

# Oracle Tuxedo on Docker Containers



## Oracle Tuxedo on Docker

We are announcing that Oracle Tuxedo is now certified to run on Docker containers. As part of this certification, we are releasing Dockerfiles and supporting sample scripts on GitHub for both [Clustered](#) and [SHM/non-Clustered](#) modes to build images of the Oracle Tuxedo. These images are built as an extension of existing Oracle Linux images. You can use these script create your own.

[Docker](#) is a platform that enables users to build, package, ship and run distributed applications. Docker users package up their applications, and any dependent libraries or files, into a Docker image. Docker images are portable artifacts that can be distributed across Linux environments. Images that have been distributed can be used to instantiate containers where applications can run in isolation from other applications running in other containers on the same host operating system.

The table below describes the certification provided for two Oracle Tuxedo versions. You can use these combinations of Oracle Tuxedo, Linux and Docker versions when building your Docker images.

Oracle Tuxedo Version	HOST OS	Kernel Version	Docker Version
12.1.3.0.0	Oracle Linux 7	Unbreakable Enterprise kernel Release 3 (3.8.13+)	1.3.9+
12.2.2.0.0	Oracle Linux 7	Unbreakable Enterprise kernel Release 3 (3.8.13+)	1.3.9+

## Oracle Tuxedo Docker Images

We are releasing Dockerfiles and supporting scripts to build Oracle Tuxedo Docker images on GitHub. These images are built as an extension of existing Oracle Linux image 6, with the Oracle Tuxedo 12cR2 (12.2.2.0.0) installations.



Fig 1. Oracle Tuxedo Docker Image

## Custom Built Oracle Tuxedo Images

You can create your own Oracle Tuxedo Docker images. To help you with this, we have posted Dockerfiles and scripts on [GitHub Oracle Tuxedo Docker files](#) as the examples for you to get started.

What are the prerequisites to build the custom Tuxedo images:

1. Oracle Linux base image.
2. Dockerfiles and scripts from GitHub.
3. Oracle Tuxedo installer.

## Dockerfiles and Scripts on GitHub

### Dockerfile

The Dockerfile to create an Oracle Tuxedo install image performs the following functions:

1. Extend the Oracle Linux base image
2. Installs the Tuxedo in silent mode
3. Set up a Tuxedo sample domain with 3 nodes

### Scripts

The scripts aid in the creation of the Oracle Tuxedo image and serve as examples to extend the Oracle Linux base image with the configuration of an Oracle Tuxedo domain.

## Clustering Oracle Tuxedo on Docker Containers

These Docker containers enable users to create clustered and non-clustered Oracle Tuxedo domain configurations. Each server running in the domain runs in its own Docker container and is capable of communicating as required with other servers on the same host.

## The advantages of this topology

- Good for traditional-like deployments.
- Easy to deploy containers from Oracle Tuxedo domain images.
- Easy to scale up and down the cluster.
- Good for developers.

## How to Build and Run

On [GitHub](#) you will find Dockerfiles and supporting scripts needed to build an Oracle Tuxedo install image and to extend this image to create an Oracle Tuxedo domain

Download the entire directory structure of [Clustered](#) and [non-Clustered](#) to build your Oracle Tuxedo images and start your containers.

## Dockerfiles

Two Dockerfiles for Oracle Tuxedo 12cR2 under the `/TuxedoUsers/Tuxedo-samples/tree/master/dockermpp` and `/TuxedoUsers/Tuxedo-samples/tree/master/docker` subdirectories, one Dockerfile to build an Oracle Tuxedo install image and a second Dockerfile to build an Oracle Tuxedo sample domain with 3 nodes.

## Scripts – non-Clustered

Under the subdirectories `/TuxedoUsers/Tuxedo-samples/tree/master/docker`, download `tuxedo_docker.zip`

**Dockerfile** - build Tuxedo image using Tuxedo installation.

**Dockerfile.template** – the template used to generate final Dockerfile.

**README.md** – instructions for how to run the sample.

**build.sh** – the script to generate Dockerfile, response file and related build script.

**tuxedo\_docker.zip** – contains all the necessary files to build the Tuxedo on docker.

**install\_tuxedo12.1.3.rsp** – response file for Tuxedo silent installation.

## How to use:

1. Into an empty directory:
  - i. Download the Tuxedo 12.1.3 Linux 64 bit installer from OTN
  - ii. Download `tuxedo_docker.zip` from this github directory
  - iii. Optionally download the latest Tuxedo rolling patch from My Oracle Support
2. Unzip `tuxedo_docker.zip`
3. Execute `build.sh`

You should end up with a docker image tagged `oracle/tuxedo`

You can then start the image in a new container with: `docker run -i -t oracle/tuxedo /bin/bash` which will put you into the container with a bash prompt. If you want to test the new container, simply execute the `simpapp_runme.sh` in an empty directory and the script will build and run the Tuxedo `simpapp` application.

