ORACLE

TIMESTEN
IN-MEMORY DATABASE

Deploying Oracle TimesTen In-Memory Database on the Oracle Bare Metal Cloud Platform

Quick Start White Paper | April 2017 | Version 1.0

ORACLE WHITE PAPER | APRIL 2017

ORACLE

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Questions or comments on this paper? Please email: timesten-info_ww@oracle.com

Table of Contents

"Oracle named a leader in In-Memory databases in the latest The Forrester WaveTM: In-Memory Databases, Q1 2017."

SOURCE: THE FORRESTER WAVETM: In-Memory Databases, Q1 2017,
Noel Yuhanna with Gene Leganza, Shreyas Warrier and Emily Miller
February 28, 2016

Oracle TimesTen In-Memory Database on Oracle Bare Metal Cloud overview

Many software engineering organizations are faced with the challenge of building systems to handle extremely high throughput (10s of thousands writes/sec) while maintaining extremely low latency (< 10 microseconds). The Oracle TimesTen In-Memory Database running on the Oracle Bare Metal Cloud Service (BMCS) easily handles these types of workloads in a secure and highly-available environment.

The Oracle TimesTen In-Memory Database is a best-in-class in-memory relational database that provides:

» Extremely low response times, in the low microseconds, for complex SQL operations
» Highly-configurable ACID transaction and durability model
» Full relational database capabilities

The Oracle BMCS offers hourly metered bare metal instances. By eliminating the hypervisor, Oracle can deliver better performance at a lower cost than traditional IaaS providers. In addition to compute unencumbered by a hypervisor, Oracle BMCS offers instances with up to 28TBs of locally attached NVMe storage. Each 28TB instance (9 NVMe storage units) is capable of over 3 million, 4KB IOPs. Oracle BMCS also offers VM instances for greater flexibility in choice of shape.

Instances in the Oracle BMCS are attached to a 10Gb non-blocking network with no oversubscription. Each node has access to the full performance of the hardware, there are no "noisy neighbors" or hypervisors to share resources with. Instances in the same region are always less than 1ms from each other, while instances in the same availability domain are less than 100 microseconds from each other.

This quick start white paper is designed as a reference guide for deploying the Oracle TimesTen In-Memory Database on the Oracle Bare Metal Cloud Service platform. The following sections describe the preliminary setup of the BMCS environment and then how to run scripts that install and configure the Oracle TimesTen In-Memory Database.

Assumptions

Consumers of this document should:

» Be familiar with the fundamentals of the Oracle BMCS:
  » https://docs.us-phoenix-1.oraclecloud.com
  » The BMCS walkthrough is highly recommended if this is the first time you are using the platform:
  » https://docs.us-phoenix-1.oraclecloud.com/Content/GSG/Reference/overviewworkflow.htm
  » Have a basic understanding of Oracle TimesTen In-Memory Database.
  The Oracle TimesTen In-Memory Database Introduction Guide is available at:

http://docs.oracle.com/cd/E21901_01/timesten.1122/e21631/toc.htm

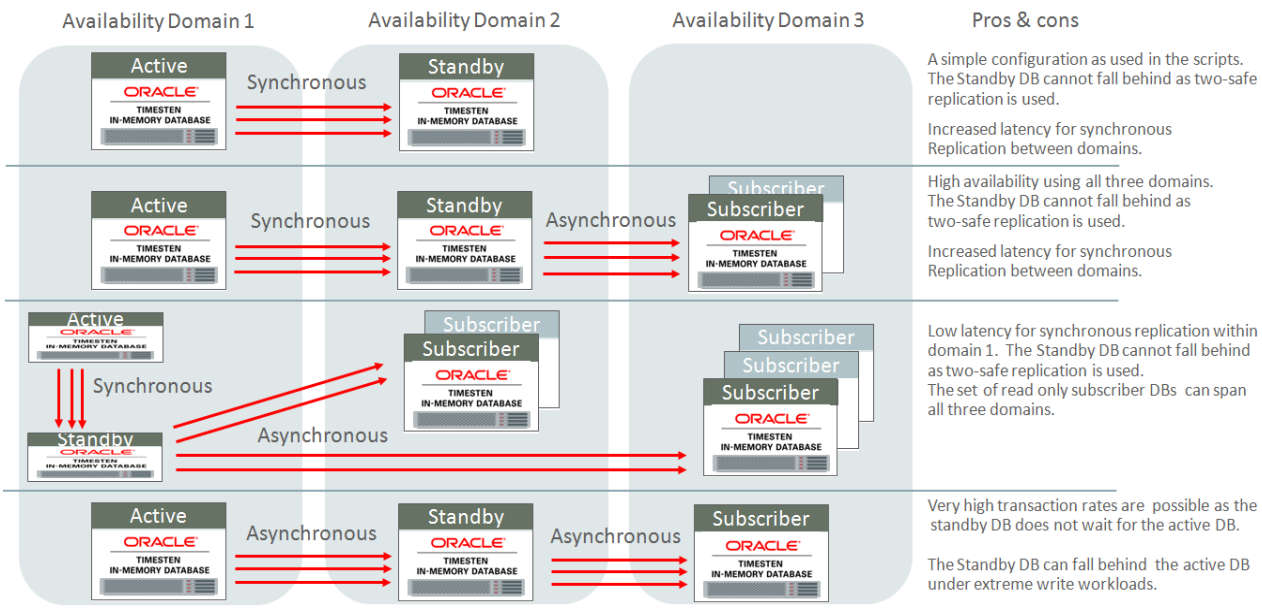
Planning an Oracle TimesTen In-Memory Database deployment on BMCS

