

Installer un cluster GlassFish avec Session FailOver

télécharger le jar d'installation sur :

<http://download.java.net/javaee5/v2ur1/promoted/Linux/glassfish-installer-v2ur1-b09d-linux.jar>

Installer le serveur

lancer l'installation

```
% java -Xmx256m -jar filename.jar
```

suivre les instructions

Se positionner dans le répertoire qui à été créé par l'installateur

```
% cd glassfish
```

donner les droits d'exécution au scripts ant intégré

```
% chmod -R +x lib/ant/bin
```

editer le fichier setup-cluster.xml pour le configurer selon vos besoins.

```
% lib/ant/bin/ant -f setup-cluster.xml pour construire un serveur glassfish prenant en compte les clusters
```

3-sudo ./asadmin start-domain --user admin <domainname> pour lancer le serveur d'application sur le domaine que vous avez configuré .

Créer les agents

1-sudo ./asadmin create-node-agent <nodeagentname> --host <hostname> --port <portnumber>
pour créer un nouvel agent

2-sudo ./asadmin start-node-agent <nodeagentname> pour démarrer votre nouvel agent

Créer le cluster

1-Créer un cluster

```
<glassfish-install-dir>/bin/asadmin create-cluster --user admin <clustername>
```

2-créer une instance rattaché au cluster que vous venez de créer

```
<glassfish-install-dir>/bin/asadmin create-instance --user admin --nodeagent <nodeagentname> --cluster <clustername> <instancename1>
```

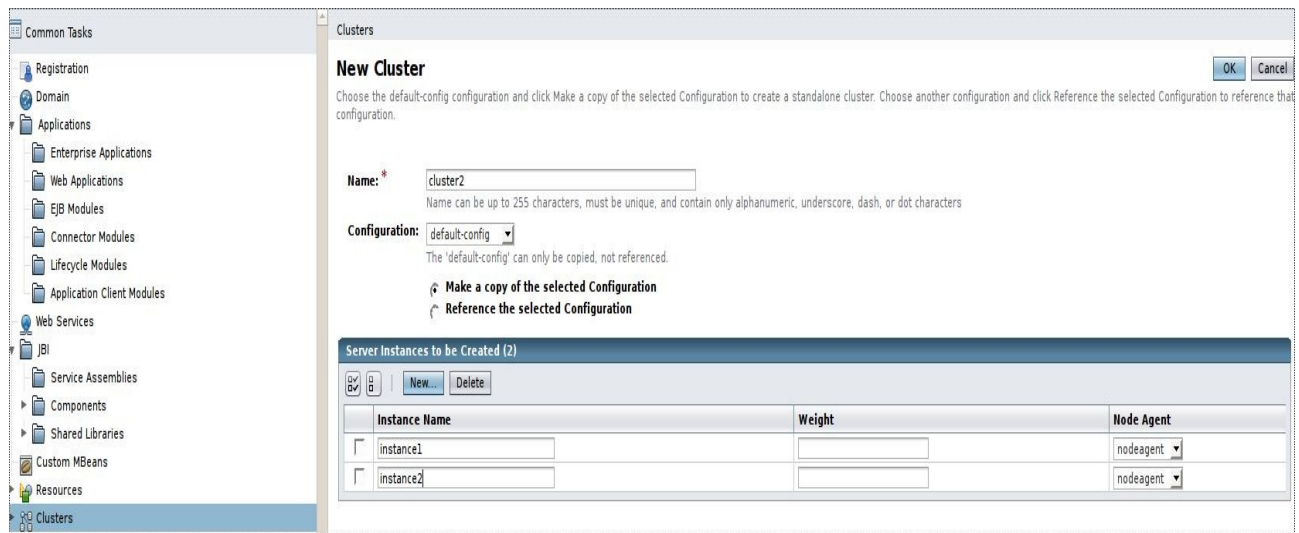
3-créer une seconde instance rattaché au cluster que vous venez de créer

```
<glassfish-install-dir>/bin/asadmin create-instance --user admin --nodeagent <nodeagentname> --cluster <clustername> <instancename2>
```

4-démarrer le cluster

```
<glassfish-install-dir>/bin/asadmin start-cluster <clustername>
```

Nous avons créé le cluster en ligne de commande mais il est tout à fait possible de le créer via l'interface d'admin



Déployer une application sur le cluster

il faut s'assurer que le cluster est configuré correctement, normalement l'option heartbeat est coché par défaut

Clusters > cluster1

General	Applications	Instances	Resources	JBI
---------	--------------	-----------	-----------	-----

General Information

To start or stop any server instance, its node agent must be running. Refer to the online help for more information.

Start Cluster Stop Cluster Migrate EJB Timers ...

Name: cluster1

Configuration: cluster1-config

HTTP Load Balancer:

Heartbeat: **Enabled**
Start GMS service as a lifecycle module in each server in the cluster

Heartbeat Port:
Communication port GMS uses to listen for group events

Heartbeat Address:
Address (only multicast supported) at which GMS will listen for group events

Status: 2 Instance(s) Running
0 Instance(s) Stopped

le descripteur de déploiement de votre application doit contenir l'element <distributable/>

```
<web-app version="2.5" ..... >
<display-name>myclusterable webapp</display-name>
<distributable/>
```

