Install the Grouper container with maturity level 0

- See the specsheet

Get a server and database

Here is an example with AWS for database and server, 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

Here is an example of using a postgres database on docker

Install the container

1. Install docker (note, using a server with systemd is easier)
2. See if docker is running

```
bin $ docker info
Client:
 Debug Mode: false

Server:
 Containers: 5
  Running: 0
  Paused: 0
```

3. List containers. Containers are instances of docker images, which is the product delivered from Grouper.

```
bin $ docker ps --all
CONTAINER ID        IMAGE                                COMMAND                  CREATED
STATUS
PORTS                                                                                NAMES
ca762df952a6        i2incommon/grouper 2.5.02              "/usr/local/bin/entr…"   9 months ago
Exited (137) 9 months
ago
grouper-ui
bin $
```

4. Remove unneeded containers if necessary

```
bin $ docker rm -f ca762df952a6
ca762df952a6
```

5. (Reference command) List images

```
[root@ip-172-30-3-152 ~]# docker image ls
REPOSITORY           TAG                 IMAGE ID            CREATED             SIZE
my-grouper-2.5.02    latest              918f5dd01bd5        6 hours ago         1.23GB
i2incommon/grouper   2.5.XX              f939770b7d91        34 hours ago        1.23GB
[root@ip-172-30-3-152 ~]#
```

6. (Reference command) Remove old images

```
[root@ip-172-30-3-152 ~]# docker rmi 918f5dd01bd5
[root@ip-172-30-3-152 ~]# docker rmi f939770b7d91
```

7. See which version of Grouper to run
8. Pull the image

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

9. 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"
```

10. Create a directory to mount files and folder in and out of container. 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
```

11. Create a local database. Preferred is postgres, but could be mysql or oracle too. (e.g. mysql, utf8, bin collation, create a user and password, and grant all to the new database from username and password)
12. 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
```

13. 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~bLþ8pOz8H?EzW(\mysql_native_password^CConnection closed by foreign host.
[root@0d9054515bed WEB-INF]#
```

14. 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
    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
```

15. 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 = ******************
```

16. 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

17. Configure logging

    There are a few strategies to the maturity level 0 logging. If you have all grouper services in one container for your host, create one log dir on the host:

```
2.5 $ mkdir -p /opt/grouperContainer/logs
2.5 $ chmod o+rwx /opt/grouperContainer/logs
```

    If you have multiple containers in a host, then make a log dir for each, and you will change the docker run command to "mount" the inner dir (/opt/grouper/logs) to the outer dir. The chmod is needed if docker runs as a different user. On macs you might not need the chmod.

```
(SEPARATE-CONTAINERS)
2.5 $ mkdir -p /opt/grouperContainer/logs/grouper-ws-logs
2.5 $ chmod o+rwx /opt/grouperContainer/logs/grouper-ws-logs
2.5 $ mkdir -p /opt/grouperContainer/logs/grouper-daemon-logs
2.5 $ chmod o+rwx /opt/grouperContainer/logs/grouper-daemon-logs
2.5 $ mkdir -p /opt/grouperContainer/logs/grouper-ui-logs
2.5 $ chmod o+rwx /opt/grouperContainer/logs/grouper-ui-logs
```

    Below you will see who writes to logs (docker user), and you can change the owner and permissions

    Make sure this env variable is provided (v2.5.26+):

```
-e GROUPER_LOG_TO_HOST=true
```

