Install and Configure the Web VM

The DCDS web application is deployed to tomcat on a server that is also running RabbitMQ. The RESTful API (em-api) can be deployed on the same server or a different tomcat server. The instructions are the same but the em-api config files need updated to point at the correct RabbitMQ message bus. See an explanation of configuration properties in the DCDS Configuration Properties section below.

Table of Contents

Prerequisites

- Apache 2.4.7
- Tomcat 8
- Rabbitmq 3.4.4
- Oracle JDK 7

Loading the proxy_wstunnel Module

Use the following command to load the proxy_wstunnel Module:

```
sudo a2enmod proxy_wstunnel
```

Including 3rd Party Software in iweb

Please place the following libraries in the directory iweb-modules/core/src/main/js/lib

- Atmosphere 2.2.12 : https://github.com/Atmosphere/atmosphere-javascript
- JQuery 1.11.1 : https://jquery.com/download/
- ExtJS 6.0.1 : https://www.sencha.com/products/extjs/#overview

The paths for each library are defined in iweb-modules/core/src/main/js/iweb/CoreModule.js

Configuration Files

Core.properties

Navigate to **/dcds-web/web-app/src/main/config/core.properties** in the repository and move the file ***core.properties*** to **/opt/data/dcds/config** on the Web VM. The configuration file should look similar to **Figure 1** and requires the fields in brackets, such as <hostname> and <cookie domain> to be configured.

core.properties

```
endpoint.rest=https://<hostname of data vm>/api/v1
endpoint.geoserver=https://<hostname of geoserver vm>/geoserver
#Example .dcds.com
private.cookie.domain=<cookie domain>

iplanet.key=iPlanetDirectoryPro
iplanet.path=/
iplanet.domain=<same as the cookie domain>

openam.key=AMAuthCookie
openam.path=/
openam.domain=<same as the cookie domain>

rabbitmq.hostname=localhost
rabbitmq.username=guest
rabbitmq.userpwd=guest
rabbitmq.exchange.name=iweb.amq.topic

# In minutes
token.timeout=720

# Report properties. Storage isn't really used on the web side, so will
factor out at some point
report.general.storage=/opt/data/dcds/static/reports/general/
report.general.url=https://<web vm>/static/reports/general/

report.damage.storage=/opt/data/dcds/static/reports/damage
report.damage.url=https://<web vm>/static/reports/damage/
main.site.label=<label on top left after login and tab page header on login
screen>
main.site.logo=<image displayed on login screen>
help.site.url=<URL for help page on login sreen>
registration.help.info.logo=<info displayed in registration page on who to
contact for assistance>
```

Apache Configuration

Configure the properties in the Apache configuration file:

```
# Proxy requests for static content to the data vm
ProxyPass /static https://<datavm>/static
ProxyPassReverse /static https://<datavm>/static

ProxyPass /dcds/mediator ws://<web ip>:8080/dcds/mediator
ProxyPass /dcds http://<web ip>:8080/dcds
ProxyPassReverse /dcds http://<web ip>:8080/dcds

ProxyPass /em-api https://<data vm>/api
ProxyPassReverse /em-api https://<data vm>/api

ProxyPass / http://<web ip>:8080/dcds
ProxyPassReverse / http://<web ip>:8080/dcds

# NOTE: static content is no longer hosted on web
#ProxyPass /static !
#<Directory "/opt/data/dcds/static">
#        Require all granted
#</Directory>
```

RabbitMQ Loopback Configuration

> Starting with RabbitMQ version 3.3.0, the default 'guest' user is no longer allowed to connect to a RabbitMQ instance remotely. The workaround is to either create a new RabbitMQ user for your connections (the preferred solution) or to add a config to allow all users to connect remotely.

In order to configure the loopback settings, use the following procedure:

1. Add the following line to your **/etc/rabbitmq/rabbitmq.config** file. If this file does not exist, create it.

   ```
   [{rabbit, [{loopback_users,[]}]}].
   ```

   Note: The period at the end is required.
2. Restart the RabbitMQ server to allow it to pick up the changes.
3. Following the restart, you should be able to connect tho this instance remotely with the 'guest' user.

