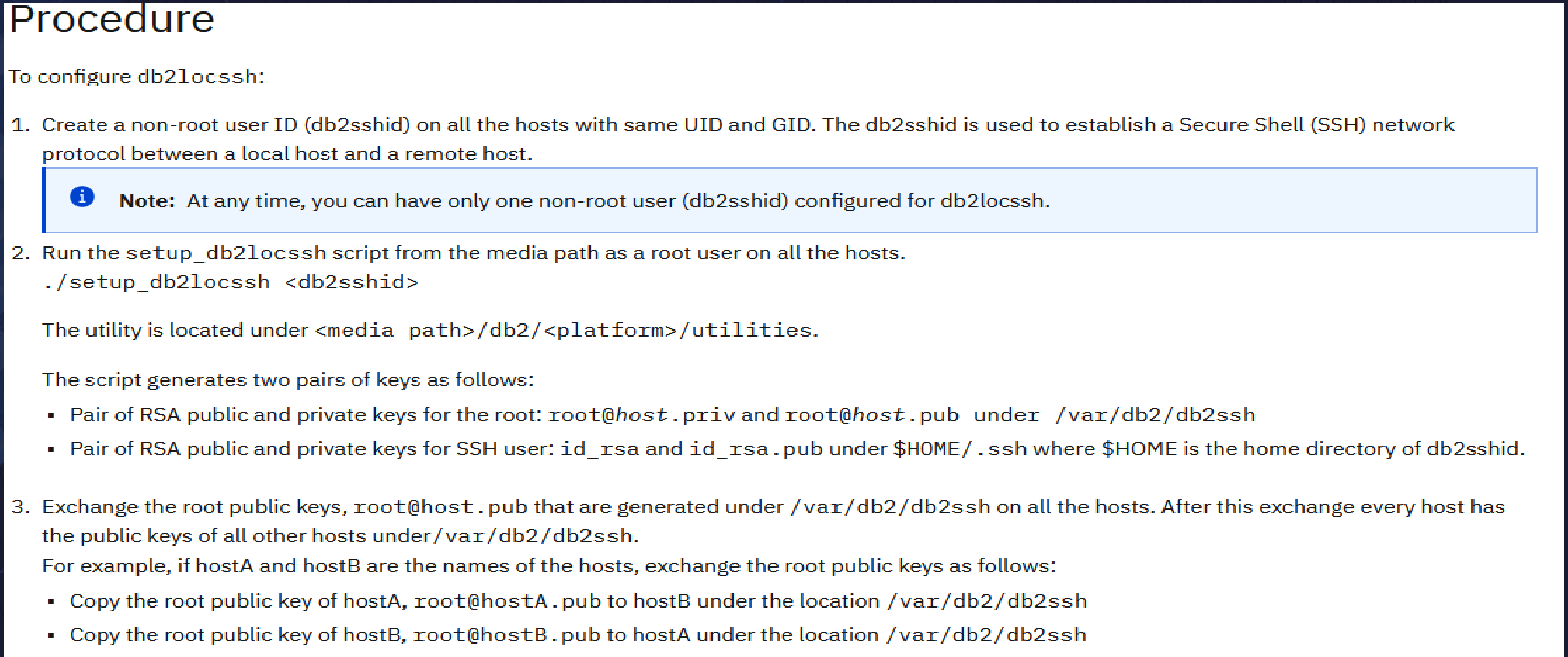
```
PermitRootLogin yes
PasswordAuthentication yes
```

Add to /etc/ssh/ssh_config

```
Port 22
Protocol 2,1
```

Setup db2locssh

- https://www.ibm.com/docs/en/db2/11.5?topic=environment-setting-up-db2locssh



Procedure

To configure db2locssh:

1. Create a non-root user ID (db2sshid) on all the hosts with same UID and GID. The db2sshid is used to establish a Secure Shell (SSH) network protocol between a local host and a remote host.

   Note: At any time, you can have only one non-root user (db2sshid) configured for db2locssh.

2. Run the `setup_db2locssh` script from the media path as a root user on all the hosts.

   ```
   ./setup_db2locssh <db2sshid>
   ```

   The utility is located under `<media path>/db2/<platform>/utilities`.

   The script generates two pairs of keys as follows:

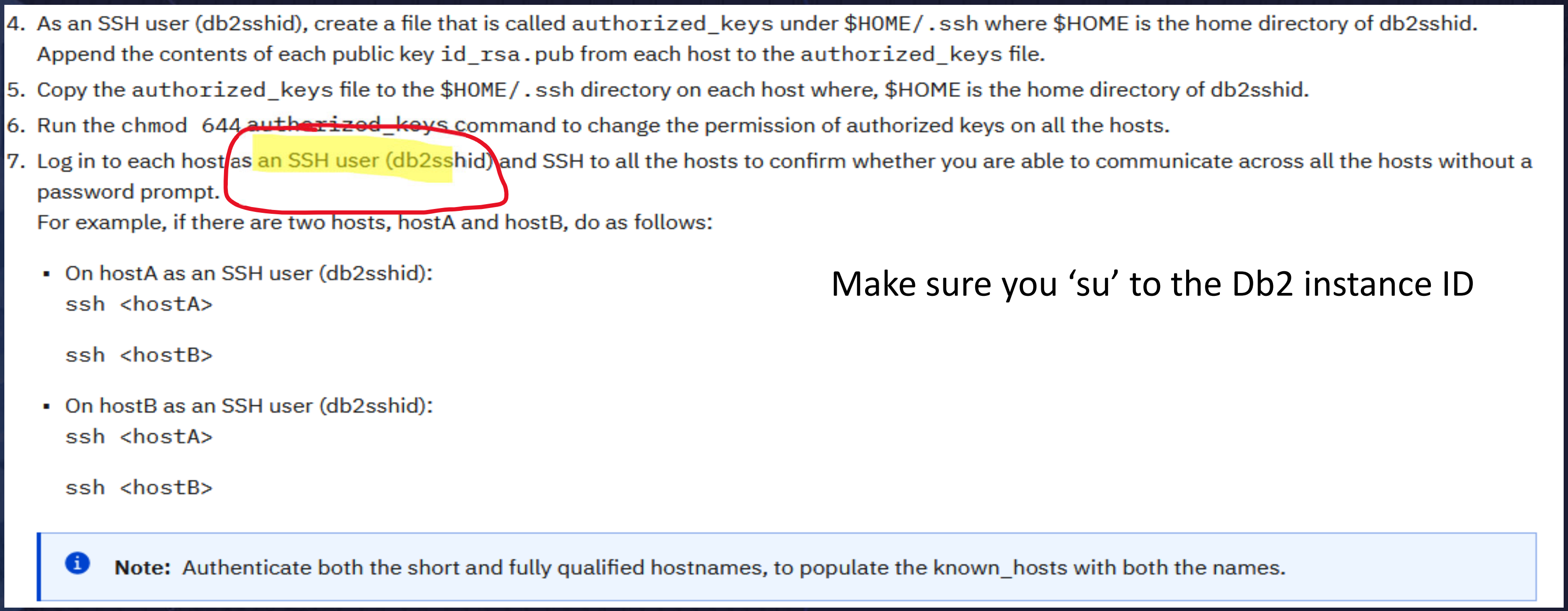
   - Pair of RSA public and private keys for the root: `root@`*`host`*`.priv` and `root@`*`host`*`.pub under /var/db2/db2ssh`
   - Pair of RSA public and private keys for SSH user: `id_rsa` and `id_rsa.pub` under `$HOME/.ssh` where $HOME is the home directory of db2sshid.

3. Exchange the root public keys, `root@host.pub` that are generated under `/var/db2/db2ssh` on all the hosts. After this exchange every host has the public keys of all other hosts under `/var/db2/db2ssh`.

