# *Installation*

Before we can start working with Apache ServiceMix, we have to get it installed and running on our local machine first.

1. System requirements

For running Apache ServiceMix itself, you'll need
• Java Runtime Environment (JRE) 1.6.x (Java 6) or
Java Runtime Environment (JRE) 1.7.x (Java 7)
• About 100 MB of free disk space for the default assembly

If you're developing your own integration applications and OSGi bundles, you'll also need
• Java Developer Kit (JDK) 1.6.x (Java 6) or
Java Developer Kit (JDK) 1.7.x (Java 7)
• Apache Maven 3.0.4 or higher

***java installation (java 1.7)***

apt-get update && apt-get -y upgrade
echo oracle-java7-installer shared/accepted-oracle-license-v1-1 select true | /usr/bin/debconf-set-selections
apt-get install -y python-software-properties software-properties-common
add-apt-repository -y ppa:webupd8team/java
apt-get -y update
apt-get install -y nano wget unzip locate oracle-java7-installer
update-java-alternatives --set java-7-oracle
apt-get install oracle-java7-set-default && java -version

2. Downloading Apache ServiceMix

Apache ServiceMix 7.0.0-SNAPSHOT is available under the Apache License v2 and can be downloaded from

***http://servicemix.apache.org/downloads.html***.

In here : Apache ServiceMix 7.0.0.M2 (zip)

Depending on your operation system, you should download either the tar.gz or the zip file:
• tar.gz for Linux/Unix/MacOS X
• zip for Windows

3. unzip it

4. Starting Apache ServiceMix

go to servicemix home directory > ./bin/servicemix

5. Web console

To get the web console installed in ServiceMix, install the feature from your console
***karaf@root> features:install webconsole***

Afterwards, you can verify that the feature is marked installed in the overview. You'll notice that the webconsole-base feature has also been installed as a requirement for the webconsole feature itself.

You will now be able to point your browser to http://localhost:8181/system/console and login with user smx and password smx to access the web console. From the webconsole, you can also start and stop bundles, install optional features again, ...

## ***clustering***

6. clustering > create 2 instance with same image for clustering

adding repo for cellar for doing cluster

***feature:repo-add cellar***
***feature:install cellar***

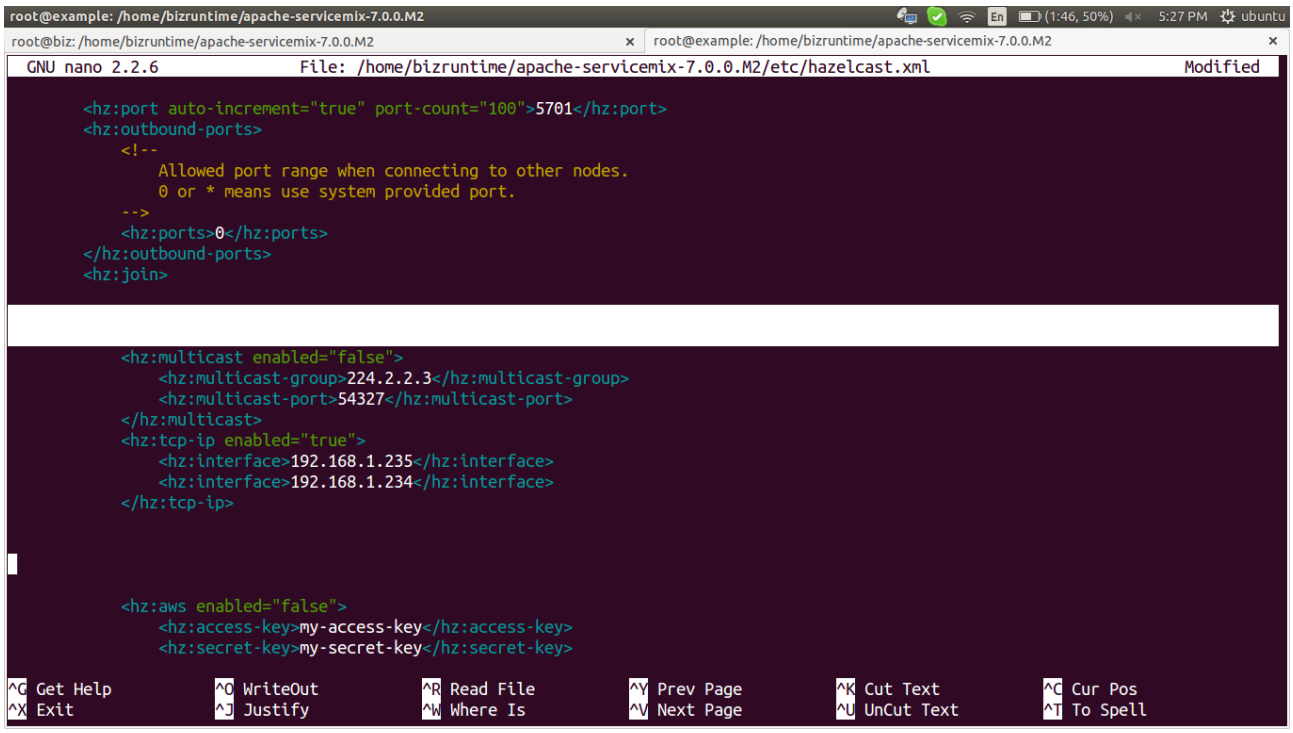
7. edit the configuration file
***nano {apache-servicemix-homedir}/etc/hazelcast.xml***

