

Install the Grouper v2.4 container with maturity level 0

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

If you dont have a software that will run a container install it. Assume linux. Assume docker.

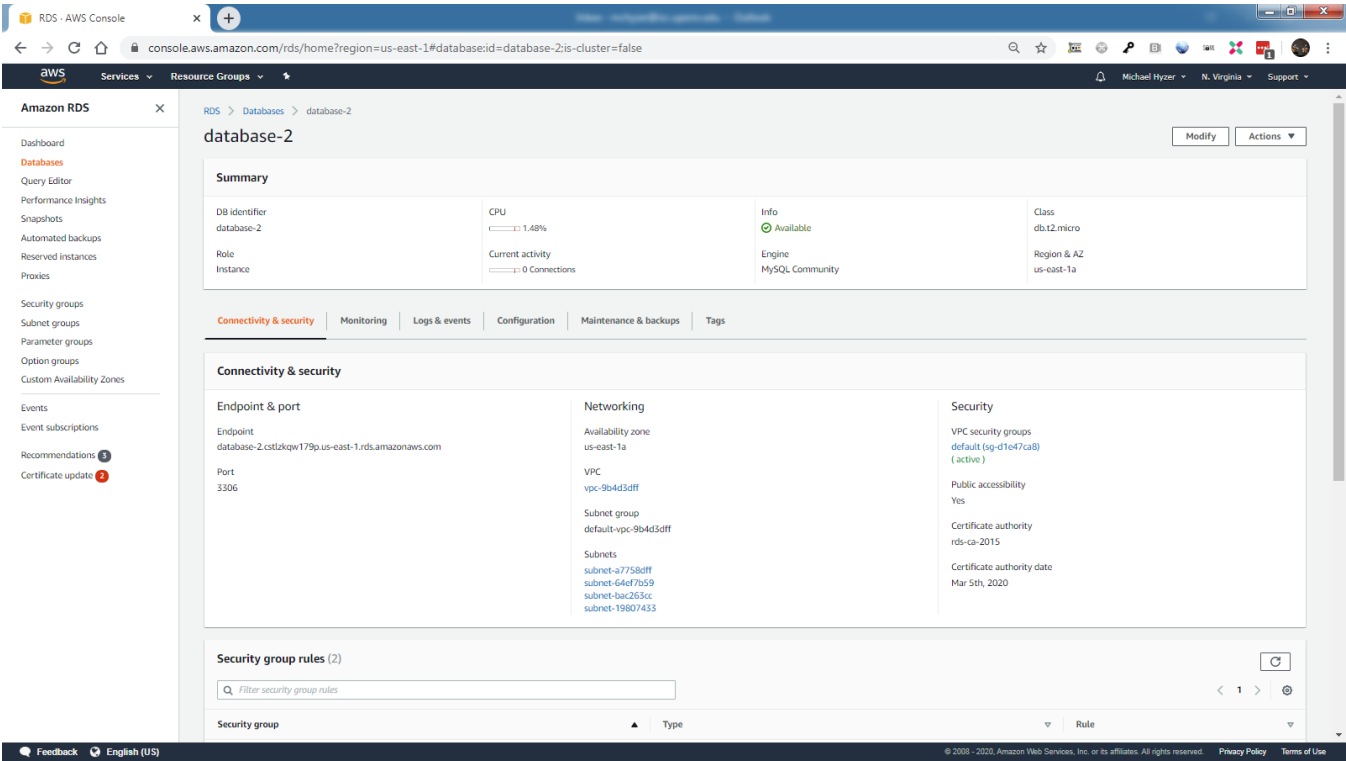
- [Get a database](#)
- [Create a database, user, and pass with sqlyog \(or whatever db client\)](#)
- [Latest i2-incommon container](#)
- [Conf and logs](#)
- [Init database](#)
- [UI](#)
- [Daemon](#)
- [WS](#)
- [Upgrade container \(or change version\)](#)

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

Get a database

You need an oracle, mysql, or postgres database with a user/pass where the user can do whatever they want in their schema (e.g. create tables)

Well, for this example, might as well go from scratch in aws, but this could be on prem or wherever. Create a free-tier database in AWS. Mysql RDS, publicly accessible.



RDS Create database Standard create MySQL Free tier T2 micro 20g publicly accessible

Allow from everywhere (just for poc)

Edit inbound rules

Type

Protocol

Port Range

Source

Description

MYSQL/Auror

TCP

3306

Custom

130.91.219.176/32

e.g. SSH for Admin Desktop

MYSQL/Auror

TCP

3306

Custom

0.0.0.0/0

e.g. SSH for Admin Desktop

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel

Save

Create a database, user, and pass with sqlyog (or whatever db client)

Mysql example

```
create database grouper_v2_5 character set UTF8 collate utf8_bin;
create user 'grouper_v2_5'@'localhost' identified by '*****';
grant all on grouper_v2_5.* to 'grouper_v2_5'@'localhost';
flush privileges;
```

Note: maybe your DBAs will do this for you

Connect to MySQL Host

New...