    This file in generally the same so you log to the same place (/opt/grouper/logs) in your container, just matters where you mount that outside

```
2.5 $ vi /opt/grouperContainer/slashRoot/opt/grouper/grouperWebapp/WEB-INF/classes/log4j.properties
## Log messages to stderr
log4j.appender.grouper_stderr = org.apache.log4j.ConsoleAppender
log4j.appender.grouper_stderr.Target = System.err
log4j.appender.grouper_stderr.layout = org.apache.log4j.PatternLayout
log4j.appender.grouper_stderr.layout.ConversionPattern = %d{ISO8601}: [%t] %-5p %C{1}.%M(%L) - %x - %m%n

## Grouper API error logging
log4j.appender.grouper_error = org.apache.log4j.DailyRollingFileAppender
log4j.appender.grouper_error.File = /opt/grouper/logs/grouper.log
log4j.appender.grouper_error.DatePattern = '.'yyyy-MM-dd
log4j.appender.grouper_error.MaxBackupIndex = 30
log4j.appender.grouper_error.layout = org.apache.log4j.PatternLayout
log4j.appender.grouper_error.layout.ConversionPattern = %d{ISO8601}: [%t] %-5p %C{1}.%M(%L) - %x - %m%n

log4j.appender.grouper_daemon = org.apache.log4j.DailyRollingFileAppender
log4j.appender.grouper_daemon.File = /opt/grouper/logs/grouperDaemon.log
log4j.appender.grouper_daemon.DatePattern = '.'yyyy-MM-dd
log4j.appender.grouper_daemon.MaxBackupIndex = 30
log4j.appender.grouper_daemon.layout = org.apache.log4j.PatternLayout
log4j.appender.grouper_daemon.layout.ConversionPattern = %d{ISO8601}: [%t] %-5p %C{1}.%M(%L) - %x - %m%n

log4j.appender.grouper_pspng = org.apache.log4j.DailyRollingFileAppender
log4j.appender.grouper_pspng.File = /opt/grouper/logs/pspng.log
log4j.appender.grouper_pspng.DatePattern = '.'yyyy-MM-dd
log4j.appender.grouper_pspng.MaxBackupIndex = 30
log4j.appender.grouper_pspng.layout = org.apache.log4j.PatternLayout
log4j.appender.grouper_pspng.layout.ConversionPattern = %d{ISO8601}: [%t] %-5p %C{1}.%M(%L) - %x - %m%n


# Loggers

## Default logger; will log *everything*
log4j.rootLogger = WARN, grouper_stderr, grouper_error

log4j.logger.edu = ERROR, grouper_stderr
log4j.logger.com = ERROR, grouper_stderr
log4j.logger.org = ERROR, grouper_stderr

log4j.logger.edu.internet2.middleware.grouper.app.loader.GrouperLoaderLog = DEBUG, grouper_daemon
log4j.additivity.edu.internet2.middleware.grouper.app.loader.GrouperLoaderLog = false

log4j.logger.edu.internet2.middleware.grouper.pspng = INFO, grouper_pspng
log4j.additivity.edu.internet2.middleware.grouper.pspng = false
```

18. 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
```

19. Self-signed SSL (no need to do this anymore, this is now an env var instead)

```
-e GROUPER_SELF_SIGNED_CERT=true
```

    This is the old way:

```
slashRoot $ mkdir -p /opt/grouperContainer/slashRoot/etc/httpd/conf.d
slashRoot $ vi /opt/grouperContainer/slashRoot/etc/httpd/conf.d/ssl-enabled.conf

SSLProtocol             all -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite          ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-
POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-
ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256
SSLHonorCipherOrder     on
SSLCompression          off
# OCSP Stapling, only in httpd 2.3.3 and later
SSLUseStapling          on
SSLStaplingResponderTimeout 5
SSLStaplingReturnResponderErrors off
SSLStaplingCache        shmcb:/var/run/ocsp(128000)
Listen 443 https
<VirtualHost *:443>
  RewriteEngine on
  RewriteRule   "^/$"  "/grouper/"  [R]
  SSLEngine on
  #SSLCertificateChainFile /etc/pki/tls/certs/localhost.crt
  SSLCertificateFile /etc/pki/tls/certs/localhost.crt
  SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
  # HSTS (mod_headers is required) (15768000 seconds = 6 months)
  Header always set Strict-Transport-Security "max-age=15768000"
</VirtualHost>
```

20. Make a Dockerfile and subcontainer (this is maturity level 1)
21. Take out shib, adjust the proxy directives (this is now an env var)

This is the old way

```
slashRoot $ docker cp 058adff0568c:/etc/httpd/conf.d/grouper-www.conf /opt/grouperContainer/slashRoot/etc
/httpd/conf.d/grouper-www.conf
slashRoot $ vi /opt/grouperContainer/slashRoot/etc/httpd/conf.d/grouper-www.conf

Timeout 2400
ProxyTimeout 2400
ProxyBadHeader Ignore

ProxyPass /grouper ajp://localhost:8009/grouper  timeout=2400
ProxyPass /grouper-ws ajp://localhost:8009/grouper  timeout=2400
ProxyPass /grouper-ws-scim ajp://localhost:8009/grouper  timeout=2400

