

Joget Clustering with Session Replication on Redhat Openshift with JBoss EAP 7

- Description
- Steps
 - Create MySQL Database
 - Create Red Hat JBoss Data Grid
 - Create Red Hat JBoss EAP 7
 - Add Storage to JBoss EAP
 - Deploy Joget WAR file to Red Hat Openshift
- References

Description

This documentation demonstrates how to run Joget in clustering mode with transparent failover, and it is deployed on Red Hat Openshift with JBoss EAP 7. In order to achieve session replication, we externalized HTTP sessions using Red Hat JBoss Data Grid. This allows the sessions of logged in users remain intact while we scale up/down the deployment Joget, or even during the rolling update.

Steps

Create MySQL Database

Create a MySQL Database server with these commands:

```
export DB_APP_NAME=jogetdb
export MYSQL_DATABASE=jwdb
export MYSQL_USER=joget
export MYSQL_PASSWORD=joget
oc new-app openshift/mysql:5.7 --name $DB_APP_NAME -e MYSQL_USER=$MYSQL_USER -e MYSQL_PASSWORD=$MYSQL_PASSWORD -e
MYSQL_DATABASE=$MYSQL_DATABASE
```

Create Red Hat JBoss Data Grid

Create a new Red Hat JBoss Data Grid app in your project using this command:

```
oc new-app --template=datagrid71-basic -p CACHE_NAMES=http-session-cache -p MEMCACHED_CACHE=memcached
```

Create Red Hat JBoss EAP 7

Once above app is available, create a Red Hat JBoss EAP 7 with this command:

```
oc new-app --template=eap71-basic-s2i -p CONTEXT_DIR= -e JGROUPS_PING_PROTOCOL=openshift.DNS_PING -e OPENSIFT_DNS_PING_SERVICE_NAME=eap-app-ping -e
OPENSIFT_DNS_PING_SERVICE_PORT=8888 -e CACHE_NAME=http-session-cache
```

Add Storage to JBoss EAP

We need a common storage for Joget data folder wflow, hence we will add a storage to the new deployed Red Hat JBoss EAP. Make sure the storage access mode is Read-Write-Many as multiple pods will

connect to this volume in clustering environment, and set the Mount Path to /home/jboss/wflow:

```
export STORAGE_NAME=wflow
echo === create persistent storage claim ===
cat <<EOF > pvc.yaml
apiVersion: "v1"
kind: "PersistentVolumeClaim"
metadata:
  name: "wflow"
spec:
  accessModes:
    - "ReadWriteMany"
  resources:
    requests:
      storage: "1Gi"
EOF
oc create -f pvc.yaml
rm pvc.yaml

echo === mount storage ===
oc set volume dc/eap-app --add --type=persistentVolumeClaim --claim-name=$STORAGE_NAME --mount-path=/home/jboss/wflow
```

Deploy Joget WAR file to Red Hat Openshift

Since deployment-scanner in JBoss EAP is not designed to use a shared deployment folder, therefore we need to create a new build and deploy Joget WAR file to Openshift. To do that, first [download](#) and install a copy of Joget locally to obtain the jw.war in the apache-tomcat/webapps directory.

Then run this command:

```
oc start-build eap-app --from-file=jw.war
```

Once it is done, it should trigger a new deployment. Check the log of new pod and make sure the jw.war is deployed.

Try to log in to Joget via \$routeProvider/jw/setup, where \$routeProvider is the result of the command: `oc get routes eap-app`. Complete the [database setup](#) on Joget using the database credentials used when creating the MySQL instance, then you should be able to scale up or down the deployments, and your login session will still be kept during these processes.

References

<https://developers.redhat.com/blog/2018/05/04/externalized-http-session-in-openshift-3-9/>

<https://github.com/mvocale/http-session-counter-openshift>