

v2.5 Upgrade Instructions from v2.4

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

Summary

When selecting which Grouper v2.5 container to use (which build number), review the [release notes wiki](#). You should install the latest stable v2.5.* release (v2.5.43 as of 2021/02/24). When you do a minor build update in the future, look at this wiki to verify the stability of the version

v2.5 is a minor upgrade from the latest v2.4 container. Some defaults have changed in the properties files, and the [container layout has drastically changed](#) but it should be easy to adjust your docker file.

If you use v2.4 not in a container, then you will have to start using the container. You don't need orchestration or a container practice in your organization, you can still use the same server you use now, just install docker and use the [maturity level 0 advice](#) to run Grouper. This should not be a barrier to running Grouper. If you are forbidden from running a container, at your institution and still want v2.5, it is possible to install docker, get the container, copy files out, and remove docker (sounds painful right? hope you don't have to do that 😊).

If you are in v2.2.1+, then it is similar to v2.4 not in a container. The DDL upgrade to 2.5 can run automatically from v2.2.1, but you should follow the "[v2.4 Upgrade Instructions from v2.3](#)" for everything except DDL. (and "[v2.3 Upgrade from v2.2](#)" if applicable) (Note: you need Grouper v2.5.36+ if you are in 2.2.1)

If you are in 2.2.0 or before, you need to upgrade to v2.2.1 before upgrading to v2.5 (or notify the Grouper team for advice)

There are a lot of specifics here based on where you are in Grouper, this document will attempt to unravel that.

Upgrade from v2.4 to v2.5

- If you don't have a morphString.properties, add one to the classpath (e.g. /opt/grouper/conf) and put a random alphanumeric upper/lower 16 char secret in there for all JVMs (UI/WS/daemon/GSH)
- If you are not using [configuration in the database](#), you should migrate to that
 - You have various envs, dev, test, prod. In one env, you have various JVMs: UI, WS, daemon, GSH. Multiply that out and you have a lot of config files.
 - With configuration in the database you don't need those config files and the configuration is in the database and editable from the UI (no need to deploy your container). So all the JVMs (UI, WS, daemon, GSH) in one env (dev, test, prod) will automatically be consistent.
 - The config files not in the database are: grouper.hibernate.properties, morphString.properties, grouper.text.en.us.properties
 - Take an env (e.g. dev) and look at all the config files of each type (e.g. grouper-loader.properties), consolidate those, and import into the UI, and don't provide that config file anymore
 - This will make config file changes easier (and mostly runtime), you will not have issues when different JVMs are inconsistent (don't need to copy config file to multiple containers)
- Database upgrade. There are a few low risk database changes (a few new tables, views, indexes). Grouper v2.4 will run against a v2.5 registry fyi. Grouper v2.3 should also. (disabled groups in v2.5 could be enabled in v2.4)
 - If Grouper is running the DDL automatically, or you run it from gsh manually, or you run the script in your DB UI tool or whatever, if it fails part-way through, you need to grab the rest of the DDL scripts (from WEB-INF/ddlScripts) and run the rest manually. Grouper will not be able to start where it left off and you need to fix it.
 - Views grouper_groups_v or grouper_roles_v will be changed. Oracle and mysql will replace those views, postgres will drop and create. If you use Postgres, see if there are any grants to those views and recreate them after DDL upgrade. For all three see if views or objects select from those views and make sure everything is intact afterwards (keep source of objects that use grouper objects and keep grants of grouper objects)
 - [Grouper DDL auto-upgrade](#). It is recommended to set this in grouper.hibernate.properties to auto-upgrade the database. will work from v2.3+ to v2.5 and will auto upgrade from here on in. Note, your database username that grouper uses needs to be permitted to make DDL changes in its database. You might need to get the DBAs to adjust that user. If you set this in grouper.hibernate.properties, turn on the container, and it will upgrade the database automatically. Any future v2.5 DDL will be backwards compatible with all v2.5.* containers
 - If you want the legacy DDL of manual updates, then turn on the container, and run "gsh -registry -check" and review and run that script. [Some examples for various databases are here](#). Compare the generated script with one of these scripts and run against your database. Note: each time you update your container you should check the release notes page about DDL requirements. We will be changing DDL with various 2.5 builds periodically. Auto-DDL is strongly recommended.
- grouper.base.properties: security.show.folders.where.user.can.see.subobjects = false by default. This is the recommended setting. It means everyone can see all folders whether they have objects inside or not. If you want the old default behavior, set that in grouper.properties
- grouper.base.properties. Do you have the rule that [vetoes assignments in folder if subject not in group](#)? The default in v2.5 (different than v2.4) is to enforce that by change log and daemon. This is recommended and you probably want this. But it could remove assignments when you turn Grouper on. Which is probably what you want
- Tomcat basic auth and apache basic auth can be replaced. Do you use tomcat-users.xml or apache user file? You should switch to [Grouper basic auth](#) (note you don't have to switch).
- Custom Java
 - You should check to see if your Java still compiles until 2.5. It should, but check anyways. Tweak it if you need to or ask for advice on slack. You might want to rebuild anyways.
 - Note that the daemon runs in tomee now, so calls like ClassLoader.getResource... will not work
- Container changes

