

TimesTen Demo VM (TTDemoVM) User Guide

Contents

Overview.....	2
Pre-requisites.....	3
Supported VM Network Configurations.....	3
Installation and Configuration.....	4
Importing the VM.....	4
Configuring the VM.....	5
VM window size and display resolution.....	5
Starting the VM for the first time.....	5
Using the TimesTen Demo VM.....	6
Getting Started.....	6
TimesTen Classic Configurations.....	7
TimesTen Cache Configurations.....	7
TimesTen Scaleout Configurations.....	8
TimesTen Sample Programs.....	9
Accessing the VM and the TimesTen hosts.....	9
Hands on Labs.....	9
Summary.....	10
Appendix A: Advanced VM Configuration.....	11
Increasing the RAM and CPU cores allocated to the VM.....	11
Advanced Network Configurations.....	12
The <i>Enhanced</i> network configuration.....	12
The <i>Full</i> network configuration.....	14
Appendix B: Accessing the VM from the Host.....	15
Appendix C: The 'vmconfig' Utility.....	16
Command syntax.....	16
Example 1: Configure the VM for two TimesTen cache databases with replication.....	18
Example 2: Configure the VM for four TimesTen classic databases with replication.....	18
Example 3: Configure the VM for a 7 node TimesTen Scaleout grid.....	18
Example 4: View information about the currently configured scenario.....	18
Example 5: Stop the current scenario.....	19
Example 6: Clear the current scenario.....	19
Appendix D: VM Filesystem Layout and Contents.....	20
Appendix E: Additional Tools and Utilities.....	21
Utilities for Scaleout scenarios.....	21
runSample.....	21
sampleStats.....	21
Appendix F: Further Information, Help and Support.....	22

Overview

The **TTDemoVM** VirtualBox VM was created for the purpose of demonstrating the Oracle TimesTen In-Memory Database. The VM includes everything needed to work with several different configurations of TimesTen database and it also contains several 'Hands on Labs' plus a set of sample programs.

NOTE: The demo VM provides convenient access to the rich functionality of TimesTen but it should always be borne in mind that this is a single VM with limited resources designed to be able to run on modestly specified host systems. It is not a suitable environment for performing benchmarks or other performance testing.

The Hands on Labs are ideally suited for someone who wants to quickly get the flavor of TimesTen as each lab typically takes 30-60 minutes to complete.

If your primary interest is just using the Hands on Labs then you should refer to the companion document **TimesTen Demo VM - Quick Start** or alternatively, you can proceed using this document as follows:

- a) Read the **Pre-requisites** section and ensure that all the requirements are met.
- b) Follow the **Importing the VM** section under **Installation and Configuration**.
- c) Skip to the **Hands on Labs** section at the end of **Using the TimesTen Demo VM**.

For a more detailed overview of the VM and how it can be used, continue reading here...

Three distinct VM / network configurations are supported, which determine whether or not you can access the VM from the host system via SSH, VNC etc. and whether the VM has access to external networks and the Internet.

Within the VM there are a number of different usage scenarios supported which cover TimesTen Classic, TimesTen Cache and TimesTen Scaleout. The VM also hosts an Oracle Database instance and a ZooKeeper server which are required for some of the scenarios.

The Oracle Database and ZooKeeper server run directly under the VM's Linux OS while the TimesTen instances run in individual hosts (**tt_{hostN}**) each within its own container within the VM. Up to 7 TimesTen hosts/containers are supported, depending on the specific scenario.

The performance of the 'network' which interconnects the VM host and the containerized hosts is approximately equivalent to that of a 25 Gbit/s physical ethernet network.

An easy to use but feature rich configuration management system has been implemented within the VM to make it extremely easy to switch between the different usage scenarios within a matter of moments. In addition the scenarios that support a Hands-on-Lab can be selected and controlled by double clicking the appropriate icons on the Desktop.

Each scenario is provisioned with scripts that make it easy to deploy a basic version of the scenario's intended use case. Consult the scenario README file (<ttthost1:/tt/config/README.txt>) for detailed instructions.

For those interested in the deeper technical details, these can be found in the various appendices; the main body of this document focuses on the 'how' rather than the 'why'.

Pre-requisites

A host system with at least 2 physical CPU cores (4 physical cores or more are recommended), at least 8 GB RAM (12 GB or more is recommended), at least 70 GB free disk space and a display resolution of at least 1280 x 720 (1920x1080 or better is recommended). Placing the Virtual Machine files on SSD or flash storage is recommended for best performance.

As initially imported, the VM is configured to use 2 CPU cores and 6 GB RAM. If your host exceeds the minimum spec then you can, and should, increase the CPU and/or memory allocation for the VM, which will improve its performance (see **Appendix A: Advanced VM Configuration**).

IMPORTANT: If running the VM on a host with the minimum configuration (2 cores, 8 GB RAM) it is strongly recommended that you close all other applications on the host before starting the VM otherwise you may encounter issues such as insufficient memory.

The host OS must be one which supports VirtualBox 6.1 such as 64-bit Windows 7 Professional or later, macOS 10.13 or later or Oracle/RedHat Linux 6 64-bit or later.

VirtualBox 6.1 or later for your host OS must be already installed and functional. Using the latest available version of VirtualBox is strongly recommended. Installation of the VirtualBox Extension Pack is also strongly recommended. If you are using a newer version of VirtualBox than that used to create the VM image then after importing the VM image to create your VM you should update the VirtualBox guest additions within the VM as detailed [here](#).