Deploying DCDS

Deploy the file dcds.war to the directory **/var/lib/to mcat8/webapps**.

Configuring and Installing CollabFeedManager

The CollabFeedManager runs on the web machine and listens for feature updates to collaboration rooms. It creates a datalayer in geoserver that can be imported into and exported from dcds. Use the following steps to configure the CollabFeedManager:

1. Run the following commands in the / directory. Create a directory to run dcds components.

```
> mkdir -p /opt/dcds/deploy/collab-feed-manager
```

2. Untar the file

```
# Untar all files to dcds deploy
> tar -xvzf collab-feed-manager-X.X.X.tar.gz -C
/opt/dcds/deploy/collab-feed-manager
```

3. Configure the properties file
   a. Open the properties file

```
> vi
/opt/dcds/deploy/collab-feed-manager/collab-feed-manager-VERSION
/collab-feed-manager.properties
```

   b. Configure to dbHost point to a database vm instance used by MapServer

```
db.name=DBNAME
db.user=USER
db.pass=PASS
db.host=HOST
db.port=5432


geoserver.url=http://HOST:8080/geoserver/rest
geoserver.username=
geoserver.password=
geoserver.workspace=<workspace name>
geoserver.datastore=


# syncInterval is in milliseconds
syncInterval=60000
collabSrcUrl=rabbitmq://localhost:5672/iweb.amq.topic?exchangeTy
pe=topic&routingKey=iweb.DCDS.#&username=guest&password=guest
kmlTemplatePath=config/kmlTemplate.kml
kmlFilepath=/data/dcds/static/collabfeedkmltest/
kmlUrl=http://HOST/static/collabfeedkmltest/
# Location of the log4j properties file to use
log4jPropertyFile=config/log4j.properties
```

4. Start the component

```
# Copy collab-feed-manager to dcds deploy
> cd /opt/dcds/deploy/collab-feed-manager/
> nohup ./start.sh > logs/collab-feed-manager.log &
```

Configuring and Installing the Email-consumer

Deployment

After building, the target directory of **dcds-core-processor/email-consumer/** will have a ***deployable.tar.gz*** file.

1. Transfer that file to the **web** VM.
2. DCDS components are usually deployed to **/opt/dcds/deploy/**
3. Untar the deployable.tar.gz file

Email Consumer Configuration

The only file that should need configuring is the email-consumer.properties file. The Host, RabbitMQ and SMTP server need to be defined.

The key 'nodeAndName' should be constructed like '[hostname]-email-consumer]'

RabbitMQ

If RabbitMQ has been installed on the **web** VM, the rabbit property should be fine. Otherwise, in key 'srcUrl' replace **localhost** with the FQDN of the RabbitMQ host.

SMTP

The SMTP server host, port and credentials need to be defined. if **GMAIL** is the SMTP server, do not include @gmail.com in the 'mail.username' key.

Example email-consumer.properties

```
srcUrl=rabbitmq://localhost:5672/iweb.amq.topic?exchangeType=topic&request
edHeartbeat=0&routingKey=iweb.#.email.#&autoAck=false&username=guest&passw
ord=guest&autoDelete=true

# SSL mail config
mail.smtp.port=587
mail.smtp.host=smtp.gmail.com

# credentials for mail server (e.g. gmail)
# do not include @gmail.com in username
mail.username=noreply-dcds
mail.password=somepassword

# Location of the log4j properties file to use
log4jPropertyFile=config/log4j.properties

mach=rabbitmq://localhost:5672/iweb.amq.topic?exchangeType=topic&requested
Heartbeat=0&routingKey=DCDS.mach.components.emailConsumer&autoAck=false&us
ername=guest&password=guest
nodeAndName=dcds-testing-email-consumer
```