TimesTen Scaleout 18.1 PoC Guide

Contents

1. TimesTen Product Overview

Oracle TimesTen In-Memory Database (TimesTen) is a next generation full-featured, memory-optimized, relational database. It provides applications with the instant responsiveness and very high throughput required by database-intensive applications. TimesTen databases reside entirely in physical memory (RAM) at runtime. Applications access the TimesTen database using standard SQL interfaces. TimesTen database can elastically scale according to the demand of your business. For customers with existing data residing on the Oracle Database, TimesTen can be deployed as an in-memory cache database with automatic data synchronization between TimesTen and the Oracle Database.

TimesTen can be deployed in three different ways:

-- TimesTen Classic: A standalone database (with optional high availability) primarily focused on instant responsiveness

-- TimesTen Cache: TimesTen Classic can also be used as a cache to an existing Oracle Database using the TimesTen Cache features

-- TimesTen Scaleout: A SQL based, distributed, elastically scalable, shared-nothing RDBMS with automatic fault tolerance and transparent data distribution

In this PoC guide, we will focus on TimesTen 18.1 new feature – TimesTen Scaleout introduction:

In TimesTen Scaleout, a TimesTen database is deployed in a multi-node environment as a distributed database. Applications with high throughput requirements can run in this mode by taking advantage of the concurrent parallel cross-node processing, transparent data location and elastic scalability of TimesTen. High availability and fault tolerance are provided by TimesTen's K-safety feature.

Key features of TimesTen Scaleout:

-- Built on the foundation of a mature, robust and high performance in-memory database (TimesTen)

-- Combines the power of many computers into a single logical database with a shared nothing architecture

-- Transparent, automatic data distribution with a single database image for simplicity

-- Automatic high-availability via K-safety

-- Fully distributed, high-performance ACID transactions provide data consistency at all times

-- Centralized deployment, management and monitoring for easy administration

-- Uses standard database APIs and standard SQL

Supported platform:

Oracle Linux, Red Hat Enterprise Linux 6.4+ or 7+, SUSE 12+

(glibc 2.12+)

Network:

Support TCP/IP or IPoIB
Two Network Interfaces are recommended.
However, in this PoC lab, we will use only one network interface to simplify the hardware.

Refer to

https://blogs.oracle.com/timesten/easy-steps-to-configure-a-timesten-scaleout-environment-for-experimentation

And TimesTen In-Memory Database Scaleout User's Guide

https://docs.oracle.com/database/timesten-18.1/TTGRD/install.htm#TTGRD745

2. PoC Environment

2.1 Host Planning

hostname	IP address	Usage
ttgridmgmt01	192.168.10.98	Active management / ZooKeeper 1 (JDK 8 required)
ttgridmgmt02	192.168.10.99	Standby management / ZooKeeper 2 (JDK 8 required)
ttgridclnt01	192.168.10.100	Client / ZooKeeper 3 (JDK 8 required)
ttgriddata01	192.168.10.101	Data host 1 (Enough CPU and DRAM with fast storage using file systems like ext4, xfs or btrfs)
ttgriddata02	192.168.10.102	Data host 2 (Enough CPU and DRAM with fast storage using file systems like ext4, xfs or btrfs)
ttgriddata03	192.168.10.103	Data host 3 (Enough CPU and DRAM with fast storage using file systems like ext4, xfs or btrfs)
ttgriddata04	192.168.10.104	Data host 4 (Enough CPU and DRAM with fast storage using file systems like ext4, xfs or btrfs)
ttgriddata05	192.168.10.105	Data host 5 (Enough CPU and DRAM with fast storage using file systems like ext4, xfs or btrfs)
ttgriddata06	192.168.10.106	Data host 6 (Enough CPU and DRAM with fast storage using file systems like ext4, xfs or btrfs)

Edit /etc/hosts by adding

```
ttgridmgmt01 192.168.10.98
ttgridmgmt02 192.168.10.99
ttgridclnt01    192.168.10.100
ttgriddata01    192.168.10.101
ttgriddata02    192.168.10.102
ttgriddata03    192.168.10.103
ttgriddata04    192.168.10.104
ttgriddata05    192.168.10.105
ttgriddata06    192.168.10.106
```