a) change ***<hz:multicast enabled="true">*** to
***<hz:multicast enabled="false">***

b) change ***<hz:tcp-ip enabled="false">*** to
***<hz:tcp-ip enabled="true">***

provide the node ip first and the another ip next

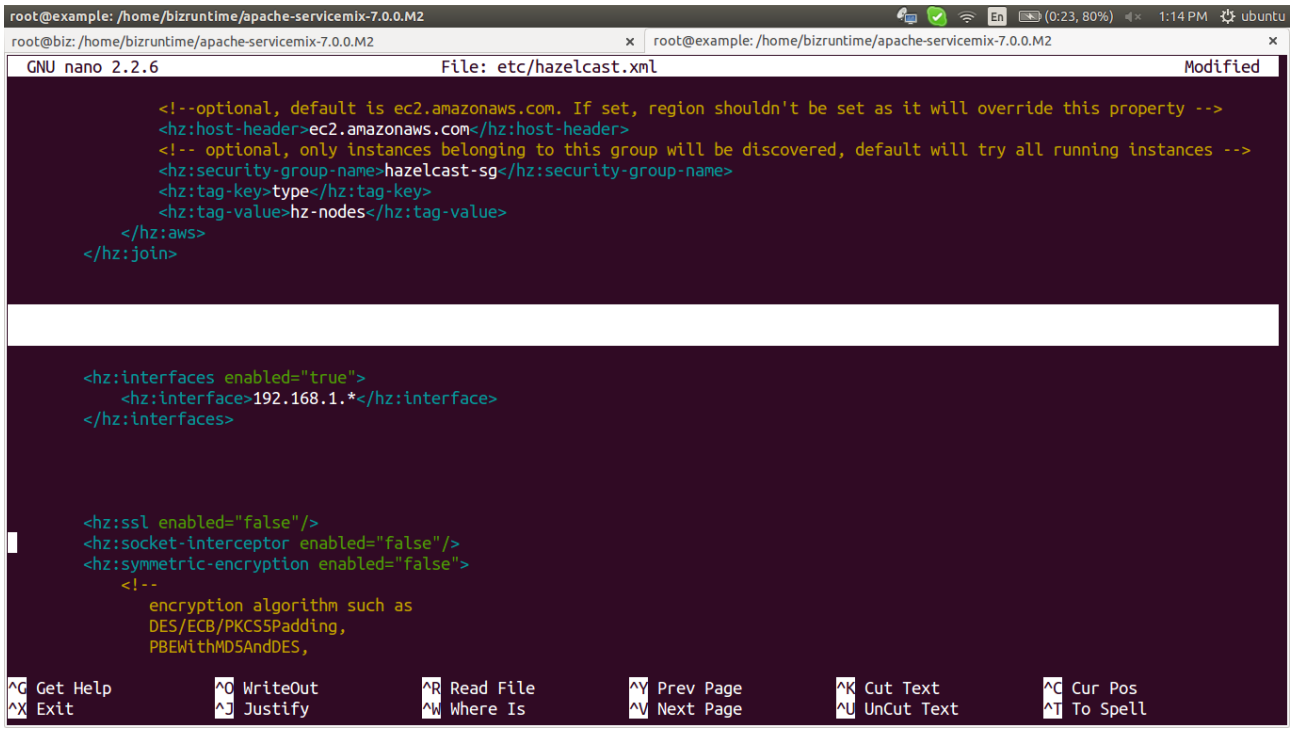
c) add the ip's of all clustering nodes as follows under ***<hz:tcp-ip enabled="true">***
***<hz:interface>192.168.1.235</hz:interface>***
***<hz:interface>192.168.1.234</hz:interface>***