Supported VM Network Configurations

There are 3 supported network configurations for the VM:

Standard - one 'internal' network

Everything runs within the VM. All user activity occurs within the VM. There is no connectivity between the VM and the host system. All access to the VM is via the VM's Linux OS GUI console accessed via VirtualBox.

This is the default configuration for the VM when it is initially imported.

Enhanced - one 'internal' and one 'host only' network

Like the standard configuration, except that access to the VM's Linux OS is also possible from the machine hosting the VM. This means you can run SSH or VNC sessions from the host machine into the VM and there is limited TimesTen client/server connectivity between the host machine, the VM and its containerized hosts. In this configuration the VM's Linux OS does not have access to any external network.

Full - one 'internal', one 'host only' and one 'NAT' network

Like the enhanced configuration, except that the VM's Linux OS also has access to the outside world, including the Internet, depending on the host system's connectivity.

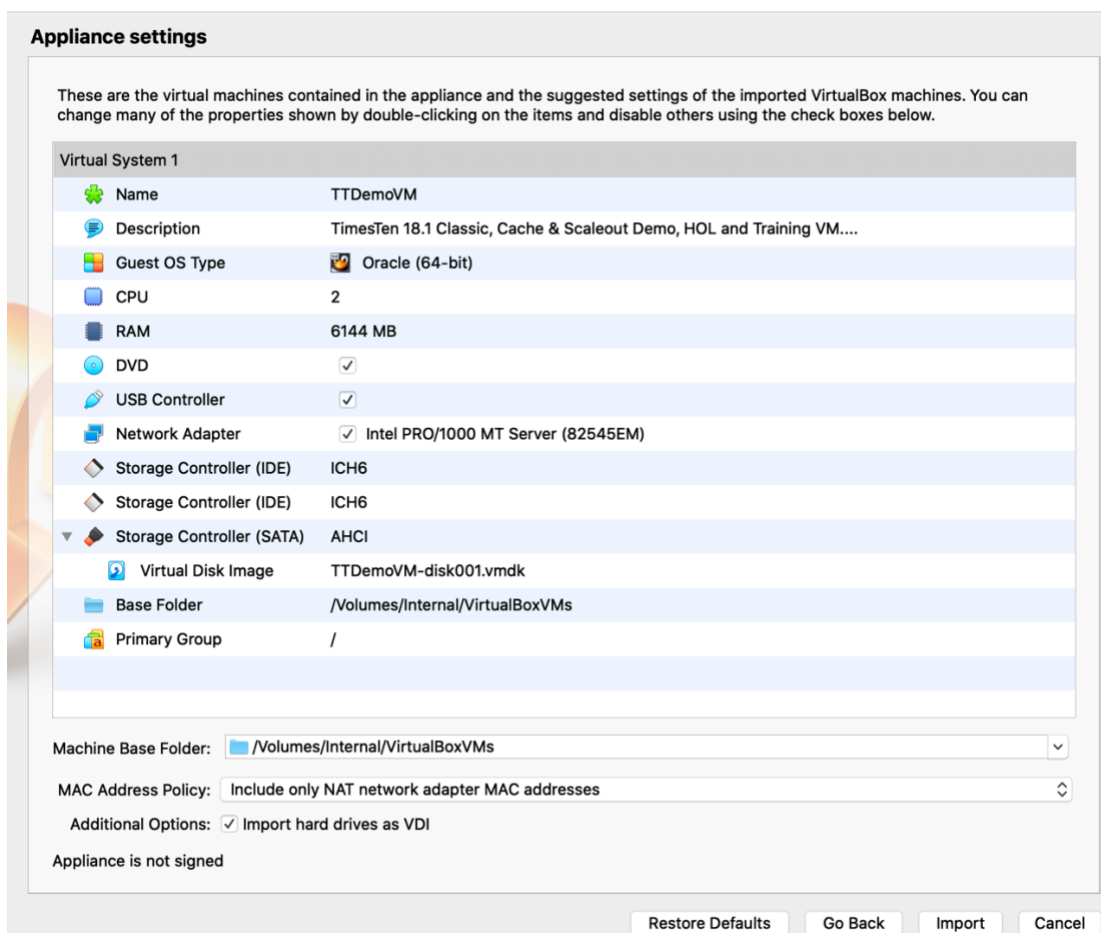
Installation and Configuration

Importing the VM

All configurations are supported via a single unified VM image (TTDemoVM.ova).

First obtain the OVA file. One way to do this is to download it from oracle.com (<https://www.oracle.com/database/technologies/timesten-181-vm.html>).

Next you need to import the OVA file to create the virtual machine. Use the VirtualBox **File / Import Appliance...** menu option.



Configuring the VM

After importing the VM some simple configuration actions may need to be performed, depending on the desired configuration. If you are happy with the VM window size, the **standard** network configuration and the default 6 GB RAM and 2 CPU cores allocated to the VM then nothing more needs to be done.

If you wish to increase the resources allocated to the VM, or use one of the more advanced networking configurations, then see **Appendix A: Advanced VM Configuration**. You can change the VM configuration to change the resource allocation or networking configuration at any time (except when the VM is running).