Save

Rename...

Delete

Saved Connections

aws_temp

MySQL

HTTP

SSH

SSL

Advanced

MySQL Host Address

e-2.cstlzkqw179p.us-east-1.rds.amazonaws.com

Username

admin

Password

.....

Save Password

Port

3306

Database(s)

(Separate multiple databases with a semicolon ';' . Leaving Database(s) blank will display all databases)

Use Compressed Protocol

?

Session Idle Timeout

?

Default

28800

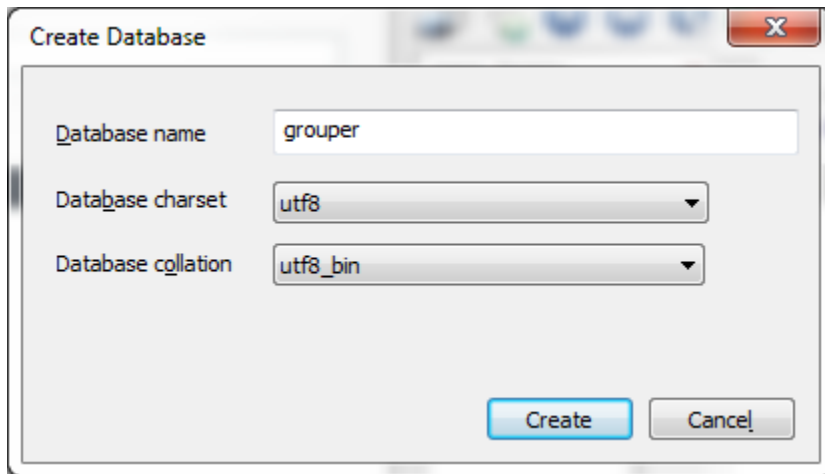
(seconds)

Connect

Cancel

Test Connection

Create database



A screenshot of a 'Create Database' dialog box. The dialog has a title bar with the text 'Create Database' and a close button (X). Inside the dialog, there are three labeled fields: 'Database name' with a text input containing 'grouper', 'Database charset' with a dropdown menu showing 'utf8', and 'Database collation' with a dropdown menu showing 'utf8_bin'. At the bottom right, there are two buttons: 'Create' and 'Cancel'.

Database name	grouper
Database charset	utf8
Database collation	utf8_bin

Create Cancel

Create user

User Manager

User: admin@%

Delete User

Add New User

Help

admin@%

Global Privileges

Object Level Privileges

+

information_schema

+

grouper

+

innodb

+

mysql

+

performance_schema

+

sys

Create a new user with limitations

Username

grouper

Host

%

Password

.....

Retype Password

.....

Maximum number of queries per hour

0

Maximum number of updates per hour

0

Maximum number of connections per hour

0

Maximum number of user connections

0

(Note - 0 means no limitations)

