Labbox on Docker

Depending on how you're viewing this document, the links in the doc below may not work, including the ToC. Just scroll to the section you want or copy and open a new browser tab.

Labbox on Docker

Labbox is a specialized pre-configured install of Tenable.sc designed for Tenable SEs and Partner Engineers to be able to quickly spin up and demo the product.

Tenable Employees can get the latest install files here. Partners should get the current install files from the Tenable Partner Portal. For full setup instructions, details, or troubleshooting, read on.

Required Software

Docker Desktop (>= 3.X)

- Docker Desktop for Mac (https://hub.docker.com/editions/community/docker-ce-desktop-mac)
- Docker Desktop for Windows (https://hub.docker.com/editions/community/docker-ce-desktop-windows)

- Docker Desktop for Windows (https://hub.docker.com/editions/community/docker-ce-desktop-windows)
  - This should also work under Docker for Linux and Docker on Windows Server, but is not expressly documented or tested. Docker Compose is required for these installs.

Docker for Desktop should be allocated at least 8 GB of RAM; the default is 2 GB. In some instances, you may need to increase this further (https://docs.docker.com/docker-for-mac/#resources for Mac and https://docs.docker.com/docker-for-windows/#resources for Windows. Changing these settings will require Docker to be re-launched.)

Optional Software

Some users may like software like Captain that shows the status of containers in your menu bar and allows you to easily start and stop containers.

Initial Setup: Building Labbox Into A Docker Image

See the note at the top of this README on where to obtain the Labbox files.

Create a directory on your system to hold the project, and decompress the zip file into that directory. You should see a few files and a 'utils' directory.

Change into the project directory and build the Docker image by typing:

```
$ docker build ./ -f Dockerfile -t tenablesc-labbox-image-ol8 --platform linux/amd64 && docker network
create "docker_labbox_default-ol8"
```

This should take less than 30 minutes to complete and requires internet access.

This step should only be required once, the first time you build up labbox via Docker, specifically called out for an update, or unless otherwise instructed.

If you rebuild the base Labbox image, your container will try to rebuild itself at next launch.

Migrate to Labbox on Ol8

Download this new version of labbox and decompress it into a new folder in Documents (or wherever you want).

1. Backup SC on Labbox - follow the 'Backup' directions in this document.
2. Stop the old Labbox container; either via Docker Desktop or the following command the old labbox folder.
   ```
   $ docker-compose stop
   ```
3. Change into the new project directory and build the new Ol8 Labbox image.
   ```
   $ docker build ./ -f Dockerfile -t tenablesc-labbox-image-ol8 --platform linux/amd64 && docker
   network create "docker_labbox_default-ol8"
   ```
4. Follow the 'Creating/Starting the Container' directions.
5. Follow the 'Restoring (Destructive!)' directions.

Creating And Operating The Container

Creating/Starting the Container

After following the build steps (see above), simply run this command:

```
$ docker compose up -d
```

Older versions of Docker will require this format of the command:

```
$ docker-compose up -d
```

This will start the labbox container.

- If this is the first time the container has been run, it will take *up to an hour* to download and install Tenable.sc and download the data for Labbox; the docker-compose up command will run relatively quickly. To see the status of the build, run the monitoring command below. This requires internet access and will download approximately 4-5 GB of data.
- If this is a subsequent container start (it has been built before), it should take less than 30 seconds for Tenable.sc to be available in your browser at https://127.0.0.1:8443. If it's not available after 30 seconds, please review the logs with the monitoring command below.

Credentials

Credentials for Labbox are as follows:
```
admin:password
secmanager:password
```

secmanager:password

There are several other accounts in Tenable.sc; feel free to reset/add/delete accounts as desired.

Stopping the Container

To stop the container (but not delete it) run this command:

```
$ docker-compose stop
```

Monitoring

To monitor what is going on in the container, view the Logs section in Docker Desktop.

You can also type:

```
$ docker logs tenablesc-labbox-ol8 -f
```

This will auto-tail the logs from the running container; CTRL-C will exit out of this (or stop the container by another terminal window/method). Remove the -f flag if you just want a snapshot of the current logs.

