Tungsten Cluster Master Class

Intermediate: Monitoring & Troubleshooting

Matthew Lang, Customer Success Director, Americas

continuent

The MySQL Availability Company

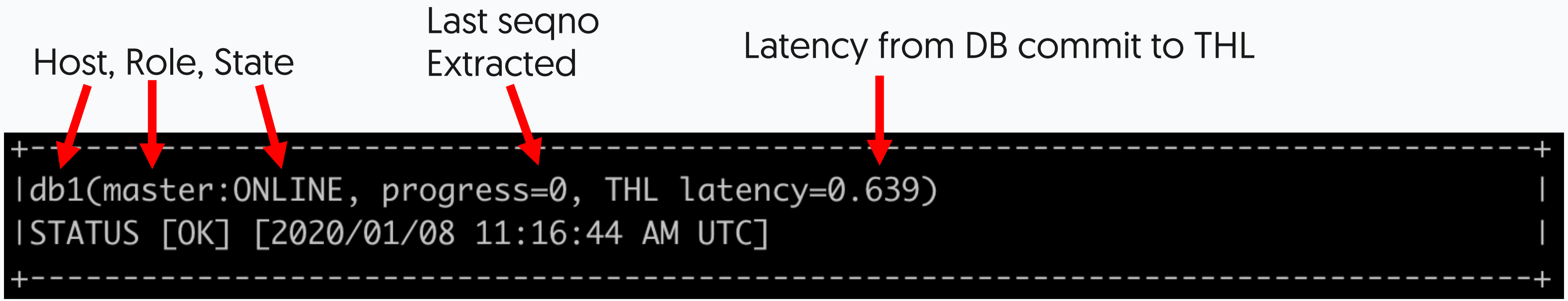
Topics

In this short course, we will

- Discuss tools used to monitor cluster health
- Discuss tools used to identify issues
- How to get more information about issues using the logs
- Resolve common replication issues
- Resolve common clustering issues
- Get more information about replication lag

Monitoring Cluster Health

cctrl [review]

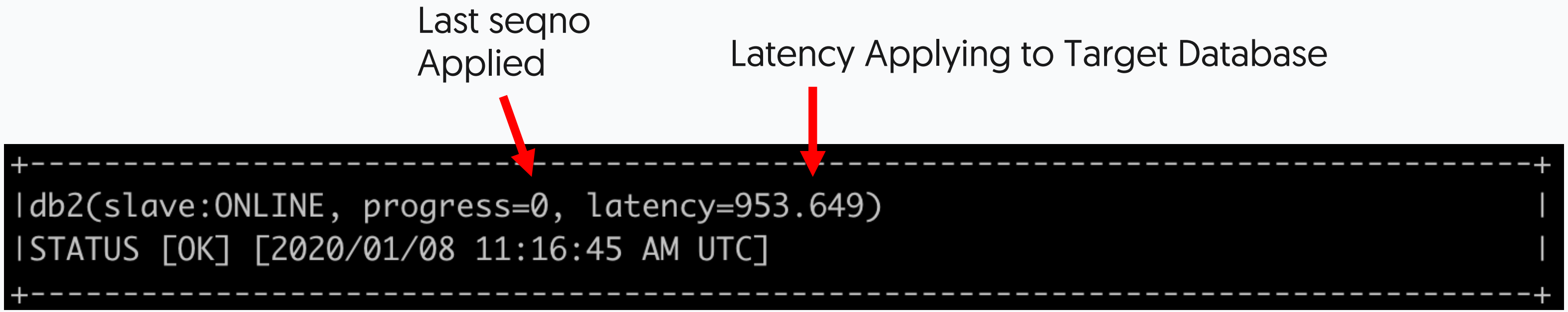


Host, Role, State

Last seqno Extracted

Latency from DB commit to THL

```
+----------------------------------------------------------------------------------+
|db1(master:ONLINE, progress=0, THL latency=0.639)                                 |
|STATUS [OK] [2020/01/08 11:16:44 AM UTC]                                          |
+----------------------------------------------------------------------------------+
```