Oracle TimesTen In-Memory Database Replication Architecture

Oracle TimesTen In-Memory Database provides substantial flexibility in choices of replication and durability options. Databases are typically deployed in a two node Active Standby Pair configuration. A single instance configuration can be used for development and testing. Applications that require the most stringent durability should use synchronous replication, while applications that require higher transaction throughput rates should use asynchronous replication. Asynchronous replication provides higher throughput while synchronous replication provides higher availability.

For the highest availability, the active and standby nodes are placed in separate availability domains. A read-only subscriber is run in a 3rd availability domain and can be made the active node as part of a disaster recovery solution. A configuration that yields better transaction response time, but still uses synchronous replication, would place the active and standby in the same availability domain, while having an asynchronously updated read-only subscriber in a second availability domain.

The following table summarizes the trade-offs in availability versus performance. The scripts described below set up the configuration in the first row of the table.



Choosing your shape

There are 3 instance shapes for Oracle BMCS –

Oracle BMCS Instance Shapes

BM.StandardIO1.36	BM.HighIO1.36	BM.DenseIO1.36
32GB local disk storage 36 cores 256GB memory	12.8TB local NVMe storage 36 cores 512GB memory	28.8TB local NVMe storage 36 cores 512GB memory

In addition, there are 5 VM instance shapes.

If your database is larger than 8GB you'll need a Block Volume

Shape	OCPU	Memory (GB)	Local Disk (TB)
VM.Standard1.1	1	7	32GB internal or Block Volume only
VM.Standard1.2	2	14	32GB internal or Block Volume only
VM.Standard1.4	4	28	32GB internal or Block Volume Only
VM.Standard1.8	8	56	32GB internal or Block Volume Only
VM.Standard1.16	16	112	32GB internal or Block Volume Only

Standard instances are suitable for more read intensive applications. HighIO and DenseIO shapes have local direct NVMe attached storage and are recommended for write oriented and high-performance workloads.

This quick start white paper shows the steps for allocating an Active Standby Pair configuration. The database server nodes will be set up in different Availability Domains (ADs) for enhanced reliability.

Outline of steps to get Oracle TimesTen In-Memory Database up and running on BMCS

The following steps show the steps to get Oracle TimesTen In-Memory Database up and running on Oracle Bare Metal Cloud Service.

1. Set up the BMCS network and provision BMCS instances
2. Acquire Oracle TimesTen In-Memory Database license
3. Copy the Oracle TimesTen In-Memory Database install scripts, Oracle JDK, and Oracle TimesTen In-Memory Database software to your local machine
4. Run the Oracle TimesTen In-Memory Database install script

At this point you can use the database(s) or you can continue to configure an Active Standby Pair.

5. Run the scripts to create an Active Standby Pair as described in the Oracle TimesTen In-Memory QuickStart Guide
6. Verify that an application can successfully connect with the Oracle TimesTen In-Memory Database.

The details for these steps are described in the following sections.

Create the required BMCS Network Resources

After setting up your account in the Oracle Bare Metal Cloud Service at https://console.us-az-phoenix-1.oracleiaas.com/, navigate to the BMCS UI to set the Internet Gateway, Route Table, and Subnet configuration as described below. These settings should be equivalent to the default settings when you create a new Virtual Cloud Network. See also the BMCS tutorial, Creating a Virtual Cloud Network in the BMCS Getting Started Guide.

1. Under Networking, select, Create a new Virtual Cloud Network
    a. Name – Oracle_TimesTen_VCN
    b. Select: Create VCN plus related resources
    c. Scroll to the bottom and select, Create Virtual Cloud Network.

2. Open Oracle TimesTen In-Memory ports for **ingress** by creating security rules
    a. Go to Networking, then navigate to Virtual Cloud Networks for your compartment
    b. Click on your Virtual Cloud Network (Oracle_TimesTen_VCN)
    c. Click Security Lists then click Default Security List for Oracle_TimesTen_VCN

d. Click Edit All Rules and add an **ingress** list rule with:

   - Source CIDR: 10.0.0.0/16
   - IP PROTOCOL: TCP
   - SOURCE PORT RANGE: All
   - DESTINATION PORT RANGE: 46337-46339

e. Scroll to the bottom and click, Save Security List Rules

Create the required BMCS Compute Instances

The examples below use a HighIO shape consisting of 36 cores, 512GB RAM and 12.8 TB of NVMe locally attached storage. Choose the shape with the memory that best accommodates your needs. The scripts provided require Oracle Linux 7. It is necessary to use a public subnet as the scripts use yum to install Linux packages needed to configure storage or run TimesTen.