Operations

Note: All of these commands are run from the CLI in the host OS, from the folder where labbox lives, unless otherwise specified.

Update Everything (Destructive!)

When you need labbox to update everything, type:

```
$ docker exec tenablesc-labbox-ol8 touch /update-all
```

This is the equivalent command to "labbox sync all" or "labbox sync full" and is DESTRUCTIVE to labbox (you will lose all customizations). The container must be running; the sync will initiate within 60 seconds.

To see the status of the job, run the monitoring command above.

Update Repos

When you need labbox to update just repository data, type:

```
$ docker exec tenablesc-labbox-ol8 touch /update-repos
```

This is the equivalent command to "labbox sync repos". The container must be running; the sync will initiate within 60 seconds.

To see the status of the job, run the monitoring command above.

Gapfill

When you want to try and fix gapfill issues (note that labbox tries to fix this at start up)

```
$ docker exec tenablesc-labbox-ol8 touch /gapfill
```

This is the equivalent command to "labbox gapfill". The container must be running; the job will initiate within 60 seconds.

To see the status of the job, run the monitoring command above.

Upgrade Tenable.sc Only

This will attempt to ensure that Labbox is running the latest GA version of Tenable.sc. It will perform a backup before an upgrade is attempted.

```
$ docker exec tenablesc-labbox-ol8 touch /tsc_upgrade
```

The container must be running; the job will initiate within 60 seconds.

To see the status of the job, run the monitoring command above.

Backup

This will create a normal Tenable.sc backup at utils/backups/ This is likely to be ~5GB in size.

```
$ docker exec tenablesc-labbox-ol8 touch /tsc_backup
```

This is the equivalent command to the instructions in the Tenable.sc docs above. The container must be running; the job will initiate within 60 seconds.

To see the status of the job, run the monitoring command above.

Several commands (like upgrading to EA or Testing mode) will also automatically initate a backup. If you want to explictly disable backups,

create a file called 'disable_backup' in the utils folder.

Restoring (Destructive!)

This will restore a normal Tenable.sc backup at utils/backups/; generally the most recent if there are multiple. If you get odd errors; ensure there is only one backup file in this folder.

Restoring a backup is only supported for current GA and older releases until 5.23.0 (ex: if you try to restore a backup using the automated functionality from an EA version of Tenable.sc, it won't work.)

Run this command BEFORE starting the container from the same folder that your normal Labbox files are in.

Mac

```
$ touch utils/tsc_restore
```

Windows

```
fsutil file createnew utils/tsc_restore 0
```

This is the equivalent command to the instructions in the Tenable.sc docs. The container must **not** be running; the restore job will initiate on the next container start.

To see the status of the job, run the monitoring command above.

EA Testing

This is to test a pre-build of the next version of Tenable.sc.

IMPORTANT NOTES:

- This assumes you're starting with a fully up-to-date (labbox files) and functional labbox environment.
- We attempt to create a backup before upgrading. Restoring a backup onto a fresh install is the only supported method of regressing to a previous version of Tenable.sc.
- See the restore instructions above if you need to go backwards to your old version.
- After upgrading to the EA; DO NOT SYNC DATA OR CONFIGURATIONS; this will very likey break lots of things and you'll have to start over.
- If you're testing out an IA/EA build, that means you're expected to provide feedback on bugs/features.

Under the 'utils' folder, there should be a folder called 'ea' (create it if it doesn't exist). Place the pre-release RHEL/CentOS 7 rpm package in this folder, then run the command below. Do not rename the RPM.

```
$ docker exec tenablesc-labbox-ol8 touch /ea_testing
```

The container must be running; the job will initiate within 60 seconds.

To see the status of the job, run the monitoring command above.

Old Build Testing

This is to test older versions of Tenable.sc.