VM window size and display resolution

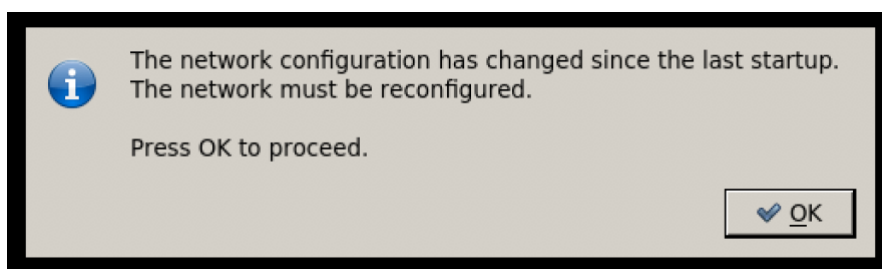
As imported, the VM is configured with a display resolution of 1280x720 and a scaling factor of 200%. These settings are suitable for a host with a reasonably good screen resolution and HiDPI (e.g. retina) display. If your host has a lower resolution, or non-HiDPI, display then the VM's window may be too large. You can change these settings using the View option on the VM's VirtualBox menu when it is running. Changes to these settings are persistent across VM restarts.

If your host has a maximum resolution of 1280x720 then it is recommended to use 'full screen' mode when viewing the VM to maximize the available screen area.

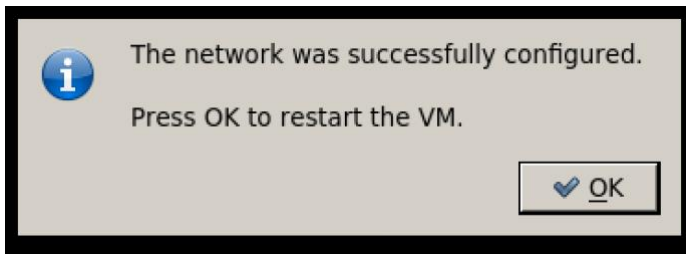
Depending on your host's screen resolution and the VM display settings, you might find that when you first log in some of the icons on the Desktop have been re-arranged by the Linux window manager and overlap each other. If this is the case, just drag them to suitable locations and set the VM display settings to your taste.

Starting the VM for the first time

Start the VM from the VirtualBox GUI. Once the VM has booted, login to the VM as user **oracle** with password **oracle**. If this is the first time the VM has been started since it was imported, or since its network configuration has been modified in the VirtualBox settings, then you will see this message:



Click **OK** to allow the OS network settings to be automatically determined and configured. After a few seconds you will see this message:



Click **OK** and the VM will reboot. Once it has restarted it is ready for use.

Using the TimesTen Demo VM

Getting Started

Login to the VM as **oracle/oracle**. All activities within the VM are performed as the **oracle** user, which also has sudo capability without requiring any password to be entered.

After logging into the VM as **oracle** you will see desktop icons to (a) start SQL Developer, (b) start a Terminal session, (c) open this README file.

In addition you will see a set of icons for each Hands on Lab (Classic, Cache and Scaleout) allowing you to Reset, Start and Stop the Hands- on Lab plus an icon to display the current state of the Hands on Lab and to display the instructions for the Hands on Lab.

For details of the VM's filesystems and their contents please see **Appendix D: VM Filesystem Layout and Contents**.

The VM supports configurations for TimesTen Classic (both standalone and with replication), TimesTen Cache (both standalone and with replication) and TimesTen Scaleout. In all cases the TimesTen instances run in their own hosts which are implemented as containers (one instance per containerized host) in order to present a realistic environment.

The containerized hosts are named **tthostN** where **N** ranges between 1 and the number of hosts in the scenario.

Additional components run directly under the VM's Linux OS; these include:

- A TimesTen Classic client instance, located in **/demo/inst/vminst**, which is primarily intended to provide a client to access the instances running in the containerized hosts.
- A ZooKeeper server, which is used by the various Scaleout configurations.
- An Oracle Database instance, which is used by the various TimesTen Cache configurations.

The configuration and management of all of these components, plus the containerized hosts, is handled automatically by the **vmconfig** utility. For detailed information about this utility please consult **Appendix C: The 'vmconfig' Utility**. When controlling the Hands on Labs using the Desktop icons, under the covers the icons invoke scripts which in turn invoke the necessary **vmconfig** commands.

The **vmconfig** utility, and this document, use some specific terminology as follows:

Configuration: The overall TimesTen functional configuration which corresponds to the three deployment modes of the TimesTen product; **classic**, **cache** and **scaleout**.

Scenario: A specific setup within a configuration which specifies the number of TimesTen containerized hosts and implies a specific usage scenario. Some examples are **classic1**, **cache3**, **scaleout7**.

Each scenario supports one instance per containerized host with a pre-defined configuration. The default scenario configurations are sized for a VM memory allocation of 6 GB. If you have allocated more memory to your VM then you can potentially increase the memory parameters in the sample TimesTen DSN(s). Ultimately you are free to create any configuration that you choose but, in that case, you are responsible for setting things up as you require.

All scenarios provide two separate networks (a 'private/internal' network and a 'public/external' network) connecting the containerized hosts and the VM itself. Both networks are IPv4/IPv6 dual stack. The network between the VM and the host system is also IPv4/IPv6 dual stack (provided that the host OS supports IPv6 and the VirtualBox 'host only' network is suitably configured).

Using the **vmconfig** utility you can switch between scenarios and start/stop them using just a few simple commands.

The following sections detail the supported configurations and their associated scenarios.