d) edit interfaces part in that configuration
change ***<hz:interfaces enabled="false">*** to
***<hz:interfaces enabled="true">***

and add the interface

```
<hz:interfaces enabled="true">
    <hz:interface>192.168.1.*</hz:interface>
</hz:interfaces>
```



(optional – no need to change cluster to any other option)
8. edit the configuration file
***nano {apache-servicemix-homedir}/etc/org.apache.karaf.cellar.groups.cfg***

its depends on your work

eg:

```
default.bundle.sync = cluster
default.config.sync = cluster
default.feature.sync = cluster
default.obr.urls.sync = cluster
```

check the nodes are available or not by
***cluster:node-list***

```
karaf@root>cluster:node-list
  | Id                | Alias | Host Name      | Port
-----------------------------------------------------
x | 192.168.1.235:5701 |       | 192.168.1.235 | 5701
  | 192.168.1.234:5701 |       | 192.168.1.234 | 5701
```

9) there is a group default which is local cluster to create cluster with remote machine, create a new group

***cluster:group-list***

```
karaf@root>cluster:group-list
  | Group   | Members
-----------------------------------------------------
x | default | 192.168.1.234:5701(x) 192.168.1.235:5701
```

10) create a new group

***cluster:group-create test***

11) join to that group ( run this on both system for add the nodes to that group)

***cluster:group-join test***
***cluster:group-set test***

12) check the group info by

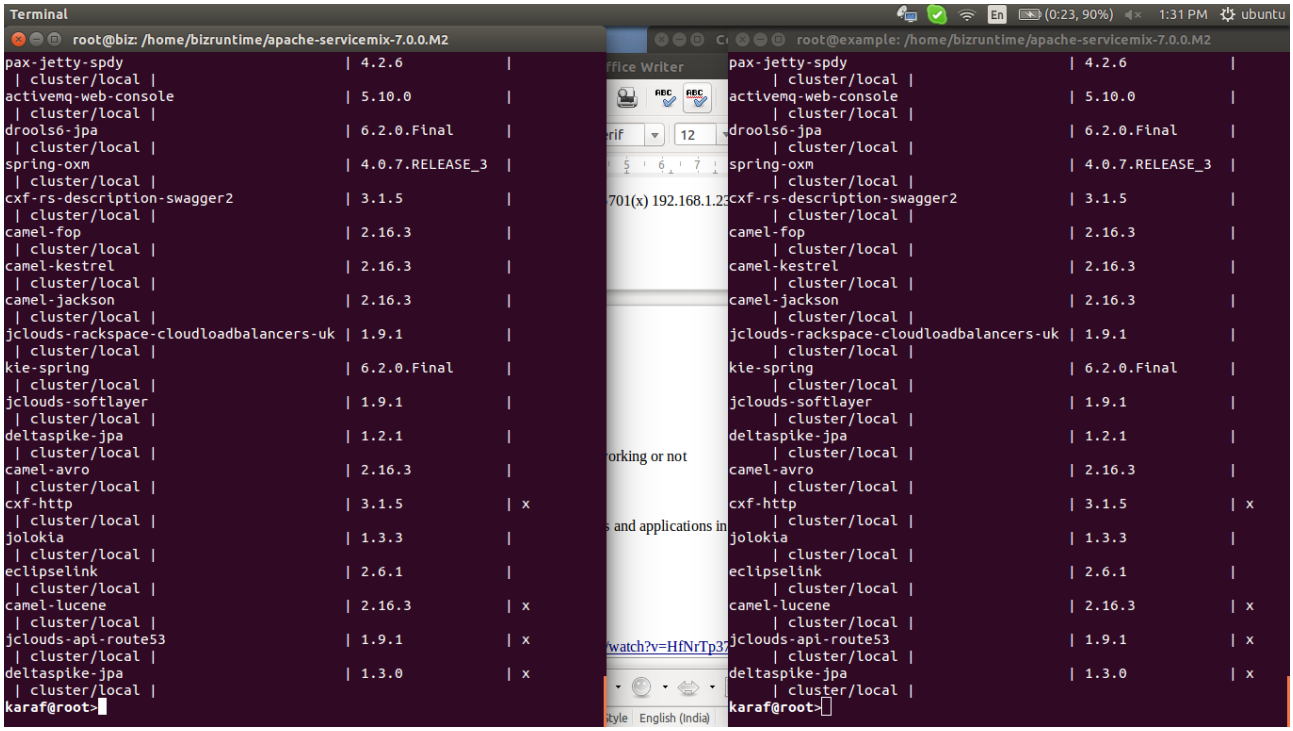
***cluster:group-list***

```
karaf@root>cluster:group-list
  | Group   | Members
-----------------------------------------------------
  | default |
x | test    | 192.168.1.235:5701(x) 192.168.1.234:5701
```