IMPORTANT NOTES:

- This assumes you're starting with a fully up-to-date (labbox files) and functional labbox (the container starts without errors) environment.
- We attempt to create a backup before upgrading. Restoring a backup onto a fresh install is the only supported method of regressing to a previous version of Tenable.sc.
- See the restore instructions above if you need to go backwards to your old version.
- After downgrading; DO NOT SYNC DATA OR CONFIGURATIONS; this will try to upgrade t.sc and it likely won't work.
- This will install a unconfigured version of the version of Tenable.sc you provide (as low as 5.19.1 is supported). No data exists in this setup.
- Tenable Employees can access old installers here.
- Partners can access old installers from downloads.tenable.com; grab the Oracle Linux 8 installer.
- Tenable.sc credentials will be the same for older versions as for the regular version of Labbox
- Depending on the version of Tenable.sc used; you may notice some discrepancies (being prompted to change passwords, etc.)

Under the 'utils' folder, there should be a folder called 'setup' (create it if it doesn't exist). Place the RHEL/CentOS 7 rpm package in this folder, then run the command below. Do not rename the RPM.

```
$ docker exec tenablesc-labbox-ol8 touch /tsc_testing
```

The container must be running; the job will initiate within 60 seconds.

To see the status of the job, run the monitoring command above.

Custom Actions/Commands

If you need to do anything within the container itself, while it is running type:

```
$ docker exec -it tenablesc-labbox-ol8 /bin/bash
```

Useful utilities installed within the docker image include:

- iftop
- htop
- screen
- rsync
- labbox

General Labbox Notes

- Gapfill runs on every container start, in an attempt to ensure that gaps are eliminated.
- The 'utils' directory is shared with the running docker container under /labbox
- It is not recommended to run multiple versions of labbox at the same time due to port and resource constraints.

Multiple Instances of Labbox for Docker

Multiple instances of Labbox are supported, if the following guidelines are followed:

- Do not run more than one instance at a time
- Name each labbox folder appropriately and consistently
- Each instance will use the same base image; if your base image gets rebuilt or deleted, your instances will fail or try to rebuild themselves.
- Use the new name you define in all subsequent commands for that container, like `docker logs -f tenablesc-labbox-ol8-ea`

Instructions to run multiple instances:

1. Have a working labbox instance.
2. Copy your labbox folder and rename it; for example `labbox-docker-ol8-ea`
3. Edit the docker-compose.yml file in the new folder and make the following adjustments (effectively appending `-ea` to several fields following the naming pattern in step 2. The actual pattern does not matter, as long as it's consistent):

   - container_name from `tenablesc-labbox-ol8` to `tenablesc-labbox-ol8-ea`
   - volumes from `tenablescdata-ol8` to `tenablescdata-ol8-ea`
   - volumes from `tenableyumcache-ol8` to `tenableyumcache-ol8-ea`
   - volume names from `tenablescdata-ol8` to `tenablescdata-ol8-ea`
   - volume names from `tenableyumcache-ol8` to `tenableyumcache-ol8-ea`

An example of how the new docker-compose.yml might look:

```
version: "3.8"
services:
  tenablesc:
    image: tenablesc-labbox-image
    hostname: labbox.locallab
    container_name: tenablesc-labbox-ea
    platform: linux/amd64
    ports:
      - "8443:443"
    volumes:
      - tenablescdata-ea:/opt/sc/
    volumes:
      - tenableyumcache-ea:/var/cache/yum
    volumes:
      - ./utils/:/labbox/
    tmpfs:
      - /tmp:mode=1770,size=21474836480
      - /run
volumes:
  tenablescdata:
    name: tenablescdata-ea
```

```
tenableyumcache:
  name: tenableyumcache-ea
```

4. From the command line, in the new folder, run `docker-compose up -d` as you would normally.
5. Monitor the status either from Docker Desktop or by running `docker logs -f tenablesc-labbox-ea` (note the appended 'ea' defining the new container name)

Update Labbox Files

Most updates for Labbox on Docker should be straight forward. Unless other explicit instructions are given, simply replace the files in the labbox folder with updated ones. They will be automatically updated within the container at next start, if needed.

