Hazelcast Management Center Reference Manual

Version 3.10.3

Table of Contents

Welcome to the Hazelcast IMDG Management Center Manual. This manual includes information on how to use Hazelcast Management Center.

Hazelcast Management Center enables you to monitor and manage your cluster members running Hazelcast. In addition to monitoring the overall state of your clusters, you can also analyze and browse your data structures in detail, update map configurations and take thread dumps from members. You can run scripts (JavaScript, Groovy, etc.) and commands on your members with its scripting and console modules.

1. Browser Compatibility

Hazelcast Management Center is tested and works on the following browsers:

- Chrome 65 and newer
- Firefox 57 and newer
- Safari 11 and newer
- Internet Explorer 11 and newer

2. Upgrading Notes

Upgrading to 3.10.x

Starting with Hazelcast Management Center 3.10:

- Hazelcast Management Center's default URL has been changed from `localhost:8080/mancenter` to `localhost:8080/hazelcast-mancenter`.
- Default home directory location has been changed from `<user-home>/mancenter-<version>` to `<user-home>/hazelcast-mancenter-<version>`.
- Name of the WAR file is named has been changed from `mancenter-{version}.war` to `hazelcast-mancenter-{version}.war`.

3. Deploying and Starting

You have two options to start Hazelcast Management Center:

1. Deploy the file `hazelcast-mancenter-3.10.3.war` on your Java application server/container.
2. Start Hazelcast Management Center from the command line. This means Hazelcast cluster members should know the URL of the `hazelcast-mancenter` application before they start.

> ℹ️ Hazelcast Management Center is compatible with Hazelcast cluster members having the same or the previous minor version. For example, Hazelcast Management Center version 3.10.x works with Hazelcast cluster members having version 3.9.x or 3.10.x.

Starting with version 3.10, you need Java Runtime Environment 1.8 or later for running Hazelcast Management Center.

3.1. Starting with WAR File

Here are the steps.

- Download the latest Hazelcast Management Center ZIP from http://www.hazelcast.org/download/ under Management Center section. The ZIP contains the `hazelcast-mancenter-3.10.3.war` file under the directory `hazelcast-management-center-3.10.3`.

- You can directly start `hazelcast-mancenter-3.10.3.war` file from the command line. The following command will start Hazelcast Management Center on port 8080 with context path 'hazelcast-mancenter' (http://localhost:8080/hazelcast-mancenter).

```
java -jar hazelcast-mancenter-3.10.3.war 8080 hazelcast-mancenter
```

3.2. Starting with a License

When starting Management Center from the command line, a license can be provided using the system property `hazelcast.mc.license`. For example by using the command line parameter:

```
java -Dhazelcast.mc.license=<key> -jar hazelcast-mancenter-3.10.3.war
```

When this option is used the license provided will take precedence over any license set and stored previously using the user interface. Previously stored licenses are not affected and will be used again when the Management Center is started without the `hazelcast.mc.license` property. This also means no new license can be stored when the property is used.

3.3. Configure Disk Usage Control

Starting with 3.10, the disk space used by Management Center is constraint to avoid exceeding available disk space. When the set limit is exceeded Management Center deals with this in two ways:

- persisted statistics data is removed, starting with oldest (one month at a time)

- persisted alerts are removed for filters that report further alerts

Usually, either of the above automatically resolves the situation and makes room for new data. Depending on the disk usage configuration and the kind of data that contributes to exceeding the limit it can occur that the limit continues to be exceeded. In this case Management Center will not store new alerts or metrics data. Other data (like configurations and account information) is still stored as they hardly cause larger data volumes.

An active blockage is reported in the UI as an error notification, as shown below:

However, storage operations will not explicitly fail or report errors since this would constantly cause interruptions and error logging - both in the UI and logs.

One way to resolve a blockage is deleting the data manually, e.g., deleting a filter that caused many alerts in the alerts view. Another way is to restart Management Center with a higher limit or in `purge` mode (if not used before).