RewriteEngine on
RewriteCond %{REQUEST_URI} "^/$"
RewriteRule . %{REQUEST_SCHEME}://%{HTTP_HOST}/grouper/ [R=301,L]

#<Location /grouper>
#  AuthType shibboleth
#  ShibRequestSetting requireSession 1
#  ShibRequireSession on
#  ShibUseHeaders On
#  require shibboleth
#</Location>
```

22. We need to setup the docker run commands. There are bunch of choices here, e.g. are you running ui/ws/daemon in one container, or a container for each? This assumes the starting point on host is /opt/grouperContainer, but could be other places

Docker run option	Sample	Description
--detach		run in background
--mount for logs	--mount type=bind,src=/opt/grouperContainer/logs,dst=/opt/grouper/logs	mount logs if one grouper container on host
	--mount type=bind,src=/opt/grouperContainer/logs/grouper-ui-logs,dst=/opt/grouper/logs	mount ui logs if multiple grouper containers on host
	--mount type=bind,src=/opt/grouperContainer/logs/grouper-ws-logs,dst=/opt/grouper/logs	mount ws logs if multiple grouper containers on host
	--mount type=bind,src=/opt/grouperContainer/logs/grouper-daemon-logs,dst=/opt/grouper/logs	mount daemon logs if multiple grouper containers on host
--mount for overlays	--mount type=bind,src=/opt/grouperContainer/slashRoot,dst=/opt/grouper/slashRoot	this slashRoot directory will take any files and overlay into the container e.g. to overlay the grouper-www.conf file, put it in the host in /opt/grouperContainer/slashRoot/etc/httpd/conf.d/grouper-www.conf
--name	--name grouper-daemon --name grouper-ui --name grouper-ws	put a name label for the container that you can use to differentiate from other containers in your docker service. generally we use grouper-ui, grouper-daemon, grouper-ws you can use whatever you want
--hostname	--hostname myGrouperUi2	sets the hostname which is used in logs and audits
-e	-e GROUPER_SELF_SIGNED_CERT=true -e GROUPER_MAX_MEMORY='3g' -e GROUPER_RUN_SHIB_SP=false	add in environment variables documented here
timezone	-e TZ=America/New_York	might want to add a timezone
network settings	--net=host	depending on how your containers need to talk to each other, adjust the docker network settings. See Docker documentation
restart policies	--restart always	automatically start containers (e.g. after reboot). See Docker documentation for various options.
port mapping	--publish 443:443	listen on the host port and translate that to the container port generally you want apache 443
	--publish 80:80	if you want apache non-ssl (maybe external load balancer on trusted network?)
	--publish 8009:8009	if you want to publish the jk tomcat port for an apache outside of the container
	--publish 8443:443	8443 on outside if you have multiple containers on one host and 443 is already used
docker-image	i2incommon/grouper:2.5.XX	put the docker image label from grouper if you dont have a Dockerfile and subcontainer
	my-grouper-2.5.XX:latest	put the docker image label of you subcontainer if you made one from Dockerfile or grouper image
command	ui ws daemon scim <none> ui-ws quickstart	command documented here each for SEPARATE-CONTAINERS
	-e GROUPER_DAEMON='true ui-ws'	command to run ALL-IN-ONE container

23. (ALL-IN-ONE), you need one shell script. Use the table above to customize. The one below is just an example.

```
grouperContainer $ vi /opt/grouperContainer/grouperAllDockerRun.sh

#!/bin/bash  (or whatever shell)

docker run --detach --restart always --mount type=bind,src=/opt/grouperContainer/logs,dst=/opt/grouper
/logs \
--mount type=bind,src=/opt/grouperContainer/slashRoot,dst=/opt/grouper/slashRoot -e
GROUPER_RUN_SHIB_SP='false' \
-e GROUPER_SELF_SIGNED_CERT='true' -e GROUPER_DAEMON=true --name grouper-all --publish 443:443 i2incommon
/grouper:2.5.XX ui-ws

grouperContainer $ chmod +x /opt/grouperContainer/grouperAllDockerRun.sh
```

24. (SEPARATE-CONTAINERS) Make your UI docker run command in a shell script. Use the table above to customize. The one below is just an example.