Last seqno Applied

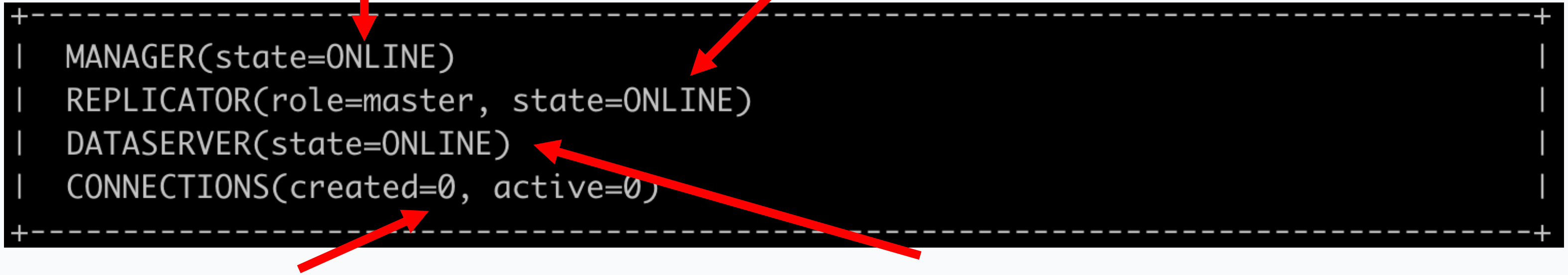
Latency Applying to Target Database

```
+----------------------------------------------------------------------------------+
|db2(slave:ONLINE, progress=0, latency=953.649)                                    |
|STATUS [OK] [2020/01/08 11:16:45 AM UTC]                                          |
+----------------------------------------------------------------------------------+
```

cctrl

Status of manager process

Role and status of replicator process

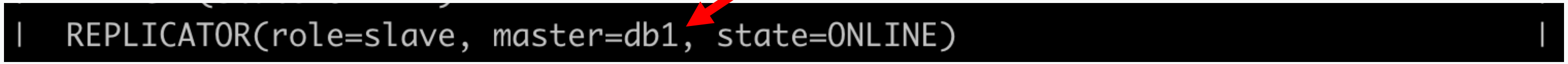


```
+----------------------------------------------------------------------------------+
|  MANAGER(state=ONLINE)                                                            |
|  REPLICATOR(role=master, state=ONLINE)                                            |
|  DATASERVER(state=ONLINE)                                                         |
|  CONNECTIONS(created=0, active=0)                                                 |
+----------------------------------------------------------------------------------+
```

Connections through Connectors to DB

Status of MySQL Database

Role and status of replicator process. On a replica, shows primary the replica is connected to



```
|  REPLICATOR(role=slave, master=db1, state=ONLINE)                                 |
```