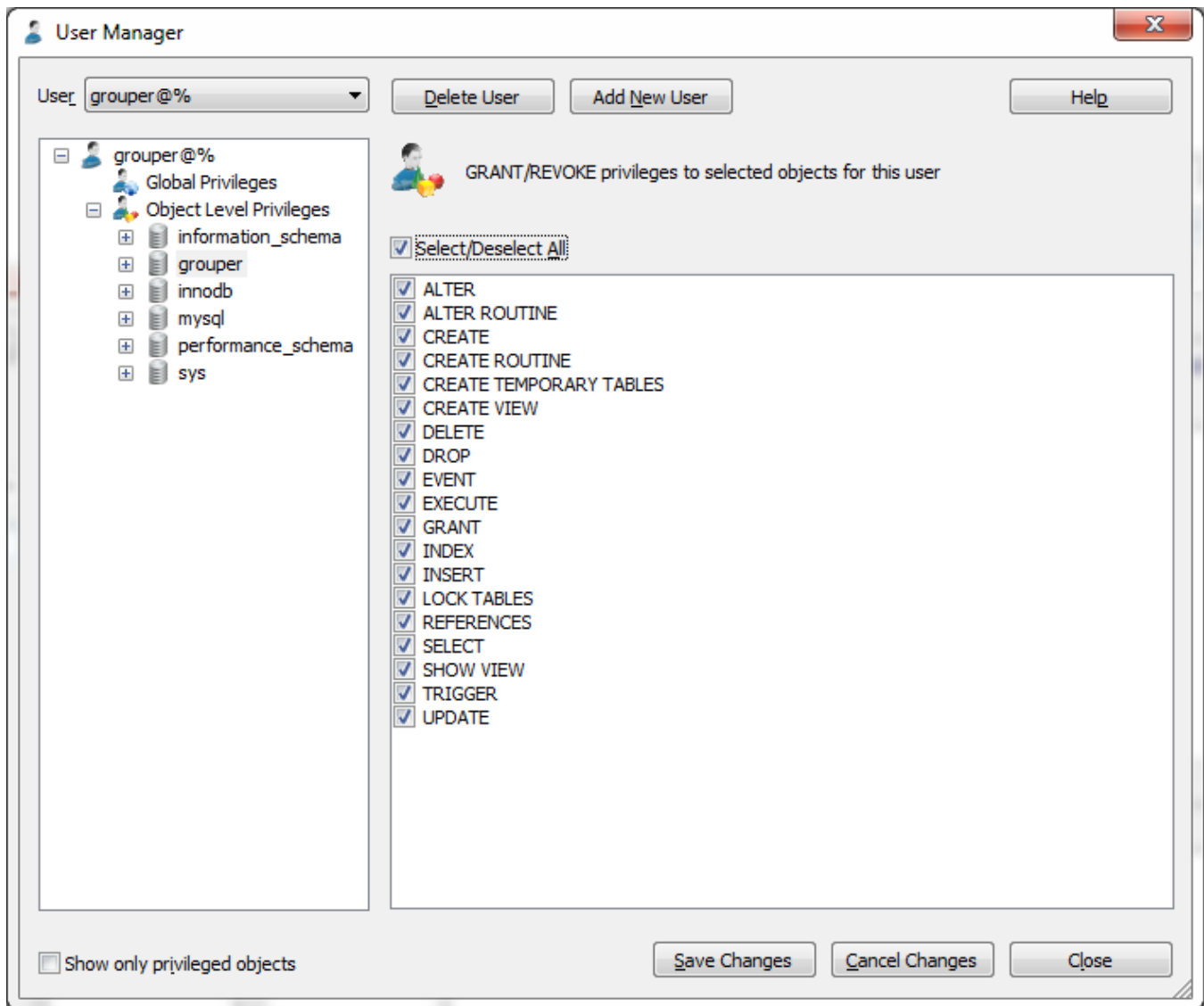
☐ Show only privileged objects

Create

Cancel

Close

Allow user to access database



Latest i2-incommon container

See the latest grouper tap container <https://hub.docker.com/r/tier/grouper/tags>

tier/grouper ☆

By tier • Updated a day ago

Docker Container for Grouper that can act as any Grouper component

Container

Overview Tags

Filter Tags

Sort by Latest

latest
Last updated a day ago by tieradmin

docker pull tier/grouper:latest

DIGEST
f71b0b4159be

OS/ARCH
linux/amd64

COMPRESSED SIZE
699.55 MB

2.4.0-a89-u55-w11-p12-20200110-rc1
Last updated a day ago by tieradmin

docker pull tier/grouper:2.4.0-a89-u55-w

DIGEST
1e5b93415e4c

OS/ARCH
linux/amd64

COMPRESSED SIZE
699.55 MB

Start and pull from docker

```
[root@ip-172-30-0-157 ~]# /sbin/service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-30-0-157 ~]# systemctl enable docker
[root@ip-172-30-0-157 ~]# docker pull tier/grouper:2.4.0-a89-u55-w11-p12-20200110-rc1
```

More instructions/docs:

<https://github.internet2.edu/docker/grouper>

Conf and logs

Basically we are identifying a folder on our server which is reserved for grouper. There are config files on the host machine which will be read from the container. There are log dirs which will be written to from the container. Later will map which dirs on the host machine connect to which dirs in the container. It is helpful if they are not the exact same path so you can keep them straight.

```
mkdir -p /opt/groupercontainer/conf
mkdir -p /opt/groupercontainer/conf/grouperText
touch /opt/groupercontainer/conf/grouperText/grouper.text.en.us.properties
mkdir -p /opt/groupercontainer/logs
mkdir -p /opt/groupercontainer/logs/grouper-ui
mkdir -p /opt/groupercontainer/logs/grouper-daemon
mkdir -p /opt/groupercontainer/logs/grouper-ws
chmod -R 777 /opt/groupercontainer/logs
mkdir -p /opt/groupercontainer/httpd
mkdir -p /opt/groupercontainer/tomcat
```

