Fedora Cluster Installation in AWS

Description:

This installation guide describes the detail steps and commands for set up a replication mode Fedora Cluster in the Amazon AWS. The installation procedure as follows: 1. Create and update an Ubuntu instance. 2. Install Java & maven. 3. Install Tomcat. 4. Install Apache and mod_jk. 6. Install Fedora 4 and replication mode configuration. 7. EC2 load balancer.

1. Create an Ubuntu instance in EC2.
    a. Login to EC2 dashboard and select launch instance, Ubuntu server AMI, and select instance type you want. Amazon document: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-instance_linux.html
    b. Configure the “Security Groups” to open 80 and 8080 port.
    c. Login to the instance and update software
        i. > sudo apt-get update
        ii. > sudo apt-get upgrade -y
2. Install Java & Maven
    a. > sudo apt-get install default-jre -y
    b. > sudo apt-get install default-jdk -y
    c. > sudo apt-get install maven -y
3. Install Tomcat
    a. > sudo apt-get install tomcat7 -y
    b. > sudo apt-get install tomcat7-admin -y
    c. config tomcat manager: http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html#Configuring_Manager_Application_Access
        i. Edit $CATALINA_BASE/conf/tomcat-users.xml
4. Install Apache and mod_jk
    a. > sudo apt-get install apache2
    b. > sudo apt-get install libapache2-mod-jk
    c. > sudo vim /etc/tomcat7/server.xml
    uncomment the following line
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    d. > sudo vim /etc/apache2/workers.properties

    # Define 1 real worker using ajp13

    worker.list=worker1

    # Set properties for worker (ajp13)
    worker.worker1.type=ajp13
    worker.worker1.host=localhost
    worker.worker1.port=8009
    e. > sudo vim /etc/apache2/mods-available/jk.conf

    change the JkWorkersFile property to /etc/apache2/workers.properties
    f. > sudo vim /etc/apache2/sites-enabled/000-default.conf

    <VirtualHost *:80>
    .....
    .....
    JkMount /fcrepo4* worker1
    </VirtualHost *:80>

5. Install Fedora 4

    a. > wget http://repo1.maven.org/maven2/org/fcrepo/fcrepo-webapp/4.0.0-beta-03/fcrepo-webapp-4.0.0-beta-03.war
    b. > sudo mkdir /usr/local/fedora-data
    c. > sudo chown -R tomcat7:tomcat7 /usr/local/fedora-data/
    d. Edit tomcat7: Add JAVA_OPTS="${JAVA_OPTS} -Dfcrepo.home=/usr/local/fedora-data"
    e. Edit catalina.properties: Add fcrepo.home=/usr/local/fedora-data
    f. > sudo cp /source/fcrepo-webapp-4.0.0-beta-03.war /var/lib/tomcat7/webapps/.
    g. > sudo service tomcat7 restart

6. Cluster configuration:

    a. Edit tomcat7:
    JAVA_OPTS="$JAVA_OPTS -Xmx1024m -XX:MaxPermSize=256m -Dfcrepo.modeshape.configuration=file:///config/clustered/repository.json"

        JAVA_OPTS="$JAVA_OPTS -Dfcrepo.infinispan.cache_configuration=config/infinispan/clustered/infinispan.xml"

        JAVA_OPTS="${JAVA_OPTS} -Dfcrepo.home=/tmp/wherever"

        JAVA_OPTS="${JAVA_OPTS} -Djgroups.tcp.address=<private-ip-address-of-ec2-instance>"

```
JAVA_OPTS="${JAVA_OPTS} -Dfcrepo.ispn.numOwners=2 -Djava.net.PreferIPv4Stack=true"


# The jgroups-ec2.xml file is included in ispn's jars
JAVA_OPTS="${JAVA_OPTS} -Dfcrepo.ispn.jgroups.configuration=jgroups-ec2.xml"


# This property overwrites the S3 bucketname variable in jgroups-ec2.xml
JAVA_OPTS="${JAVA_OPTS} -Djgroups.s3.bucket=<some-bucket-that-you-have-already-created>"
```

Fedora document: https://wiki.duraspace.org/display/FF/Deploying+a+Fedora+Cluster#DeployingaFedoraCluster-DeployinginAWS

7. EC2 load balancer

    a. Login to EC2 dashboard and select “Load Balancers”.
    b. Create Load Balancer.
    c. Add instances into newly created Load Balancer.