1. Create BMCS compute instance 1 to run Oracle TimesTen In-Memory DB active node
    a. Name – Oracle_TimesTen_DB_AD1_0
    b. Image – Oracle-Linux-7.3-2017.04.18-0
    c. Shape – BM.HighIO1.36
    d. Availability Domain – PHX-AD-1
    e. Cloud Network – Oracle_TimesTen_VCN
    f. Public Subnet – NIbo:PHX-AD1
    g. SSH Key – <public half of your key pair; e.g ~/.ssh/id_rsa.pub on the system where you will run the scripts>
2. Create BMCS compute instance 2 to run Oracle TimesTen In-Memory DB standby node (Optional)
    a. Name – Oracle_TimesTen_DB_AD2_0
    b. Image – Oracle-Linux-7.3-2017.04.18-0
    c. Shape – BM.HighIO1.36
    d. Availability Domain – PHX-AD-2
    e. Cloud Network – Oracle_TimesTen_VCN
    f. Public Subnet – NIbo:PHX-AD2
    g. SSH Key – <public half of your key pair>
3. Create BM compute instance 3 to run Oracle TimesTen In-Memory DB read-only subscriber (Optional)
    a. Name – Oracle_TimesTen_DB_AD3_0
    b. Image – Oracle-Linux-7.3-2017.04-18-0
    c. Shape – BM.HighO1.36
    d. Availability Domain – PHX-AD-3
    e. Cloud Network – Oracle_TimesTen_VCN
    f. Public Subnet – NIbo:PHX-AD3
    g. SSH Key – <public half of your key pair>

Create any BMCS Block Volume devices desired (Optional)

Block Volumes are good choices for persistent storage when using a standard image or a VM. Though they enable substantially less throughput rates than NVMe devices, they provide durability, availability and better performance than the small amount of ephemeral storage that comes with a standard image. Block Volumes are local to an Availability Domain. You will want to create a Block Volume device for each of your nodes. Block Volumes are available in 256GB or 2TB capacities.

Allow for 4-5 times the expected PermSize of your TimesTen database [2x-checkpoint files; 1x-transaction logs; 1.5x-ttBackup or ttMigrate files]. Allocating Block Volumes is not necessary if you use a HighIO, DenseIO, or other shape that has NVMe storage. The example below uses 256GB, however if you have a VM with 112GB you need more storage. In case you need more storage, you can either:

- Add a second 256GB Block Volume where you store checkpoint files on one volume and store log files and backup on the other volume.
- Use a single volume with a 2TB capacity.

Using the Bare Metal Console, navigate to the Storage Service, then to the Block Volumes drop-down list. Select your compartment from the list in the left frame. Then, select Create Block Volume and specify the following parameters:

a. Name – Oracle_TimesTen_BV_AD1

b. Availability Domain - PHX-AD-1

c. Size – 256.0GB

Repeat these steps for allocating Block Volumes for each Availability Domain.

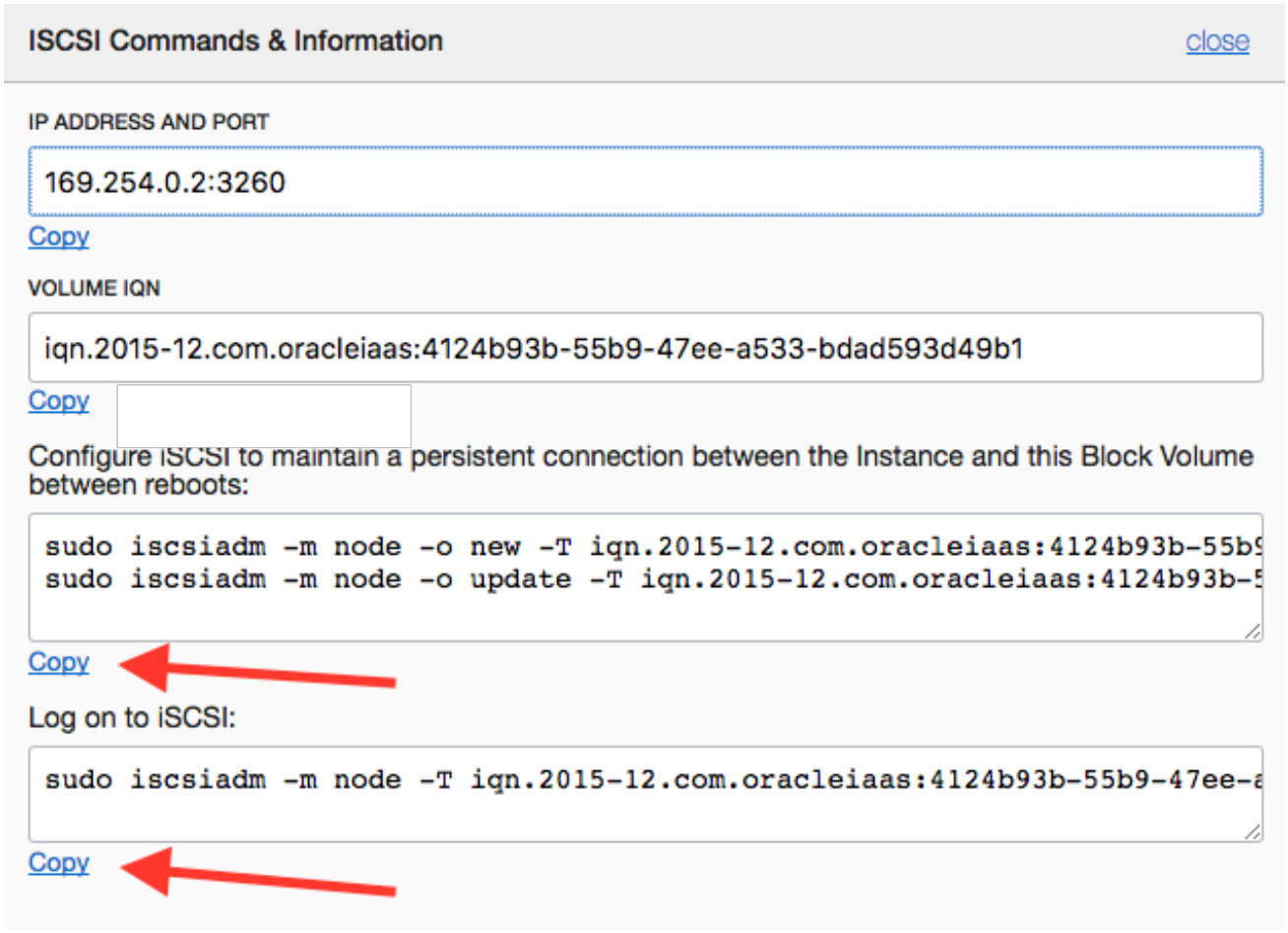
Additional steps required for configuring Block Volume Storage