edit /opt/groupercontainer/conf/grouper.hibernate.properties (the conf dir is copied by the TAP container to the place where it is read by grouper)

```

# The grouper hibernate config uses Grouper Configuration Overlays (documented on wiki)
# By default the configuration is read from grouper.hibernate.base.properties
# (which should not be edited), and the grouper.hibernate.properties overlays
# the base settings. See the grouper.hibernate.base.properties for the possible
# settings that can be applied to the grouper.hibernate.properties

#####
## DB settings
#####

# e.g. mysql:          jdbc:mysql://localhost:3306/grouper
# e.g. p6spy (log sql): [use the URL that your DB requires]
# e.g. oracle:         jdbc:oracle:thin:@server.school.edu:1521:sid
# e.g. hsqldb (a):      jdbc:hsqldb:dist/run/grouper;create=true
# e.g. hsqldb (b):      jdbc:hsqldb:hsqldb://localhost:9001/grouper
# e.g. postgres:       jdbc:postgresql://localhost:5432/database
# e.g. mssql:          jdbc:sqlserver://localhost:3280;databaseName=grouper

hibernate.connection.url      = jdbc:mysql://database-2.cstlzkqw179p.us-east-1.rds.amazonaws.com:3306/grouper

hibernate.connection.username = grouper

hibernate.connection.password = *****

```

Edit /opt/groupercontainer/conf/morphString.properties (put random long alphanumeric string for encrypt key)

```

# Put a random alphanumeric string (Case sensitive) for the password encryption. e.g. fh43IRJ4Nf5
# or put a filename where the random alphanumeric string is. e.g. c:/whatever/key.txt
encrypt.key = *****

# set this to true if you have slashes in your passwords and dont want to look in external files or unencrypt
encrypt.disableExternalFileLookup = false

```

Edit /opt/groupercontainer/conf/log4j.properties (here we tell the contain to write to the dir on the host machine so logs persist if the container is destroyed)

#\${grouper.home} will be substituted with the System property "grouper.home", which must have a trailing \ or /
depending on your OS. Of course you can use absolute paths if you prefer

```
#
# log4j Configuration
# $Id: log4j.properties,v 1.4 2016/04/23 00:00:07 mchyzner Exp $
#

# Appenders

## 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/grouper/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
```

Take out shib, edit /opt/groupercontainer/httpd/grouper-www.conf (we took the file in the container (grouper-www.conf), removed the shib part, and will map it back into container to overlay


```

Timeout 2400
ProxyTimeout 2400
ProxyBadHeader Ignore

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

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

```

Make SSL work with self-signed: /opt/groupercontainer/httpd/ssl-enabled.conf (pointing to localhost self signed files)

```

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>

```

Enable tomcat basic auth: /opt/groupercontainer/tomcat/server.xml (we need to enable basic auth, copied the file from container, uncommented two sections, will overlay back)

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<!-- Note: A "Server" is not itself a "Container", so you may not
define subcomponents such as "Valves" at this level.
Documentation at /docs/config/server.html
-->
<Server port="8005" shutdown="SHUTDOWN">
    <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
    <!-- Security listener. Documentation at /docs/config/listeners.html
    <Listener className="org.apache.catalina.security.SecurityListener" />
-->

```

```

<!--APR library loader. Documentation at /docs/apr.html -->
<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
<!-- Prevent memory leaks due to use of particular java/javax APIs-->
<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

<!-- Global JNDI resources
      Documentation at /docs/jndi-resources-howto.html
-->
<GlobalNamingResources>
  <!-- Editable user database that can also be used by
        UserDatabaseRealm to authenticate users
    -->
  <Resource name="UserDatabase" auth="Container"
            type="org.apache.catalina.UserDatabase"
            description="User database that can be updated and saved"
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
            pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>

<!-- A "Service" is a collection of one or more "Connectors" that share
      a single "Container" Note: A "Service" is not itself a "Container",
      so you may not define subcomponents such as "Valves" at this level.
      Documentation at /docs/config/service.html
-->
<Service name="Catalina">

  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
        maxThreads="150" minSpareThreads="4"/>
  -->

  <!-- A "Connector" represents an endpoint by which requests are received
        and responses are returned. Documentation at :
        Java HTTP Connector: /docs/config/http.html
        Java AJP Connector: /docs/config/ajp.html
        APR (HTTP/AJP) Connector: /docs/apr.html
        Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
  <Connector port="8080" protocol="HTTP/1.1" URIEncoding="UTF-8"
            connectionTimeout="20000"
            redirectPort="8443" />
  <!-- A "Connector" using the shared thread pool-->
  <!--
  <Connector executor="tomcatThreadPool"
            port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
  -->