   For example, if hostA and hostB are the names of the hosts, exchange the root public keys as follows:

   - Copy the root public key of hostA, `root@hostA.pub` to hostB under the location `/var/db2/db2ssh`
   - Copy the root public key of hostB, `root@hostB.pub` to hostA under the location `/var/db2/db2ssh`

Setup db2locssh con’t

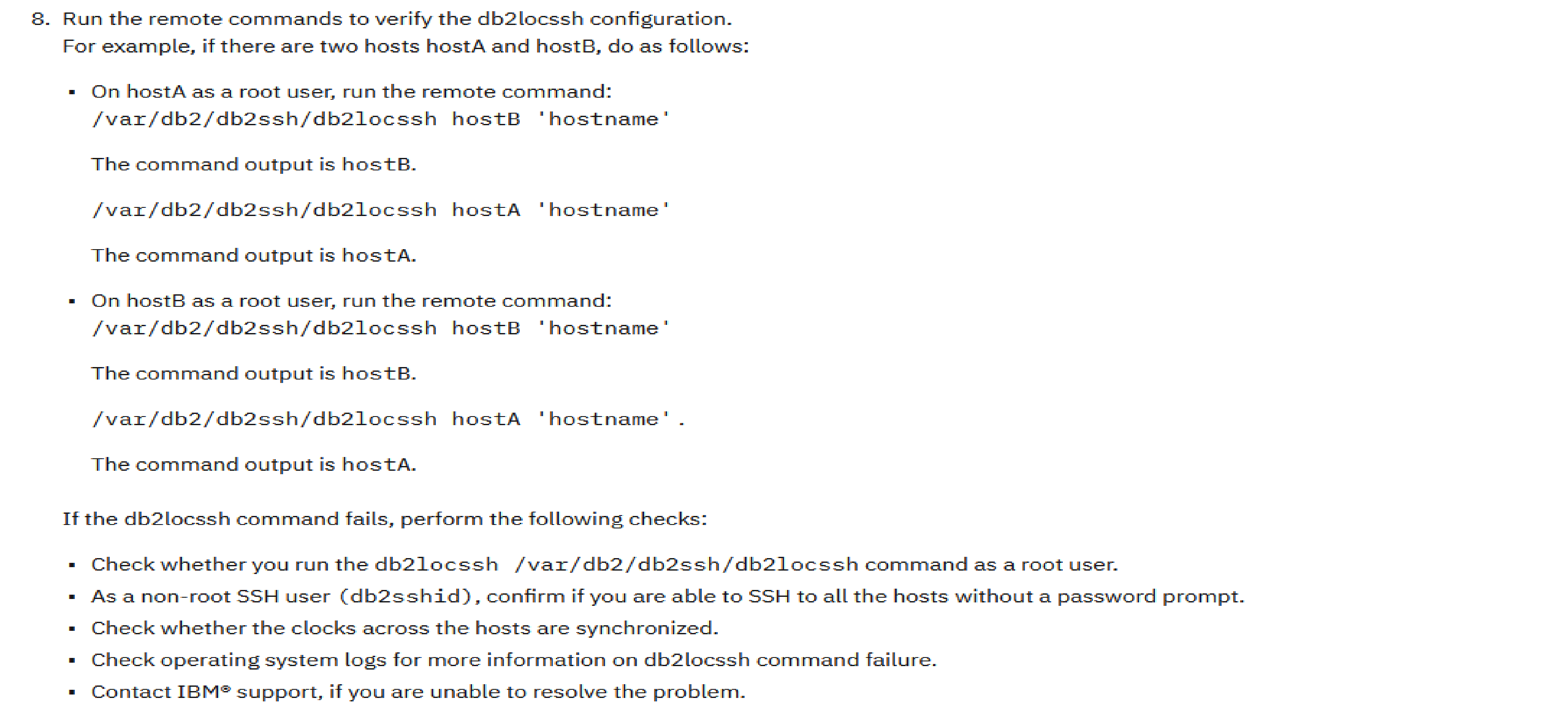


4. As an SSH user (db2sshid), create a file that is called `authorized_keys` under `$HOME/.ssh` where $HOME is the home directory of db2sshid. Append the contents of each public key `id_rsa.pub` from each host to the `authorized_keys` file.
5. Copy the `authorized_keys` file to the `$HOME/.ssh` directory on each host where, $HOME is the home directory of db2sshid.
6. Run the chmod 644 authorized_keys command to change the permission of authorized keys on all the hosts.
7. Log in to each host as an SSH user (db2sshid) and SSH to all the hosts to confirm whether you are able to communicate across all the hosts without a password prompt.
   For example, if there are two hosts, hostA and hostB, do as follows:
   - On hostA as an SSH user (db2sshid):
     ```
     ssh <hostA>

     ssh <hostB>
     ```
   - On hostB as an SSH user (db2sshid):
     ```
     ssh <hostA>

     ssh <hostB>
     ```

ℹ **Note:** Authenticate both the short and fully qualified hostnames, to populate the known_hosts with both the names.

Make sure you ‘su’ to the Db2 instance ID

Setup db2locssh con’t



8. Run the remote commands to verify the db2locssh configuration.
   For example, if there are two hosts hostA and hostB, do as follows:

   - On hostA as a root user, run the remote command:

     ```
     /var/db2/db2ssh/db2locssh hostB 'hostname'
     ```

     The command output is hostB.

     ```
     /var/db2/db2ssh/db2locssh hostA 'hostname'
     ```

     The command output is hostA.

   - On hostB as a root user, run the remote command:

     ```
     /var/db2/db2ssh/db2locssh hostB 'hostname'
     ```

     The command output is hostB.

     ```
     /var/db2/db2ssh/db2locssh hostA 'hostname'.
     ```

     The command output is hostA.

   If the db2locssh command fails, perform the following checks:

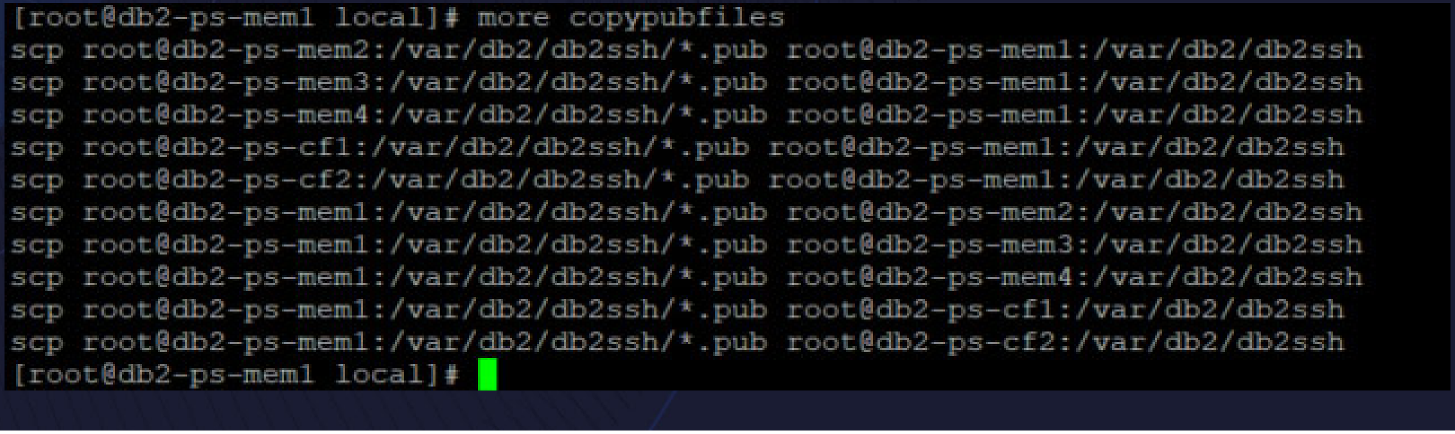
   - Check whether you run the `db2locssh /var/db2/db2ssh/db2locssh` command as a root user.
   - As a non-root SSH user (`db2sshid`), confirm if you are able to SSH to all the hosts without a password prompt.
   - Check whether the clocks across the hosts are synchronized.
   - Check operating system logs for more information on db2locssh command failure.
   - Contact IBM® support, if you are unable to resolve the problem.

