

# Hazelcast - Configuración Java

Para poder configurar el cluster y que no se tomen valores por defecto hay varias maneras de configurarlos:

- Declarativamente a través de un archivo XML (generalmente llamado `hazelcast.xml`)
- Programáticamente desde el cliente Java
- Utilizando propiedades del sistema
- Usando contexto de Spring

A continuación vamos a ver el **modo programático desde Java**. De esta forma, utilizando las clases de la biblioteca de Hazelcast podemos prescindir completamente del archivo XML de configuración mencionado en la teórica.

- **Cluster y Cliente**
  - `GroupConfig`: para indicar las credenciales con `setName` y `setPassword`
- **Cluster**
  - `NetworkConfig`: para indicar cómo se conectará el cluster
    - `InterfacesConfig`: indicar la máscara de red correspondiente
    - `JoinConfig`: indicar con qué estrategia se conectarán entre sí los nodos del cluster. Podemos utilizar `MulticastConfig` o `TcpIpConfig`. En el caso de la última será necesario mencionar una por una todas las direcciones ip posibles de los nodos.
  - `ManagementCenterConfig`: Opcional. Hazelcast ofrece una *webapp* para monitorear en tiempo real el cluster. Nota: Es necesario utilizar una versión coincidente con la de `hazelcast-all` (Ver más abajo).
  - `Config`: clase que encapsula toda la configuración del cluster
- **Cliente**
  - `ClientNetworkConfig`: indicar la dirección IP de al menos uno de los nodos del cluster
  - `ClientConfig`: clase que encapsula toda la configuración del cliente
    - `instanceName`: En caso de que se necesite que cada cliente utilice una colección distinta.

## Ejemplo completo de configuración del cluster

```
public class Server {  
  
    private static final Logger logger = LoggerFactory.getLogger(Server.class);  
  
    public static void main(String[] args) {  
        logger.info("hz-config Server Starting ...");  
  
        // Config  
        Config config = new Config();  
  
        // Group Config  
        GroupConfig groupConfig = new  
GroupConfig().setName("g0").setPassword("g0-pass");  
        config.setGroupConfig(groupConfig);  
    }  
}
```

```
// Network Config
MulticastConfig multicastConfig = new MulticastConfig();

JoinConfig joinConfig = new JoinConfig().setMulticastConfig(multicastConfig);

InterfacesConfig interfacesConfig = new InterfacesConfig()
.setInterfaces(Collections.singletonList("192.168.1.*")).setEnabled(true);

NetworkConfig networkConfig = new
NetworkConfig().setInterfaces(interfacesConfig).setJoin(joinConfig);

config.setNetworkConfig(networkConfig);

// Management Center Config
ManagementCenterConfig managementCenterConfig = new ManagementCenterConfig()
.setUrl("http://localhost:8080/mancenter/")
.setEnabled(true);
config.setManagementCenterConfig(managementCenterConfig);

// Start cluster
Hazelcast.newHazelcastInstance(config);
}
}
```

### Ejemplo completo de configuración del cliente

```
public class Client {

    private static final Logger logger = LoggerFactory.getLogger(Client.class);

    public static void main(String[] args) {
        logger.info("hz-config Client Starting ...");

        // Client Config
        ClientConfig clientConfig = new ClientConfig();

        // Group Config
        GroupConfig groupConfig = new
GroupConfig().setName("g0").setPassword("g0-pass");
        clientConfig.setGroupConfig(groupConfig);

        // Client Network Config
        ClientNetworkConfig clientNetworkConfig = new ClientNetworkConfig();
        String[] addresses = {"192.168.1.51:5701"};
        clientNetworkConfig.addAddress(addresses);
        clientConfig.setNetworkConfig(clientNetworkConfig);

        HazelcastInstance hazelcastInstance =
HazelcastClient.newHazelcastClient(clientConfig);

        String mapName = "testMap";

        IMap<Integer, String> testMapFromMember = hazelcastInstance.getMap(mapName);
        testMapFromMember.set(1, "test1");
    }
}
```

```
    IMap<Integer, String> testMap = hazelcastInstance.getMap(mapName);
    System.out.println(testMap.get(1));

    // Shutdown
    HazelcastClient.shutdownAll();
}

}
```

### Ejemplo completo de cluster y cliente para tests unitarios

```
public class HazelcastTest {

    private TestHazelcastFactory hazelcastFactory;
    private HazelcastInstance member, client;

    @Before
    public void setUp() {
        hazelcastFactory = new TestHazelcastFactory();

        // Group Config
        GroupConfig groupConfig = new
GroupConfig().setName("gX").setPassword("gX-pass");

        // Config
        Config config = new Config().setGroupConfig(groupConfig);

        member = hazelcastFactory.newHazelcastInstance(config);

        // Client Config
        ClientConfig clientConfig = new ClientConfig().setGroupConfig(groupConfig);

        client = hazelcastFactory.newHazelcastClient(clientConfig);
    }

    @Test
    public void simpleTest() {
        String mapName = "testMap";

        IMap<Integer, String> testMapFromMember = member.getMap(mapName);
        testMapFromMember.set(1, "test1");

        IMap<Integer, String> testMap = client.getMap(mapName);
        String value = testMap.get(1);
        Assert.assertEquals("test1", value);
    }

    @After
    public void tearDown() {
        hazelcastFactory.shutdownAll();
    }

}
```

### Dependencias adicionales para tests unitarios

```
<dependency>
  <groupId>com.hazelcast</groupId>
  <artifactId>hazelcast</artifactId>
  <version>${hazelcast.version}</version>
  <classifier>tests</classifier>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>com.hazelcast</groupId>
  <artifactId>hazelcast-client</artifactId>
  <version>${hazelcast.version}</version>
  <classifier>tests</classifier>
  <scope>test</scope>
</dependency>
```

