

Deployment of HA applications Singleton

Goals

After completing this section, students should be able to do the following:

- Configure and implement a highly available application.

The need for cluster singletons

A singleton is a software development design pattern in which there is a single shared instance of an object for the entire application. Common implementation strategies for the singleton design pattern create an instance for each cluster member, and this can have unintended consequences for an application, such as data consistency issues.

Sometimes the application logic requires only one instance to run for the entire cluster. A developer was able to extract this instance to run as a service deployed on a single non-clustered EAP instance, and accessed by a clustered application, but then the service would become a single point of failure.

Many developers believe that a `@Singleton` EJB works correctly as a clustered singleton, but this is NOT true per the EJB specification.

Declares that each JVM in a distributed application server gets an instance. All ways of implementing a singleton using only standard JVM or JEE functions also create one instance per JVM, that is, one instance for each EAP server in a cluster.

When an application requires a singleton to function as a singleton for all clusters, and also be highly available, developers and administrators of EAP should collaborate to extract the singleton parts of the clustered application and adapt it to use one of the following proprietary EAP 7 approaches:

- Singleton implementations (similar to EAP 5 feature)
- Singleton Services (enhanced version of EAP 6 feature)

Both approaches allow an application to function as an active-passive HA application within an EAP cluster. They differ in configuration details and required changes to application packaging and source code.

The singleton subsystem

Both singleton implementations and services are managed by the singleton subsystem. It uses an Infinispan cache to record all known singleton implementations and services and which cluster member runs each application. The EAP server instance running a singleton application is called the master server for that application.

The singleton subsystem can be configured with different election policies that define the master. EAP 7 provides two types of election policy:

Chapter 12. Deploying clustered applications

- **single** – The first node to join the cluster runs the singleton application.
- **random**: A random node is selected to run the singleton application.

An election policy can optionally specify a preferred server, which, when available, will be the master of ALL singleton applications in that policy. The policy refers to the name of the node, as specified by the `jboss.node.name` system property.

The default EAP 7 configuration files offer a simple choice policy called `default` with no preferred server. This policy is shown in the following list:

```
...
    <subsystem xmlns="urn:jboss:domain:singleton:1.0">
      <singleton-policies default="default"> <singleton-
        policy name="default" cache-container="server"> <simple-election-policy/> </
          singleton-policy>

      </singleton-policies> </
    </subsystem>
...
```

The following commands show how to create a new choice policy named `custom` of type `random`, which is preferred by the server named `servera1`:

```
/subsystem=singleton/singleton-policy=custom/election-policy=random:add()
```

```
/subsystem=singleton/singleton-policy=custom/election-policy=random:list-add(\ name=name-preferences,
value=servera1)
```

If the master server fails, the singleton subsystem runs a new election for all singleton applications that were using the failed server as their master. These applications are then restarted on new master servers. Each singleton application must provide its own means to recover or recreate data in memory, lost on the failed master server.

A possible problem with a singleton application is when there is a network partition, also known as a **split-brain scenario**: two farms in the same cluster cannot connect to each other, but servers in one farm have no problem connecting to servers in the same cluster. Each set of servers considers that all the servers in the other set have failed and continues to function as a surviving cluster.

A **split-brain scenario** has two (or more) independent clusters, where there should be only one cluster. This can easily lead to data consistency issues. To prevent this, an election policy can specify a **quorum**; that is, the minimum required number of cluster members. If a quorum is not reached, all remaining cluster members are powered off.

To set up a quorum of three servers, use the following command:

```
/subsystem=singleton/singleton-policy=custom:write-attribute(name=quorum, value=3)
```

The quorum must be at least $N/2+1$, where N is the anticipated total number of cluster members.

Singleton implementations

Singleton implementations are similar to the `HASingletonDeployer` from EAP 5 and earlier. There was no similar feature in EAP 6. It is a way to mark an implementation as singleton across the whole cluster, without the need to use proprietary EAP 7 APIs.