```
grouperContainer $ vi /opt/grouperContainer/grouperUiDockerRun.sh

#!/bin/bash  (or whatever shell)

docker run --detach --restart always --mount type=bind,src=/opt/grouperContainer/logs/grouper-ui-logs,
dst=/opt/grouper/logs \
--mount type=bind,src=/opt/grouperContainer/slashRoot,dst=/opt/grouper/slashRoot -e
GROUPER_RUN_SHIB_SP='false' \
-e GROUPER_SELF_SIGNED_CERT='true' --name grouper-ui --publish 443:443 i2incommon/grouper:2.5.XX ui

grouperContainer $ chmod +x /opt/grouperContainer/grouperUiDockerRun.sh
```

25. (SEPARATE-CONTAINERS) Make your WS docker run command in a shell script. Use the table above to customize. The one below is just an example.
Note, since the UI and WS are on the same host, we use the host port of 8443 for web services. In production they will have different domain names and 443 so this is for dev only

```
2.5 $ vi /opt/grouperContainer/grouperWsDockerRun.sh

#!/bin/bash  (or whatever shell)

docker run --detach --restart always --mount type=bind,src=/opt/grouperContainer/logs/grouper-ws-logs,
dst=/opt/grouper/logs \
--mount type=bind,src=/opt/grouperContainer/slashRoot,dst=/opt/grouper/slashRoot -e
GROUPER_SELF_SIGNED_CERT='true' \
 --name grouper-ws --publish 8443:443 i2incommon/grouper:2.5.XX ws

2.5 $ chmod +x /opt/grouperContainer/grouperWsDockerRun.sh
```

26. (SEPARATE-CONTAINERS) Make your Daemon docker run command in a shell script. Use the table above to customize. The one below is just an example.

```
grouperContainer $ vi /opt/grouperContainer/grouperDaemonDockerRun.sh
#!/bin/bash  (or whatever shell)
docker run --detach --restart always --mount type=bind,src=/opt/grouperContainer/logs/grouper-daemon-
logs,dst=/opt/grouper/logs \
--mount type=bind,src=/opt/grouperContainer/slashRoot,dst=/opt/grouper/slashRoot --name grouper-daemon
i2incommon/grouper:2.5.XX daemon

grouperContainer $ chmod +x /opt/grouperContainer/grouperDaemonDockerRun.sh
```

27. (SEPARATE-CONTAINERS) Make your Scim docker container. Note, Scim is not a commonly used feature of Grouper. Until you need it you should not use it. Note, since the UI and Scim are on the same host, we use the host port of 8444 for scim. In production they will have different domain names and 443 so this is for dev only

```
2.5 $ vi /opt/grouperContainer/grouperScimDockerRun.sh

#!/bin/bash  (or whatever shell)

docker run --detach --restart always --mount type=bind,src=/opt/grouperContainer/logs,dst=/opt/grouper
/logs \
--mount type=bind,src=/opt/grouperContainer/slashRoot,dst=/opt/grouper/slashRoot -e
GROUPER_SELF_SIGNED_CERT='true' \
--name grouper-scim --publish 8444:443 i2incommon/grouper:2.5.XX scim

2.5 $ chmod +x /opt/grouperContainer/grouperScimDockerRun.sh
2.5 $ /opt/grouperContainer/grouperScimDockerRun.sh
2.5 $
```

28. Start container init database, note if not creating new container with Dockerfile, just use the container label i2incommon/grouper:2.5.XX

```
(SEPARATE-CONTAINERS)
grouperContainer $ /opt/grouperContainer/grouperDaemonDockerRun.sh
grouperContainer $ docker ps
CONTAINER ID                IMAGE
COMMAND                                  CREATED
STATUS                                PORTS
NAMES
a63006217009        my-grouper-2.5.XX:latest   "/usr/local/bin/entrâ€¦"   11 minutes ago       Up 10
minutes       80/tcp, 443/tcp                grouper-daemon
grouperContainer $