TimesTen Classic Configurations

The **classic** configurations support TimesTen Classic instances on up to 4 TimesTen hosts. The defined classic scenarios and their intended usage is as follows:

- classic1** A single standalone TimesTen Classic instance.
- classic2** Two classic instances with active-standby pair replication.
- classic3** Three classic instances; an active-standby pair plus one readonly subscriber.
- classic4** Four classic instances; an active-standby pair plus two readonly subscribers.

The containerized hosts all include pre-defined scripts to create an instance, create a database, setup replication etc. For details on the scripts and the procedure to roll out the environment consult the file **README.txt** in the **/tt/config** directory on **tthost1**.

The scripts are intended to be a starting point to help you get a fully functional environment up and running quickly and easily. You are of course free to customize/enhance them as desired. You should create any TimesTen instances in **/tt/inst** and database files should be placed under **/tt/db**.

At the VM level, the VM instance (**/demo/inst/vminst**) will have a suitable client configuration installed that matches the scenario, as will SQL Developer. If the scenario supports a **Hands on Lab** that will also be installed ready for use.

TimesTen Cache Configurations

The **cache** configurations support TimesTen Cache instances on up to 4 TimesTen hosts. These scenarios also utilize the integrated Oracle Database instance hosted in the VM. The defined cache scenarios and their intended usage is as follows:

- cache1** A single standalone TimesTen Cache instance.

- cache2** Two cache instances with active-standby pair replication.
- cache3** Three cache instances; an active-standby pair plus one readonly subscriber.
- cache4** Four cache instances; an active-standby pair plus two readonly subscribers.

The containerized hosts all include pre-defined scripts to create an instance, create a database, setup cache and replication etc. For details on the scripts and the procedure to roll out the environment consult the file **README.txt** in the **/tt/config** directory on **tthost1**.

The scripts are intended to be a starting point to help you get a fully functional environment up and running quickly and easily. You are of course free to customize/enhance them as desired. You should create any TimesTen instances in **/tt/inst** and database files should be placed under **/tt/db**.

At the VM level, the VM instance (**/demo/inst/vminst**) will have a suitable client configuration installed that matches the scenario as will SQL Developer. If the scenario supports a **Hands on Lab** that will also be installed ready for use.

The integrated Oracle Database instance has been preconfigured for use by TimesTen Cache. You should confine your activities to the pluggable database **XEPDB1** which is pre-configured for use with TimesTen cache. The cache administration user is **ttcacheadm** with password **ttcacheadm**. A regular user, **appuser**, with password **appuser**, has also been defined.

You are free to make any changes to the database contents within the XEPDB1 database (users, tables, etc.) that you desire as the database is reset to a 'clean' condition (just the two pre-defined users and cache management schema) whenever the scenario is changed using a **vmconfig set ...** command and subsequently started.

TimesTen Scaleout Configurations

The **scaleout** configurations support TimesTen Scaleout instances on up to 7 TimesTen hosts. These scenarios utilize the ZooKeeper server hosted in the VM. The defined scaleout scenarios and their intended usage is as follows:

- scaleout3** One management instance and two data instances.
- scaleout5** One management instance and four data instances.
- scaleout7** One management instance and six data instances.

The management host (**tthost1**) includes pre-defined configuration files and a script for deploying a Scaleout grid and database using **ttGridRollout**. These files can be found in **/tt/config**; consult the **README.txt** file in that directory for more information.

The first data host (**tthost2**) includes files and a script for populating the tables in the demo database. These files are located in **/tt/data**; consult the **README.txt** file in that directory for more information.

TimesTen instances are created in **/tt/inst** and the database files are placed in **/tt/db**.

At the VM level, the VM instance (**/demo/inst/vminst**) will have a suitable client configuration installed that matches the scenario as will SQL Developer. If the scenario supports a **Hands on Lab** that will also be installed ready for use.

TimesTen Sample Programs

The VM contains a pre-built copy of the TimesTen Sample programs from the TimesTen public Github repository '[oracle-timesten-samples](#)'. These samples include a guided QuickStart for both Classic and Cache plus a set of sample programs for both TimesTen Classic and TimesTen Scaleout.

In the VM the samples can be found at **/demo/oracle-timesten-samples** while within the containerized hosts they can be found at **/shared/oracle-timesten-samples**. The samples are shared between the VM and the containerized hosts, and are read only within the containers, so any changes need to be made at the VM level.

Accessing the VM and the TimesTen hosts

You can only access the TimesTen containerized hosts when they are running and even then, generally only from within the VM itself. You can access them in various ways:

1. Using SSH, SCP and SFTP. The hosts are configured for passwordless SSH (key based SSH) from the VM, and between one another, for the user 'oracle'.
2. Using TimesTen client tools such as ttIsqlCS, ttBulkCpCS etc from the VM client instance (**/demo/inst/vminst**).
3. Using the SQL Developer installed within the VM (which uses the VM's TimesTen client instance).

If you are using the **Enhanced** or **Full** network configurations then you have some additional access capabilities (see **Appendix B: Accessing the VM from the Host** for more details):

4. You can access the VM from the host system using tools such as SSH, SCP, SFTP and.
5. You can access the VM's Oracle database from the host using tools such as SQL*Plus, SQL Developer etc.
6. You can access the TimesTen instances in the containerized hosts from the host system using TimesTen client/server via SSH port forwarding. This must be explicitly enabled in the VM by running the command **vmconfig remcs on** (for more details see **Appendix C: The 'vmconfig' Utility**). This connectivity has some limitations; primarily TimesTen client failover and/or re-direction/load balancing (Scaleout only) will not function and also performance is reduced due to the overhead of SSH encryption/decryption.