You can use the following system properties to configure Management Center's disk usage control:

- `-Dhazelcast.mc.disk.usage.mode`: Available values are `purge` and `block`. If the mode is `purge`, persisted statistics data is removed (as stated in the beginning of this section). If it is `block`, persisted statistics data is not removed. Its default value is `purge`.

- `-Dhazelcast.mc.disk.usage.limit`: The *high water mark* in `KB`, `MB` or `GB`. Its default value adapts to the available disk space and the space already used by database files. At a maximum it will default to `512MB` unless existing data already exceeds this maximum. In that case the already used space is used as limit. The minimal allowed limit is `2MB`.

- `-Dhazelcast.mc.disk.usage.interval`: Specifies how often the disk usage is checked to see if it exceeds the limit (`hazelcast.mc.disk.usage.limit`). It is in milliseconds and its default value is `1000` milliseconds. Set values have to be in range of `50` to `5000` ms.

It is important to understand that the limit given is a *soft* limit, a *high water mark*. Management Center will act if it is exceeded but it might be exceeded by a margin between two measurements. Do not set it to the absolute maximum disk space available. A smaller interval increases accuracy but also performance overhead.

In case of a misconfiguration of any of the three properties Management Center will log the problem and abort startup immediately.

3.4. Enabling TLS/SSL when starting with WAR file

When you start Management Center from the command line, it will serve the pages unencrypted by using "http", by default. To enable TLS/SSL, use the following command line parameters when starting the Management Center:

- `-Dhazelcast.mc.tls.enabled=true` (default is false)
- `-Dhazelcast.mc.tls.keyStore=path to your keyStore`
- `-Dhazelcast.mc.tls.keyStorePassword=password for your keyStore`
- `-Dhazelcast.mc.tls.trustStore=path to your trustStore`
- `-Dhazelcast.mc.tls.trustStorePassword=password for your trustStore`

You can leave trust store and trust store password values empty to use the system JVM's own trust store.

Following is an example on how to start Management Center with TLS/SSL enabled from the command line:

```
java -Dhazelcast.mc.tls.enabled=true -Dhazelcast.mc.tls.keyStore=/some/dir/selfsigned
.jks -Dhazelcast.mc.tls.keyStorePassword=yourpassword -jar hazelcast-mancenter-3.10.3
.war
```

You can access Management Center from the following HTTPS URL on port 8443:
https://localhost:8443/hazelcast-mancenter

To override the HTTPS port, you can give it as the second argument when starting Management Center. For example:

```
java -Dhazelcast.mc.tls.enabled=true -Dhazelcast.mc.tls.keyStore=/dir/to/certificate
.jks -Dhazelcast.mc.tls.keyStorePassword=yourpassword -jar hazelcast-mancenter-3.10.3
.war 80 443 hazelcast-mancenter
```

This will start Management Center on HTTP port 80 and HTTPS port 443 with context path /hazelcast-mancenter. Note that accessing port 80 with an http:// prefix will redirect the users to an https:// URL on port 443. It means that the users will use HTTPS regardless of the version of the URL they use.

> You can encrypt the keyStore/trustStore passwords and pass them as command line arguments in encrypted form for improved security. See Variable Replacers for more information.

3.5. Disabling HTTP Port

For increased security, you can disable the HTTP port completely and use the HTTPS port only. Use the following command line argument:

```
-Dhazelcast.mc.tls.enableHttpPort=false
```

> If you want to change the HTTPS port, you still need to provide both port numbers. When this parameter is set, only the HTTPS port will be used. This is required to preserve compatibility with older Management Center versions.