(ALL-IN-ONE)
grouperContainer $ /opt/grouperContainer/grouperAllDockerRun.sh
grouperContainer $ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a63006217009 my-grouper-2.5.XX:latest "/usr/local/bin/entrâ€¦" 11 minutes ago Up 10 minutes 80/tcp, 443
/tcp grouper-all
grouperContainer $
```

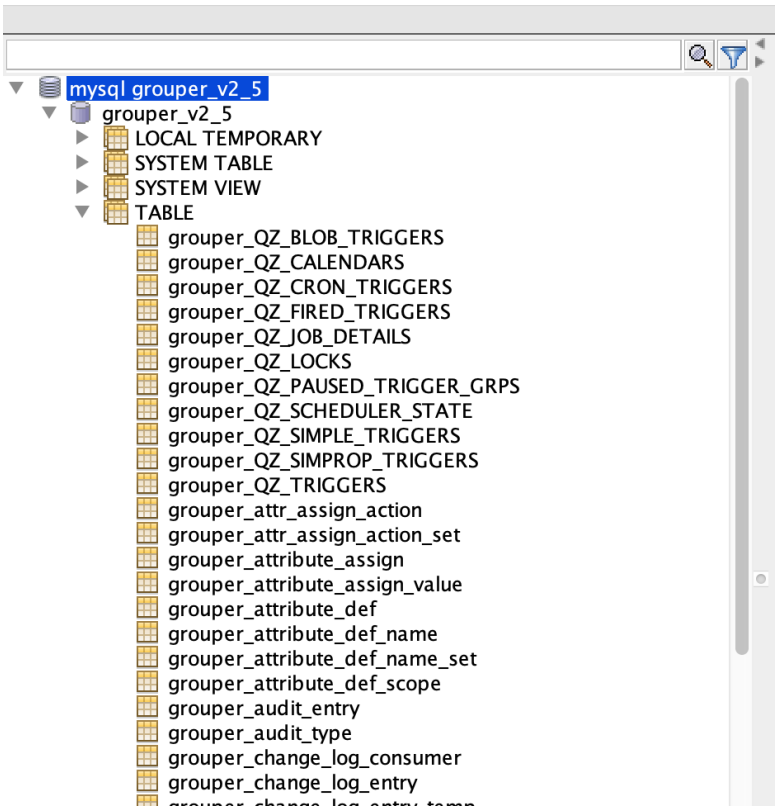
29. Set the owner and permissions of the log dir

```
after running container, see what user the docker container used to write logs to the host.  Find a log
file and see the owner/group.  Then change owner of logs folder to whatever the container writes as

(SEPARATE-CONTAINERS)
[root@ip-172-30-3-152 logs]# ls -latr /opt/grouperContainer/logs/grouper-daemon-logs/grouper.log
-rw-r-----. 1 polkitd systemd-coredump 0 Apr 8 05:56 grouper.log

(ALL-IN-ONE)
[root@ip-172-30-3-152 logs]# ls -latr /opt/grouperContainer/logs/grouper.log
-rw-r-----. 1 polkitd systemd-coredump 0 Apr 8 05:56 grouper.log
(BOTH)
[root@ip-172-30-3-152 logs]# chown -R polkitd.systemd-coredump /opt/grouperContainer/logs
[root@ip-172-30-3-152 logs]# chmod -R 755 /opt/grouperContainer/logs
[root@ip-172-30-3-152 logs]#
```

30. Check logs if there is a problem. Look for errors. Shell into the container to troubleshoot

```
(SEPARATE-CONTAINERS)
grouperContainer $ docker logs grouper-daemon | grep -i error
grouperContainer $ cat /opt/grouperContainer/logs/grouper-daemon-logs/grouper.log
(ALL-IN-ONE)
grouperContainer $ docker logs grouper-all | grep -i errorgrouperContainer $ cat /opt/grouperContainer
/logs/grouper.log


(BOTH)
grouperContainer $ docker exec -it grouper-daemon /bin/bash
grouperContainer $ ps -ef | grep tomee
```

31. Note: the database is initialized. See the tables in the database



32. If tables arent there, go in and run gsh

```
(SEPARATE-CONTAINERS)
grouperContainer $ docker exec -it grouper-daemon /bin/bash
[root@0d9054515bed WEB-INF]# cd /opt/grouper/grouperWebapp/WEB-INF/bin/
[root@0d9054515bed WEB-INF]# cd bin
[root@0d9054515bed bin]# ./gsh.sh -registry -check -runscript