  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
        This connector uses the NIO implementation. The default
        SSLImplementation will depend on the presence of the APR/native
        library and the useOpenSSL attribute of the
        AprLifecycleListener.
        Either JSSE or OpenSSL style configuration may be used regardless of
        the SSLImplementation selected. JSSE style configuration is used below.
    -->
  <!--
  <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
            maxThreads="150" SSLEnabled="true">
    <SSLHostConfig>
      <Certificate certificateKeystoreFile="conf/localhost-rsa.jks"
                    type="RSA" />
    </SSLHostConfig>
  </Connector>
  -->

  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2
        This connector uses the APR/native implementation which always uses

```

OpenSSL for TLS.

Either JSSE or OpenSSL style configuration may be used. OpenSSL style configuration is used below.

```
-->
<!--
<Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"
    maxThreads="150" SSLEnabled="true" >
    <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
        <Certificate certificateKeyFile="conf/localhost-rsa-key.pem"
            certificateFile="conf/localhost-rsa-cert.pem"
            certificateChainFile="conf/localhost-rsa-chain.pem"
            type="RSA" />
    </SSLHostConfig>
</Connector>
-->

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" tomcatAuthentication="false" URIEncoding="UTF-
8" />

<!-- An Engine represents the entry point (within Catalina) that processes
every request. The Engine implementation for Tomcat stand alone
analyzes the HTTP headers included with the request, and passes them
on to the appropriate Host (virtual host).
Documentation at /docs/config/engine.html -->

<!-- You should set jvmRoute to support load-balancing via AJP ie :
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine name="Catalina" defaultHost="localhost">

    <!--For clustering, please take a look at documentation at:
        /docs/cluster-howto.html (simple how to)
        /docs/config/cluster.html (reference documentation) -->
    <!--
    <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
    -->

    <!-- Use the LockOutRealm to prevent attempts to guess user passwords
        via a brute-force attack -->
    <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
            resources under the key "UserDatabase". Any edits
            that are performed against this UserDatabase are immediately
            available for use by the Realm. -->

        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
            resourceName="UserDatabase"/>

    </Realm>

    <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true">

        <!-- SingleSignOn valve, share authentication between web applications
            Documentation at: /docs/config/valve.html -->
        <!--
        <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
        -->

        <!-- Access log processes all example.
            Documentation at: /docs/config/valve.html
            Note: The pattern used is equivalent to using pattern="common" -->
        <!-- Managing through Apache HTTPD Server config
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
            prefix="localhost_access_log" suffix=".txt"
            pattern="%h %l %u %t \"%r\" %s %b" />
        -->
-->
```

```

    </Host>
  </Engine>
</Service>
</Server>

```

Make a tomcat users config: /opt/groupercontainer/tomcat/tomcat-users.xml (set a password for *****). You can add more users if you like

```

<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
    version="1.0">
  <role rolename="grouper_user"/>
  <user username="GrouperSystem" password="*****" roles="grouper_user"/>
</tomcat-users>

```

Use basic auth for tomcat ui, edit: /opt/groupercontainer/tomcat/grouper-ui-web.xml (copy file from container, add in sections to bottom)

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:j2ee="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
    version="2.4">
  <filter>
    <filter-name>GrouperUi</filter-name>
    <filter-class>edu.internet2.middleware.grouper.ui.GrouperUiFilter</filter-class>
  </filter>
  <filter>
    <filter-name>CSRFGuard</filter-name>
    <filter-class>org.owasp.csrfguard.CsrfGuardFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>GrouperUi</filter-name>
    <url-pattern>*.jsp</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>GrouperUi</filter-name>
    <url-pattern>/grouperUi/app/*</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>GrouperUi</filter-name>
    <url-pattern>/grouperUi/appHtml/*</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>GrouperUi</filter-name>
    <url-pattern>/grouperExternal/app/*</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>GrouperUi</filter-name>
    <url-pattern>/grouperExternal/public/UiV2Public.index</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>GrouperUi</filter-name>
    <url-pattern>/grouperExternal/public/UiV2Public.postIndex</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>CSRFGuard</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <listener>
    <listener-class>edu.internet2.middleware.grouper.ui.GrouperSessionAttributeListener</listener-class>
  </listener>
  <listener>
    <listener-class>org.owasp.csrfguard.CsrfGuardServletContextListener</listener-class>
  </listener>
</web-app>