→ Utilizar la versión **3.8.6** de Hazelcast

### Instalación Rápida de Management Center

Descargar el WAR del **Management Center 3.8.5** (compatible con la versión 3.8.6 de Hazelcast) desde aquí: [ZIP \(22.0 MB\)](#)

Se requiere un **Tomcat** que utilice **JDK 8**. Puede utilizar la siguiente imagen de Docker: <https://hub.docker.com/layers/library/tomcat/9.0.82-jre8/images/sha256-14e413b6e4396e531a30c2b515757268e706c7227df678558f48b5e2131616f7?context=explore>

Copiar el .war descargado en el directorio webapps/ correspondiente

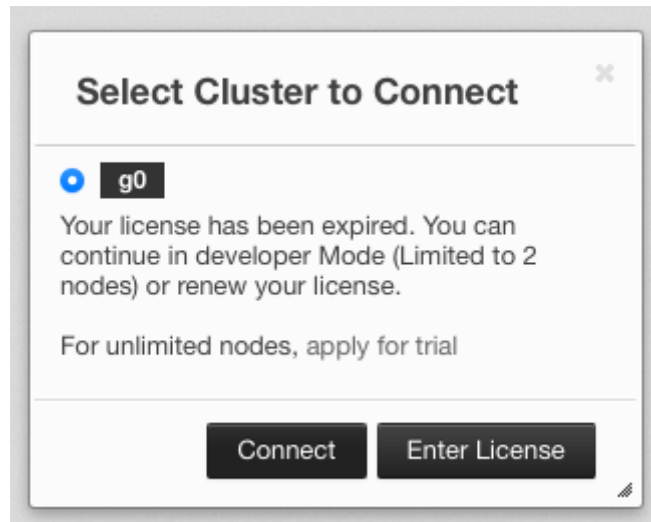
Ingresar a localhost:XXXX/mancenter-3.8.5/ para acceder a la aplicación

Luego deberá crear un usuario

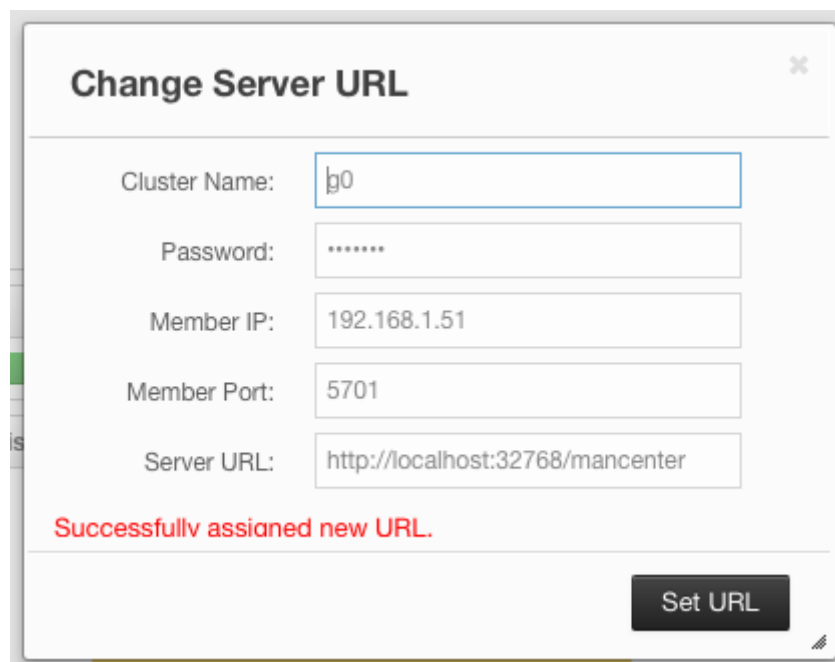
Al correr un nodo, agregar los siguientes flags a la JVM:

```
--add-opens java.management/sun.management=ALL-UNNAMED
--add-opens jdk.management/com.sun.management.internal=ALL-UNNAMED
```

Si ya cuenta con un nodo corriendo localmente, aparecerá una ventana para monitorearlo. La utilidad tiene una limitación de hasta dos nodos.



Sino, desde la opción **Update Cluster URL**, ingresar el **Cluster Name** y **Password** correspondientes a la **GroupConfig** del Server.



Para el ejemplo de la configuración, luego de correr el Client podrá consultar el mapa “testMap” desde la sección “Maps”. Luego en “Map Browser” puede consultar por una clave particular:

### Map Browser

Integer

Browse

<b>Value:</b>	test1	<b>Class:</b>	java.lang.String
<b>Cost:</b>	0.11 KB	<b>Creation Time:</b>	Sun Oct 10 01:06:11 UTC 2021
<b>Expiration Time:</b>	Sun Aug 17 07:12:55 UTC 292278994	<b>Hits:</b>	1
<b>Access Time:</b>	Sun Oct 10 01:06:11 UTC 2021	<b>Update Time:</b>	Sun Oct 10 01:06:11 UTC 2021
<b>Version:</b>	0	<b>Valid:</b>	

### Links Útiles

- Starting the Member and Client  
<https://docs.hazelcast.org/docs/3.8.6/manual/html-single/index.html#starting-the-member-and-client>
- Management Center:  
<https://docs.hazelcast.org/docs/3.8.6/manual/html-single/index.html#management-center>
- Join Configuration:  
<https://docs.hazelcast.org/docs/3.8.6/manual/html-single/index.html#join>
- Creating Cluster Groups  
<https://docs.hazelcast.org/docs/3.8.6/manual/html-single/index.html#creating-cluster-groups>