2.2 ZooKeeper Membership Planning

HOSTNAME	Membership server (client/peer/leader)	Installation location	Configuration
ttgridmgmt01	2181 / 2888 / 3888	/ttgrid/zk/zookeeper-3.4.10	/ttgrid/zk/zookeeper-3.4.10/conf/zoo.cfg /ttgrid/zk/zookeeper-3.4.10/data/myid /ttgrid/inst/mgmtinst/grid/conf/membership.conf （myid file should be created with number in each of the host. E.g. In ttgridmgmt01, myid file includes 1, ttgridmgmt02's myid includes 2, the ttgridclnt01' myid file includes 3）
ttgridmgmt02	2181 / 2888 / 3888	/ttgrid/zk/zookeeper-3.4.10	
ttgridclnt01	2181 / 2888 / 3888	/ttgrid/zk/zookeeper-3.4.10	

2.3 Management Host Planning

Hostname	Management instance (daemon/client/management)	Installation location	Instance base location

ttgridmgmt01	6624 / 6625 / 3754	/ttgrid/install	/ttgrid/inst
ttgridmgmt02	6624 / 6625 / 3754	/ttgrid/install	/ttgrid/inst

2.4 Data Host Planning

Hostname	Data instance (daemon/client)	Installation location	Instance base location
ttgriddata01	63388/20233	/ttgrid/install	/ttgrid/inst
ttgriddata02	63388/20233	/ttgrid/install	/ttgrid/inst
ttgriddata03	63388/20233	/ttgrid/install	/ttgrid/inst
ttgriddata04	63388/20233	/ttgrid/install	/ttgrid/inst
ttgriddata05	63388/20233	/ttgrid/install	/ttgrid/inst
ttgriddata06	63388/20233	/ttgrid/install	/ttgrid/inst

2.5 Database Planning

Database Name	Checkpoint File location	Log file location	Characterset	PermSize / TempSize	Log Buffer Size
sampledb	/ttgrid/db	/ttgrid/db/log	AL32UTF8	4096 (MB) / 1024 (MB)	1024 (MB)

Note: TimesTen software installations will be in the **/ttgrid/install** directory.
The zookeeper software will be installed in the **/ttgrid/zk** directory.
Instances will be created under **/ttgrid/inst** directory.
User database files will be stored under **/ttgrid/db** (data instance hosts only).
Repository storage will be under **/ttgrid/repos** (client hosts only).

3. OS Configuration for each host

3.1 OS User and Group

Make sure all the hosts using the identical UID and GID
e.g.
Create a new group

```
# groupadd –g 2000 timesten
```

Create a new use

```
# useradd -u 2000 -g timesten timesten

# passwd timesten
```

Check OS Package

```
# rpm -q libaio
```

3.2 OS Kernel Settings

3.2.1 Edit the OS kernel setting file: /etc/sysctl.conf

e.g.

```
# vi /etc/sysctl.conf
```

```
# General settings for TimesTen Scaleout
kernel.sem = 4000 400000 2000 2560
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 49152 65535
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576

# Shared memory settings for TimesTen Scaleout
kernel.shmmni = 4096
vm.hugetlb_shm_group = 1000

#  settings for 8 GB RAM
kernel.shmmax = 8589934592
kernel.shmall = 2097152
# uncomment next line to allocate 6 GB of huge pages
#vm.nr_hugepages = 3072

#  settings for 16 GB RAM
#kernel.shmmax = 17179869184
#kernel.shmall = 4194304
# uncomment next line to allocate 14 GB of huge pages
#vm.nr_hugepages = 7168

#  settings for 32 GB RAM
#kernel.shmmax = 34359738368
#kernel.shmall = 8388608
```

```
# uncomment next line to allocate 30 GB of huge pages
#vm.nr_hugepages = 15360
```

Execute the command to make settings take effective

```
# sudo /sbin/sysctl -p
```

Repeat the same configuration in each of the hosts

Note:

Maximum Shared segments

shmmax: The maximum size of a single shared memory segment expressed in bytes.
The value must be large enough to accommodate the size of the total shared memory segment for the element.

shmall: The total size of all shared memory segments system wide. The value is expressed in multiples of the page size (4 KB) and must be greater or equal to the value of shmmax.

It is recommended that you set the value of shmall to less than or equal to the total amount of physical RAM. To display the total amount of physical memory, run the Linux cat /proc/meminfo command.