How I setup db2locssh

- Step 0 – Ensure you have the install image on EACH node
  - In my case on /local

- Step 1 – as root on EACH node run setup_db2locssh
  - /local/server_dec/db2/linuxamd64/install/setup_db2locssh db2sdin1

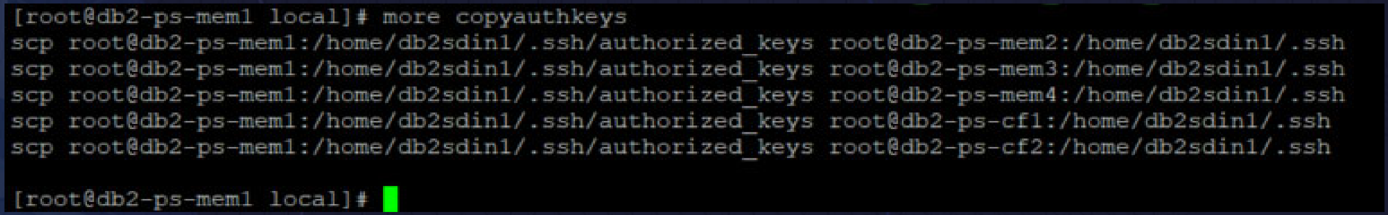
- Step 2 – copy all of the root@host.pub files onto node 1 then copy all files to each node



```
[root@db2-ps-mem1 local]# more copypubfiles
scp root@db2-ps-mem2:/var/db2/db2ssh/*.pub root@db2-ps-mem1:/var/db2/db2ssh
scp root@db2-ps-mem3:/var/db2/db2ssh/*.pub root@db2-ps-mem1:/var/db2/db2ssh
scp root@db2-ps-mem4:/var/db2/db2ssh/*.pub root@db2-ps-mem1:/var/db2/db2ssh
scp root@db2-ps-cf1:/var/db2/db2ssh/*.pub root@db2-ps-mem1:/var/db2/db2ssh
scp root@db2-ps-cf2:/var/db2/db2ssh/*.pub root@db2-ps-mem1:/var/db2/db2ssh
scp root@db2-ps-mem1:/var/db2/db2ssh/*.pub root@db2-ps-mem2:/var/db2/db2ssh
scp root@db2-ps-mem1:/var/db2/db2ssh/*.pub root@db2-ps-mem3:/var/db2/db2ssh
scp root@db2-ps-mem1:/var/db2/db2ssh/*.pub root@db2-ps-mem4:/var/db2/db2ssh
scp root@db2-ps-mem1:/var/db2/db2ssh/*.pub root@db2-ps-cf1:/var/db2/db2ssh
scp root@db2-ps-mem1:/var/db2/db2ssh/*.pub root@db2-ps-cf2:/var/db2/db2ssh
[root@db2-ps-mem1 local]#
```

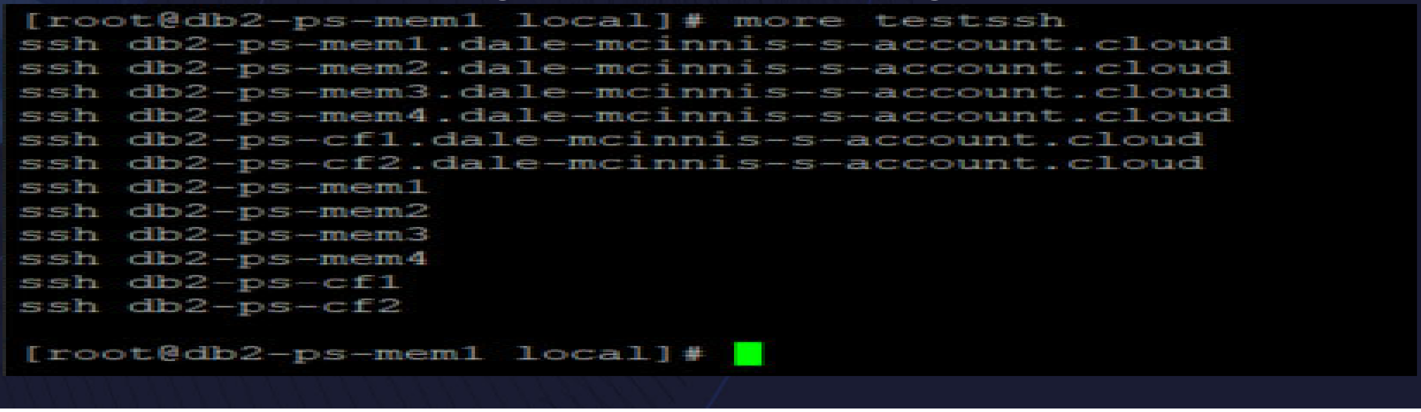
How I setup db2locssh con’t

- Step 3 – as root
  - Create $HOME/.ssh/authorized_keys file using the contents of $HOME/.ssh/id_rsa.pub from ALL of the nodes
  - Copy the same file to all other nodes



```
[root@db2-ps-mem1 local]# more copyauthkeys
scp root@db2-ps-mem1:/home/db2sdin1/.ssh/authorized_keys root@db2-ps-mem2:/home/db2sdin1/.ssh
scp root@db2-ps-mem1:/home/db2sdin1/.ssh/authorized_keys root@db2-ps-mem3:/home/db2sdin1/.ssh
scp root@db2-ps-mem1:/home/db2sdin1/.ssh/authorized_keys root@db2-ps-mem4:/home/db2sdin1/.ssh
scp root@db2-ps-mem1:/home/db2sdin1/.ssh/authorized_keys root@db2-ps-cf1:/home/db2sdin1/.ssh
scp root@db2-ps-mem1:/home/db2sdin1/.ssh/authorized_keys root@db2-ps-cf2:/home/db2sdin1/.ssh

[root@db2-ps-mem1 local]#
```