- For your overlays of existing files, look at the new container files, and make sure that the changes you made do not overwrite other things in the file. e.g. server.xml, grouper-www.conf, web.xml, etc
- There is no more /opt/tomcat. It is /opt/tomee now. It uses Tomcat 8.5 so things should generally be the same, but if you were overlaying files into /opt/tomcat, then you should redo those changes for tomee (diff your overlay with tomee, and make sure you are only changing your changes, not introducing other changes from the old container)
- If you are doing [WS/UI authentication in tomcat \(e.g. ldap\)](#), you need to merge with the new server.xml and make sure the connector tomcatAuthentication is true (defaults to false now). Also make sure the web.xml is right
- There is only one webapp now, not one webapp for UI/WS/SCIM
- There is no longer a command line daemon
- If you ran v2.4 in a container, then you will need to adjust your mounts and Dockerfile
- The path to Grouper is: /opt/grouper/grouperWebapp
- If used this previous path: /opt/grouper/conf, change to /opt/grouper/grouperWebapp/WEB-INF/classes
- If used this previous path: /opt/grouper/lib, it will not work. If the jar is for the UI/daemon/GSH. e.g. a new change log consumer, use /opt/grouper/grouperWebapp/WEB-INF/libUiAndDaemon. If it is a driver that should be in those things and WS and SCIM, then put it in /opt/grouper/grouperWebapp/WEB-INF/lib instead
- Note: there is no Oracle driver anymore in the container (unless you used installer and agreed to Oracle terms). You need to download the oracle driver and put in /opt/grouper/grouperWebapp/WEB-INF/lib. Might want to use this: <https://repo1.maven.org/maven2/com/oracle/ojdbc8/ojdbc8/19.3.0.0/ojdbc8-19.3.0.0.jar>
- If you used /opt/grouper/grouper.ui, grouper.ws, grouper.apiBinary, grouper.scim, you need to adjust those. There is one webapp dir in /opt/grouper/grouperWebapp
- It is recommended to just put files here: /opt/grouper/slashRoot especially for mounting (will copy the structure to the root dir / in the container)
 - If you do not have a Dockerfile, you only need one mount path to the container
 - Example for classpath file: /opt/grouper/conf/grouper.hibernate.properties or /opt/grouper/slashRoot/opt/grouper/conf/grouper.hibernate.properties
- All things run in tomee (not daemon command line anymore). So this is how to set memory for all envs. Note, it used to be different for daemon envs, so adjust those accordingly. Daemon should have 12gigs at least

```
ENV GROUPER_MAX_MEMORY="3g"
```

Test the memory setting in all your containers:

```
# ps -ef | grep tom      (get pid)
# sudo -u tomcat jmap -heap <pid>      (see max heap, should be approx what you expect)
```

- If you copy files into the container, you should end your (Dockerfile or whatever) script by setting the owner of the webapp dir

```
RUN chown -R tomcat:tomcat /opt/grouper/grouperWebapp
```

- vt-ldap is no longer supported. [Make sure you are not using it in grouper-loader.properties](#)

This gets you to v2.5.X. Now look at the [v2.5.X upgrade steps and see which ones apply to you](#)

See Also

[Release Notes for Grouper 2.5](#)