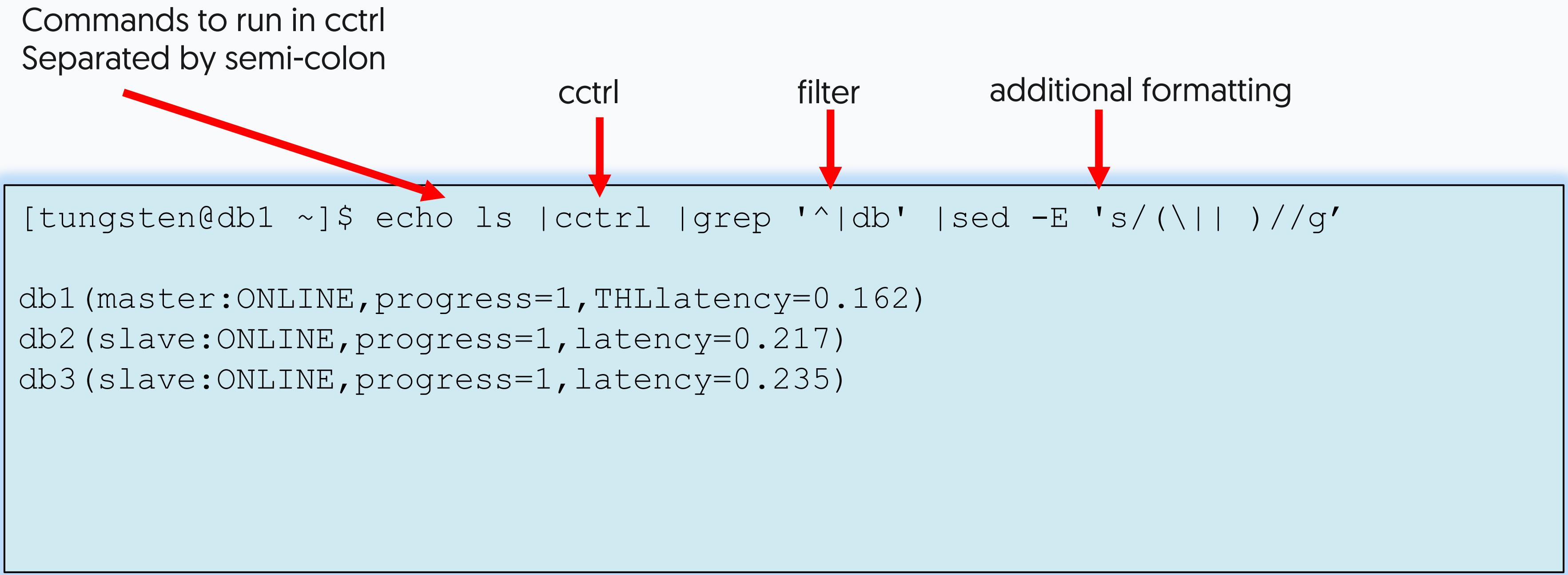
Datasource States

- `ONLINE`
- `OFFLINE`
  - Node will not accept connections through the connector for reads or writes.
  - When cluster policy is `AUTOMATIC`, an offline datasource will be automatically brought back into the cluster.
  - If the cluster policy is `MAINTENANCE`, the datasource remains offline until explicitly brought back online.
- `SHUNNED`
  - Similar to OFFLINE
  - However, a SHUNNED datasource will NOT be automatically recovered into the cluster.
  - Nodes can manually SHUNNED, or can be SHUNNED due to an error
  - Typically used to perform maintenance on the underlying system.
- `FAILED`
  - A service has failed on the host, such as the MySQL server
  - After correcting the issue, the node can be bought into the cluster using the `recover` command

Replicator States (most common)

- `ONLINE`
- `OFFLINE:NORMAL`
  - Replicator has been put into an OFFLINE state
  - no Replicator issues
  - If cluster policy is `AUTOMATIC`, manager will bring replicator `ONLINE`
- `GOING-ONLINE:SYNCHRONIZING`
  - Secondary Replicator is trying to connector to source, however source may be unavailable
  - Source Replicator is offline or perhaps there's a connectivity/firewall issue between the Replicators
- `OFFLINE:ERROR`
  - Replicator has gone OFFLINE due to an error
  - `trepctl status` will show a description of the error

Getting fancy with cctrl



Commands to run in cctrl
Separated by semi-colon

cctrl

filter

additional formatting

```
[tungsten@db1 ~]$ echo ls |cctrl |grep '^|db' |sed -E 's/(\|| )//g’

db1(master:ONLINE,progress=1,THLlatency=0.162)
db2(slave:ONLINE,progress=1,latency=0.217)
db3(slave:ONLINE,progress=1,latency=0.235)
```

`check_tungsten_*` scripts

- Nagios style scripts that report on key cluster health metrics:
- `check_tungsten_backups` – backups present on any datasource
- `check_tungsten_latency` – reports warning or critical if latency is above thresholds
- `check_tungsten_online` – tungsten services online
- `check_tungsten_policy` – tungsten cluster policy automatic
- `check_tungsten_progress` – replicator is making progress in cluster
- `check_tungsten_services` – tungsten services running

check_tungsten_* examples

```
[tungsten@db1 ~]$ check_tungsten_policy
CRITICAL: Policy is MAINTENANCE

[tungsten@db1 ~]$ check_tungsten_online
OK: All services are online

[tungsten@db1 ~]$ check_tungsten_services -r
CRITICAL: Replicator is not running
```

Log files

The big “3”

```
CONTINUENT_ROOT=/opt/continuent
```

1. Manager log: `$CONTINUENT_ROOT/tungsten/tungsten-manager/log/tmsvc.log`
2. Replicator log: `$CONTINUENT_ROOT/tungsten/tungsten-replicator/log/trepsvc.log`
3. Connector log: `$CONTINUENT_ROOT/tungsten/tungsten-connector/log/connector.log`