Hands on Labs

For selected scenarios, the VM provides a 'Hands on Lab' (HOL). This is a series of step by step hands on exercises to help you learn the basics of deploying and using TimesTen in that specific configuration. A HOL typically takes 30-60 minutes to complete.

To use a HOL, configure the VM for the desired HOL by double clicking the appropriate 'Reset' icon on the Desktop. Resetting a HOL makes that HOL the currently active HOL and sets it back to its initial state.

In order to successfully reset a HOL, the current HOL (if any) must be in the stopped state and both SQL Developer and FireFox must be closed.

Once a HOL has been reset it can be started and stopped using the associated 'Start' and 'Stop' icons. The 'HOL Info' icon, when double clicked, reports on the currently selected HOL and its state while the 'Hands on Lab Instructions' icon opens the step by step instructions for the currently selected HOL.

Currently, the following scenarios support a Hands on Lab:

- **classic3**
- **cache1**
- **scaleout7**

Summary

The previous sections have described the basics of how to install, configure and use the TimesTen Demo VM.

The rest of this document contains more detailed information on various topics as a reference for those who are interested.

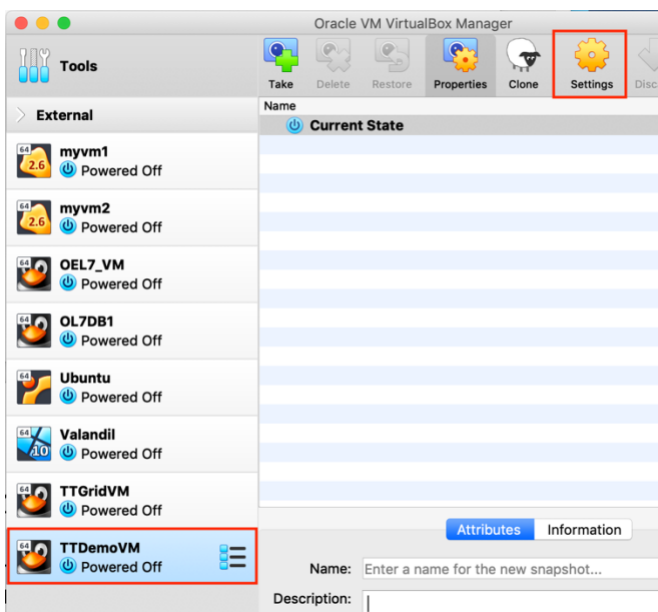
Appendix A: Advanced VM Configuration

NOTE: All VirtualBox GUI screenshots in this document were captured on macOS. If your host system is running Windows or Linux, then you may see some slight differences in the UI compared to the screenshots.

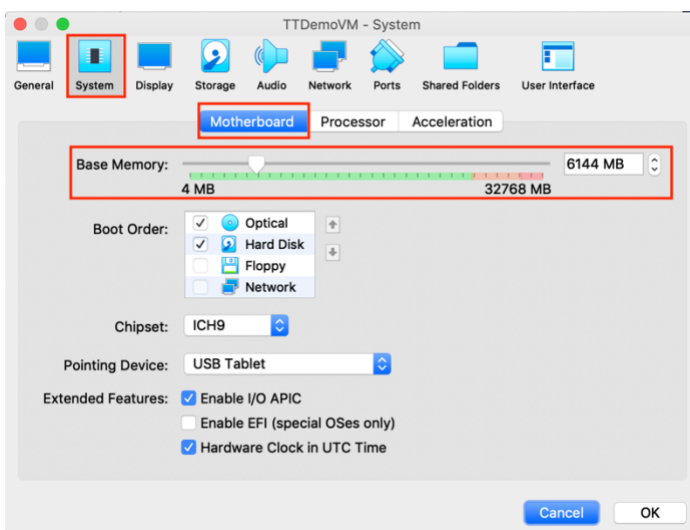
Increasing the RAM and CPU cores allocated to the VM

If your host system has sufficient resources, you can improve the performance and capacity of the VM by allocating more memory and/or more CPU cores. This is done via the VirtualBox GUI in the VM Settings.

Click on the TTDemoVM entry in the list of Virtual Machines and then click on the **Settings** icon in the tool bar to modify the settings:

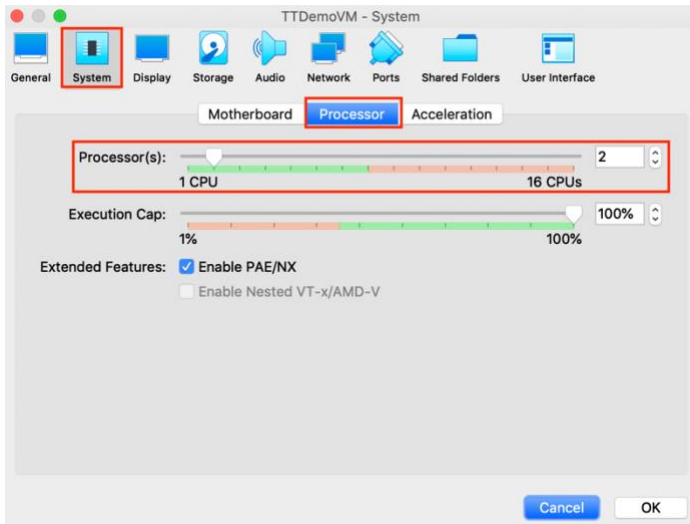


When the settings dialogue appears, click the **System** icon and then click the **Motherboard** tab. Adjust the memory assigned to the VM using the controls:



You should not allocate less than 6 GB of memory nor should you exceed the VirtualBox recommended maximum (shown in green in the settings area).