(ALL-IN-ONE)
grouperContainer $ docker exec -it grouper-all /bin/bash
[root@0d9054515bed WEB-INF]# cd /opt/grouper/grouperWebapp/WEB-INF/bin/
[root@0d9054515bed bin]# ./gsh.sh -registry -check -runscript
```

33. Shell in there

```
(SEPARATE-CONTAINERS)
2.5 $ docker exec -it grouper-daemon /bin/bash
(ALL-IN-ONE)
2.5 $ docker exec -it grouper-all /bin/bash
```

34. Create passwords for UI. Change the four asterisks (****) with the password you want. You can change "GrouperSystem" with another subject that resolves

```
[root@d588628876f7 bin]# sudo -u tomcat bash
[tomcat @d588628876f7 bin]> cd /opt/grouper/grouperWebapp/WEB-INF/bin
[tomcat @d588628876f7 bin]> ./gsh.sh
gsh % new GrouperPasswordSave().assignApplication(GrouperPassword.Application.UI).assignUsername
("GrouperSystem").assignPassword("*****").save();
```

```
[root@d588628876f7 bin]# cd /opt/grouper/grouperWebapp/WEB-INF/bin
[root@d588628876f7 bin]# sudo -u tomcat bash
[tomcat @d588628876f7 bin]> vi createUiPass.gsh

grouperPasswordSave = new GrouperPasswordSave();
grouperPasswordSave.assignUsername("GrouperSystem").assignPassword("*****").assignEntityType("username");
grouperPasswordSave.assignApplication(GrouperPassword.Application.UI);
new Authentication().assignUserPassword(grouperPasswordSave);

[tomcat @d588628876f7 bin]> ./gsh.sh createUiPass.gsh
[tomcat @d588628876f7 bin]> rm createUiPass.gsh
```

35. Create a WS/SCIM username and password in container (you should already be in there). Change the four asterisks (****) with the password you want. You can change "GrouperSystem" with another subject that resolves

```
[root@d588628876f7 bin]# cd /opt/grouper/grouperWebapp/WEB-INF/bin
[root@d588628876f7 bin]# sudo -u tomcat bash
[tomcat @d588628876f7 bin]> ./gsh.sh
gsh % new GrouperPasswordSave().assignApplication(GrouperPassword.Application.WS).assignUsername
("GrouperSystem").assignPassword("*****").save();
```

```
[root@d588628876f7 bin]# sudo -u tomcat bash
[tomcat @d588628876f7 bin]> vi createWsPass.gsh

grouperPasswordSave = new GrouperPasswordSave();
grouperPasswordSave.assignUsername("GrouperSystem").assignPassword("*****").assignEntityType("username");
grouperPasswordSave.assignApplication(GrouperPassword.Application.WS);
new Authentication().assignUserPassword(grouperPasswordSave);

[tomcat @d588628876f7 bin]> ./gsh.sh createWsPass.gsh
```

36. (SEPARATE-CONTAINERS) Run the UI container

```
grouperContainer $ /opt/grouperContainer/grouperUiDockerRun.sh
```

37. The Grouper UI should now be accessible at https://whatever.server.edu:443/grouper and will prompt for credentials via basic auth.
38. (SEPARATE-CONTAINERS) Run the WS container

```
2.5 $ /opt/grouperContainer/grouperWsDockerRun.sh
2.5 $
```

39. (SEPARATE-CONTAINERS) Run the SCIM container (if using SCIM, you probably dont need this)

```
2.5 $ /opt/grouperContainer/grouperScimDockerRun.sh
2.5 $
```

40. (SEPARATE-CONTAINERS) If daemon not running (it probably is), run the daemon

```
2.5 $ /opt/grouperContainer/grouperDaemonDockerRun.sh
```

41. If you need to make changes to run command, or get a new image, you can remove a container. Change the "run" shell script, and run it again

```
grouperContainer $ docker ps --all
CONTAINER ID        IMAGE                      COMMAND                  CREATED
STATUS              PORTS                                                  NAMES
058adff0568c        my-grouper-2.5.XX:latest   "/usr/local/bin/entr…"   3 minutes ago       Up 3
minutes        80/tcp, 0.0.0.0:8080->8080/tcp, 0.0.0.0:8443->443/tcp   grouper-ui
grouperContainer $ docker rm -f 058adff0568c
058adff0568c
```

42. Add a subject source
43. Add Shibboleth
44. Add members to the wheel group