Block Volumes require additional configuration to make an iSCSI attachment from the compute node to the Block Volume Storage device. The details are described in the BMCS Documentation under Connecting to a Volume.

1. In the BMCS Console, navigate to find the **Compute**, and then select **Instances**.
2. Select **the row containing your instance** and click on the instance name. For example, Oracle_TimesTen_DB_AD1_0.
3. Select **Attach Block Volume.**
   a. From the drop-down list, select the name of the Block Volume. For example, Oracle_TimesTen_BV_AD1.
4. From the Attached Block Volumes table, select the name of the Block Volume. Select **SCSI Commands and Information.**
   a. Copy and paste the commands to a file**. For more information, see the graphic below.**
5. Use the following commands to copy the file to your instance and execute the script:
   a. scp <SCSI-commands> opc@<external-ip-addr>:
   b. ssh –t opc@<external-ip-addr> sudo sh /home/opc/<SCSI-commands>

The `fdisk –l` command should show you a device attachment. For example:

```
Disk /dev/sdb: 274.9 GB, 274877906944 bytes, 536870912 sectors
```



NOTE: If you have the BMCS Command Line Interface (CLI) installed, a python3 script is provided that uses the CLI to generate the attached commands for you to execute. The script and its usage are described in the appendix.

Create a summary table with the IP addresses of your compute instances

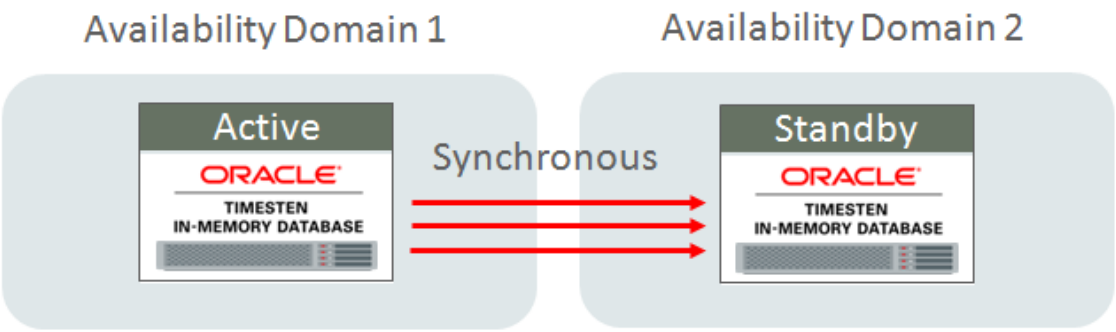
Instance	Public IP	Private IP
Oracle_TimesTen_DB_AD1_0		
Oracle_TimesTen_DB_AD2_0		
Oracle_TimesTen_DB_AD3_0		

Examine the compute instance entries with the BMCS console to complete the table above. Alternatively, a python3 script that uses the BMCS Command Line API is provided and described in the appendix.

NOTE: Oracle TimesTen In-Memory Database nodes and associated applications communicate across the private address of the instances, not the public IPs. Communication over the public IP adds latency to the connection and limits the bandwidth. The RFC1918 private IP guarantees access to the full network bandwidth and the lowest possible latency. The public IPs can be used for external access, such as connecting through ssh to the machines or as parameters to the Oracle TimesTen In-Memory Database install scripts.

Installing the Oracle TimesTen In-Memory database on BMCS

The following sections describe the steps needed to install the Oracle TimesTen In-Memory database on BMCS. You can download scripts to set up single instances on one or multiple hosts or an active standby pair configuration, as shown in the graphic below.