An application package is considered a singleton implementation if it contains the proprietary deployment descriptor `/META-INF/singleton-deployment.xml`. This file refers to the choice policy to be used for the singleton application, and the following example refers to the custom policy from the previous example:

```
<singleton-deployment xmlns="urn:jboss:singleton-deployment:1.0" policy="custom"/>
```

Although this approach does not require an application to use proprietary EAP 7 APIs, it still requires that the original clustered application have access to the extracted singleton application, for example, using remote EJB or JMS calls.

singleton services

Singleton services are implemented in the same way as internal EAP services, that is, their source code uses the basic WildFly APIs. Although more intrusive than singleton implementations, singleton services offer the following advantages in very specific use cases:

- The election policy can be defined by the application, without the need to configure the singleton subsystem.
- Allows EAP modules to start singleton services throughout the cluster, so similar to the old JBossMQ from EAP 4. That is, it allows an EAP module to function as a clustered active-passive service.

Teaching students how to program a singleton service is beyond the scope of this book.



References

For more information about HA Singletons, see the upstream documentation:

Features of the Singleton HA Wildfly

10 <https://docs.jboss.org/author/display/WFLY10/HA+Singleton+Features>

For more information about singleton EJBs and clustered environments:

JSR-345: EJB 3.2 Specification

<https://jcp.org/aboutJava/communityprocess/final/jsr345/index.html>

For more information about developing singleton services:

EJB application development by EAP 7

<https://access.redhat.com/documentation/en/red-hat-jboss-enterprise-application-platform/7.0/developing-ejb-applications/developing-ejb-applications>

Quiz: HA Singleton Implementation

Choose the correct answer to the following questions:

1. What file does a JEE implementation make into an EAP 7 singleton application? (Choose one option).

to. /META-INF/cluster-singleton.xml b. /META-INF/singleton-deployment.xml c. /META-INF/singleton.xml d. /META-INF/singleton-app.xml e. /META-INF/jboss-deployment.xml

2. Which of the following can be configured as part of a clustered singleton election policy? (Choose three options).

to. Choose a random server as the master.
b. Choose a specific server as the master.
Choose the newest running server as the master.
CD Choose the oldest running server as the master.
and. Choose the server with the highest available CPU as the master.

3. Which of these cases can be implemented using a singleton implementation? (Choose two options.)

to. Run an application only on the cluster member with the largest JVM heap and free space.
b. Run an application on a single cluster member to avoid database deadlock issues.
c. Run an application only on the cluster member with a specific hardware accelerator card.
d. Run an application only on the cluster member with the lowest network latency to an external back-end server.
and. Run an application on a single cluster member to avoid multiple concurrent writes to the same shared file.

4. A cluster is expected to have between 7 and 9 active members. What would be the minimum partition size for a singleton master election policy on this cluster? (Choose one option).

to. Three
b. Four
Five
CD Six

Solution

Choose the correct answer to the following questions:

1. What file does a JEE implementation make into an EAP 7 singleton application? (Choose one option).

to. /META-INF/cluster-singleton.xml b. /META-INF/singleton-deployment.xml c. /META-INF/singleton.xml d. /META-INF/singleton-app.xml e. /META-INF/jboss-deployment.xml

2. Which of the following can be configured as part of a clustered singleton election policy? (Choose three options).

to. Choose a random server as the master.
b. Choose a specific server as the master.
c. Choose the newest running server as the master.
d. Choose the oldest running server as the master.
and. Choose the server with the highest available CPU as the master.

3. Which of these cases can be implemented using a singleton implementation? (Choose two options.)

to. Run an application only on the cluster member with the largest JVM heap and free space.
b. Run an application on a single cluster member to avoid database deadlock issues.
c. Run an application only on the cluster member with a specific hardware accelerator card.
d. Run an application only on the cluster member with the lowest network latency to an external back-end server.
and. Run an application on a single cluster member to avoid multiple concurrent writes to the same shared file.

4. A cluster is expected to have between 7 and 9 active members. What would be the minimum partition size for a singleton master election policy on this cluster? (Choose one option).

to. Three
b. Four
c. Five
d. Six