# SCALE-2020 NuoDB Hands-on Lab

## Prereqs

### Kubernetes clusters

*CSP1 cluster 0 (Region 0, e.g. London):*
<url/creds>

*CSP2 cluster 0 (Region 0, e.g. London):*
<url/creds>

*CSP1 cluster 1 (Region 1, e.g. Amsterdam):*
<url/creds>

Spec per cluster:
3 workers, 8vCPU/16Gb

Note:  additional info for setting up K8S clusters and running the lab exercises will be provided to the lab participants at our SFTP site.

**SFTP**
scale2020@sftp.nuodb.com:<pwd provided to lab participants>

## Lab 0 - Setup

### Useful commands

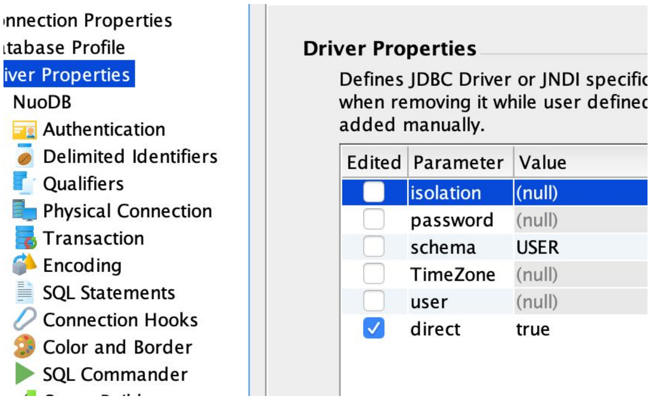
```
oc apply -f nuodb-scc-vpn.yaml -n nuodb
helm install thp ../../nuodb-helm-charts/stable/transparent-hugepage/ --values values-?.yaml -n nuodb
helm install storage ../../nuodb-helm-charts/stable/storage-class/ --values values-?.yaml -n nuodb
helm install admin ../../nuodb-helm-charts/stable/admin/ --values values-?.yaml -n nuodb
helm install database ../../nuodb-helm-charts/stable/database/ --values values-?.yaml -n nuodb
helm install ycsb ../../nuodb-helm-charts/incubator/demo-ycsb/ --values values-ycsb.yaml -n nuodb
oc adm policy add-scc-to-user privileged system:serviceaccount:nuodb:insights-grafana -n nuodb
helm install insights ../../nuodb-dashboards-influx/charts/monitoring-influx/ --values values-insights.yaml -n nuodb
```

### Deploy hockey schema

```
nuosql demo@nuodb --user dba --password dba --file
/opt/nuodb/samples/quickstart/sql/create-db.sql
nuosql demo@nuodb --user dba --password dba --file
/opt/nuodb/samples/quickstart/sql/Players.sql
nuosql demo@nuodb --user dba --password dba --file
/opt/nuodb/samples/quickstart/sql/Scoring.sql
nuosql demo@nuodb --user dba --password dba --file /opt/nuodb/samples/quickstart/sql/Teams.sql
```

### Connect DBVis

```
kubectl get svc -n nuodb -l 'database=demo'
```



**Driver Properties**

Defines JDBC Driver or JNDI specific
when removing it while user defined
added manually.

| Edited | Parameter | Value |
|---|---|---|
| ☐ | isolation | (null) |
| ☐ | password | (null) |
| ☐ | schema | USER |
| ☐ | TimeZone | (null) |
| ☐ | user | (null) |
| ☑ | direct | true |

## Lab 1 - Resilience

### Environment Overview

Walk through:

1. Openshift components
    a. Admin stateful set
    b. SM stateful set
    c. TE deployment
    d. Storage
    e. Services
    f. Jobs
    g. Configmaps
2. NuoDB Helm charts repo https://github.com/nuodb/nuodb-helm-charts
    a. Dockerhub repo https://hub.docker.com/r/nuodb/nuodb-ce
3. Values file (local)
4. Multi cloud
    a. NuoDB Domain across clouds
5. DBVisualizer connectivity
    a. Quick example of multi-master in two DBVis queries