3.6. Mutual authentication

Mutual authentication allows cluster members to have their keyStores and Management Center to have their trustStores so that Management Center can know which members it can trust. To enable mutual authentication, you need to use the following command line parameters when starting the Management Center:

```
-Dhazelcast.mc.tls.mutualAuthentication=REQUIRED
```

And at member side, you need to set the following JVM arguments when starting the member:

```
-Djavax.net.ssl.keyStore=path to your keyStore -Djavax.net.ssl.keyStorePassword
=yourpassword
```

Please see the below example snippet to see the full command to start Management Center:

```
java -Dhazelcast.mc.tls.enabled=true -Dhazelcast.mc.tls.keyStore=path to your keyStore
-Dhazelcast.mc.tls.keyStorePassword=password for your keyStore -Dhazelcast.mc.tls
.trustStore=path to your trustStore -Dhazelcast.mc.tls.trustStorePassword=password for
your trustStore -Dhazelcast.mc.tls.mutualAuthentication=REQUIRED -jar hazelcast-
mancenter-3.10.3.war
```

And the full command to start the cluster member:

```
java -Djavax.net.ssl.keyStore=path to your keyStore -Djavax.net.ssl.keyStorePassword
=yourpassword -Djavax.net.ssl.trustStore=path to your trustStore -Djavax.net.ssl
.trustStorePassword=yourpassword -jar hazelcast.jar
```

The parameter `-Dhazelcast.mc.tls.mutualAuthentication` has two options:

- `REQUIRED`: If the cluster member does not provide a keystore or the provided keys are not included in Management Center's truststore, the cluster member will not be authenticated.
- `OPTIONAL`: If the cluster member does not provide a keystore, it will be authenticated. But if the cluster member provides keys that are not included in Management Center's truststore, the cluster member will not be authenticated.

3.6.1. Excluding Specific TLS/SSL Protocols

When you enable TLS on the Management Center, it will support the clients connecting with any of the TLS/SSL protocols that the JVM supports by default. In order to disable specific protocols, you need to set the `-Dhazelcast.mc.tls.excludeProtocols` command line argument to a comma separated list of protocols to be excluded from the list of supported protocols. For example, to allow only TLSv1.2, you need to add the following command line argument when starting the Management Center:

```
-Dhazelcast.mc.tls.excludeProtocols=SSLv3,SSLv2Hello,TLSv1,TLSv1.1
```

When you specify the above argument, you should see a line similar to the following in the Management Center log:

```
2017-06-21 12:35:54.856:INFO:oejus.SslContextFactory:Enabled Protocols [TLSv1.2] of
[SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2]
```

3.7. Configuring Session Timeout

If you have started Management Center from the command line by using the WAR file, by default, sessions that are inactive for 30 minutes are invalidated. To change this, you can use the `-Dhazelcast.mc.session.timeout.seconds` command line parameter.

For example, the following command starts Management Center with a session timeout period of 1 minute:

```
java -Dhazelcast.mc.session.timeout.seconds=60 -jar hazelcast-mancenter-3.10.3.war
```

If you have deployed Management Center on an application server/container, you can configure the default session timeout period of the application server/container to change the session timeout period for Management Center. If your server/container allows application specific configuration, you can use it to configure the session timeout period for Management Center.

3.8. Enabling Multiple Simultaneous Login Attempts

Normally, a user account on Management Center can't be used from multiple locations at the same time. If you want to allow others to log in, when there's already someone logged in with the same username, you can start Management Center with the `-Dhazelcast.mc.allowMultipleLogin=true` command line parameter.

3.9. Disable Login Configuration

In order to prevent password guessing attacks, logging in is disabled temporarily after a number of failed login attempts. When not configured explicitly, default values are used, i.e., logging in is disabled for 5 seconds when a username is failed to log in consecutively 3 times. During this 5 seconds of period, logging in will not be allowed even when the correct credentials are used. After 5 seconds, the user will be able to log in using the correct credentials.

Assuming the configuration with the default values, if the failed attempts continue (consecutively 3 times) after the period of disabled login passes, this time the disable period will be multiplied by 10 and logging in will be disabled for 50 seconds; the whole process repeats itself until the user logs in successfully. By default, there's no upper limit to the disable period, but can be configured by using the `-Dhazelcast.mc.maxDisableLoginPeriod` parameter.