Votre application doit aussi fournir le fichier web-inf/sun-web.xml qui contient les propriétés pour le gestionnaire de session:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Application Server
9.0
Servlet 2.5//EN" "http://www.sun.com/software/appserver/dtds/sun-web-
app_2_5-0.dtd">
<sun-web-app>
  <session-config>
    <session-manager persistence-type="replicated">
      <manager-properties>
        <property name="persistenceFrequency" value="web-method"/>
      </manager-properties>
      <store-properties>
        <property name="persistenceScope" value="session"/>
      </store-properties>
    </session-manager>
  </session-properties/>
  <cookie-properties/>
</session-config>
```

Quelques explications sur la configuration du fichier web-inf/sun-web.xml

La propriété « **persistence-type** » doit être de type « **replicated** » pour permettre la réplication en mémoire des sessions dans votre cluster.

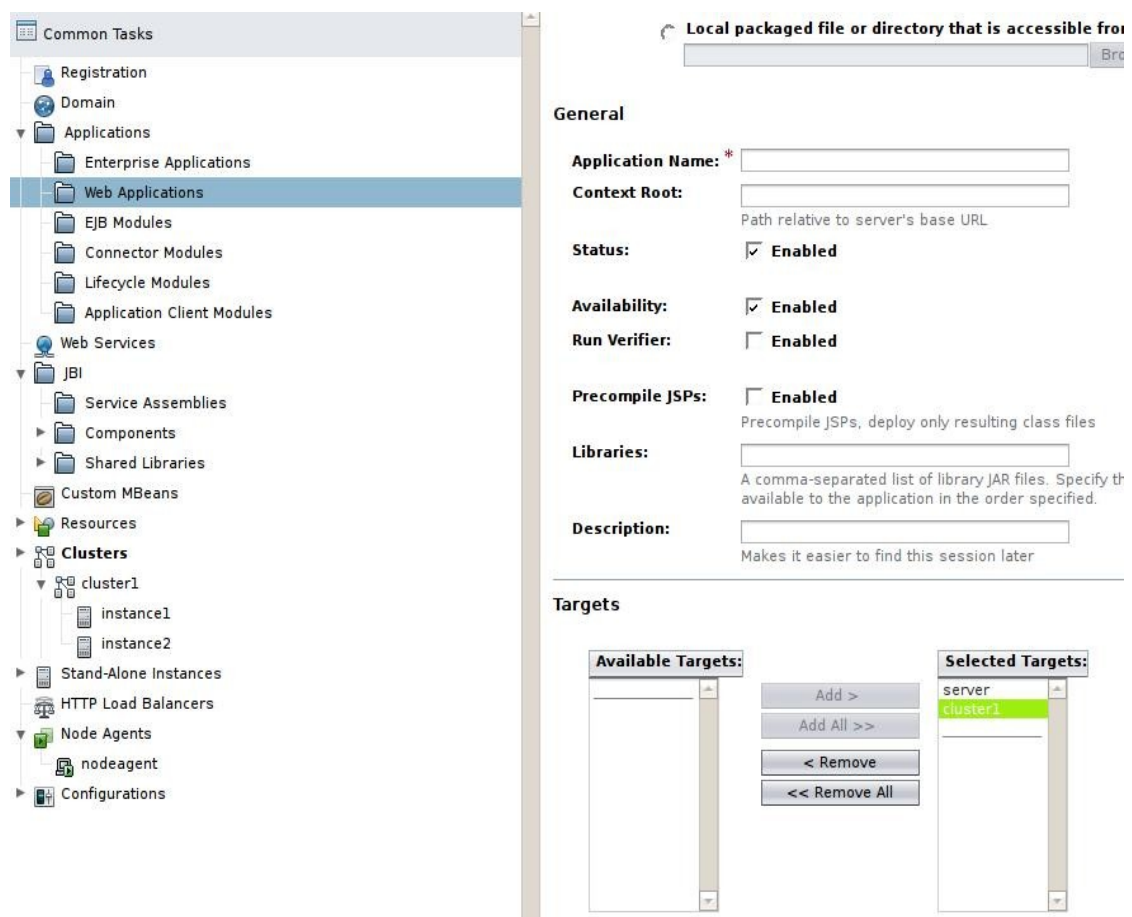
Définir la propriété « **persistenceFrequency** » avec la valeur « **web-méthod** » signifie qu'après chaque requête client les sessions sont reproduites.

On peut configurer cette propriété avec une autre valeur « **time-based** ».

La propriété « **PersistenceScope** » configurée avec la valeur « **session** » signifie qu'après chaque requête l'ensemble des Session HTTP seront reproduites.

D'autres valeurs peuvent être applicables à la propriété « **PersistenceScope** », ces valeurs sont « **modified-session** » et « **modified-attribute** ».

Il ne vous reste plus qu'à déployer votre application, aller dans l'interface d'admin de votre serveur et dans les options de déploiement assurez-vous de sélectionner la case à cocher **Availability**.



Si les choses se sont déroulées correctement, vous devriez voir dans votre fichier `server.log` la trace suivante :

```
[# | 2007-04-17T10:31:11.255+0200 | INFO | sun-appserver9.1 |  
javax.enterprise.system.container.web |
```

```
_ThreadID=24;_ThreadName=RMI TCP connection(19)-<ip-  
address>/clusterjsp;replicated;  
web-method;session;|WEB0130: Enabling ha-based persistence for web module  
[/clusterjsp]'s  
sessions: persistence-type = [replicated] / persistenceFrequency = [web-  
method] /  
persistenceScope = [session]|#]
```

Et voilà à ce stade vous avez un cluster avec session FailOver, maintenant vous pouvez accéder à votre applications sur les deux instances en vous connectant sur chacune d'entre elle (ex <host1>:<httpport> and <host2>:<httpport>).

Si votre application modifie vos sessions dans une instance, elle seront répliquer sur la seconde instance de votre cluster.