## Scripts – Clustered

Under the subdirectories `/TuxedoUsers/Tuxedo-samples/tree/master/dockermpp`, download all the files.

**build.sh** – check environment and generate the response file for silent mode, Dockerfile, etc.

**Dockerfile.template** – the template used to generate final Dockerfile.

**fix\_locations.sh** - the script modifies the installer scripts and response files to match the local environment.

**README.md** – instructions for how to run the sample.

**simpapp\_runme.sh** – shell script to build and run the Tuedo sample simpapp.

**start\_tlisten.sh** – start tlisten in the specified ports on the host's network address.

**tuxedo.docker** – parameter list and definition for shell script and templates.

**tuxedo12.1.3.rsp.template** – template for response file used in Tuxedo silent installation mode.

**tuxedo12.1.3\_silent\_install.sh.template** - template file for tuxedo12.1.3\_silent\_install.sh.

How to use

1. First run the docker project non-Clustered that creates a Tuxedo docker image containing the simpapp application
2. Pull the files from directory
3. Build the containers: docker-compose build
4. Start the containers: docker-compose up -d
5. You should have 3 Docker containers (dockerm\_node1\_1, dockerm\_node2\_1, dockerm\_node3\_1) up and running at this point
6. Open a shell in dockerm\_node1\_1: docker exec -it dockerm\_node1\_1 /bin/bash
7. Source the setenv.sh script: source setenv.sh
8. Execute the simpappmp\_runme.sh script: sh ../simpappmp\_runme.sh
9. You should now have a 3 node Tuxedo cluster with each node running a single copy of simpsserv.
10. If you want to verify that the load was executed on multiple nodes do:
  - tmadmin
  - > d site1
  - > psc
  - > d site2
  - > psc
  - > d site3
  - > psq
  - > q
11. You should see that some of the client requests were done on one node, and some on another.

Additional Considerations Running Oracle Tuxedo with Docker

When a Docker container is restarted its IPAddresses changes, Oracle Tuxedos running in the Docker container will now have a new address. Applications as well as others servers that were communicating with the server before the container restart, will be unable to communicate. Configuring a DNS server on Docker

and configure Tuxedo domains to use DNS names is a solution to the IPAddress change after a container restart.

Oracle Tuxedo configuration, server logs, file stores etc are all kept in the container file system. When a Docker container is destroyed you will lose your entire file system. There are two alternatives to this:

- Use the host file-system to store the container's local file system.
- Maintain a "data-only" container to store your domain file system.

To minimize the dependency on the file system we recommend:

- Keep your stores such as TLog and Queue stores in the database

To patch or upgrade your Oracle Tuxedo images created with the Oracle Tuxedo installation image, follow the following steps:

1. Upgrade/patch by extending the Oracle Tuxedo install Docker image.
2. Use the Docker **cp** (copy) command to copy your domain folder to destination either the host or a "data-only" container
3. Remove Container
4. Run new container from extended image (with upgrade/patch).
5. Use the Docker **cp** (copy) command to copy back your domain folder to the upgraded container

Security concerns have been raised regarding Docker and Linux containers.

- One area of concern is whether it is possible to isolate code running in separate containers from each other. There are no known issues impacting the ability to run Oracle Tuxedo in such an environment at this time.
- Another area of concern with respect to security is the source of Docker images. One should only obtain Docker images from trusted sources and one needs to be aware of the frequency of updates and the nature of the controls on Docker Hub.
- Customers should stay current with Docker and Linux technology and remain aware of security issues that are raised in each.
- Docker containers default network mode of "Bridge Networking" does not support multicast. Docker Containers "Host Networking" supports multicast but provides less isolation since it uses the host networking stack. We recommend the use of unicast as the Oracle Tuxedo clustering protocol when running in Docker Containers.

## Conclusion

Docker technology offers the promise of simplifying operations and reducing cost due to the portable characteristics of its artifacts and ease of distribution across Linux environments. Oracle has responded to the growing interest of our customers by certifying Oracle Tuxedo to run in Docker containers, and by providing Dockerfiles, and scripts that support the creation of Oracle Tuxedo configurations running in Docker containers. We hope these are useful, and will seek to improve the scope of our support for Docker environments over time.