The size of the element is based on the values of the PermSize, TempSize, LogBufMB and Connections connection attributes. The element sizing formula is:

*PermSize+TempSize+LogBufMB+1+(0.042 * Connections)*

Semaphores

Each user and system connection (a database connection) is assigned one semaphore, such that the total semaphores for a database are:

Total semaphores = user connections (N) + system connections (48) +
other required connections (107)

Total semaphores = N + 155
The semaphore settings are located in the kernel.sem configuration directive in /etc/sysctl.conf:

kernel.sem = SEMMSL SEMMNS SEMOPM SEMMNI
where:

SEMMSL is the maximum number of semaphores per array. Configure this value to 155 plus the number of connections.
*SEMMNS is the maximum number of semaphores system wide. Use the formula SEMMNS = (SEMMNI * SEMMSL) as a guideline.*
SEMOPM is the maximum number of operations for each semop call.
SEMMNI is the maximum number of arrays.

3.2.2 Configure the limits.conf file

```
# vi /etc/security/limits.conf
```

```
# /etc/security/limits.conf
#
#This file sets the resource limits for the users logged in via PAM.
#It does not affect resource limits of the system services.
…
timesten hard nproc 16384
timesten soft nproc 16384
timesten hard nofile 65536
timesten soft nofile 65536
timesten hard memlock 33554432
timesten soft memlock 33554432
# End of file
```

Repeat the same configuration in each of the hosts

3.3 Fileware settings

```
# vi /etc/selinux/config
```

```
SELINUX=disabled
```

```
# service iptables stop

# vi /etc/selinux/config
```

Repeat the same configuration in each of the hosts

3.4 File System Planning

According to the Plannings above, create the following directories on each of the host:

```
# umask 022

# mkdir -p /ttgrid

# chown -R timesten:timesten /ttgrid
```

```
# su – timesten

$ mkdir -p /ttgrid/config

$ mkdir -p /ttgrid/zk

$ mkdir -p /ttgrid/inst

$ mkdir -p /ttgrid/install

$ mkdir -p /ttgrid/db

$ mkdir -p /ttgrid/db/log
```

Repeat the same configuration in each of the hosts

3.5 Download the Software and install the TimesTen Scaleout **on ttgridmgmt01 ONLY**

To get started, download TimesTen 18.1 from the Oracle Technology Network **on ttgridmgmt01 ONLY**
http://www.oracle.com/technetwork/database/database-technologies/timesten/downloads/index.html

```
$ ssh ttgridmgmt01

$ cd /ttgrid/install

$ unzip /tmp/ timesten181121.server.linux8664.zip
```

3.6 Setting passwordless SSH

The instance administrator must be able to use SSH to log on to all hosts without a password for the ttGridAdmin utility to be able to create and interact with every member of the grid.

Run the ttGridAdmin gridSshConfig command to set up passwordless SSH access for the current user. Ensure that you execute the command with the user you selected as the instance administrator.
e.g.

```
$ /ttgrid/install/tt18.1.1.2.1/bin/ttgridadmin gridSshConfig -internalAddress ttgriddata01 ttgriddata02 ttgriddata03
ttgriddata04 ttgriddata05 ttgriddata06
```

3.7 NTP Clock

Consulting to system administrator to configure the NTP clock or other ways to gurantee the OS clock synchronized between the hosts

4. Membership Server Setup – ZooKeeper

4.1 On the three selected hosts, install JDK 8

Refer to the JAVA download and installation instruction
https://www.java.com/en/download/help/linux_install.xml

After Java installed, setup the environment:

```
PATH=$PATH:$HOME/bin
export PATH
export JAVA_HOME=/home/timesten/jdk1.8.0_181
export JAVA_BIN=$JAVA_HOME/bin
export JAVA_LIB=$JAVA_HOME/lib
export CLASSPATH=.:$JAVA_LIB/tools.jar:$JAVA_LIB/dt.jar
export PATH=$JAVA_BIN:$PATH
```

Verify the version via

```
$  java -version
```

4.2 Setup Apache ZooKeeper

```
$ cd /ttgrid/zk

$ tar  -xzvf /ttgrid/install/tt18.1.1.2.1/3rdparty/zookeeper-3.4.10.tar.gz
```

Create a myid text file in the /ttgrid/zk/zookeeper-3.4.10/data directory on ttmgmt01 for its membership server. The myid text file contains the value 1.

Create a myid text file in the /swdir/zkdir/zookeeper-3.4.10/data directory on ttmgmt02 for its membership server. The myid text file contains the value 2.

Create a myid text file in the /swdir/zkdir/zookeeper-3.4.10/data directory on ttclnt01 for its membership server. The myid text file contains the value 3.

