

# Install the Grouper container with maturity level 1

<a href="#">Wiki Home</a>	<a href="#">Grouper Release Announcements</a>	<a href="#">Grouper Guides</a>	<a href="#">Grouper Deployment Guide</a>	<a href="#">Community Contributions</a>	<a href="#">Internal Developer Resources</a>
---------------------------	---	--------------------------------	--	---	--

- See the [specs sheet](#)

Maturity level 1 leads you toward how to do container orchestration properly. With maturity level 0, you are using the container "as is" but relying on external "mounts" which means you have configs outside of the container that depend on the host that runs the container. In this model you bake all of that into the container and it is a more stand-alone package that will be more reliable and consistent (i.e. the state of the configs are in the immutable container).

## Get a server and database

[Here is an example with AWS](#), basically for this example you need a Unix-based server (or Mac), and a postgres (recommended), or mysql or oracle database. Install Docker as well

## Install docker

See the [maturity level 0 document](#) for installing docker, and basic commands of working with images and containers.

## Install the container

1. [See which version of Grouper to run](#)
2. Pull the image

```
bin $ docker pull i2incommon/grouper:2.5.XX
```

3. Make sure the digest is correct (from release notes page)

```
[root@ip-172-30-3-152 ~]# docker image inspect i2incommon/grouper:2.5.XX | grep i2incommon/grouper@sha256
    "i2incommon/grouper@sha256:b675bb410bf873483497b9b231e7a5db208645e58a3a42a8048381a33b79fd19"
```

4. Create a directory to hold files to put in your subcontainer. You might have one of these directories that is shared for ws/ui/daemon.

```
2.5 $ mkdir -p /opt/grouperContainer
2.5 $ mkdir -p /opt/grouperContainer/slashRoot/opt/grouper/grouperWebapp/WEB-INF/classes
```

5. Set grouper.hibernate.properties. Note, for DB URL, "localhost" is the container itself, not the enclosing server. You need to use an IP address that the container can communicate with. Look in the grouper.hibernate.properties for documentation on setting up the url.

```

2.5 $ vi /opt/grouperContainer/slashRoot/opt/grouper/grouperWebapp/WEB-INF/classes/grouper.hibernate.
properties

hibernate.connection.url = jdbc:mysql://192.168.86.71:3306/grouper_v2_5?useSSL=false

hibernate.connection.username          = grouper_v2_5

hibernate.connection.password         = *****

# what version should we auto install DDL up to.  You should put the major and minor version here (e.g.
2.5.*).  Or you could go to a build number if you like,
# or nothing to not auto DDL.  e.g. 2.5.32      or      2.5.*
# {valueType: "string"}
registry.auto.ddl.upToVersion = 2.5.*

# UI basic auth is for quick start. Set to false when you migrate to shib or something else
grouper.is.ui.basicAuthn=true
grouper.is.ws.basicAuthn=true
grouper.is.scim.basicAuthn = true

```

6. If you cant connect to the database, go in the container (instructions later 😊) and test the communication with telnet

```

grouperContainer $ docker exec -it grouper-daemon /bin/bash
[root@0d9054515bed WEB-INF]# yum install telnet
[root@0d9054515bed WEB-INF]# telnet database-2.cstlzkqw179p.us-east-1.rds.amazonaws.com 3306
Trying 172.30.3.40...
Connected to database-2.cstlzkqw179p.us-east-1.rds.amazonaws.com.
Escape character is '^]'.
X
5.5.5-10.4.8-MariaDBK&I~bLp8pOz8H?EzW(\mysql_native_password^CConnection closed by foreign host.
[root@0d9054515bed WEB-INF]#

```

7. The container contains jdbc drivers for hsql, msyql and postgres. If you're using Oracle, you'll need to add the jar.  
Might want to use: [https://raw.githubusercontent.com/Internet2/grouper/GROUPER\\_2\\_4\\_BRANCH/grouper/lib/jdbcSamples/ojdbc6\\_g.jar](https://raw.githubusercontent.com/Internet2/grouper/GROUPER_2_4_BRANCH/grouper/lib/jdbcSamples/ojdbc6_g.jar)  
Might want to use: <https://repo1.maven.org/maven2/com/oracle/ojdbc/ojdbc8/19.3.0.0/ojdbc8-19.3.0.0.jar>

```

2.5 $ ls -al /opt/grouperContainer/slashRoot/opt/grouper/grouperWebapp/WEB-INF/lib/ojdbc6_g.jar

```

8. Set morphString.properties unique key for encryption

```

2.5 $ vi /opt/grouperContainer/slashRoot/opt/grouper/grouperWebapp/WEB-INF/classes/morphString.properties
# random 16 char alphanumeric upper/lower
encrypt.key = *****

```

9. Decide how many containers

Strategy	Containers	Notes
SEPARATE-CONTAINERS	ui ws daemon scim	More like a production env Uses more memory Can control, bring up down, configure each separately Need to manage ports. Generally 443 for UI, 8443 for WS, 8444 for Scim
ALL-IN-ONE	all	Runs everything in one container. Don't do this in prod Uses less memory When anything is up or down all is up or down Can use 443 for UI, WS, Scim
UI-WS	ui-ws daemon	This is not documented here. Don't do this in prod You can have a hybrid and put whatever components in whatever containers you want

10. Assume logs go to docker. If you want to mount external logs, follow directions from [maturity level 0](#)
11. Allow grouper db config from all. You can decide if you trust your authn and mfa if you want to leave this open, or lock it down to your vpn or whatever.

```
2.5 $ vi /opt/grouperContainer/slashRoot/opt/grouper/grouperWebapp/WEB-INF/classes/grouper-ui.properties

grouperUi.configurationEditor.sourceIpAddresses = 0.0.0.0/0
```

12. Make a Dockerfile and subcontainer

```
slashRoot $ vi /opt/grouperContainer/Dockerfile

# this matches the version you decided on from release notes
ARG GROUPER_VERSION=2.5.XX

FROM i2incommon/grouper:${GROUPER_VERSION}

# this will overlay all the files from /opt/grouperContainer/slashRoot on to /
COPY slashRoot /

RUN chown -R tomcat:root /opt/grouper \
    && chown -R tomcat:root /opt/tomee
```

13. Make container. Note you could have one subcontainer (recommended if possible), and deploy that to UI/WS/daemon (either ALL-IN-ONE or SEPARATE-CONTAINERS)

```
grouperContainer $ docker build -t my-grouper-2.5.XX /opt/grouperContainer
Sending build context to Docker daemon 216.1kB
Step 1/2 : FROM i2incommon/grouper:2.5.XX
--> 04ced0374ad5
--> Running in 7bd1a51c3552
Removing intermediate container 7bd1a51c3552
--> ff79b4b2afb9
Successfully built ff79b4b2afb9
Successfully tagged my-grouper-2.5.XX:latest
```

14. See [maturity level 0 for Docker run command](#) (approx step 15), make your shell script(s). Note, you do not need mounts. e.g. for ui

```
docker run --detach --restart always -e RUN_SHIB_SP='false' \
-e SELF_SIGNED_CERT='true' --name grouper-ui --publish 443:443 my-grouper-2.5.XX:latest ui
```

15. Setup the database run grouper