Next click the **Processor** tab. Adjust the number of CPU cores using the controls:



You should not allocate less than 2 CPUs nor should you exceed the VirtualBox recommended maximum (shown in green in the settings area).

Click the **OK** button to save the changes.

Advanced Network Configurations

The VM OS includes an automatic networking configuration mechanism which auto-detects the network configuration (within certain limitations) and configures the VM OS accordingly. This mechanism works correctly provided that only the three supported network configurations detailed here are used and they are configured exactly as documented here. Other configurations may appear to work but some functionality will be compromised.

The network auto configuration mechanism will be activated in the following situations:

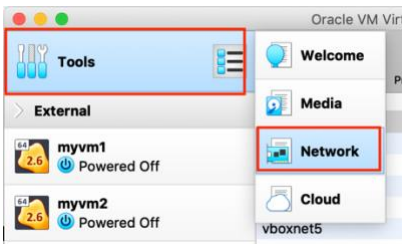
1. The first time the VM boots after it has been imported.
2. The first time the VM boots after the network configuration has been changed to a different (supported) configuration.

*The **Enhanced** network configuration*

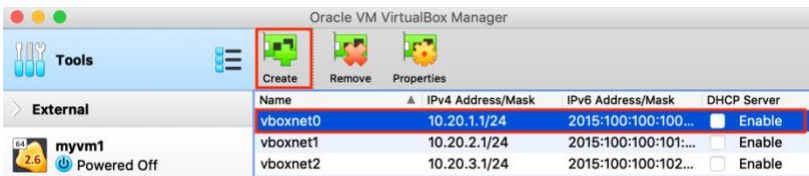
In this configuration the VM has two network adapters. The first is the 'internal network' adapter used by standard configuration and the second is a VirtualBox 'host only' adapter which allows for network communication between the VM and the host system (but not between the VM and the outside world beyond the host system).

You can use an existing host-only network, if you already have one configured, provided that its settings meet the criteria detailed below. Alternatively, you can create a new host-only network. The instructions here cover creating a new network (strongly recommended).

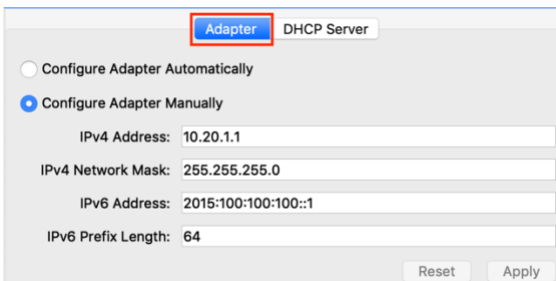
In the VirtualBox GUI, choose **Tools** and then **Network**:



If you are creating a new host-only network, click the Create button in the toolbar to create it otherwise click on an existing network to re-configure it. In this example we will use **vboxnet0**.

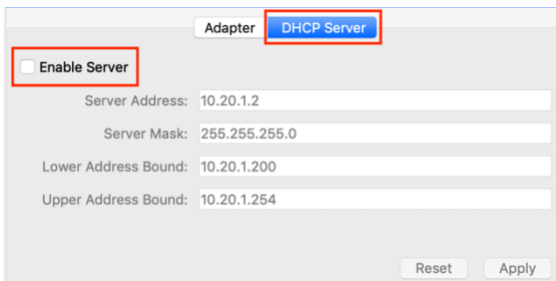


Click on the **Properties** icon to display the network's settings then click on the **Adapter** tab:



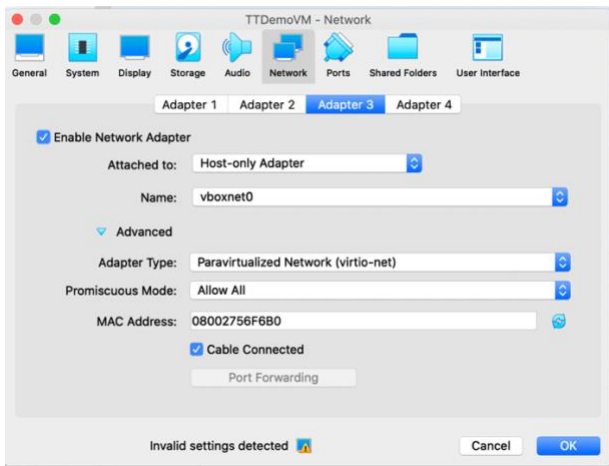
Configure the settings exactly as shown.

Next click on the **DHCP** tab and uncheck the box to disable the DHCP server for this adapter.



Finally click the **Apply** button to save the changes.

Open the **Settings** for the **TTDemoVM** virtual machine, click on the **Network** tab and click on the **Adapter 3** tab:



Configure the settings exactly as shown but substitute the name of your host-only network if it is not **vboxnet0** (the value for the MAC address will differ from that shown, that is okay).

Click **OK** to save the changes.