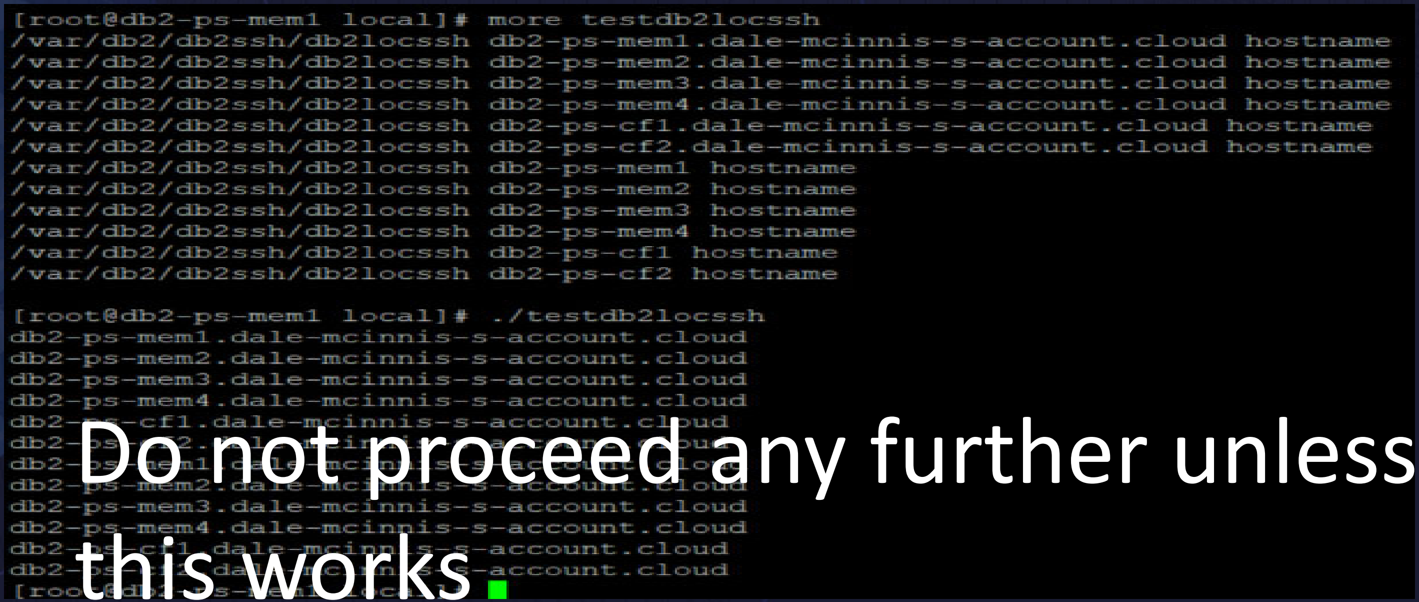
- Step 4 – as Db2 Instance Owner (db2sdin1)
  - ssh to each node, using both the short and long name



```
[root@db2-ps-mem1 local]# more testssh
ssh db2-ps-mem1.dale-mcinnis-s-account.cloud
ssh db2-ps-mem2.dale-mcinnis-s-account.cloud
ssh db2-ps-mem3.dale-mcinnis-s-account.cloud
ssh db2-ps-mem4.dale-mcinnis-s-account.cloud
ssh db2-ps-cf1.dale-mcinnis-s-account.cloud
ssh db2-ps-cf2.dale-mcinnis-s-account.cloud
ssh db2-ps-mem1
ssh db2-ps-mem2
ssh db2-ps-mem3
ssh db2-ps-mem4
ssh db2-ps-cf1
ssh db2-ps-cf2

[root@db2-ps-mem1 local]#
```

How I setup db2locssh con’t

- Step 5 – as root
  - Db2locssh to each node using both short and long name

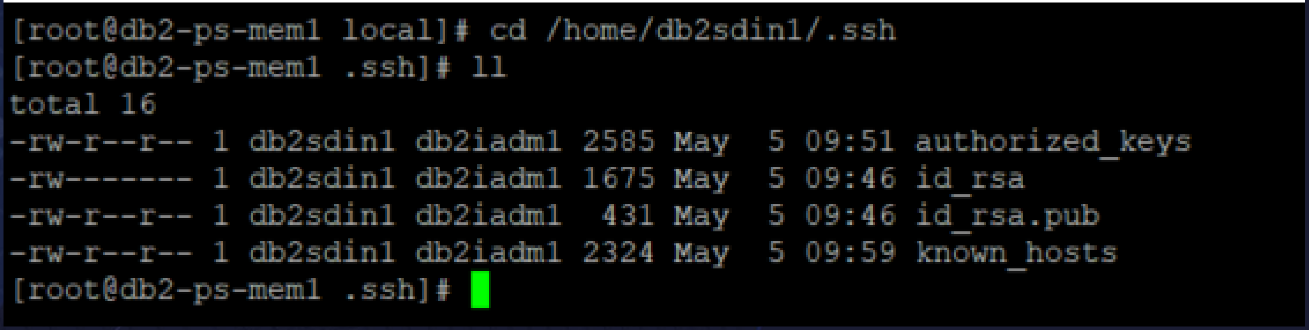


```
[root@db2-ps-mem1 local]# more testdb2locssh
/var/db2/db2ssh/db2locssh db2-ps-mem1.dale-mcinnis-s-account.cloud hostname
/var/db2/db2ssh/db2locssh db2-ps-mem2.dale-mcinnis-s-account.cloud hostname
/var/db2/db2ssh/db2locssh db2-ps-mem3.dale-mcinnis-s-account.cloud hostname
/var/db2/db2ssh/db2locssh db2-ps-mem4.dale-mcinnis-s-account.cloud hostname
/var/db2/db2ssh/db2locssh db2-ps-cf1.dale-mcinnis-s-account.cloud hostname
/var/db2/db2ssh/db2locssh db2-ps-cf2.dale-mcinnis-s-account.cloud hostname
/var/db2/db2ssh/db2locssh db2-ps-mem1 hostname
/var/db2/db2ssh/db2locssh db2-ps-mem2 hostname
/var/db2/db2ssh/db2locssh db2-ps-mem3 hostname
/var/db2/db2ssh/db2locssh db2-ps-mem4 hostname
/var/db2/db2ssh/db2locssh db2-ps-cf1 hostname
/var/db2/db2ssh/db2locssh db2-ps-cf2 hostname

[root@db2-ps-mem1 local]# ./testdb2locssh
db2-ps-mem1.dale-mcinnis-s-account.cloud
db2-ps-mem2.dale-mcinnis-s-account.cloud
db2-ps-mem3.dale-mcinnis-s-account.cloud
db2-ps-mem4.dale-mcinnis-s-account.cloud
db2-ps-cf1.dale-mcinnis-s-account.cloud
db2-ps-cf2.dale-mcinnis-s-account.cloud
db2-ps-mem1.dale-mcinnis-s-account.cloud
db2-ps-mem2.dale-mcinnis-s-account.cloud
db2-ps-mem3.dale-mcinnis-s-account.cloud
db2-ps-mem4.dale-mcinnis-s-account.cloud
db2-ps-cf1.dale-mcinnis-s-account.cloud
db2-ps-cf2.dale-mcinnis-s-account.cloud
[root@db2-ps-mem1 local]#
```

Do not proceed any further unless this works

How I setup db2locssh con’t

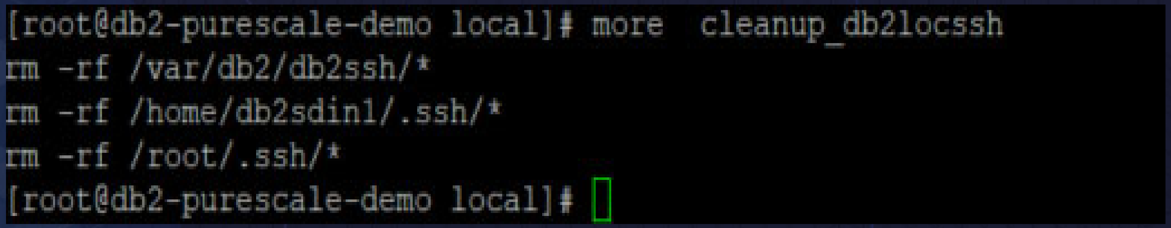
- Once complete, you should have the following files on EACH node
  - authorized_keys = copy of the id_rsa.pub contents for all nodes
  - known_hosts = stash file for each node’s key



```
[root@db2-ps-mem1 local]# cd /home/db2sdin1/.ssh
[root@db2-ps-mem1 .ssh]# ll
total 16
-rw-r--r-- 1 db2sdin1 db2iadm1 2585 May  5 09:51 authorized_keys
-rw------- 1 db2sdin1 db2iadm1 1675 May  5 09:46 id_rsa
-rw-r--r-- 1 db2sdin1 db2iadm1  431 May  5 09:46 id_rsa.pub
-rw-r--r-- 1 db2sdin1 db2iadm1 2324 May  5 09:59 known_hosts
[root@db2-ps-mem1 .ssh]#
```

What if this fails?

