

Tungsten Cluster Master Class

Intermediate: Backup & Recovery

Matthew Lang, Customer Success Director, Americas



The MySQL Availability Company

Topics

In this short course, we will discuss:

- How to develop a backup plan
- Methods and tools for taking a backup
- Verifying the back contains the last binary position, and the importance of this.
- Restore backups into the cluster
- Provision a replica from an existing Datasource

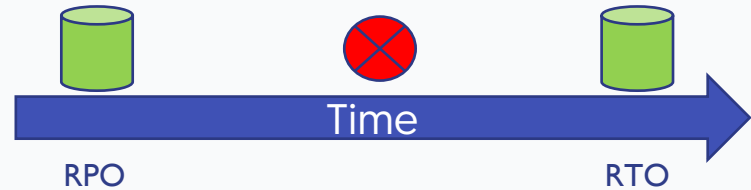


Backup Plan



Backup Plan – Considerations

- How much data can you realistically afford to lose in a disaster?
 - Business decision that will have an impact on backups
 - The less you can lose, the more complex the backup plan
 - RPO – Recovery Point Objective
- How long can you realistically be offline in a disaster?
 - Restoring a backup takes time
 - Less time offline adds complexity to the plan
 - RTO – Recovery Time Objective
- Where will you keep your backups?
 - Requires provisioning of space
 - Must be durable
 - Must factor in backup retention time



Backup Plan – Best practices

Do NOT Do This!

- “We don’t take backups, we use replication instead”
- Backups should not be considered viable until they have been restored
 - Backups need to be tested
 - Backup process can also test restore
- Restore process for production needs to be documented
 - Avoid panic during a disaster
 - Test regularly



Backup Plan – Best practices

- 3-2-1 rule:
 - 3 copies total copies of your data
 - 2 local copies
 - Good: Separate disk
 - Better: Separate servers
 - Even Better: Separate LANs
 - Yes, this applies for cloud too.
 - For cloud, 2nd copy in another account
 - 1 offsite copy
 - Cloud if using on site production
 - On site or another cloud if using cloud
 - Protects again unauthorized access



Backup Tools

- mysqldump
 - Installed with all MySQL deployments
 - Easy to make backup, easy to restore
 - Quick for small datasets
 - Slow for large datasets, especially restore
 - Locks tables
 - Adds load to MySQL server
- xtrabackup
 - Free
 - Online, does not lock tables
 - Can load server, however does not put load on MySQL itself
 - Ideal for larger datasets
 - A lot of options, though confusing



Backup tools 2

- Snapshots (lvm, cloud, via hypervisor, etc)
 - Ideal for very large datasets
 - Extremely quick
 - Requires working at the OS or hardware level
 - Need to backup the snapshot – extra step
 - Having several snapshots will impact disk performance
- File copy (rsync and others)
 - Fast
 - Works with large datasets
 - Familiar tools
 - Requires scripting to automate
 - Must stop MySQL server to get a viable backup



Taking and Restoring Backups



Within a Cluster

```
[LOGICAL:EXPERT] /alpha > datasource host3 backup
```

Using the 'mysqldump' backup agent.

```
Replicator 'host3' starting backup Backup of dataSource 'host3' succeeded;  
uri=storage://file-system/store-0000000001.properties
```

- Select a secondary when using this to keep load off of the primary
- Backups by default will be saved into /opt/continuent/backups
- Backup command will use xtrabackup if available, otherwise it will use mysqldump
- A properties file is also created, which includes backup method and checksum



Automate backups within the cluster

- Use the `cluster_backup` command within cron on all database nodes
- `cluster_backup` will ONLY run on the coordinator; it will gracefully exit on other hosts
- The coordinator will select a secondary to backup
- Exactly like running `datasource host backup` in `ctrl`

```
shell> crontab -l  
00 00 * * * /opt/continuent/tungsten/cluster-home/bin/cluster_backup >>/opt/continuent/service_logs/cluster_backup.log 2>&1
```



Restoring a backup within a Tungsten Cluster

```
[LOGICAL] /alpha > datasource host3 shun  
[LOGICAL] /alpha > replicator host3 offline  
[LOGICAL] /alpha > datasource host3 restore
```

- This restore process will restore the latest available backup
- Only searches for backups on the host specified, does NOT search all nodes for the latest backup

```
[LOGICAL] /alpha > datasource host3 shun  
[LOGICAL] /alpha > replicator host3 offline  
[LOGICAL] /alpha > datasource host3 restore storage://file-  
system/store-0000000004.properties
```