1. Download the new version and decompress it.
2. Stop the docker container if running
   `$ docker-compose stop`
3. Replace the files in your docker_labbox folder with the new versions.
4. Start the docker container
   `$ docker-compose up -d`
5. You may watch the progress of the update in Docker Desktop or by running:
   `$ docker logs tenablesc-labbox-ol8 -f`

Troubleshooting

The Labbox license is expired

Download the latest copy of labbox and replace the `license.key` file in your utils directory with the one in the download. Then start, or restart, Labbox. You should see a note in the logs that Labbox has detected a new license and updated it accordingly.

Then run the following (ensure Labbox is running) to update the activation codes and update plugins.
`$ docker exec tenablesc-labbox-ol8 /usr/bin/labbox config_update tsc`

/usr/lib64/python3.6/tarfile.py:2251: RuntimeWarning: The default behavior of tarfile extraction has been changed to disallow common exploits (including CVE-2007-4559). By default, absolute/parent paths are disallowed and some mode bits are cleared. See https://access.redhat.com/articles/7004769 for more details.

This may be safely ignored; it will be resolved in a future version of Labbox.

docker.errors.DockerException: Error while fetching server API version: ('Connection aborted.', ConnectionRefusedError(61, 'Connection refused')) [82616] Failed to execute script docker-compose

Docker isn't running. Ensure that you see Docker in the menu bar and when you click on it you see a green circle and "Docker Desktop is running"

Docker Fatal Error; Docker daemon failed to start; sending kill signal 15 to container

This usually means that Docker ran out of memory. Ensure you've allocated at least 4GB of RAM to docker and review your active memory usage. Do not try to run Labbox-VirutalBox and Labbox-Docker at the same time on a standard laptop; you will have a bad experience.

The tenablesc-labbox logs end at "Decrypting Package..." or "Downloading Package"

See Docker Fatal Error; Docker daemon failed to start; sending kill signal 15 to container

Tenable.sc opens when the build is completed, but redirects you to install a license.

See Docker Fatal Error; Docker daemon failed to start; sending kill signal 15 to container

Trending charts do not load and errors are seen in the logs saying "Error opening file for buffer data... namedb.db (msg: No such file or directory).

Follow the steps outlined here after accessing the docker container directly using the steps above under Custom Actions/Commands.

The image build fails with "Cannot retrieve metalink for repository: epel/x86_64. Please verify its path and try again"

Re-run the image creation command and try and re-create the image. Note that the image build process requires internet (but not VPN) access.

If you get this repeatedly and are on Windows, you may have to uncomment "#RUN /usr/bin/echo "ip_resolve=4" >> /etc/yum.conf" in the Dockerfile to get the base image to build properly. This forces yum to use IPv4 to pull updates.

The container start fails with "ERROR: for tenablesc Cannot create container for service tenablesc: Conflict. The container name "/tenablesc-labbox" is already in use by container"

In general, this means that you're trying to spin up another instance of labbox for docker, but you already have one that exists. This is not expressly supported today. The workaround is to either launch the old instance of your container or delete and re-create a new instance. Also see Multiple Instances of Labbox for Docker.

Gathering logs

Mac

Open Terminal and change directories to your labbox folder. Run the following command:

```
sh utils/mac_debug.sh
```

Windows or Linux

```
$ docker logs tenablesc-labbox-ol8 \> labbox.log
```

Starting Over

The following commands will stop the the running instance, if up, and delete all remnants of labbox, allowing you to start from scratch. This should be run from the labbox directory.

```
$ docker-compose stop; docker rm tenablesc-labbox -f; docker volume rm tenablescdata -f; docker volume
rm tenableyumcache -f; docker rmi tenablesc-labbox-image -f
```

Examples of successful (and expected) command output

Building the Labbox Docker Image (perform once)

```
[+] Building 207.8s (12/12) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 2.29kB
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/centos:7.9.2009
 => [auth] library/centos:pull token for registry-1.docker.io
 => [internal] load build context
 => => transferring context: 1.45kB
 => CACHED [compile-image 1/3] FROM docker.io/library/centos:7.9.2009@sha256:be65f488b7764ad3638f236b7
 => => resolve docker.io/library/centos:7.9.2009@sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b
 => [compile-image 2/3] COPY utils/python_setup.sh /root/python_setup.sh
 => [compile-image 3/3] RUN /usr/bin/chmod +x /root/python_setup.sh         &&     /usr/bin/yum -y ins
 => [build-image 2/4] COPY --from=compile-image /root /root
 => [build-image 3/4] COPY utils/start.sh /
 => [build-image 4/4] RUN /usr/bin/yum -y install epel-release
 => exporting to image
 => => exporting layers
 => => writing image sha256:ab44aaded3d0fd8387a2c42f79cddb168bf5bf21884f42b346e0edee62c33ee9
 => => naming to docker.io/library/tenablesc-labbox
```