Be prepared to cleanup and try again



```
[root@db2-purescale-demo local]# more  cleanup_db2locssh
rm -rf /var/db2/db2ssh/*
rm -rf /home/db2sdin1/.ssh/*
rm -rf /root/.ssh/*
[root@db2-purescale-demo local]# 
```

Start at Step 1 again

Useful commands to reset the environment

- Sometimes Spectrum Scale will leave a tag on the drives and will not reuse them, quickest solution is to wipe out the drive using dd command.
  - dd if=/dev/zero bs=1M of=/dev/dm-0

- Sometimes the ssh keys get messed up – not sure why
  - delete all of the /root/.ssh/* and /home/db2sdin1/.ssh/* files
  - rerun the testssh and testdb2locssh scripts

Create db2 instance

- On the first member execute

```
cd /opt/ibm/db2/V11.5/instance
```

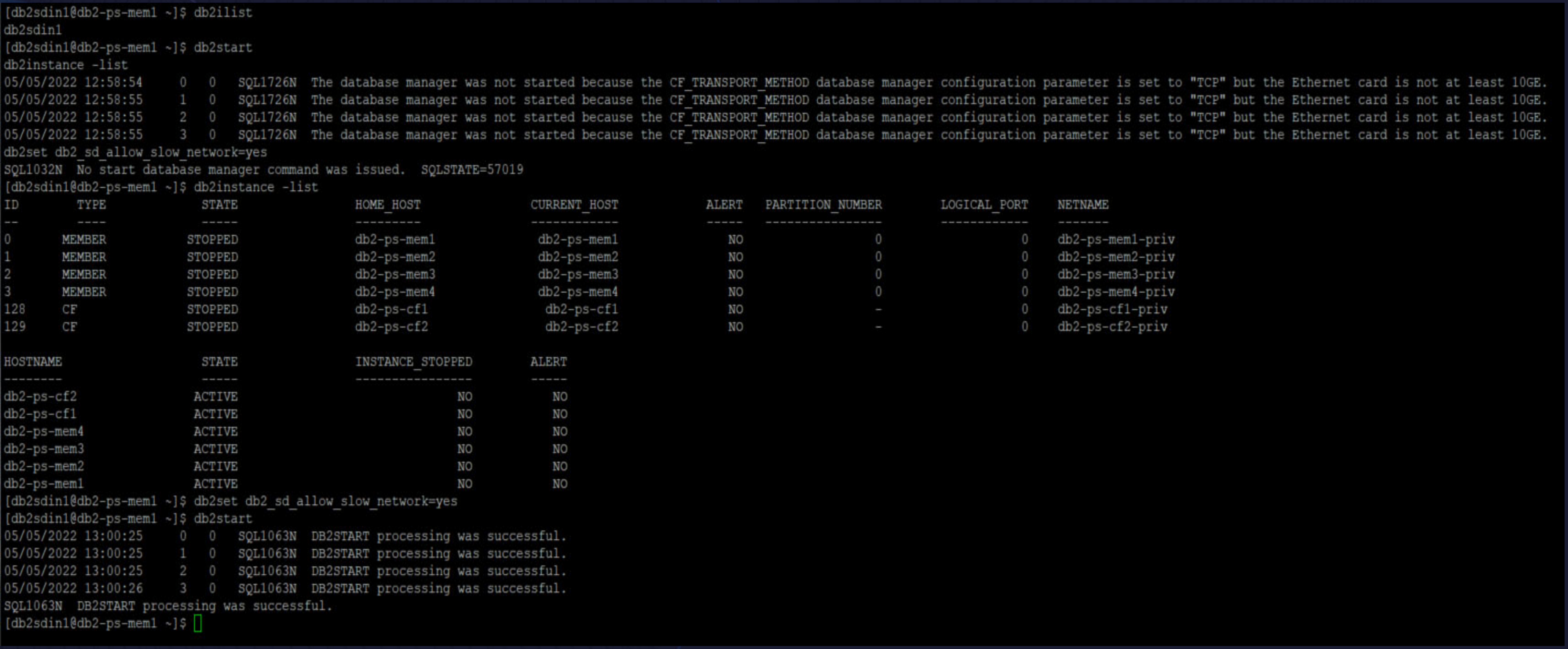
```
./db2icrt -d -m db2-ps-mem -mnet db2-ps-mem-priv -m db2-ps-mem2 -mnet db2-ps-mem2-priv -m db2-ps-mem3 -mnet
db2-ps-mem3-priv -m db2-ps-mem4 -mnet db2-ps-mem4-priv -cf db2-ps-mem5 -cfnet db2-ps-mem5-priv -cf db2-ps-mem6 -
cfnet db2-ps-mem6-priv -instance_shared_dev /dev/dm-2 -tbdev /dev/dm-1 -u db2sdfe1 db2sdin1
```

This will install Db2 on all subsequent nodes for you – so be prepared for this to take a while

Do not be surprized if your telnet session terminates.

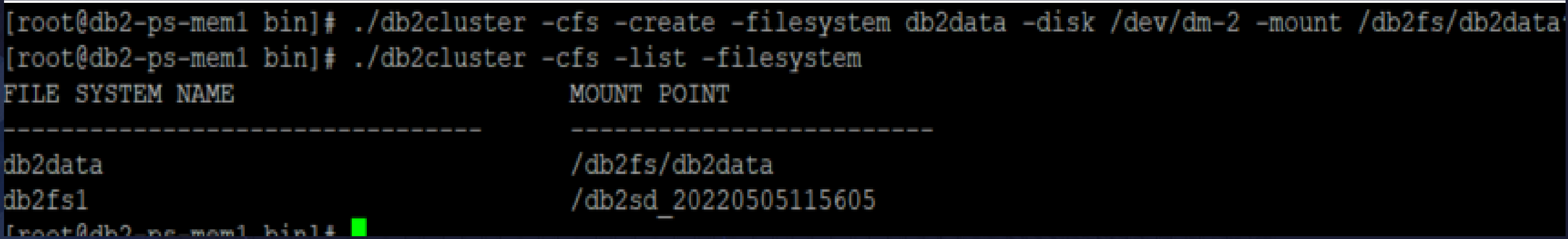
Telnet back in and check the /tmp/db2icrt.log.xxx to see where is left off