```

```

    <listener-class>org.owasp.csrfguard.CsrfGuardHttpSessionListener</listener-class>
</listener>
<servlet>
    <servlet-name>StatusServlet</servlet-name>
    <display-name>Status Servlet</display-name>
    <servlet-class>edu.internet2.middleware.grouper.j2ee.status.GrouperStatusServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
    <servlet-name>UiServlet</servlet-name>
    <servlet-class>edu.internet2.middleware.grouper.j2ee.GrouperUiRestServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
    <servlet-name>OwaspJavaScriptServlet</servlet-name>
    <servlet-class>org.owasp.csrfguard.servlet.JavaScriptServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>StatusServlet</servlet-name>
    <url-pattern>/status</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>UiServlet</servlet-name>
    <url-pattern>/grouperUi/app/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>UiServlet</servlet-name>
    <url-pattern>/grouperExternal/app/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>UiServlet</servlet-name>
    <url-pattern>/grouperExternal/public/UiV2Public.index</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>UiServlet</servlet-name>
    <url-pattern>/grouperExternal/public/UiV2Public.postIndex</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>OwaspJavaScriptServlet</servlet-name>
    <url-pattern>/grouperExternal/public/OwaspJavaScriptServlet</url-pattern>
</servlet-mapping>
<security-constraint>
    <web-resource-collection>
        <web-resource-name>UI</web-resource-name>
        <url-pattern>/grouperUi/app/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>grouper_user</role-name>
    </auth-constraint>
</security-constraint>
<security-constraint>
    <web-resource-collection>
        <web-resource-name>UI</web-resource-name>
        <url-pattern>/grouperUi/appHtml/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>grouper_user</role-name>
    </auth-constraint>
</security-constraint>
<security-constraint>
    <web-resource-collection>
        <web-resource-name>UI</web-resource-name>
        <url-pattern>/grouperExternal/app/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>grouper_user</role-name>
    </auth-constraint>
</security-constraint>
<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>Grouper Application</realm-name>

```

```

</login-config>
<security-role>
  <description>
    The role that is required to log in to the Grouper UI
  </description>
  <role-name>grouper_user</role-name>
</security-role>

</web-app>

```

Enable config editing from UI from anywhere (if you have a static IP or cidr use that instead). This is for grouper admins. edit: /opt/groupercontainer/conf/grouper-ui.properties

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

Docker commands

See which containers are running

```

[root@ip-172-30-0-157 init.d]# docker ps
CONTAINER ID        IMAGE                                     COMMAND
CREATED            STATUS              PORTS              NAMES
23f7fa789326        tier/grouper:2.4.0-a89-u55-w11-p12-20200110-rc1  "/usr/local/bin/entrâ€¦"  21 minutes
ago                Up 21 minutes      80/tcp, 0.0.0.0:443->443/tcp

```

Stop container (there will be grouper-ui, grouper-ws, and grouper-daemon... you can start or stop individually)

```

[root@ip-172-30-0-157 init.d]# docker stop grouper-ui
grouper-ui
[root@ip-172-30-0-157 init.d]#

```

Remove container

```

[root@ip-172-30-0-157 init.d]# docker rm grouper-ui
grouper-ui
[root@ip-172-30-0-157 init.d]#

```

Terminal in

```
[root@ip-172-30-0-157 httpd]# docker exec -it grouper-ui /bin/bash
```

Review logs

```

docker logs grouper-ui

-or for grouper logs go to-

/opt/groupercontainer/logs/grouper-ui

```

Init database

Start a container below.

Then SSH into it and init database (one time task to create grouper tables)

```
[root@ip-172-30-0-83 ~]# docker exec -it grouper-ui /bin/bash
[root@50a8af38ba7f grouper.apiBinary]# cd /opt/grouper/grouper.apiBinary/bin/
[root@50a8af38ba7f bin]# ./gsh -registry -runscript
```

Alternate way to init db by creating a throw away container

```
[root@ip-172-30-0-83 ~]# docker run --detach \
--mount type=bind,src=/opt/groupercontainer/conf,dst=/opt/grouper/conf \
--mount type=bind,src=/opt/groupercontainer/logs/grouper-ui,dst=/opt/grouper/logs \
--mount type=bind,src=/opt/groupercontainer/ddlScripts,dst=/opt/grouper/grouper.apiBinary/ddlScripts/ \
--name dbInit \
--entrypoint /opt/grouper/grouper.apiBinary/bin/gsh \
tier/grouper:2.4.0-a89-u55-w11-p12-20200110-rc1 \
-registry -check -runscript -noprompt

# NOTE: The /opt/groupercontainer/ddlScripts dir will contain the ddl file that was executed against the DB. (
If you want it. )
# Now you can delete that container. The above line will only check and correct the DB if needed, but generally
should not be done trivially.
[root@ip-172-30-0-83 ~]# docker rm dbInit
```