## Loss of a Transaction Engine

1. Locally, initiate a connection to an admin pod and run nuosql
   a. `kubectl exec pod/admin-nuodb-amazon0-0 -it -n nuodb -- /bin/sh`
   b. `nuosql demo@nuodb --user dba --password dba`
2. `SELECT * FROM VW_PLAYER_STATS LIMIT 10;`
3. Check which TE we are connected to
   a. `SELECT * FROM SYSTEM.NODES WHERE ID = GETNODEID();`
   b. Check `nuocmd show domain` for the pod name
4. In the OCP IDE, delete one of the *other* pods.
5. Verify query execution while the pod restarts (no disconnect)
   a. Run `SELECT * FROM SYSTEM.NODES` showing loss and re-apparence of the TE
   b. Run `SELECT * FROM SYSTEM.NODES WHERE ID = GETNODEID();` to see that we are still connected to the same TE
6. Delete the pod we are connected to
   a. This time use `kill -9 -1` to force terminate all process
7. Verify SQL connection is lost, but can immediately reconnect before pod is reprovisioned
   a. `RECONNECT;`
   b. Verify new node id `SELECT * FROM SYSTEM.NODES WHERE ID = GETNODEID();`

## Loss of a Storage Manager

1. Locally, initiate a connection to an admin pod and run nuosql
   a. `kubectl exec pod/admin-nuodb-amazon0-0 -it -n nuodb -- /bin/sh`
   b. `nuosql demo@nuodb --user dba --password dba`
2. `SELECT * FROM VW_PLAYER_STATS LIMIT 10;`
3. In the OCP IDE, delete an SM pod
4. Verify query execution while the pod restarts (no disconnect)
   a. Run `SELECT * FROM SYSTEM.NODES` showing loss and re-appearance of the SM
   b. Run `SELECT * FROM SYSTEM.NODES WHERE ID = GETNODEID();` to see that we are still connected to the same TE
5. View logs of replacement SM (syncing to running)

## Loss of SM storage (In Pod)

1. Show config for autoRestore to latest backup

2. Show resolution of  backupset
    a. `nuocmd get value --key /nuodb/nuobackup/demo/amazon0/latest`
    b. `nuocmd get value --key /nuodb/nuobackup/demo/amazon0/?`
3. Rename archive on hotcopy SM
    a. `mv /var/opt/nuodb/archive/nuodb/demo /var/opt/nuodb/archive/nuodb/demo-old`
    b. Observe pod termination, replacement restore, sync to running

**Loss of admin**

1. Locally, initiate a connection to an admin pod and run nuosql
    a. `kubectl exec pod/admin-nuodb-amazon0-0 -it -n nuodb -- /bin/sh`
    b. `nuosql demo@nuodb --user dba --password dba`
2. `SELECT * FROM VW_PLAYER_STATS LIMIT 10;`
3. In the OCP IDE, delete an AP pod
4. Verify query execution while the pod restarts (no disconnect)
    a. Run `SELECT * FROM SYSTEM.NODES` showing loss and re-appearance of the AP
    b. Run `SELECT * FROM SYSTEM.NODES WHERE ID = GETNODEID();` to see that we are still connected to the same TE

**YCSB**

1. Throughout the above, Yahoo Cloud Services Benchmark (YCSB) has been running on both clusters
2. Review SQL transactions throughout various component losses

**Loss of a node**

1. In compute > machines identify a node with a good mixture of AP/SM/TE pods
2. Delete the machine and wait for it to re-provision (5-10min).  Note behaviour:
    a. Because the topology only has one node per zone, the SM pods cannot reschedule immediately (EBS storage is zoned)
    b. However a TE can move nodes and re-schedule immediately
    c. This is a good reason to have more than one node per zone
3. Observe OCP pod recovery
    a. Note backoff loop may cause delay between node available and pod provisioning
4. Connect to nuosql and run a query as above during provisioning
5. Observe insights