Mining for Errors

- Example 1: Manager log

```
2020/08/03 02:56:05 | ERROR |             db2 | ERROR [MySQLIOs] - SOCKET_IO_ERROR
2020/08/03 02:56:05 | ERROR |             db2 | ERROR [MySQLIOs] - I/O exception caught
while connecting to a socket to db2:13306
2020/08/03 02:56:05 | ERROR |             db2 | ERROR [MySQLIOs] -
Exception='java.net.ConnectException: Connection refused (Connection refused)'
```

- Messages with ERROR
- In this case, the manager can’t connect to MySQL on db2, port 13306. Is MySQL running?

Mining for Errors

- Example 2: Replicator log

```
2020/08/03 03:01:23.614 | Event application failed: seqno=3 fragno=0 message=java.sql.SQLException: Statement failed on
slave but succeeded on master
2020/08/03 03:01:23.614 | create database matt
2020/08/03 03:01:23.614 | com.continuent.tungsten.replicator.applier.ApplierException: java.sql.SQLException: Statement
failed on slave but succeeded on master
2020/08/03 03:01:23.614 |       at
com.continuent.tungsten.replicator.applier.MySQLDrizzleApplier.applyStatementData(Unknown Source)
2020/08/03 03:01:23.614 |       at com.continuent.tungsten.replicator.applier.JdbcApplier.apply(Unknown Source)
2020/08/03 03:01:23.614 |       at com.continuent.tungsten.replicator.applier.ApplierWrapper.apply(Unknown Source)
2020/08/03 03:01:23.614 |       at com.continuent.tungsten.replicator.pipeline.SingleThreadStageTask.apply(Unknown Source)
```

- Java stack trace
- In this case, there was an error applying to replica (database already exists?)

Mining for Errors

- Example 3: Connector log

```
2020/08/03 03:20:09 | ERROR [App] - Configuration error:
2020/08/03 03:20:09 | com.continuent.tungsten.common.config.cluster.ConfigurationException:
2020/08/03 03:20:09 | Bad entry at line 40 of file '../../tungsten-connector/conf/user.map',
text='@{CONN_PASSWORD_LINES}'. Found 1 items while expecting 3 or 4. The format of the line should be '<user> <password>
<data service> [<preferred site>]'
2020/08/03 03:20:09 |   at org.continuent.myosotis.configuration.UserMap.readConfigFrom(Unknown Source)
2020/08/03 03:20:09 |   at org.continuent.myosotis.configuration.UserMap.readConfig(Unknown Source)
2020/08/03 03:20:09 |   at org.continuent.myosotis.configuration.Configuration.parseProps(Unknown Source)
```