UI

```
[root@ip-172-30-0-157 ~]# docker run --detach --publish 443:443 \
--mount type=bind,src=/opt/groupercontainer/conf,dst=/opt/grouper/conf \
--mount type=bind,src=/opt/groupercontainer/logs/grouper-ui,dst=/opt/grouper/logs \
--mount type=bind,src=/opt/groupercontainer/httpd/ssl-enabled.conf,dst=/etc/httpd/conf.d/ssl-enabled.conf \
--mount type=bind,src=/opt/groupercontainer/httpd/grouper-www.conf,dst=/etc/httpd/conf.d/grouper-www.conf \
--mount type=bind,src=/opt/groupercontainer/tomcat/tomcat-users.xml,dst=/opt/tomcat/conf/tomcat-users.xml \
--mount type=bind,src=/opt/groupercontainer/tomcat/server.xml,dst=/opt/tomcat/conf/server.xml \
--mount type=bind,src=/opt/groupercontainer/tomcat/grouper-ui-web.xml,dst=/opt/grouper/grouper.ui/WEB-INF
/web.xml \
--restart always --name grouper-ui \
tier/grouper:2.4.0-a89-u55-w11-p12-20200110-rc1 \
ui
```

Now try UI, login as GrouperSystem and the pass is whatever you configured above

<https://ec2-54-210-221-100.compute-1.amazonaws.com/grouper/>

Sign in

https://ec2-34-239-141-228.compute-1.amazonaws.com

Username

Password

Sign in

Cancel



Search

Logged in as GrouperSysAdmin · Log out · Help

+ Create new group

Quick links

My groups
My folders
My favorites
My services
My activity
Miscellaneous

Browse folders

Root
etc

Home

Grouper

Institute of Higher Education

This website allows you to manage groups associated with your organization and the members of those groups. For a list of answers to frequently asked questions, refer to the [support documentation](#).

Recent activity

Recent activity	Activity Date
Added attribute Grouper user data to a membership for member GrouperSysAdmin.	2020/01/15 18:58 PM
Added GrouperSysAdmin as a member of the grouperUiUserData group.	2020/01/15 18:58 PM
Edited group grouperUiUserData .	2020/01/15 18:58 PM
Added group grouperUiUserData .	2020/01/15 18:58 PM
Added folder grouperUi .	2020/01/15 18:58 PM

My favorites

[View all favorites](#)

Groups I manage

instrumentationDataCollectorsGroup
etc : attribute : instrumentationData

instrumentationDataInstancesGroup
etc : attribute : instrumentationData

grouperMessageRole
etc : attribute : messages

upgradeTasksMetadataGroup
etc : attribute : upgradeTasks

My services

[View all services](#)

Daemon

Run


```
docker run --detach \
--mount type=bind,src=/opt/groupercontainer/conf,dst=/opt/grouper/conf \
--mount type=bind,src=/opt/groupercontainer/logs/grouper-daemon,dst=/opt/grouper/logs \
--restart always --name grouper-daemon \
tier/grouper:2.4.0-a89-u55-w11-p12-20200110-rc1 \
daemon
```

See it working in the UI: misc all daemon jobs see change log temp to change log

ec2-54-210-221-100.compute-1.amazonaws.com/grouper/grouperUi/app/UiV2Main.index?operation=UiV2Admin.viewLogs&jobName=CHANGE_LOG_changeLogTe...

INTERNET

Search

Logged in as GrouperSysAdmin · Log out · Help

+ Create new group

Quick links

- My groups
- My folders
- My favorites
- My services
- My activity
- Miscellaneous

Browse folders

- Root
- etc

Home > Miscellaneous > All daemon jobs > Daemon logs

Daemon logs

Filter for: CHANGE_LOG_changeLogTempToChangeLog

Start time between: yyyy-mm-dd hh:mi:ss

End time between: yyyy-mm-dd hh:mi:ss

Last updated between: yyyy-mm-dd hh:mi:ss

Subjobs: ☐ Show subjobs

Status: ☐ Success ☐ Error ☐ Started ☐ Running ☐ Warning ☐ Config error ☐ Subject problems

Number of rows: 400

Apply filter Reset

1 logs found for job name: CHANGE_LOG_changeLogTempToChangeLog

Status	Loaded group	Job type	Start time	End time	Millis	Millis get data	Millis load data	Total count	Add count	Update count	Delete count	Unresolvable count
Success	N/A	overall	2020-01-16 02:58:50.0	2020-01-16 02:59:02.0	11675			916	0	0	0	0