Starting Db2

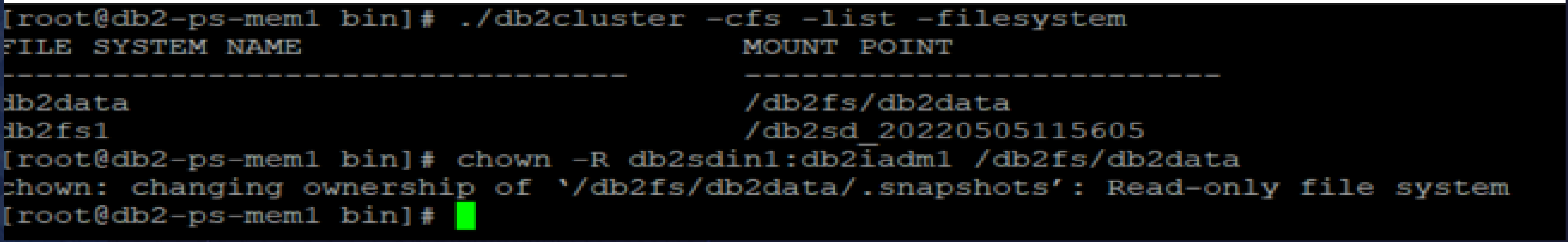


```
[db2sdin1@db2-ps-mem1 ~]$ db2ilist
db2sdin1
[db2sdin1@db2-ps-mem1 ~]$ db2start
db2instance -list
05/05/2022 12:58:54     0   0   SQL1726N  The database manager was not started because the CF_TRANSPORT_METHOD database manager configuration parameter is set to "TCP" but the Ethernet card is not at least 10GE.
05/05/2022 12:58:55     1   0   SQL1726N  The database manager was not started because the CF_TRANSPORT_METHOD database manager configuration parameter is set to "TCP" but the Ethernet card is not at least 10GE.
05/05/2022 12:58:55     2   0   SQL1726N  The database manager was not started because the CF_TRANSPORT_METHOD database manager configuration parameter is set to "TCP" but the Ethernet card is not at least 10GE.
05/05/2022 12:58:55     3   0   SQL1726N  The database manager was not started because the CF_TRANSPORT_METHOD database manager configuration parameter is set to "TCP" but the Ethernet card is not at least 10GE.
db2set db2_sd_allow_slow_network=yes
SQL1032N  No start database manager command was issued.  SQLSTATE=57019
[db2sdin1@db2-ps-mem1 ~]$ db2instance -list
ID        TYPE             STATE                HOME_HOST               CURRENT_HOST            ALERT   PARTITION_NUMBER        LOGICAL_PORT    NETNAME
--        ----             -----                ---------               ------------            -----   ----------------        ------------    -------
0       MEMBER           STOPPED                db2-ps-mem1              db2-ps-mem1               NO                  0                   0    db2-ps-mem1-priv
1       MEMBER           STOPPED                db2-ps-mem2              db2-ps-mem2               NO                  0                   0    db2-ps-mem2-priv
2       MEMBER           STOPPED                db2-ps-mem3              db2-ps-mem3               NO                  0                   0    db2-ps-mem3-priv
3       MEMBER           STOPPED                db2-ps-mem4              db2-ps-mem4               NO                  0                   0    db2-ps-mem4-priv
128     CF               STOPPED                db2-ps-cf1                db2-ps-cf1               NO                  -                   0    db2-ps-cf1-priv
129     CF               STOPPED                db2-ps-cf2                db2-ps-cf2               NO                  -                   0    db2-ps-cf2-priv

HOSTNAME                   STATE                INSTANCE_STOPPED        ALERT
--------                   -----                ----------------        -----
db2-ps-cf2                ACTIVE                              NO           NO
db2-ps-cf1                ACTIVE                              NO           NO
db2-ps-mem4               ACTIVE                              NO           NO
db2-ps-mem3               ACTIVE                              NO           NO
db2-ps-mem2               ACTIVE                              NO           NO
db2-ps-mem1               ACTIVE                              NO           NO
[db2sdin1@db2-ps-mem1 ~]$ db2set db2_sd_allow_slow_network=yes
[db2sdin1@db2-ps-mem1 ~]$ db2start
05/05/2022 13:00:25     0   0   SQL1063N  DB2START processing was successful.
05/05/2022 13:00:25     1   0   SQL1063N  DB2START processing was successful.
05/05/2022 13:00:25     2   0   SQL1063N  DB2START processing was successful.
05/05/2022 13:00:26     3   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
[db2sdin1@db2-ps-mem1 ~]$
```

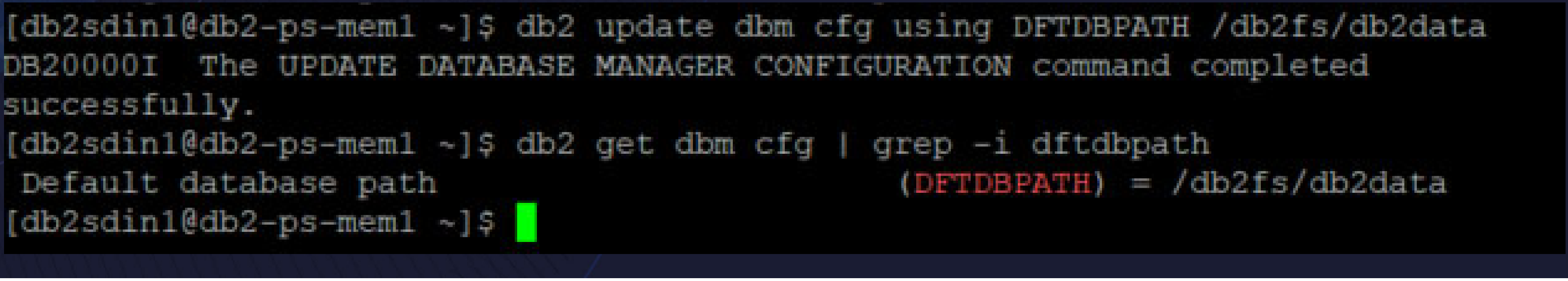
Add a new Spectrum Scale filesystem for the DB



```
[root@db2-ps-mem1 bin]# ./db2cluster -cfs -create -filesystem db2data -disk /dev/dm-2 -mount /db2fs/db2data
[root@db2-ps-mem1 bin]# ./db2cluster -cfs -list -filesystem
FILE SYSTEM NAME                        MOUNT POINT
---------------------------------       -------------------------
db2data                                 /db2fs/db2data
db2fs1                                  /db2sd_20220505115605
[root@db2-ps-mem1 bin]#
```



```
[root@db2-ps-mem1 bin]# ./db2cluster -cfs -list -filesystem
FILE SYSTEM NAME                        MOUNT POINT
---------------------------------       -------------------------
db2data                                 /db2fs/db2data
db2fs1                                  /db2sd_20220505115605
[root@db2-ps-mem1 bin]# chown -R db2sdin1:db2iadm1 /db2fs/db2data
chown: changing ownership of '/db2fs/db2data/.snapshots': Read-only file system
[root@db2-ps-mem1 bin]#
```



```
[db2sdin1@db2-ps-mem1 ~]$ db2 update dbm cfg using DFTDBPATH /db2fs/db2data
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
[db2sdin1@db2-ps-mem1 ~]$ db2 get dbm cfg | grep -i dftdbpath
 Default database path                       (DFTDBPATH) = /db2fs/db2data
[db2sdin1@db2-ps-mem1 ~]$
```