- Specifying a properties files allows you to select a particular backup to restore



External backups

- Use your favorite method to take a backup
- When backing up a primary, you MUST record the binlog position of the backup
 - Binary log position may be different than the values in `trp_commit_seqno`
 - For `mysqldump`, include `--master-data=2`, which will include a line in the dump:

```
CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000002', MASTER_LOG_POS=622;
```
 - `xtrabackup` always records the position in `xtrabackup_binlog_info`
 - For file copy or snapshot backups on a primary:
 - quiesce the database, then
 - run `SHOW MASTER STATUS;` in MySQL to obtain the binary log position
 - Execute backup
 - Record the binary log position with the backup



Restoring an external backup

- SHUN the node if it's not already Shunned or Failed (`datasource host shun`)
- Take the replicator offline (`treptcl -all-services offline`)
- MUST remove ALL THL: `thl -service service-name purge`
- Stop the MySQL server
- Perform the restore (load a previous dump, xtrabackup copy-back, file copy, etc)
- If backup was taken from a primary, we must perform some additional steps (on following pages)
- Verify ownership and permissions of database files
- Start MySQL
- Finally, RECOVER the node



Restoring from a primary



The Tungsten tracking schema

- Tungsten creates a “tracking” schema in the MySQL database
- GTID and binlog positions are stored within this schema
- When a database is restored, Tungsten uses the data in the tracking schema to set the replicator position

```
mysql> select * from tungsten_east.trep_commit_seqno;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| task_id | seqno | fragno | last_frag | source_id | epoch_number | eventid |
applied_latency | update_timestamp | shard_id | extract_timestamp |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          0 | 190 | 0 | 1 | db1 | 0 | mysql-bin.000002:0000000000048972;32826
|          1 | 2017-10-17 03:20:39 | music | 2017-10-17 03:20:38 |
```



Setting the correct position

- When backing up a busy primary, the actual binary log position may be different than the values in the `trep_commit_seqno` table
- Compare the binary log values in `trep_commit_seqno` with the values obtained from the backup (`mysqldump master-data` or `xtrabckup_binlog_info`)
- If the values are the same, no further action is needed
- If the values differ, note the sequence number in `trep_commit_seqno` table
- Find the binlog position in the THL, using the sequence number from the above step as a starting point
- Note this new sequence number obtained from the THL, and use `dsctl` on the slave node to set the replicator position with this new sequence number



Setting the correct position (example)

```
mysql> select seqno,epoch,source_id, event_id from tungsten_east.trep_commt_seqno;  
seqno      epoch      source_id  eventid  
32033674   32032892   host1      mysql-bin.000032:0000000473860407;-1
```

Examine binary log position in backup. In our example, the position is `mysql-bin.000032:473863524`.

Now on the primary:

```
Shell> thl list -service alpha -low 32033674 -headers | grep 473863524  
32033678 32030709 0 true 2014-10-17 16:58:11.0 mysql-bin.000032:0000000473863524;-1 db1-east.continuent.com
```

On the secondary, use
Info from above:

```
Shell> dsctl set -service alpha -epoch 32030709 -source-id host1 -seqno 32033678 -event-id "mysql-  
bin.000032:0000000473863524;-1" -reset
```



tungsten_provision_slave



tungsten_provision_slave

- **tungsten_provision_slave** allows you to easily provision a node from a remote host
- Is run on the secondary to be provisioned
- Runs the backup on the remote hosts specified by the `source` option
- Copies the backup to the local server and restores it
- Will recover replication services and have the newly provisioned node (re)join the cluster
- Defaults to `mysqldump`, but can use `xtrabackup`
- Will perform all steps previously discussed when provision from a primary node

```
shell@host3> tungsten_provision_slave --source host2
```

- When run on `host3`, will provision from host `host2`
- Best to run in a `screen` session!



Demo: tungsten_provision_slave



Summary

What we have learnt today

- Factors to consider for your backup plan
 - RPO
 - RTO
 - 3-2-1 rule
 - Testing
- Review of backup tools
- Backup and restore within the cluster
- External backups and restore
- Special considerations when restoring a backup taken from a Primary
- Easy provisioning with `tungsten_provision_slave`



Next Steps

In the next session we will

- Review Multi Cluster Topologies
- Understand Multi Cluster Architecture
- Explore Configuration Differences



THANK YOU FOR LISTENING

continuent.com

Matthew Lang, Customer Success Director, Americas



The MySQL Availability Company