14) check the clustering working or not

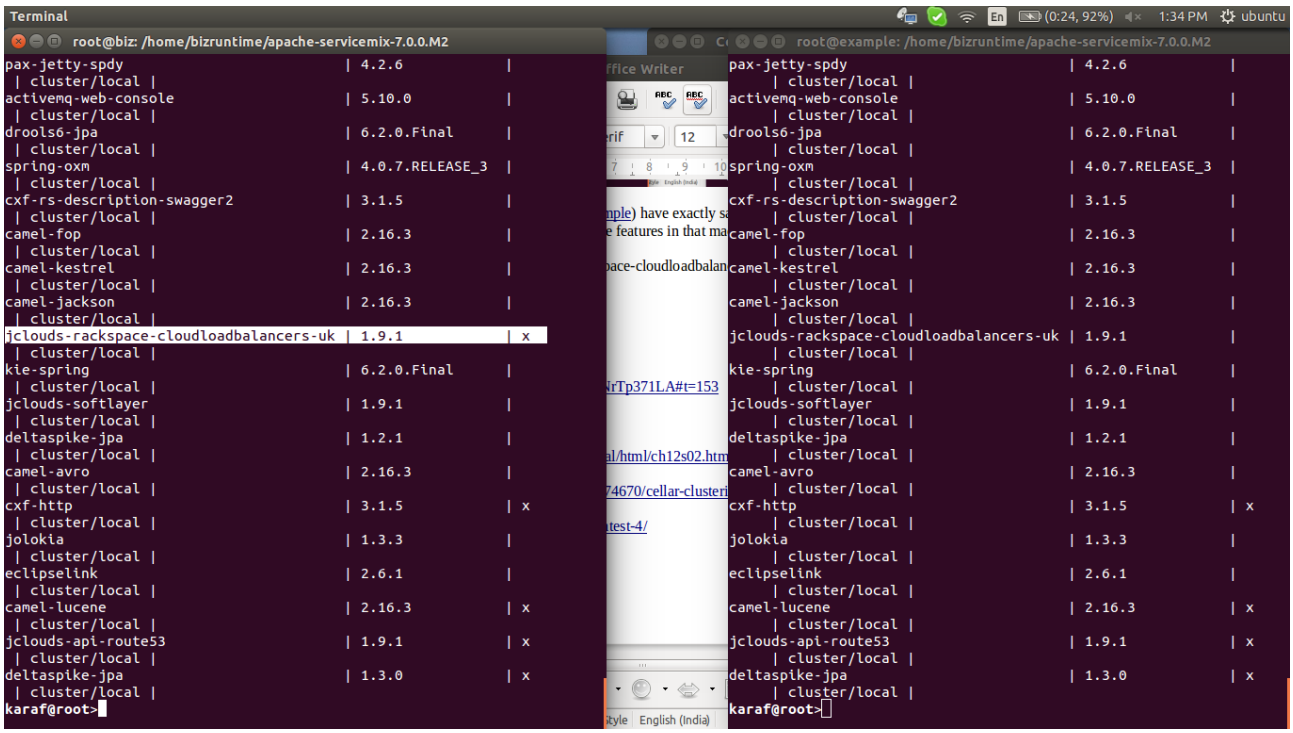
***cluster:feature-list test***

shows info abouts services and applications in test group

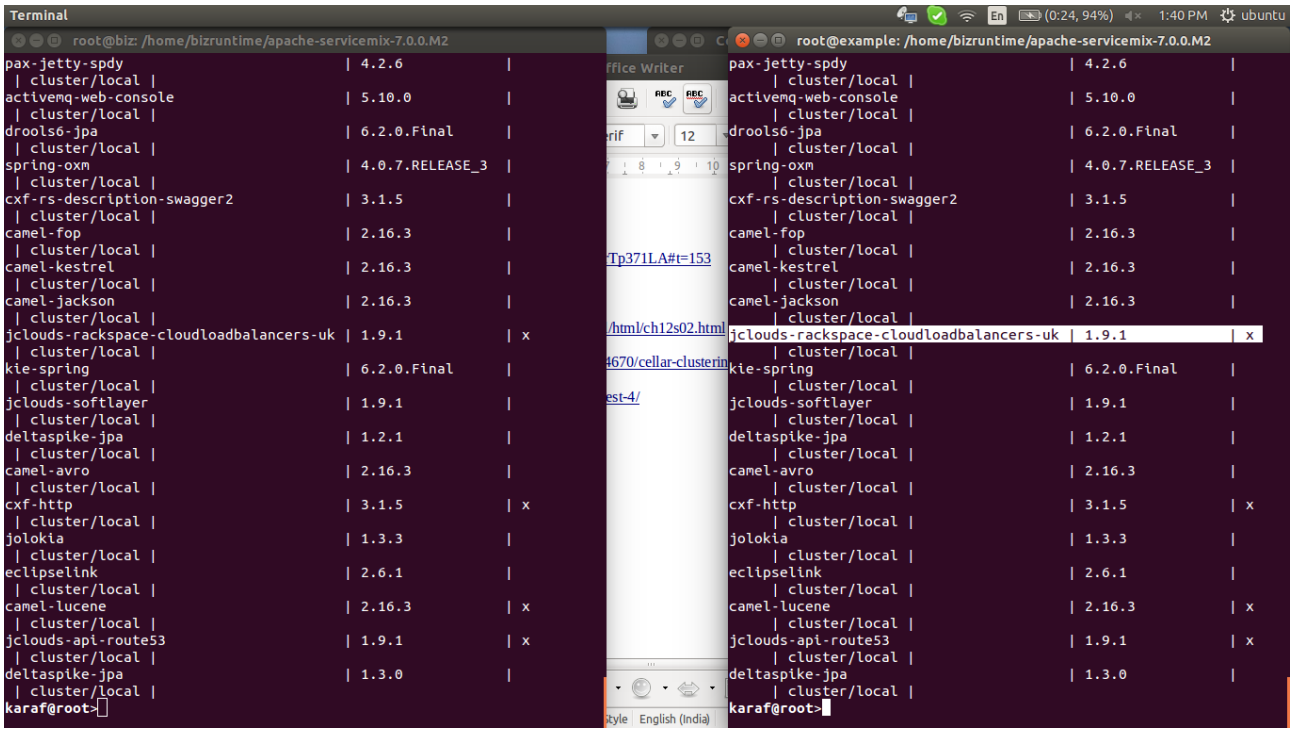



both machine (root@biz and root@example) have exactly same applications initially
install an app in one machine and list the features in that machine again

***cluster:feature-install test jclouds-rackspace-cloudloadbalancers-uk***




now we can see that app in first machine, now list second machine's features




now it reflect on the second machine also
so clustering working properly

links:>>>

https://www.youtube.com/watch?v=HfNrTp371LA#t=153

clustering ::

http://docs.hazelcast.org/docs/2.4/manual/html/ch12s02.html

http://stackoverflow.com/questions/30574670/cellar-clustering-in-servicemix

https://karaf.apache.org/manual/cellar/latest-4/