Create your database – on any member

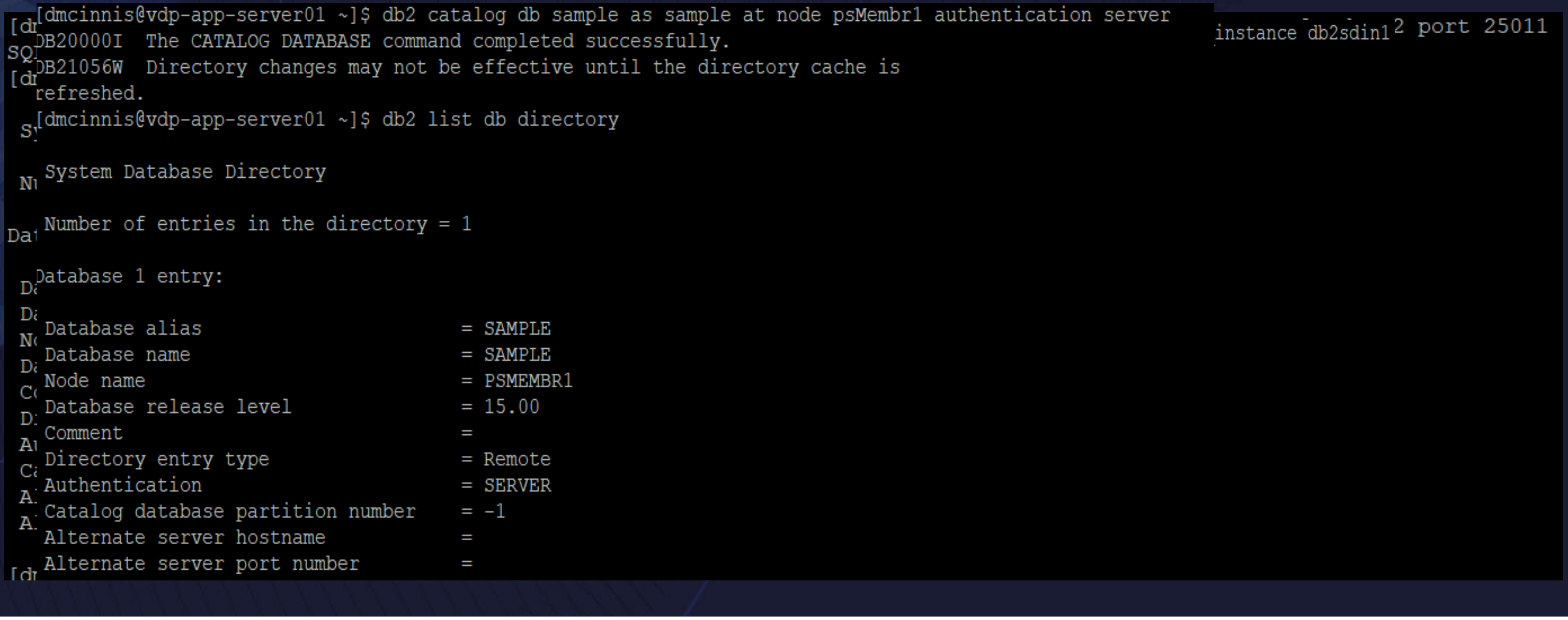
- db2 create database psdb on /db2fs/db2data DBPATH ON /db2fs/db2data

Move over to the App Server

- At this point the Db2 pureScale cluster is up and running and the database has been created.
- Now we move over to the client to setup the following
  - Workload balancer
  - Automatic client reroute
  - Client affinity

Connecting to the pureScale cluster

- Separate VM node configured with the Db2 Client runtime
  - Catalog the remote node (member 1)
  - Catalog the database
  - Alternate server is only supported on the server, not the client

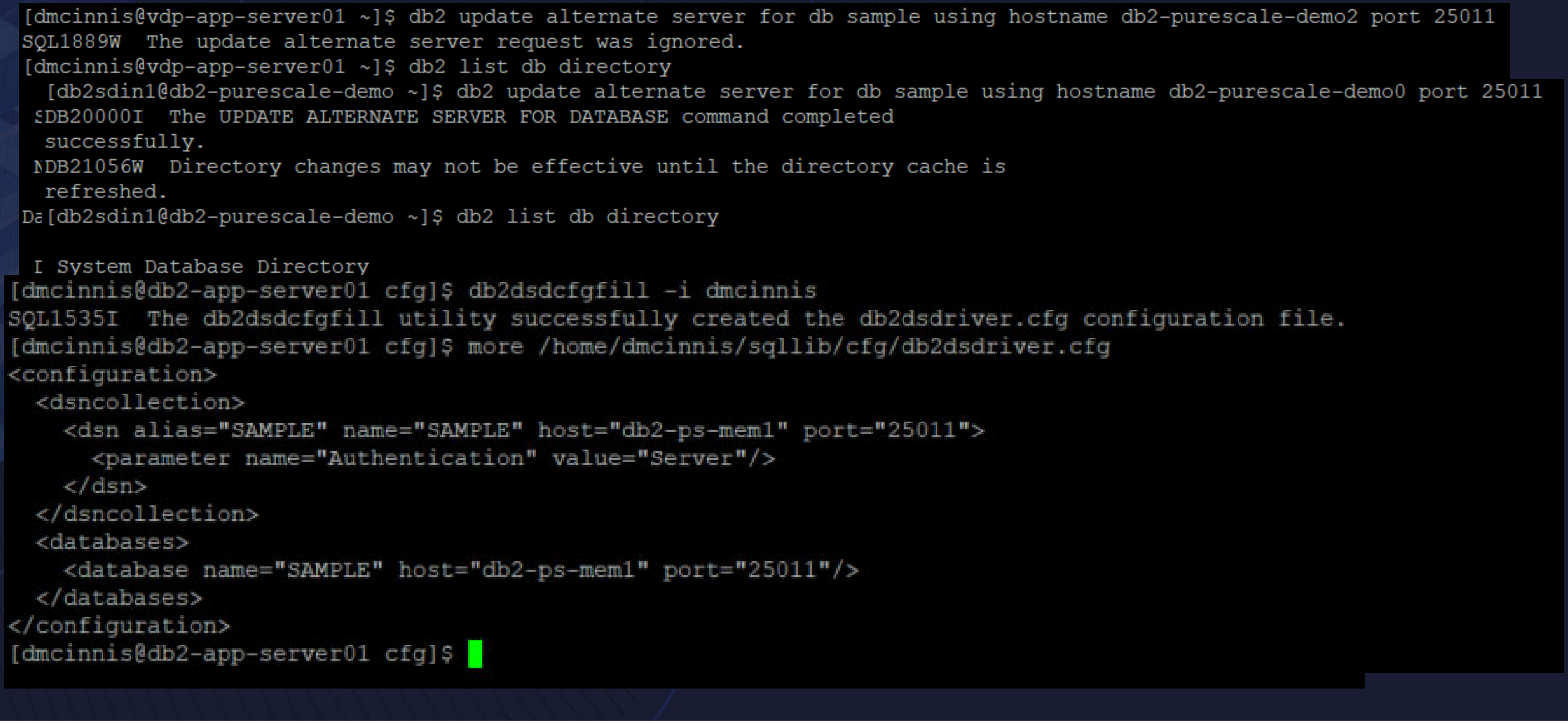


```
[dmcinnis@vdp-app-server01 ~]$ db2 catalog db sample as sample at node psMembr1 authentication server
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.
[dmcinnis@vdp-app-server01 ~]$ db2 list db directory

 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = SAMPLE
 Database name                        = SAMPLE
 Node name                            = PSMEMBR1
 Database release level               = 15.00
 Comment                              =
 Directory entry type                 = Remote
 Authentication                       = SERVER
 Catalog database partition number    = -1
 Alternate server hostname            =
 Alternate server port number         =
```

```
instance db2sdin12 port 25011
```

Connecting to the pureScale cluster

- Setting up the alternate server
    - Unable to update the DB catalog entry on the client



```
[dmcinnis@vdp-app-server01 ~]$ db2 update alternate server for db sample using hostname db2-purescale-demo2 port 25011
SQL1889W  The update alternate server request was ignored.
[dmcinnis@vdp-app-server01 ~]$ db2 list db directory
```

```
  [db2sdin1@db2-purescale-demo ~]$ db2 update alternate server for db sample using hostname db2-purescale-demo0 port 25011
 :DB20000I  The UPDATE ALTERNATE SERVER FOR DATABASE command completed
  successfully.
 NDB21056W  Directory changes may not be effective until the directory cache is
  refreshed.
Da[db2sdin1@db2-purescale-demo ~]$ db2 list db directory

 I System Database Directory
```

```
[dmcinnis@db2-app-server01 cfg]$ db2dsdcfgfill -i dmcinnis
SQL1535I  The db2dsdcfgfill utility successfully created the db2dsdriver.cfg configuration file.
[dmcinnis@db2-app-server01 cfg]$ more /home/dmcinnis/sqllib/cfg/db2dsdriver.cfg
<configuration>
  <dsncollection>
    <dsn alias="SAMPLE" name="SAMPLE" host="db2-ps-mem1" port="25011">
      <parameter name="Authentication" value="Server"/>
    </dsn>
  </dsncollection>
  <databases>
    <database name="SAMPLE" host="db2-ps-mem1" port="25011"/>
  </databases>
</configuration>
[dmcinnis@db2-app-server01 cfg]$
```