When the membership server starts up, it identifies which server it is in by the integer configured in the myid file in the ZooKeeper data directory.

```
$ ssh ttgridmgmt01

$ cd /ttgrid/zk/zookeeper-3.4.10/data

$ vi myid

1
```

```
$ ssh ttgridmgmt02

$ cd /ttgrid/zk/zookeeper-3.4.10/data

$ vi myid

2
```

```
$ ssh ttclnt01

$ cd /ttgrid/zk/zookeeper-3.4.10/data

$ vi myid

3
```

4.3 Setup Apache ZooKeeper Configuration File

```
vi /ttgrid/zk/zookeeper-3.4.10/conf/zoo.cfg
```

```
tickTime=250

initLimit=40

syncLimit=12

dataDir=/ttgrid/zk/zookeeper-3.4.10/data

clientPort=2181

autopurge.snapRetainCount=3

autopurge.purgeInterval=1

minSessionTimeout=2000

maxSessionTimeout=10000

server.1=ttgridmgmt01:2888:3888

server.2=ttgridmgmt02:2888:3888

server.3=ttgridmgmt03:2888:3888
```

Repeate the setup on ttmgmt01, ttmgmt02 and ttclnt01 hosts

4.4 Start Apache ZooKeeper

```
$ /ttgrid/zk/zookeeper-3.4.10/bin/zkServer.sh start
```

Repeate the setup on ttmgmt01, ttmgmt02 and ttclnt01 hosts

You can verify the status for each membership server by executing the `zkServer.sh status` command on each membership server:

```
$ /ttgrid/zk/zookeeper-3.4.10/bin/zkServer.sh status

ZooKeeper JMX enabled by default

Using config: /swdir/zkdir/zookeeper-3.4.10/conf/zoo.cfg

Mode: { leader | follower }
```

5. TimesTen Scaleout Configuration

Login the ttmgmt01 host, and configure the following files for rolling out the TimesTen Scaleout from one place via “ttGridRollout” tool which is under the /ttgird/install/tt18.1.1.2.1/bin directory

5.1 Create Membership Configuration File - membership.conf

```
$ ssh ttmgmt01

$ cd /ttgrid/config

$ vi membership.conf
```

```
Servers ttmgmt01!2181,ttmgmt02!2181,ttclnt01!2181
```

5.2 Create Database Definition File – sampledb.dbdef

```
$ ssh ttmgmt01

$ cd /ttgrid/config

$ vi sampledb.dbdef
```

```
DataStore=/ttgrid/db/sampledb
LogDir=/ttgrid/db/log
PermSize=4096
TempSize=1024
LogBufMB=1024
LogFileSize=1024
MemoryLock=4
DatabaseCharacterSet=AL32UTF8
ConnectionCharacterSet=AL32UTF8
Durability=0
Connections=2000
Preallocate=0
```

5.3 Create Deamon Configuration File – daemon.conf (Optional)

```
$ ssh ttmgmt01

$ cd /ttgrid/config

$ vi daemon.conf
```

```
 show_date=1
```

5.4 Create Connectable Configuration File – sampledbcs.connect (Optional)

```
$ ssh ttmgmt01

$ cd /ttgrid/config

$ vi sampledbcs.connect
```

```
ConnectionCharacterSet=AL32UTF8
TTC_TCP_KEEPALIVE_TIME_MS=200
TTC_TCP_KEEPALIVE_INTVL_MS=200
TTC_TCP_KEEPALIVE_PROBES=3
```