- Java stack trace
- In this case, there is a typo on the file user.map

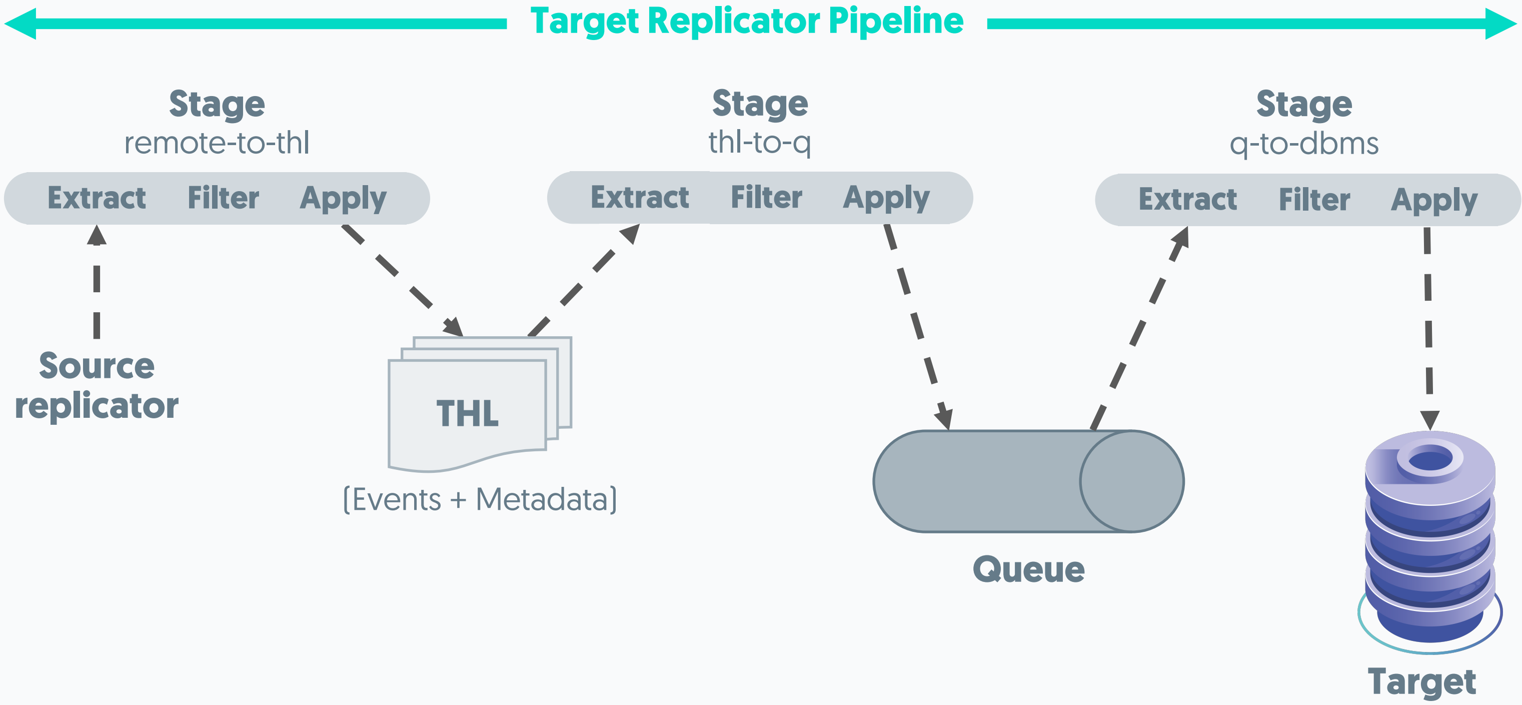
Gather log files for support

- `tpm diag`
  - Creates a .zip file containing tungsten logs, MySQL logs, and various OS settings
  - This file should be attached to a support ticket for Continuent Staff
  - For INI deployments, versions 5.3.7, 5.4.0, and 6.0.5+, `tpm diag` will collect logs from current host only, unless otherwise specified with `--hosts` or `-allhosts`
  - Feel free to examine!
- `tungsten_send_diag`
  - Like `tpm diag`, but can upload the .zip directly to Continuent
  - It can create the zip, or simply use it to upload an existing .zip file