### Network Faults - Disconnection between clouds

1. Scale down the VPN pod in the 3rd cloud
2. Observe the domain from each cloud
    a. This will cause a disconnect between the 3 clusters leaving two possible connected networks for majority evaluation - a majority and minority
    b. NuoDB has a configurable timeout for the time period until it decides a node has gone away uncleanly (this demo used 60s)
    c. During this time NuoDB will no longer process write transactions in safe commit mode
3. Observe YCSB workload
    a. Observe the pause in processing
    b. Followed by resume
4. Observe the new domain state after failure resolution has occurred
5. Reinstate the network ACL
    a. Observe after K8S backoff connectivity is automatically restored

## Lab 2 - Backup & Restore

### Backup

1. Review backup configuration in values file
2. Review backup jobs in OCP
3. Review hotcopy SM
    a. Backup location
    b. Backupset contents

### autoRestore

1. Demonstrated during SM loss
2. Show values configuration
    a. `:group-latest`
    b. SFTP path for off-site

### Restore Chart

1. Show restore chart / values
2. Explain conceptual differences between *restore* and *autoImport*
3. Explain difference between *:latest* and *:group-latest*
4. Create a new object (schema, table etc)
    a. `CREATE SCHEMA AARON;`
5. Run Restore chart

   a.
   ```
   helm install restore ../../nuodb-helm-charts/stable/restore --values values-restore.yaml -n nuodb
   ```

6. Show SM logs
   a. Tag resolution
   b. First In logic
7. Watch :latest resolution, restore, sync, running
8. Verify new object is not present

**PiT Restore**

1. Create a table and run a number of identifiable transactions
   a.
   ```
   nuosql demo@nuodb --user dba --password dba
   ```
   b.
   ```
   CREATE TABLE TEST.TRANSACTIONS (id int GENERATED ALWAYS AS IDENTITY, data string, tran_date time, sleep int, tran_id int);
   ```
   c. Insert some data with delays
      i.
      ```
      SET DELIMITER @
      ```
      ii.
      ```
      VAR $i int = 1; WHILE ($i<=10) START TRANSACTION; INSERT INTO TEST.TRANSACTIONS (tran_date, data, sleep, tran_id) SELECT NOW(),CHAR(64+$I),MSLEEP($i*1000),GETTRANSACTIONID() FROM DUAL; COMMIT; $i=$i+1; END_WHILE;@
      ```
      iii.
      ```
      SET DELIMITER ;
      ```
   d.
   ```
   SELECT * FROM TEST.TRANSACTIONS;
   ```
   e. Take a note of the data
2. Prepare a point in time archive
   a. Shell to a hotcopy SM
   b. Take  a PiT backup
      i.
      ```
      nuobackup --db-name demo --type journal --group amazon0 --timeout 600 --backup-root /var/opt/nuodb/backup
      ```
   c. View the backups in the backup set
      i.
      ```
      nuoarchive restore --report-backups /var/opt/nuodb/backup/?/
      ```
   d. Note the option to restore to an element id or snapshot (transaction id)
   e. Choose a previous journal backupset to prepare and view the transactions
      i.
      ```
      nuoarchive restore --report-timestamps --start-time 2020-07-30T22:00:00 --end-time 2020-07-31T00:00:00 /var/opt/nuodb/backup/?/
      ```
   f. Create a PiT archive

      i. `nuoarchive restore --restore-dir /var/opt/nuodb/backup/pit-restore --restore-snapshot ? /var/opt/nuodb/backup/?/`

3. Restore to the Point In Time archive
   a. tar/gz the pit-restore archive and SFTP it to a cold storage location
      i. `tar -czf pit-restore.tar.gz pit-restore`
      ii. `sftp scale2020@sftp.nuodb.com`
   b. Review the autoImport of a new deployment
      i. Note that this is also how a database can be duplicated
      ii. Or restoring the Temenos ISB
   c. Deploy the copy database and review the table contents reflect the point in time
      i. `helm install database-pit ../../nuodb-helm-charts/stable/database/ --values values-aws-pit.yaml -n nuodb`
   d. Review the data to ensure correct PiT was restored
      i. `SELECT * FROM TEST.TRANSACTIONS;`