5.5 Create a configuration file for the ttGridRollout utility

The configuration file for the ttGridRollout utility defines all the necessary parameters to successfully create and deploy a grid and database in TimesTen Scaleout.

```
$ ssh ttmgmt01

$ cd /ttgrid/config

$ vi mygrid.conf
```

```
zoo_conf = membership.conf

grid_name = mygrid

instance_location = /ttgrid/inst

dbdef_file = sampledb.dbdef

instance_config = daemon.conf

cs_connect_files = sampledbcs.connect

mgmt_instances = [
{
        "host":"ttgridmgmt01","address":"192.168.10.98","instance":"mgmtinst","daemonport":6624,"csport":6625,"
mgmtport":3754 },
{
        "host":"ttgridmgmt02","address":"192.168.10.99","instance":"mgmtinst","daemonport":6624,"csport":6625,"
mgmtport":3754 }
]

data_instances =
[
	{
	"host":"ttgriddata01","address":"192.168.10.101","dataspacegroup":1,"instance":"datainst","daemonport":63388,"csport":20233},
	{
	"host":"ttgriddata02","address":"192.168.10.102","dataspacegroup":2,"instance":"datainst","daemonport":63388,"csport":20233},
	{
	"host":"ttgriddata03","address":"192.168.10.103","dataspacegroup":1,"instance":"datainst","daemonport":63388,"csport":20233},
	{
	"host":"ttgriddata04","address":"192.168.10.104","dataspacegroup":2,"instance":"datainst","daemonport":63388,"csport":20233},
	{
	"host":"ttgriddata05","address":"192.168.10.105","dataspacegroup":1,"instance":"datainst","daemonport":63388,"csport":20233},
	{
	"host":"ttgriddata06","address":"192.168.10.106","dataspacegroup":2,"instance":"datainst","daemonport":63388,"csport":20233}
]
```

6. Rollout the TimesTen Scaleout from ttmgmt01 host

6.1 Check the configuration files

```
$ ssh ttmgmt01

$ cd /ttgrid/config

$ ls –lrt *
```

6.2 Verify configuration file contents via ttGridRollout

Use ttGridRollout to sanity check the configuration files and environment.

```
$ /ttgrid/install/tt18.1.1.2.1/bin/ttGridRollout  –dry-run mygrid.conf
```

6.3 Deploy a 3x2 grid and create sampledb database via ttGridRollout

This is the most important step to rollout a grid and create the sampledb database from ttmgmt01 host only!

```
$ /ttgrid/install/tt18.1.1.2.1/bin/ttGridRollout -wait 600 -timeout 1200 mygrid.conf
```

```
……
…..

================================================================================
6-instance (3x2) grid successfully created.

Management Instance Locations
-----------------------------
- ttgridmgmt01:/ttgrid/inst/mgmtinst
- ttgridmgmt02:/ttgrid/inst/mgmtinst

Please source ttenv script under Management Instances for grid management via "ttGridAdmin" commands.

  For example, to use the first management instance, on ttgridmgmt01:
  sh:  . /ttgrid/inst/mgmtinst/bin/ttenv.sh
  csh: source /ttgrid/inst/mgmtinst/bin/ttenv.csh


Data Instance Locations
-----------------------
- ttgriddata01.datainst ==> ttgriddata01:/ttgrid/inst/datainst
- ttgriddata02.datainst ==> ttgriddata02:/ttgrid/inst/datainst
- ttgriddata03.datainst ==> ttgriddata03:/ttgrid/inst/datainst
- ttgriddata04.datainst ==> ttgriddata04:/ttgrid/inst/datainst
- ttgriddata05.datainst ==> ttgriddata05:/ttgrid/inst/datainst
- ttgriddata06.datainst ==> ttgriddata06:/ttgrid/inst/datainst

Please source ttenv script under Data Instances for database operations.

  For example, to use datainst, on ttgriddata01:
  sh:  . /ttgrid/inst/datainst/bin/ttenv.sh
  csh: source /ttgrid/inst/datainst/bin/ttenv.csh
```

After you completed the grid database deployment, add the environment variables properly
On ttmgmt01 and ttmgmt02 host:

```
. /ttgrid/inst/mgmtinst/bin/ttenv.sh
```