Here is a scenario, in the given order, with the default values:

1. You try to login with your credentials consecutively 3 times but failed.

2. Logging in is disabled and you have to wait for 5 seconds.

3. After 5 seconds have passed, logging in is enabled.

4. You try to login with your credentials consecutively 3 times but again failed.

5. Logging in is disabled again and this time you have to wait for 50 seconds until your next login attempt.

6. And so on; each 3 consecutive login failures will cause the disable period to be multiplied by 10.

You can configure the number of failed login attempts, initial and maximum duration of the disabled login and the multiplier using the following command line parameters:

- `-Dhazelcast.mc.failedAttemptsBeforeDisableLogin`: Number of failed login attempts that cause the logging in to be disabled temporarily. Default value is 3.

- `-Dhazelcast.mc.initialDisableLoginPeriod`: Initial duration for the disabled login in seconds. Default value is 5.

- `-Dhazelcast.mc.disableLoginPeriodMultiplier`: Multiplier used for extending the disable period in case the failed login attempts continue after disable period passes. Default value is 10.

- `-Dhazelcast.mc.maxDisableLoginPeriod`: Maximum amount of time for the disable login period. This parameter does not have a default value. By default, disabled login period is not limited.

3.10. Forcing Logout on Multiple Simultaneous Login Attempts

If you haven't allowed multiple simultaneous login attempts explicitly, the first user to login with a username stays logged in until that username explicitly logs out or its session expires. In the meantime, no one else can login with the same username. If you want to force logout for the first user and let the newcomer login, you need to start Management Center with the `-Dhazelcast.mc.forceLogoutOnMultipleLogin=true` command line parameter.

3.11. Using a Dictionary to Prevent Weak Passwords

In order to prevent certain words from being included in the user passwords, you can start the Management Center with `-Dhazelcast.mc.security.dictionary.path` command line parameter which points to a text file that contains a word on each line. As a result, the user passwords will not contain any dictionary words, making them harder to guess.

The words in the dictionary need to be at least 3 characters long in order to be used for checking the passwords. The shorter words will be ignored to prevent them from blocking the usage of many password combinations. You can configure the minimum length of words by starting the Management Center with `-Dhazelcast.mc.security.dictionary.minWordLength` command line parameter and setting it to a number.

An example to start the Management Center using the aforementioned parameters is shown below:

```
java -Dhazelcast.mc.security.dictionary.path=/usr/MCtext/pwd.txt -Dhazelcast.mc
.security.dictionary.minWordLength=3 -jar hazelcast-mancenter-3.10.3.war
```

3.12. Starting with an Extra Classpath

You can also start the Management Center with an extra classpath entry (for example, when using JAAS authentication) by using the following command:

```
java -cp "hazelcast-mancenter-3.10.3.war:/path/to/an/extra.jar" Launcher 8080
hazelcast-mancenter
```

On Windows, the command becomes as follows (semicolon instead of colon):

```
java -cp "hazelcast-mancenter-3.10.3.war;/path/to/an/extra.jar" Launcher 8080
hazelcast-mancenter
```

3.13. Starting with Scripts

Optionally, you can use the scripts `start.bat` or `start.sh` to start the Management Center.

3.14. Deploying to Application Server

Or, instead of starting at the command line, you can deploy it to your application server (Tomcat, Jetty, etc.).

If you have deployed `hazelcast-mancenter-3.10.3.war` in your already-SSL-enabled web container, configure `hazelcast.xml` as follows.

```
<management-center enabled="true">
    https://localhost:sslPortNumber/hazelcast-mancenter
</management-center>
```

If you are using an untrusted certificate for your container, which you created yourself, you need to add that certificate to your JVM first. Download the certificate from the browser, after this you can add it to JVM as follows.

```
keytool -import -noprompt -trustcacerts -alias <AliasName> -file <certificateFile> -
keystore $JAVA_HOME/jre/lib/security/cacerts -storepass <Password>
```

3.15. Connecting Hazelcast members to Management Center

After you perform the above steps, make sure that http://localhost:8080/hazelcast-mancenter is up.