WLB or Client Affinity – pick only 1 per client

- When setting up the client's configuration you need to decide if you want automatic load balancing across all nodes, e.g. enableWLB=true

OR

- You can setup client affinity, defining which members your client can connect to
- These are mutually exclusive. If you set both your connection will fail with the following error:

```
[dmcinnis@db2-app-server01 cfg]$ db2 connect to psdb
SQL5162N  The db2dsdriver.cfg configuration file contains the parameter
"enableWLB", which has the same value as the parameter "clientaffinitydefined",
but these parameters cannot have the same value.
```

Modify the db2dsdriver.cfg: Add the alternate servers

```
<configuration>
 <dsncollection>
  <dsn alias=“PSDB" name=“PSDB" host="db2-ps-mem1" port="25011">
   <parameter name="Authentication" value="Server"/>
  </dsn>
 </dsncollection>
 <databases>
  <database name=“PSDB" host="db2-ps-mem1" port="25011">
   <parameter name="KeepAliveTimeout" value="20"/>
    <acr>
    <parameter name="enableACR" value="true"/>
    <parameter name="enableSeamlessACR" value="true"/>
    <parameter name="maxAcrRetries" value="2"/>
    <parameter name="acrRetryInterval" value="3"/>
    <parameter name="affinityFailbackInterval" value="60"/>
    <alternateserverlist>
        <server name="mem1" hostname="db2-ps-mem1" port="25011"/>
        <server name="mem2" hostname="db2-ps-mem2" port="25011"/>
        <server name="mem3" hostname="db2-ps-mem3" port="25011"/>
        <server name="mem4" hostname="db2-ps-mem4" port="25011"/>
    </alternateserverlist>
    <affinitylist>
        <list name="evenmembers" serverorder="mem2,mem4"/>
        <list name="oddmembers"  serverorder="mem1,mem3"/>
    </affinitylist>
    <clientaffinitydefined>
        <client name="myappserver" hostname="db2-app-server01" listname="evenmembers"/>
    </clientaffinitydefined>
   </acr>
  </database>
 </databases>
</configuration>
```

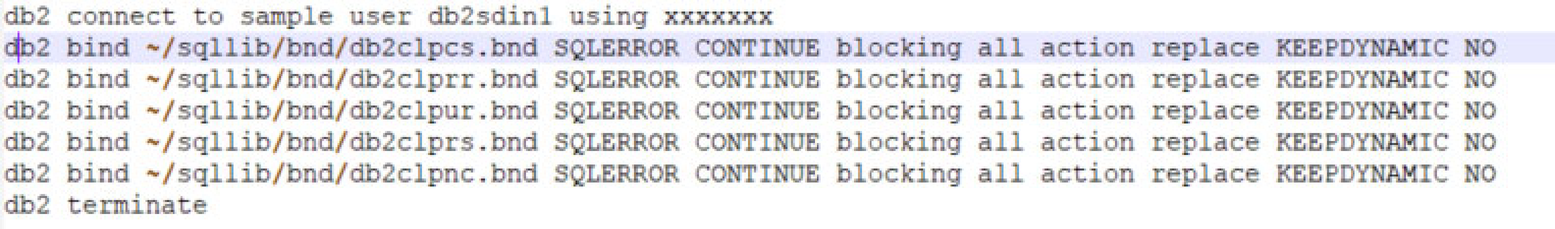
db2dsdriver.cfg with Client Affinity

```
<configuration>
 <dsncollection>
  <dsn alias=“PSDB" name=“PSDB" host="db2-ps-mem1" port="25011">
   <parameter name="Authentication" value="Server"/>
  </dsn>
 </dsncollection>
 <databases>
  <database name=“PSDB" host="db2-ps-mem1" port="25011">
   <parameter name="KeepAliveTimeout" value="20"/>
    <acr>
    <parameter name="enableACR" value="true"/>
    <parameter name="enableSeamlessACR" value="true"/>
    <parameter name="maxAcrRetries" value="2"/>
    <parameter name="acrRetryInterval" value="3"/>
    <parameter name="affinityFailbackInterval" value="60"/>
    <alternateserverlist>
        <server name="mem1" hostname="db2-ps-mem1" port="25011"/>
        <server name="mem2" hostname="db2-ps-mem2" port="25011"/>
        <server name="mem3" hostname="db2-ps-mem3" port="25011"/>
        <server name="mem4" hostname="db2-ps-mem4" port="25011"/>
    </alternateserverlist>
    <affinitylist>
        <list name="evenmembers" serverorder="mem2,mem4"/>
        <list name="oddmembers"  serverorder="mem1,mem3"/>
    </affinitylist>
    <clientaffinitydefined>
        <client name="myappserver" hostname="db2-app-server01" listname="evenmembers"/>
    </clientaffinitydefined>
   </acr>
  </database>
 </databases>
</configuration>
```

db2dsdriver.cfg with WLB

```
<configuration>
 <dsncollection>
  <dsn alias=“PSDB" name=“PSDB" host="db2-ps-mem1" port="25011">
   <parameter name="Authentication" value="Server"/>
  </dsn>
 </dsncollection>
 <databases>
  <database name=“PSDB" host="db2-ps-mem1" port="25011">
   <parameter name="KeepAliveTimeout" value="20"/>
   <wlb>
     <parameter name=“enableWLB” value=“true”/>
    </wlb>
    <acr>
    <parameter name="enableACR" value="true"/>
    <parameter name="enableSeamlessACR" value="true"/>
    <parameter name="maxAcrRetries" value="2"/>
    <parameter name="acrRetryInterval" value="3"/>
    <parameter name="affinityFailbackInterval" value="60"/>
    <alternateserverlist>
        <server name="mem1" hostname="db2-ps-mem1" port="25011"/>
        <server name="mem2" hostname="db2-ps-mem2" port="25011"/>
        <server name="mem3" hostname="db2-ps-mem3" port="25011"/>
        <server name="mem4" hostname="db2-ps-mem4" port="25011"/>
    </alternateserverlist>
    <affinitylist>
        <list name="evenmembers" serverorder="mem2,mem4"/>
        <list name="oddmembers"  serverorder="mem1,mem3"/>
    </affinitylist>
   </acr>
  </database>
 </databases>
</configuration>
```

Rebind packages to enable WLB



```
db2 connect to sample user db2sdin1 using xxxxxxx
db2 bind ~/sqllib/bnd/db2clpcs.bnd SQLERROR CONTINUE blocking all action replace KEEPDYNAMIC NO
db2 bind ~/sqllib/bnd/db2clprr.bnd SQLERROR CONTINUE blocking all action replace KEEPDYNAMIC NO
db2 bind ~/sqllib/bnd/db2clpur.bnd SQLERROR CONTINUE blocking all action replace KEEPDYNAMIC NO
db2 bind ~/sqllib/bnd/db2clprs.bnd SQLERROR CONTINUE blocking all action replace KEEPDYNAMIC NO
db2 bind ~/sqllib/bnd/db2clpnc.bnd SQLERROR CONTINUE blocking all action replace KEEPDYNAMIC NO
db2 terminate
```

Conclusion

- It is possible to run Db2 pureScale in a cloud environment
- It works best on bare metal, same as on-prem
- Ensure you get the fastest network and storage devices
- I recommend waiting for TSA to be replaced with Pacemaker/Corosyn before doing any large scale pureScale deployments
- Suitable for QA/Dev today
- Recommend waiting for RoCE support before going into production with pureScale in a cloud environment

Thank You

Speaker: Dale McInnis

Company: IBM Canada Ltd.

Email Address: dmcinnis@ca.ibm.com

Session Code: C7

Please fill out your session evaluation before leaving!