Creating the Labbox Container; based on the Labbox Image (perform once)

```
----------------------------
Labbox (Tenable.sc) Starting
----------------------------
Image v1.1 built on Fri Jun  3 15:11:11 UTC 2022
Container started on Fri Dec 30 15:06:49 UTC 2022
----------------------------
Container files at 20221230
Updated the Labbox Binary with c1191fbdc3b6f83ccd7d946d5818938c  /labbox/labbox.py
Updated the Tenable Repo with 8678570ecaf86213b9038c2e70817adf  /labbox/Tenable.repo
Updated the Tenable Repo Key with 27ef15ecc6e422effda42c0732f9b2fe  /labbox/RPM-GPG-KEY-Tenable
Updated the Tenable.sc license with aab0a003e30572ce1c564286ab3b0517  /labbox/license.key and told Ten
```

```
Updated the supervisord conf with f42fbac6cc12c3f3a090ce40e0830464   /labbox/supervisord.conf
Updated the running conf with d1d2904cdea6a3c294b9ff719b512afd   /labbox/running.sh
Change to yum config, re-making cache.
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: mirror.math.princeton.edu
 * epel: mirror.xenyth.net
 * extras: mirror.dal10.us.leaseweb.net
 * updates: mnvoip.mm.fcix.net
Metadata Cache Created
Installing new Tenable.sc console...
warning: /var/tmp/rpm-tmp.zbNawd: Header V3 RSA/SHA256 Signature, key ID 1c0c4a5d: NOKEY
Retrieving https://appliance.cloud.tenable.com/repos/7/tenable/applications/x86_64/TNSRPMS/SecurityCen
Preparing...                          ########################################
Updating / installing...
SecurityCenter-5.23.1-5.el7           ########################################


Installing Nessus plugins ... complete
Java version 1.8 or higher is required for Reporting to function properly.
/var/tmp/rpm-tmp.pthQF9: line 802: /sbin/ip: No such file or directory

Running SecurityCenter Feed update process with included package ... complete
/var/tmp/rpm-tmp.pthQF9: line 14: service: command not found
Starting Tenable.sc services...
Enforcing Labbox Tenable.sc Config...
2022-12-30 15:12:21,782 INFO Set uid to user 0 succeeded
2022-12-30 15:12:21,812 INFO RPC interface 'supervisor' initialized
2022-12-30 15:12:21,812 INFO supervisord started with pid 935
2022-12-30 15:12:22,816 INFO spawned: 'Apache' with pid 939
2022-12-30 15:12:22,818 INFO spawned: 'Jobd' with pid 940
2022-12-30 15:12:25,286 INFO success: Apache entered RUNNING state, process has stayed up for > than 2
2022-12-30 15:12:25,287 INFO success: Jobd entered RUNNING state, process has stayed up for > than 2 s
Starting a complete rebuild process...
Making sure internal components are updated...
pip is updated.
pyTenable is updated.
Syncing all of Labbox...
Checking & applying if necessary any Tenable.sc updates...
Stopping Tenable.sc to prepare for data merge...
2022-12-30 15:13:35,416 INFO waiting for Apache to stop
2022-12-30 15:13:35,417 INFO waiting for Jobd to stop
2022-12-30 15:13:35,417 INFO stopped: Apache (terminated by SIGTERM)
2022-12-30 15:13:35,435 INFO stopped: Jobd (exit status 0)
Attempting to download and overlay sc-repos package...
        - Downloading demo-sc-repos-all.tar.gz.enc...
        - Decrypting Package...
        - Decryption Successful, Package OK
        - Unpacking Package...
        - Package unpacked onto filesystem.
        - Package file removed.
Attempting to download and overlay sc-data package...
        - Downloading demo-sc-data-all.tar.gz.enc...
        - Decrypting Package...
        - Decryption Successful, Package OK
        - Unpacking Package...
        - Package unpacked onto filesystem.
        - Package file removed.
Reverting ownership to whats expected...
Restarting Tenable.sc services ...
2022-12-30 15:22:19,488 INFO spawned: 'Apache' with pid 1310
2022-12-30 15:22:19,490 INFO spawned: 'Jobd' with pid 1311
```

```
2022-12-30 15:22:21,739 INFO success: Apache entered RUNNING state, process has stayed up for > than 2
2022-12-30 15:22:21,740 INFO success: Jobd entered RUNNING state, process has stayed up for > than 2 s
Overriding user passwords [setting to "password"]...
2022-12-30 15:22:32,469 INFO exited: Jobd (exit status 0; expected)
Deleting links to existing Scanners, PVS Sensors, and LCEs...
Setting passive data expiration to 365 days...
Allowing users to manage objects (workaround to dashboard issues)...
        - Allowing admin to manage all objects.
Synchronization Complete.
Enforcing Labbox Tenable.sc Config...
Finished rebuild.
Looking for any Dashboard gaps and fixing them...
Completed rebuild process at Fri Dec 30 15:22:39 UTC 2022.
Tenable.sc needs to be restarted.
2022-12-30 15:22:39,215 INFO stopped: Apache (terminated by SIGTERM)
TenableSC:Apache: stopped
Enforcing Labbox Tenable.sc Config...
2022-12-30 15:22:39,320 INFO spawned: 'Apache' with pid 1511
2022-12-30 15:22:39,325 INFO spawned: 'Jobd' with pid 1512
2022-12-30 15:22:41,380 INFO success: Apache entered RUNNING state, process has stayed up for > than 2
2022-12-30 15:22:41,380 INFO success: Jobd entered RUNNING state, process has stayed up for > than 2 s
TenableSC:Apache: started
TenableSC:Jobd: started
Tenable.sc is up and running, available at https://127.0.0.1:8443.
```