## Lab 3 - Scaling & Ops

### Scaling

1. Deploy with reduced spec TE pods (1vcpu, 1Gb ram, 1 pod per cloud)
2. Start 1 YCSB pod per cloud
   a. Observe dashboard screen
3. Increase YCSB to 2 pods per cloud
   a. Observe limited increase in TPS
4. Increase TE to 2 pods per cloud
   a. Observe TPS increase to handle YCSB load
5. Increase TE pods to 3 per cloud
   a. Observe no increase in TPS (load was handled)
6. Scale down workload
7. Scale down TEs

### Autoscaling

1. Show HPA configuration
   a. `kubectl autoscale deployment te-database-nuodb-amazon0-demo --cpu-percent=50 --min=1 --max=10 -n nuodb`
   b. `kubectl autoscale deployment te-database-nuodb-azure0-demo --cpu-percent=50 --min=1 --max=10 -n nuodb`
   c. Note that options / support for scaling implementations varies by k8s distribution

2. Increase YCSB pods
3. Watch TE pods increase automatically (while this is happening, show Rancher below)
4. Scale down to 1 YCSB pod

## Rancher

1. Walk around rancher rig - single pane of glass for multiple cluster management
    a. Show CNI config in clouds, multiple tunnels
    b. Show Coredns configuration
    c. Show specific changes to NuoDB deployment for referencing multiple clusters

## Other Ops

1. TLS configuration
    a. YAML config
    b. Vault integration
2. Example TLS connection
    a. `nuosql demo@nuodb --user dba --password dba --connection-property trustStore=/etc/nuodb/keys/ca.cert --connection-property verifyHostname=false`
    b. Repeat with incorrect password
3. Logging & Monitoring
    a. Papertrail config & interface
    b. Show previous failed auth
    c. Show slack alerts
4. System tables
    a. `select * from system.tables;`
    b. `select * from system.users;`
    c. `select * from system.nodes;`
    d. `select * from system.globalatoms limit 10;`
    e. `select * from system.connections;`
    f. `select * from system.transactions;`
    g. `select * from system.querystats; (select MSLEEP(11000) FROM DUAL;)`
    h. https://doc.nuodb.com/nuodb/latest/sql-development/scalar-engine/sql-reference-information/sql-system-tables/
5. Dynamic logging / tracing
    a. `SET GLOBAL TRACE ON; (AND OFF)`
    b. `select * from traceschema.tracetable`
    c. `nuocmd get log-messages --db-name demo --log-options sql-statements,sql-statement-metrics`
    d. `SELECT * FROM VW_PLAYER_STATS LIMIT 10;`

   e. `SET GLOBAL TRACE OFF;`
   f. `CREATE SCHEMA TEST;`
   g. `DDL in papertrail`

6. Statistics
   a. `https://doc.nuodb.com/nuodb/latest/sql-development/scalar-engine/using-sql-trace-facility/`
7. EXPLAIN
   a. `EXPLAIN SELECT * FROM VW_PLAYER_STATS LIMIT 10;`
   b. `Index hints, query hints`
8. Scale down YCSB

## Clean delete / install

1. Remove the existing deployment
   a. `helm delete database -n nuodb`
   b. `helm delete admin -n nuodb`
   c. `kubectl delete pvc -n nuodb -l 'group=nuodb'`
   d. `kubectl get pv | grep "Released" | awk '{print $1}' | xargs kubectl delete pv`
2. Re-install as per opening section commands

## Rolling Upgrade

1. Install with  4.0.4-2-mc
   a. Show output of `nuocmd show domain`
   b. Show output of `nuocmd show database-versions --db-name demo`
2. Start YCSB workload
3. Upgrade one cloud to 4.0.5-4-mc
   a. `helm upgrade admin ../../nuodb-helm-charts/stable/admin/ --values values-?.yaml -n nuodb`
   b. `helm upgrade database ../../nuodb-helm-charts/stable/database/ --values values-?.yaml -n nuodb`
   c. Show output of `nuocmd show domain`
   d. Show output of `nuocmd show database-versions --db-name demo`
4. Upgrade other clouds
   a. Show output of `nuocmd show domain`
   b. Show output of `nuocmd show database-versions --db-name demo`