On each of the ttgriddata0<n> host

```
. /ttgrid/inst/datainst/bin/ttenv.sh
```

6.4 Verify the database connection

Login to the ttgriddata01 host

```
$ /ttgrid/inst/datainst/bin/ttenv

$ ttisql "dsn=sampledb "

create user appuser identified by appuser;

grant all to appuser;

exit;



$ ttisql “DSN=sampledbcs;uid=appuser;pwd=appuser”
```

```

CREATE TABLE appuser.account_type
(
    type            CHAR(1) NOT NULL PRIMARY KEY,
    description     VARCHAR2(100) NOT NULL
)
DUPLICATE;

CREATE TABLE appuser.account_status
(
    status          NUMBER(2,0) NOT NULL PRIMARY KEY,
    description     VARCHAR2(100) NOT NULL
)
DUPLICATE;

CREATE TABLE appuser.customers
(
    cust_id            NUMBER(10,0) NOT NULL PRIMARY KEY,
    first_name         VARCHAR2(30) NOT NULL,
    last_name          VARCHAR2(30) NOT NULL,
    addr1              VARCHAR2(64),
    addr2              VARCHAR2(64),
    zipcode            VARCHAR2(5),
    member_since       DATE NOT NULL
)
DISTRIBUTE BY HASH;

CREATE TABLE appuser.accounts
(
    account_id         NUMBER(10,0) NOT NULL PRIMARY KEY,
    phone              VARCHAR2(16) NOT NULL,
    account_type       CHAR(1) NOT NULL,
    status             NUMBER(2,0) NOT NULL,
    current_balance    NUMBER(10,2) NOT NULL,
    prev_balance       NUMBER(10,2) NOT NULL,
    date_created       DATE NOT NULL,
    cust_id            NUMBER(10,0) NOT NULL,
    CONSTRAINT fk_customer
        FOREIGN KEY (cust_id)
            REFERENCES appuser.customers(cust_id),
    CONSTRAINT fk_acct_type
        FOREIGN KEY (account_type)
            REFERENCES appuser.account_type(type),
    CONSTRAINT fk_acct_status
        FOREIGN KEY (status)
            REFERENCES appuser.account_status(status)
)
DISTRIBUTE BY REFERENCE (fk_customer);
```

7. TimesTen Scaleout DBA Management

7.1 Check current grid database status

```
$ ssh ttgridmgmt01

$ /ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin hostList

$ /ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin instanceList

$ /ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin dbStatus sampledb -all
```

7.2 Stop and Restart the database

7.2.1 Close the database to reject new connections to Sampledb database

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin dbClose sampledb -wait
```

7.2.2 Verify and disconnect any existing connections

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin instanceExec -type data ttXactAdmin sampledb > ttxactadmin.log

cat ttxactadmin.log
```

Kill the transactions returned in ttxactadmin.log on each of the data hosts before unloading

7.2.3 Unload the database

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin dbUnload sampledb –wait

Or

/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin dbUnload sampledb –force –wait
```

7.2.4 Verify the database status

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin dbStatus sampledb –all
```

7.2.5 Shutdown the grid instances

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin instanceExec -type data ttDaemonAdmin –stop
```

7.2.6 Start the grid instances

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin instanceExec -type data ttDaemonAdmin –start
```

7.2.7 Load the database to RAM

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin dbLoad sampledb -wait
```

7.2.8 Open the database for application connections

```
/ttgrid/inst/mgmtinst/bin/ttenv ttGridAdmin dbOpen sampledb –wait
```

8. Client Installation

8.1 Download and Install the TimesTen software

```
$ ssh ttclnt01

$ cd /ttgrid/install

$ unzip timesten181121.server.linux8664.zip
```

8.2 Create client instance

The target instance location will be /ttgrid/inst/clntinst

```
cd tt18.1.1.2.1/bin

./ttinstancecreate -clientonly
```

8.3 Setup the SYSODBCINI environment variable

```
. /ttgrid/inst/clntinst/bin/ttenv.sh

export SYSODBCINI=/ttgrid/inst/clntinst/conf/sys.odbc.ini
```

8.4 Adding connection string to sys.odbc.ini

The client connection information are all located under the sys.odbc.ini file under each of data instances.

e.g.
ssh ttgriddata01
cd /ttgrid/inst/datainst/conf
cat sys.odbc.ini

Then, copy the SAMPLEDBCS client information to the ttgridclnt01 host's sys.odbc.ini file

ssh ttgridclnt01
vi /ttgrid/inst/clntinst/conf/sys.odbc.ini

```
[sampledbcs]

TTC_SERVER_DSN=SAMPLEDB

TTC_SERVER1=192.168.10.101/20233

TTC_SERVER2=192.168.10.102/20233

 TTC_SERVER3=192.168.10.103/20233

TTC_SERVER4=192.168.10.104/20233
```

```
TTC_SERVER5=192.168.10.105/20233

TTC_SERVER6=192.168.10.106/20233

ConnectionCharacterSet=AL32UTF8

TTC_TCP_KEEPALIVE_TIME_MS=200

TTC_TCP_KEEPALIVE_INTVL_MS=200

TTC_TCP_KEEPALIVE_PROBES=3
```

8.5 Verify client connection to TimesTen Scaleout

```
$ ttisqlcs "dsn=sampledbcs;uid=appuser;pwd=appuser"

Command> tables;
```

Implementation PoC complete

Congratulations, you have completed the TimesTen Scaleout implementation PoC!