Follow the instructions in **Appendix B: Accessing the VM from the Host** to configure the host OS to enable network access to the VM.

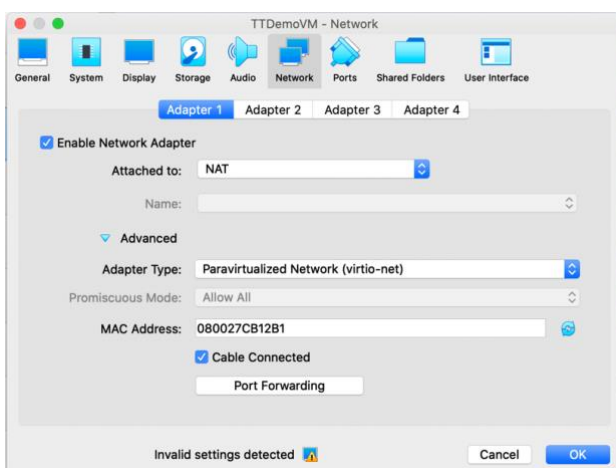
*The **Full** network configuration*

This configuration extends the Enhanced configuration by adding a NAT network which allows the VM to share the host systems connectivity so that it potentially has access to the Internet etc. This is achieved by adding a third network adapter of type **NAT** (Network Address Translation).

NOTE: Although the VM itself will be able to use the host's network connectivity, the Docker containers running within the VM can not.

First configure the VM for Enhanced networking as described in the previous section.

Open the **Settings** for the **TTDemoVM** virtual machine, click on the **Network** tab and click on the **Adapter 1** tab:



Configure the settings exactly as shown (the value for the MAC address will differ from that shown, that is okay).

Click **OK** to save the changes.

Appendix B: Accessing the VM from the Host

Sometimes it is useful to be able to access the VM from the host OS. For example:

1. To login via an SSH terminal session or VNC session.
2. To transfer files to the VM using SCP or SFTP (although using VirtualBox shared folders is usually easier and faster)

NOTE: The VM normally runs a VNC server on screen :1 (port 5901). You can check this using the command **vnclist** and you can stop and start the server using the commands **vncstart** and **vncstop**.

Providing that you have followed the instructions to configure the VM for the **enhanced** or **full** network configurations as detailed in **Appendix A: Advanced VM Configuration** then all of the above are possible; the host OS has full IPv4 (and possibly IPv6 if the host OS supports it) connectivity to the VM.

NOTE: Access to a local VM from the host may not be possible when some types of VPN solution are active on the host, depending on how the VPN software is configured.

You can refer to the VM using one of its IP addresses (10.20.1.51 / 2015:100:100:100::51) but more conveniently you can add entries to the host OS's hosts file to allow you to refer to the VM by its hostname (**ttdemovm**).

To add entries to your host system's hosts file proceed as follows.

Windows

Log in to Windows using an account that has Administrator privilege.

Using a suitable text editor (such as **Notepad** for example), open the file **C:\Windows\System32\Drivers\etc\hosts** file.

Add the following lines at the end of the file:

```
10.20.1.51          ttdemovm
2015:100:100:100::51 ttdemovm
```

Save the file, ensuring that you save it as a plain text file without any extension.

macOS and Linux

Using the **sudo** command with a suitable text editor (such as **vi** or **nano** for example), open the file **/etc/hosts**.

Add the lines shown above to the end of the file and save the file.

Appendix C: The 'vmconfig' Utility

The **vmconfig** utility is the tool that you use to manage the different configurations and scenarios available within the demo VM. It allows you to configure the VM and containerized hosts for a specific scenario, to remove the current configuration, to start and stop the current configuration and to view the status of the current configuration.

vmconfig takes care of all OS and environmental configuration for the host OS and all of the containerized hosts. This includes the ZooKeeper server and the Oracle Database instance depending on the specific configuration. This simplifies usage of the VM and allows the user to focus on the TimesTen related aspects rather than the other aspects.

Command syntax

Here is the detailed online help for the **vmconfig** utility:

Usage:

```
vmconfig list [-verbose]
vmconfig status
vmconfig vmversion
vmconfig nwconfig
vmconfig reset [-verbose]
vmconfig clearrepo
vmconfig set <cfgname> <numhosts> [-force] [-verbose]
vmconfig start
vmconfig stop
vmconfig restart
vmconfig remcs { on | off | status }
vmconfig help
```

This command must be run as the user 'oracle'.

`list [-verbose]`

Displays the list of available scenarios and, if '-verbose' is specified, their descriptions.

`status`

Displays the current scenario and the status (running/stopped) of each component

`vmversion`

Displays the version number of the VM.

`nwconfig`

Displays the current network configuration (standard, enhanced, full) of the VM.

`reset [-verbose]`

Removes scenario specific files such that there is no longer a configured scenario. Only permitted when there are no running components.

`clearrepo`

Empties the Scaleout repository directory '/demo/repos'. This is the only action that affects the contents of that directory.

set <cfgname> <numhosts> [-force] [-verbose]

Clears the current scenario and configures the system for the specified scenario. Only permitted when there are no running components. If the target scenario is the same as the current one then this is a no-op unless you specify '-force'.