Normal Labbox Container Start

```
----------------------------
Labbox (Tenable.sc) Starting
----------------------------
Image v1.1 built on Fri Jun  3 15:11:11 UTC 2022
Container started on Thu Dec 29 17:44:53 UTC 2022
----------------------------
Container files at 20221222
Updated the Labbox Binary with c1191fbdc3b6f83ccd7d946d5818938c  /labbox/labbox.py
No updates to Tenable.repo detected.
No updates to repo key detected.
Updated the Tenable.sc license with aab0a003e30572ce1c564286ab3b0517  /labbox/license.key and told Ten
No updates to supervisor conf detected.
No updates to running config detected.
Enforcing Labbox Tenable.sc Config...
Starting Tenable.sc Services...
2022-12-29 17:44:53,610 INFO Set uid to user 0 succeeded
2022-12-29 17:44:53,638 INFO RPC interface 'supervisor' initialized
2022-12-29 17:44:53,638 INFO supervisord started with pid 58
2022-12-29 17:44:54,642 INFO spawned: 'Apache' with pid 62
2022-12-29 17:44:54,645 INFO spawned: 'Jobd' with pid 63
2022-12-29 17:44:56,709 INFO success: Apache entered RUNNING state, process has stayed up for > than 2
2022-12-29 17:44:56,709 INFO success: Jobd entered RUNNING state, process has stayed up for > than 2 s
Making sure internal components are updated...
pip is updated.
pyTenable is updated.
Looking for any Dashboard gaps and fixing them...
Docker Labbox started at Thu Dec 29 17:45:00 UTC 2022.
```

Clean Up

If you need to remove everything discussed in here:

```
$ docker-compose stop
$ docker rm tenablesc-labbox-ol8
$ docker volume rm tenablescdata-ol8
$ docker volume rm tenableyumcache-ol8
```

```
$ docker network rm docker_labbox_default-ol8
$ docker rmi tenablesc-labbox-image-ol8
```