Demo

Diagnosing Replication Latency

Understanding Replicator Pipelines



The MySQL Availability Company

Main Causes of Latency

1. **Large Transaction**
2. **Schema Change**

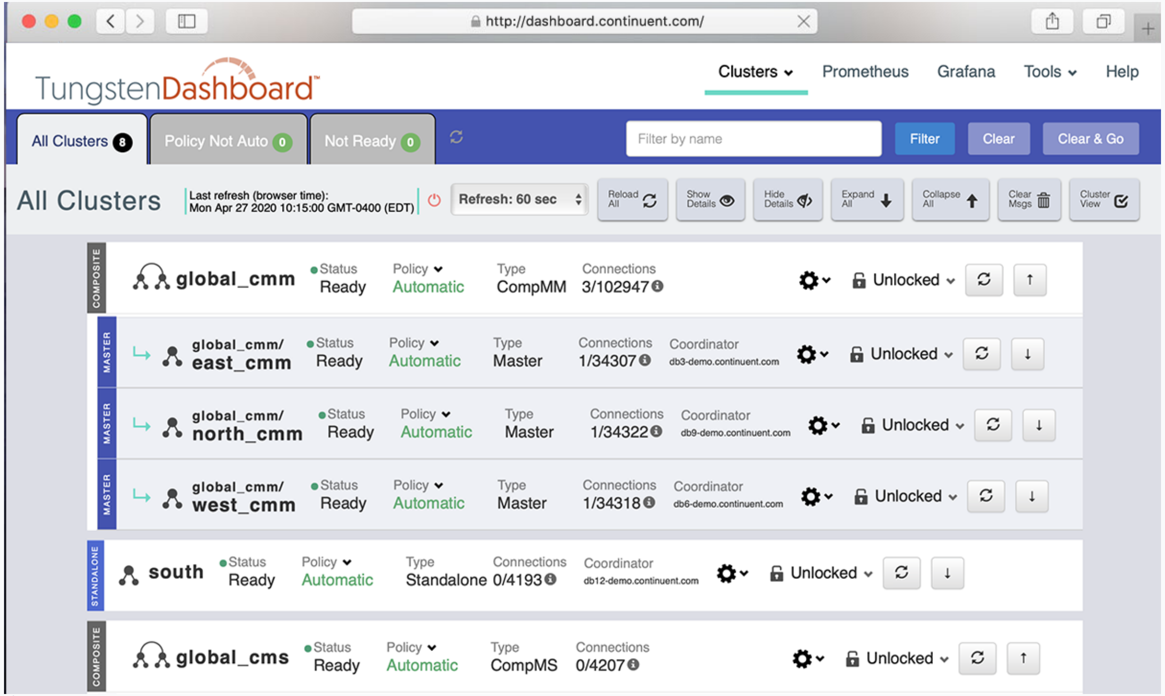
- Transactions must be applied in serially in sequence
- Newer transactions must wait for the large transaction to complete
- A schema change is a large transaction if target table must be rebuilt

Using `trepctl perf`

- Displays performance statistics for each stage in the replicator
- Supports auto refresh option `-r s` where *`s`* is the number of seconds to refresh

```
Statistics since last put online 38.418s ago
Stage         | Seqno | Latency | Events | Extraction | Filtering | Applying | Other |  Total
remote-to-thl | 3246  | 1.143s  | 42     | 37.831s    | 0.001s    |  0.403s  | 0.011s| 38.246s
                      Avg time per Event |  0.901s    | 0.000s    |  0.000s  | 0.010s|  0.911s
thl-to-q      | 3246  | 1.209s  | 1654   | 37.113s    | 0.005s    |  1.090s  | 0.098s| 38.306s
                      Avg time per Event |  0.022s    | 0.000s    |  0.000s  | 0.001s|  0.023s
q-to-dbms     | 3235  | 3.746s  | 1644   | 22.226s    | 0.019s    | 15.242s  | 0.338s| 37.825s
                      Avg time per Event |  0.014s    | 0.000s    |  0.000s  | 0.009s|  0.023s
                        Filters in stage | mysqlsessions -> pkey
```

Dashboard (Separate Training Session)



All Clusters

Last refresh (browser time):
Mon Apr 27 2020 10:15:00 GMT-0400 (EDT)

Refresh: 60 sec

	Name	Status	Policy	Type	Connections	Coordinator	
COMPOSITE	global_cmm	Ready	Automatic	CompMM	3/102947		Unlocked
MASTER	global_cmm/ east_cmm	Ready	Automatic	Master	1/34307	db3-demo.continuent.com	Unlocked
MASTER	global_cmm/ north_cmm	Ready	Automatic	Master	1/34322	db9-demo.continuent.com	Unlocked
MASTER	global_cmm/ west_cmm	Ready	Automatic	Master	1/34318	db6-demo.continuent.com	Unlocked
STANDALONE	south	Ready	Automatic	Standalone	0/4193	db12-demo.continuent.com	Unlocked
COMPOSITE	global_cms	Ready	Automatic	CompMS	0/4207		Unlocked

Summary

What we have learnt today

- Command line tools to monitor health
- Cluster and Replicator States
- How to examine log files
- Collecting log files for support
- How to Diagnose Replication Lag

Next Steps

In the next session we will

- Discuss Backup and Recovery
    - Backup through cctrl
    - Automating backups
    - How to reprovision
        - `tungsten_provision_slave`
        - Manual methods

THANK YOU FOR LISTENING

continuent.com

Matthew Lang, Customer Success Director, Americas

continuent

The MySQL Availability Company