Download Oracle TimesTen In-Memory database software

Download the software and scripts to your local machine. The scripts copy the software to the BMCS systems and perform a remote installation.

Once you have your software license, you can download Oracle TimesTen In-Memory Database from the Oracle Technology Network (OTN) downloads page:

http://www.oracle.com/technetwork/database/database-technologies/TimesTen/downloads/index.html

The tar.gz file is uploaded to the BMCS compute nodes during the install process.

Note that for customers with a license, patch releases more recent than what's available on OTN are available on ARU. As of this writing the current version is 11.2.2.8.21. When downloading from ARU, the file extension is zip instead of tar.gz

Optionally, customers with JDBC applications for example, may want to download the latest **64-bit** JDK tar.gz file for Linux x64 Platforms (Note: Oracle TimesTen In-Memory Database requires Java version 1.7 or greater). For example, go to, http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Download the 64-bit Linux (linux-x64) tar.gz file to your local machine.

Download the TimesTen scripts from:

https://app.compendium.com/api/post_attachments/923f8f3c-9bd3-47e2-b361-ea9d86722be7/view

Place the TimesTen tar.gz or zip file, JDK tar.gz file if you want it, and the install scripts in the same directory so that the install scripts can copy these files to the BMCS compute nodes.

Scripts to install and configure the Oracle TimesTen In-Memory Database

The provided scripts enable you to install and configure the Oracle TimesTen In-Memory Database from your local machine.

These scripts automatically:

1. Partition and mount the NVMe or Block Volume storage and create file systems on the BMCS nodes.
2. Configure the BMCS network and firewall ports.
3. Create a group and user named oracle. For enhanced security, the user is a system user without a home directory or a password.
4. Install operating system utilities and services needed for the Oracle TimesTen In-Memory database:
   - Oracle JDK (if provided)
   - NTP service
   - libaio, lvm2, yum-plugin-security, nc, dstat
5. Install and start up the Oracle TimesTen In-Memory Database.

Another script creates a database with a TimesTen Active Standby Pair configuration.

The Oracle TimesTen In-Memory Database install script requires an input list of IP addresses. The scripts are copied and run on each node provided in the IP list. Ensure that the user that runs the script is able to run ssh opc@<ipaddr> and connect to the node without using any -i keypath option. For Linux systems, this means that the key used for creating the instance is ~/.ssh/id_rsa.pub.To run the TimesTen install script. It's a good idea to test the connection first, as it takes time after provisioning completes before you can actually connect to the node.

```
ott-bmc-byol/scripts/ott_bmc_install <ipaddr> [<ipaddr> …]
```

The script requires a list of the public IP addresses obtained from the BMCS Compute Instance Details page. For more information, see Create a summary table with the IP address of your compute instances

ott_bmc_install.sh parameters:

parameter	description
--instance I --port P	Instance name (default is tt1122). Port for instance name (default is 46337)
--uninstall [--destructive]	Uninstall the TimesTen instance (default is last instance specified or tt1122) The copied scripts are also deleted. Any storage configured is left intact. To cleanup any storage, specify –destructive Use --destructive with caution as it removes the created disk partitions.

The ott_bmc_install script copies the local Oracle JDK tar.gz and Oracle TimesTen In-Memory Database .tar.gz to each node. The script copies the ott_bmc_byol directory and runs the node install scripts sequentially. Once the installations are complete the user can:

1. Install applications and create databases.

2. Run the provided ott-bmc-byol/scripts/timesten/mkasp script to create an active standby pair configuration, as described in the TimesTen Quickstart Guide.
3. Run other demos from the TimesTen quick start.

For example, the TimesTen installation directory on your local machine should include the following files:

- `jdk-8u121-linux-x64.tar.gz`
- `ott-bmc-byol`
- `timesten1122821.linux8664.tar.gz # or timesten1122821.linux8664.zip`

The install scripts automatically pick up the most recent TimesTen and JDK files based on their prefix and suffix patterns (ls –rt determines latest).

TimesTen is installed in the `/opt/TimesTen/<instance-name>` directory. The default directory is `/opt/TimesTen/tt1122` The `/opt` directory is selected to ensure that the installation is on a separate disk from the database and transaction log files. The database and transaction log files are placed in /u10 along with the TimesTen daemon info directory where TimesTen operational message and error logs can be found.

The `–instance <name> –port <number>` option changes the instance name and port number. The ports are described in `Script Details – Network Configuration`

To uninstall the TimesTen instance and the copied script files:

```
ott-bmc-byol/scripts/ott_bmc_install –uninstall <IPaddr> [<IPaddr …>]
```

If you had NVMe or Block Volume storage and want to return it to its initial state, use the –destructive option.

The following sections describe some of the workings of the ott_bmc_install.sh script.

Script Details – File System Setup

The BMCS is delivered as a raw bare metal machine with Oracle Linux installed. It is necessary to do some initial hardware configuration, which includes formatting and mounting of storage drives. If the node install script fails to find NVMe devices, it looks for Block Volumes. If the script does not find Block Volumes, it uses local storage. NOTE: The BMCS does not have much local storage. Use of local storage for databases is not recommended as you may run out of space on the root file system. In particular, use of databases larger than 8GB have a strong probability of resulting in a full root file system.