Valid values for 'cfgname' are:

classic - TimesTen Classic
cache - TimesTen Cache
scaleout - TimesTen Scaleout

Valid values for 'numhosts' depends on the configuration type:

classic - 1 to 4
1 = standalone
2 = active-standby pair
3 = active-standby pair with 1 subscriber
4 = active-standby pair with 2 subscribers

cache - 1 to 4
1 = standalone
2 = active-standby pair
3 = active-standby pair with 1 subscriber
4 = active-standby pair with 2 subscribers

scaleout - 3, 5 or 7
3 = 1 management, 2 data
5 = 1 management, 4 data
7 = 1 management, 6 data

start

Starts all the components in the current scenario.

stop

Stops all the components in the current scenario.

restarts

Restarts (i.e. stops and then starts) all the components in the current scenario.

remcs { on | off | status }

Enables, disables or displays the status of remote client-server access. When enabled, TimesTen clients on the host system can connect to TimesTen instances running in the containerized hosts by connecting to port 1000n (where 'n' is the n in tthostn) on the VM. This capability utilises SSH port forwarding so TimesTen client failover, redirection and load balancing will not function.

help

Displays this extended help information.

NOTES:

1. The state of the containerized hosts is reset on each start apart from the /repos, /shared and /tt filesystems which are preserved across restarts. The state of the Oracle database is also preserved

across restarts.

2. The state of the containerized hosts /tt filesystem is reset on every completed (re)set operation (other than no-ops).
3. The state of the Oracle database is reset on every completed (re)set set operation (other than no-ops).
4. The contents of the containerized hosts /repos filesystem (which maps to the VM's /demo/repos directory) is never automatically reset. You can explicitly clear it using 'vmconfig clearrepo'.

The configuration name is one of **classic**, **cache** and **scaleout**.

The scenario name is a concatenation of the configuration name and the number of hosts for example **classic2**, **cache1**, **scaleout5**.

Example 1: Configure the VM for two TimesTen cache databases with replication

```
$ vmconfig set cache 2
$ vmconfig start
info: starting scenario 'cache2'
info: starting hosts: OK
info: host initialization: OK
info: starting Oracle Database: OK
info: resetting Oracle Database contents, this may take some time...
info: Oracle Database reset: OK
```

Example 2: Configure the VM for four TimesTen classic databases with replication

```
$ vmconfig set classic 4
$ vmconfig start
info: starting scenario 'classic4'
info: starting hosts: OK
info: host initialization: OK
```

Example 3: Configure the VM for a 7 node TimesTen Scaleout grid

```
$ vmconfig set scaleout 7
$ vmconfig start
info: starting scenario 'scaleout7'
info: starting hosts: OK
info: host initialization: OK
info: starting ZooKeeper: OK
```

Example 4: View information about the currently configured scenario

```
$ vmconfig status
info: scenario = scaleout7
info: zookeeper = running
info: tthost1 = running
info: tthost2 = running
info: tthost3 = running
info: tthost4 = running
info: tthost5 = running
info: tthost6 = running
info: tthost7 = running
```

Example 5: Stop the current scenario

```
$ vmconfig stop  
info: stopping scenario 'scaleout7'  
info: stopping ZooKeeper: OK  
info: stopping hosts: OK
```

Example 6: Clear the current scenario

```
$ vmconfig reset
```

Appendix D: VM Filesystem Layout and Contents

The VM has a single filesystem (plus /boot). Notable directories within the filesystem are:

/home/oracle	'oracle' home directory
/opt/oracle	Oracle DB XE instance
/demo	All demo related files including ZooKeeper server installation, TimesTen software installations, VM instance, repositories, container filesystems, configuration framework files, content etc. etc.

You should not modify anything under **/home**, or **/opt/oracle**.

For **/demo** you should restrict your activities to **/demo/config** (your work directory) and not modify anything under any other directory unless you know exactly what you are doing.

Appendix E: Additional Tools and Utilities

The VM includes some additional tools and utilities which can be used with some of the scenarios.

Utilities for Scaleout scenarios

If you have deployed a Scaleout scenario and created the standard sample database with the sample schema and loaded the sample dataset, either by using the pre-provisioned scripts or by completing the Scaleout HoL then you can use the following utilities to run the Scaleout QuickStart samples against the database.

runSample

Usage:

```
runSample { odbc | jdbc } { cs | direct }  
          [ -procs n ] [ -duration nmins ]
```

Runs the TimesTen Scaleout sample program against the sample database.
Options are:

odbc | jdbc

The version of sample program to run.

cs

Processes execute against all data instances using Client/Server connections.

direct

Processes execute against all data instances using direct connections.

-procs n

The number of application processes to execute per data instance.
Minimum is 1, default is 1.

-duration nmins

The run duration in minutes. Minimum is 1, default is 15.

At the end of the run detailed execution statistics are displayed.

sampleStats

If you redirect the output of `runSample` to a file you can process that file using `sampleStats` to produce a summary.

Usage:

```
sampleStats runsample_output_file
```

Processes the output from a successful 'runSample' command and displays a summary of execution metrics.

Appendix F: Further Information, Help and Support

You can get information on TimesTen from the [TimesTen OTN portal](#). There you will also find [TimesTen software downloads](#) and the [TimesTen online documentation](#).

This VM is not an officially supported Oracle product and support is provided on a best effort basis.

You can ask questions about TimesTen and this VM in the [TimesTen OTN Community Forum](#) and you can find interesting articles about TimesTen on the [TimesTen Blog](#).

If you are not able to find the answer to your questions using the above resources, please contact the VM's developer directly at chris.jenkins@oracle.com.