© Institute of Higher Education

WS

Run

```
docker run --detach --publish 8443:443 \
--mount type=bind,src=/opt/groupercontainer/conf,dst=/opt/grouper/conf \
--mount type=bind,src=/opt/groupercontainer/logs/grouper-ws,dst=/opt/grouper/logs \
--mount type=bind,src=/opt/groupercontainer/httpd/ssl-enabled.conf,dst=/etc/httpd/conf.d/ssl-enabled.conf \
--mount type=bind,src=/opt/groupercontainer/httpd/grouper-www.conf,dst=/etc/httpd/conf.d/grouper-www.conf \
--mount type=bind,src=/opt/groupercontainer/tomcat/tomcat-users.xml,dst=/opt/tomcat/conf/tomcat-users.xml \
--mount type=bind,src=/opt/groupercontainer/tomcat/server.xml,dst=/opt/tomcat/conf/server.xml \
--mount type=bind,src=/opt/groupercontainer/tomcat/grouper-ui-web.xml,dst=/opt/grouper/grouper.ui/WEB-INF/web.xml \
--restart always --name grouper-ws \
tier/grouper:2.4.0-a89-u55-w11-p12-20200110-rc1 \
ws
```

Try WS from client from browser (you need real SSL or non-SSL to use the grouper client)

https://ec2-34-239-141-228.compute-1.amazonaws.com:8443/grouper-ws/servicesRest/v2_4_000/subjects/GrouperSystem

```
{ "WsGetSubjectsResults":
  { "resultMetadata":
    { "success": "T", "resultCode": "SUCCESS", "resultMessage": "Queried 1 subjects" },
    "responseMetadata":
    { "serverVersion": "2.4.0", "millis": "353" },
    "wsSubjects": [
      { "sourceId": "g:isa", "success": "T", "name": "GrouperSysAdmin", "resultCode": "SUCCESS", "id": "GrouperSystem" }
    ]
  }
}
```

If you have a real cert on WS apache, use the client like this

```
[root@ip-172-30-0-157 logs]# cd /tmp
[root@ip-172-30-0-157 tmp]# mkdir grouperClient
[root@ip-172-30-0-157 tmp]# cd grouperClient
[root@ip-172-30-0-157 grouperClient]# wget https://software.internet2.edu/grouper/release/2.4.0/grouper.
clientBinary-2.4.0.tar.gz
[root@ip-172-30-0-157 grouperClient]# tar xzvf grouper.clientBinary-2.4.0.tar.gz
[root@ip-172-30-0-157 grouperClient]# cd grouper.clientBinary-2.4.0/
[root@ip-172-30-0-157 grouper.clientBinary-2.4.0]# yum install java-1.8.0-openjdk

Edit grouper.client.properties

# url of web service, should include everything up to the first resource to
access
# e.g. http://groups.school.edu:8090/grouper-ws
/servicesRest
# e.g. https://groups.school.edu/grouper-ws
/servicesRest
grouperClient.webService.url = https://ec2-34-239-141-228.compute-1.amazonaws.com:8443/grouper-ws/servicesRest

# kerberos principal used to connect to web
service
grouperClient.webService.login = GrouperSystem

# password for shared secret authentication to web
service
# or you can put a filename with an encrypted
password
grouperClient.webService.password = *****

[root@ip-172-30-0-157 grouper.clientBinary-2.4.0]# java -jar grouperClient.jar --operation=getSubjectsWs --
subjectIds=GrouperSystem --debug=true
```

Upgrade container (or change version)

```

[root@ip-172-30-0-82 ~]# docker stop grouper-ui
grouper-ui
[root@ip-172-30-0-82 ~]# docker stop grouper-ws
grouper-ws
[root@ip-172-30-0-82 ~]# docker stop grouper-daemon
grouper-daemon
[root@ip-172-30-0-82 ~]# docker rm grouper-ui
grouper-ui
[root@ip-172-30-0-82 ~]# docker rm grouper-ws
grouper-ws
[root@ip-172-30-0-82 ~]# docker rm grouper-daemon
grouper-daemon
[root@ip-172-30-0-82 ~]# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
tier/grouper	2.4.0-a89-u55-w11-p12-20200110-rc1	4218bfea3573	2 days ago	1.34GB

```

[root@ip-172-30-0-82 ~]# docker rmi 4218bfea3573
[root@ip-172-30-0-82 ~]#

```

Pull a new tag from: <https://hub.docker.com/r/tier/grouper/tags>

```

[root@ip-172-30-0-82 ~]# docker pull tier/grouper:2.4.0-a86-u53-w10-p12-20191224-rc1

```

Startup the UI and WS and daemon per command above with new version