If NVMe or Block Storage devices are available, the script partitions the disks, creates a Volume Group (vgu10), and stripes a single logical volume across the available devices. The script mounts a `/u10` file system on the logical volume for storing database files and transaction logs. The devices are not partitioned if a `/u10` file system exists or can be mounted. The following example uses a BM.HighIO.1.36 shape:

```
% df -hm
Filesystem              1M-blocks  Used Available Use% Mounted on
devtmpfs                   257775     0    257775   0% /dev
tmpfs                      257792     0    257792   0% /dev/shm
tmpfs                      257792    66    257727   1% /run
tmpfs                      257792     0    257792   0% /sys/fs/cgroup
/dev/sda3                   38921  3733     35189  10% /
/dev/sda1                     512    10       503   2% /boot/efi
tmpfs                       51559     0     51559   0% /run/user/1000
/dev/mapper/vgu10-lvu10   12017637    53  11407098   1% /u10
```

If your node does not have NVMe or Block Volume storage devices, the df command will not display /u10.

Script Details – Network Configuration

The Oracle TimesTen In-Memory database needs open ports for client/server, server/server, and replication communication. The table below shows the default port usage. Any port you used must be part of a security list ingress rule as created in the BMCS Console above.

Component	Default Port	With script options --instance I –port P
TimesTen Main Daemon (timestend)	46337	P
TimesTen Server (ttcserver)	46338	P+1
Replication (ttimestenrepd)	46339	P+2

A set of generally open ports for P above can be chosen from the range 46337-46997.

The node install script opens up ports on the Linux 7 images using firewalld:

```
$ firewall-cmd --zone=public --add-rich-rule="rule family=ipv4 source address=\"10.0.0.0/16\" port
port=\"46337\" protocol=\"tcp\" accept"
```

Script Details – NTP Installation

NTP is installed to keep the clocks synchronized on the TimesTen database nodes. This is required for TimesTen replication.

The scripts install NTP and add ntpd to the boot configuration:

```
$ sudo yum -y install ntp
$ sudo systemctl start ntpd
$ sudo systemctl status ntpd
```

The scripts perform additional steps to use ntp servers internal to BMCS.

Script Details – Running TimesTen as a Service: systemd

One of the major reasons for choosing Oracle Linux 7 images is to take advantage of systemd. systemd starts the TimesTen daemons on system startup, shuts down the daemons on system shutdown, and restarts the daemons if they go down providing increased availability over /etc/init.d

At installation time, the TimesTen daemons are started. The scripts shut down the daemons and install a service file under /usr/lib/systemd/system/ott-<instance-name>.service. The TimesTen daemons are then started under systemd.

```
sudo systemctl enable ott-tt1122.service
sudo systemctl start  ott-tt1122.service
sudo systemctl status ott-tt1122.service
```

Use systemd to manage the TimesTen daemons instead of using the ttDaemonAdmin utility.
Use systemctl restart ott-<instance-name>.service to start or restart daemons instead of using the ttDaemonAdmin utility.
Use systemctl stop ott-<instance-name>.service to stop daemons instead of using the ttDaemonAdmin utility.

The service restart policy is set to 'on-failure' meaning the daemons will be restarted in most software failure cases, but will NOT be restarted if they exit with status 0 nor if they are killed with signals, **SIGHUP, SIGINT, SIGTERM** or **SIGPIPE.**

To prevent TimesTen from running as a service, execute the following and use the `ttDaemonAdmin` utility.

```
sudo systemctl stop ${servicename}
sudo systemctl disable ${servicename}
sudo rm -f /usr/lib/system/system/${servicename}
sudo systemctl daemon-reload
sudo systemctl reset-failed
```

Verifying the Oracle TimesTen In-Memory Database Installation

Once the installation is complete, you can log on to any of the nodes and verify the installation was successful by executing the following commands:

```
 % . /opt/TimesTen/ott-t1122-env.sh  # ott-<instance-name>-env.sh
LD_LIBRARY_PATH set to
/opt/TimesTen/tt1122/lib:/opt/TimesTen/tt1122/ttoracle_home/instantclient_11_2
ANT_HOME set to /u10/TimesTen/tt1122/3rdparty/ant
  …
%ttstatus
TimesTen status report as of Thu Apr  6 23:58:36 2017

Daemon pid 49032 port 46337 instance tt1122
TimesTen server pid 49041 started on port 46338
------------------------------------------------------------------------
Accessible by group oracle
End of report
```

Create a database as the `oracle` user:

```
% sudo –n –u oracle bash
bash-4.2$ . /opt/TimesTen/ott-tt1122-env.sh # ott-<instance-name>-env.sh

LD_LIBRARY_PATH set to ….

% ttIsql dsn=ttimdb

Copyright (c) 1996, 2017, Oracle and/or its affiliates. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.

connect "dsn=ttimdb";
Connection successful:
DSN=ttimdb;UID=oracle;DataStore=/u10/TimesTen/tt1122/datastores/ttimdb;DatabaseCharacterSet=AL32UTF
8;ConnectionCharacterSet=AL32UTF8;LogFileSize=1024;PermSize=3072;TempSize=1024;CkptRate=0;CkptFrequ
ency=0;CkptLogVolume=1024;TypeMode=0;LogBufMB=1024;LogBufParallelism=4;
(Default setting AutoCommit=1)

Command> CREATE TABLE t (i TT_INTEGER NOT NULL PRIMARY KEY);
Command> INSERT INTO t SELECT tblid FROM TABLES;
81 rows inserted.
Command> SELECT FIRST 2 * FROM t;
< 1796752 >
< 1796768 >
2 rows found.
Command> DROP TABLE t;
Command> quit;
Disconnecting...
Done.
```

If you plan to run a non-replicated configuration, set the connection attribute DurableCommits=1 to prevent transaction loss. In this type of configuration, also set the “RAM policy” so that database remains in memory. Otherwise the database will unload from memory after the last application disconnects and will need to be reloaded when the next application connects.

```
% ttDaemonAdmin –rampolicy manual –ramload dsn=ttimdb
```

If you want this system to create an active standby pair configuration, first destroy your database:

```
% ttDaemonAdmin –ramunload dsn=ttimdb # if RAM-loaded
% ttDestroy dsn=ttimdb
% ttStatus
TimesTen status report as of Fri Apr  7 00:06:32 2017

Daemon pid 49032 port 46337 instance tt1122
TimesTen server pid 49041 started on port 46338
------------------------------------------------------------------------
Accessible by group oracle
End of report
```

Installing an Oracle TimesTen In-Memory Database active standby pair

The procedure for setting up an active standby pair is documented in the TImesTen Quick Start Guide.

1.	Create a database	In order to set up an active standby pair configuration, you need to have a running database. This database should include one or more tables owned by an application user. All tables must have a primary key index. If you have an existing database, this step can be skipped.
2.	Create a user to administer the active standby pair	This database must include a user with *ADMIN* privilege. This user is the replication administrator responsible for managing the active standby configuration.
3.	Define an active standby pair replication scheme	Use the *CREATE ACTIVE STANDBY PAIR* SQL statement to define the configuration of the active standby pair including the hostnames and the data store names of both the active and standby databases.
4.	Set the state of the database to active	Use the *ttRepStateSet* built-in procedure to designate the current database as the active database.
5.	Start the replication agent	The replication agent process is responsible for replicating data between TimesTen databases. Use the *ttRepStart* built-in procedure to start the replication agent.
6.	Duplicate the active database to a standby server	. Use the *ttRepAdmin* utility to duplicate the active database to the standby server.
7.	Start the replication agent on the standby database	The replication agent process on the standby server is responsible for receiving data from the active database. Use the *ttRepStart* built-in procedure to start the replication agent. Once the agent is up, the database automatically enters STANDBY mode. While the database is in STANDBY mode, changes from the active database are propagated to the standby database.
8.	Verify the replication between active and standby	Perform updates on the active database and ensure that the changes are replicated to the standby database.

Scripts are provided that implement these procedures from a remote host. To run the script, provide two IP addresses setup as described above, a password for a TimesTen administrative user (--admpwd pwd), and a password for a TimesTen application user (--apppwd pwd). Run the script from the same host where ott_bmc_install was previous invoked:

```
cd ott-bmc-byol/scripts/timesten
./mkasp --active <IPaddr> --standby <IPaddr> --admpwd <adpwd> --apppwd <apppwd>
```

The table shows the scripts that the `mkasp` uses to complete the steps mentioned in the table above.

Steps	Script
1-5	createactivedb
6-7	createstandbydb
8	verifyactivedb verifystanddb

When the `mkasp` script finishes, a verification transaction is conducted. Verify that the row selection matches the data that

was inserted into the database. For example:

```
ttIsql dsn=ttimdb
…
Command> call ttRepStateGet;
< ACTIVE, NO GRID >
1 row found.

Command>INSERT INTO orders VALUES (6853180,1121,'9999999999', sysdate);
1 row inserted.
```

On the standby node:

```
ttIsql dsn=ttimdb
…
Command> SELECT * FROM appuser.orders WHERE order_number=6853180;
< 6853180, 1121, 9999999999, 2017-04-07 19:09:36 >
1 row found.
```

Adding more TimesTen databases to Existing BMCS Compute Instances

By default, the Oracle TimesTen In-Memory Database install scripts create a single database on a set of BMCS compute instances. In some cases, this might not be the best use of these compute resources. For example, during an initial development phase you may want separate data stores for development, test and production that run on the same BMCS hardware. Specifying distinct instances and ports can help enable these custom configurations on the same instance. You can modify the install scripts to handle specific use cases.

For increased security, database instances can be created on a private subnet. The scripts cannot reach the public yum server from a private subnet (there's no internet access). To workaround this, create a yum proxy server on a host on a public network, then set the environment variable FORCEINSTALL to a non-zero value to prevent the scripts from an error exit on a private subnet. An alternative is to use the compute service to create a custom image from a working TimesTen instance. Then create a compute instance using this custom image on a private subnet.

Further Information about the Oracle TimesTen In-Memory Database

The TimesTen Quick Start Guide provides a number of example scripts and utilities to help you learn how to use TimesTen. To learn more about TimesTen, consider setting up a third system as a read-only subscriber..

For more information about the Oracle TimesTen In-Memory Database, see:

- http://www.oracle.com/technetwork/database/database-technologies/TimesTen/overview/index.html
- http://www.oracle.com/technetwork/database/database-technologies/TimesTen/documentation/index.html

Appendix: Using the BMCS Command Line Interface (CLI)

The Bare Metal Cloud System CLI provides a convenient mechanism for issuing requests directly to the BMCS REST API. To install the SDK, you need to install python (use python3 for the ott scripts) and set up the toolkit. For more information, see https://docs.us-phoenix-1.oraclecloud.com/Content/API/SDKDocs/cli.htm.

Once you have installed the BMCS command line API, the ott-byol toolkit provides python3 wrappers to help list instances and attach block volumes.

- getipmap List public and internal IP addresses for a given compartment
- getiscsicmds [--volume OCID] List iSCSI commands needed for attaching and detaching a block volume

Before you use the commands, ensure that you have set up your ~/.oraclebmc/config file properly to use the command line API utility, `bmcs`.

To use the wrapper scripts, add a [tenancy] profile that contains a `compartment` property. For example:

```
 [DEFAULT]
tenancy=ocid1.tenancy.oc1..aaaaaaaapwgyzhcjkv2oe4mg4p7hgao56zdd27f6s35jbmm3lkbxc2req5tq
region=us-phoenix-1
user=ocid1.user.oc1..aaaaaaaaatvcbe5durq5qbv64xglrlam26jb7r2y3mecmzvuer5cuez7t3ra
fingerprint=da:b4:3a:c0:87:cd:98:83:ca:b8:0f:84:b7:3c:a4:2a
key_file=/home/user/.ssh/id_rsa

[tenancy]
compartment=ocid1.compartment.oc1..aaaa…  # compartment id (ocid)
```

To list compute instances:

```
% python3 ott-bmc-byol/scripts/cli/getipmap
ttimdbvm22 NIbo:PHX-AD-2 129.146.29.86   10.0.2.70       RUNNING
ttimdb22   NIbo:PHX-AD-2 129.146.17.252  10.0.2.20       RUNNING
ttimdbvm33 NIbo:PHX-AD-3 129.146.37.241  10.0.3.33       RUNNING
ttimdb32   NIbo:PHX-AD-3 129.146.14.144  10.0.3.23       RUNNING
```

To avoid the copying and pasting of the iSCSI commands required for attaching a Block Volume:

```
$ python3 ott-bmc-byol/scripts/cli/getiscsicmds --volume OCID 2>&1 | tee attach
# attach: blockv-imdbcs-33; NIbo:PHX-AD-3; ATTACHED; AVAILABLE; 262144 mb
iscsiadm -m node -T iqn.2015-12.com.oracleiaas:58bd17ba-90b8-4b26-8a33-abf4bd3e22bc  -p
169.254.0.2:3260 -o new
iscsiadm -m node -T iqn.2015-12.com.oracleiaas:58bd17ba-90b8-4b26-8a33-abf4bd3e22bc -o update -n
node.startup -v automatic
iscsiadm -m node -T iqn.2015-12.com.oracleiaas:58bd17ba-90b8-4b26-8a33-abf4bd3e22bc  -p
169.254.0.2:3260 -o update -n node.session.auth.authmethod -v CHAP
iscsiadm -m node -T iqn.2015-12.com.oracleiaas:58bd17ba-90b8-4b26-8a33-abf4bd3e22bc  -p
169.254.0.2:3260 -o update -n node.session.auth.username -v
ocid1.volume.oc1.phx.abyhqljshpwfxx42xnv77aswd5eamv2tnjuhs2ey7ggnr6bekejfcrj7ridq
iscsiadm -m node -T iqn.2015-12.com.oracleiaas:58bd17ba-90b8-4b26-8a33-abf4bd3e22bc  -p
169.254.0.2:3260 -o update -n node.session.auth.password -v None
iscsiadm -m node -T iqn.2015-12.com.oracleiaas:58bd17ba-90b8-4b26-8a33-abf4bd3e22bc  -p
169.254.0.2:3260 -l


# detach: blockv-imdbcs-33; NIbo:PHX-AD-3; ATTACHED; AVAILABLE; 262144 mb
# iscsiadm -m node -T iqn.2015-12.com.oracleiaas:58bd17ba-90b8-4b26-8a33-abf4bd3e22bc  -p
169.254.0.2:3260 -u
fdisk –l
```

For example, if your block bolume is attached to129.146.37.241:

```
% scp attach opc@129.146.37.241:
% ssh -t opc@129.146.37.241  sudo sh /home/opc/attach
% ssh -t opc@129.146.37.241  sudo fdisk -l
```

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US


blogs.oracle.com/oracle


facebook.com/oracle


twitter.com/oracle


oracle.com

Integrated Cloud Applications & Platform Services

Deploying TimesTen on the Bare Metal Cloud
April 2017
Author: steve.folkman@oracle.com based on a template by rick.george@oracle.com
Contributing Authors: douglas.hood@oracle.com


Oracle is committed to developing practices and products that help protect the environment

Integrated Cloud Applications & Platform Services