Configure your Hazelcast members by adding the URL of your web application to your `hazelcast.xml`. Hazelcast members will send their states to this URL.

```
<management-center enabled="true">
    http://localhost:8080/hazelcast-mancenter
</management-center>
```

You can configure it programmatically as follows.

```
Config config = new Config();
config.getManagementCenterConfig().setEnabled(true);
config.getManagementCenterConfig().setUrl("http://localhost:8080/hazelcast-mancenter"
);

HazelcastInstance hz = Hazelcast.newHazelcastInstance(config);
```

If you enabled TLS/SSL on Management Center, then you will need to configure the members with the relevant keystore & trustore. In that case you expand the above configuration as follows.

```
<management-center enabled="true">
  <url>https://localhost:sslPortNumber/hazelcast-mancenter</url>
  <mutual-auth enabled="true">
    <factory-class-name>
        com.hazelcast.nio.ssl.BasicSSLContextFactory
    </factory-class-name>
    <properties>
        <property name="keyStore">keyStore</property>
        <property name="keyStorePassword">keyStorePassword</property>
        <property name="trustStore">trustStore</property>
        <property name="trustStorePassword">trustStorePassword</property>
        <property name="protocol">TLS</property>
    </properties>
  </mutual-auth>
</management-center>
```

In the example above, Hazelcast's default SSL context factory (BasicSSLContextFactory) is used; you can also provide your own implementation of this factory.

Here are the descriptions for the properties:

- **keystore:** Path of your keystore file. Note that your keystore's type must be JKS.

- **keyStorePassword:** Password to access the key from your keystore file.

- **keyManagerAlgorithm:** Name of the algorithm based on which the authentication keys are provided.

- **keyStoreType:** The type of the keystore. Its default value is JKS.

- **truststore:** Path of your truststore file. The file truststore is a keystore file that contains a collection of certificates trusted by your application. Its type should be JKS.

- **trustStorePassword:** Password to unlock the truststore file.

- **trustManagerAlgorithm:** Name of the algorithm based on which the trust managers are provided.

- **trustStoreType:** The type of the truststore. Its default value is JKS.

- **protocol:** Name of the algorithm which is used in your TLS/SSL. Its default value is TLS.

Available values are:

- SSL

- SSLv2

- SSLv3

- TLS

- TLSv1

- TLSv1.1

- TLSv1.2

See the programmatic configuration example below:

```
Config config = new Config();
SSLContextFactory factory = new BasicSSLContextFactory();

MCMutualAuthConfig mcMutualAuthConfig = new MCMutualAuthConfig().setEnabled(true)
.setFactoryImplementation(factory)
        .setProperty("keyStore", "/path/to/keyStore")
        .setProperty("keyStorePassword", "password")
        .setProperty("keyManagerAlgorithm", "SunX509")
        .setProperty("trustStore", "/path/to/truststore")
        .setProperty("trustStorePassword", "password")
        .setProperty("trustManagerAlgorithm", "SunX509");

ManagementCenterConfig mcc = new ManagementCenterConfig()
    .setEnabled(true)
    .setMutualAuthConfig(mcMutualAuthConfig)
    .setUrl("https://localhost:8443/hazelcast-mancenter");

config.setManagementCenterConfig(mcc);

HazelcastInstance hz = Hazelcast.newHazelcastInstance(config);
```

> ℹ️ For the protocol property, we recommend you to provide SSL or TLS with its version information, e.g., TLSv1.2. Note that if you write only SSL or TLS, your application will choose the SSL or TLS version according to your Java version.

Now you can start your Hazelcast cluster, browse to http://localhost:8080/hazelcast-mancenter or https://localhost:sslPortNumber/hazelcast-mancenter (depending on installation) and setup your administrator account explained in the next section.

3.16. Managing TLS Enabled Clusters

If a Hazelcast cluster is configured to use TLS for communication between its members using a self-signed certificate, Management Center will not be able to perform some of the operations that use the cluster's HTTP endpoints (such as shutting down a member or getting the thread dump of a

member). This is so because self-signed certificates are not trusted by default by the JVM. For these operations to work, you need to configure a `truststore` containing the public key of the self-signed certificate when starting the JVM of Management Center using the following command line parameters:

- `-Dhazelcast.mc.httpClient.tls.trustStore=path to your trust store`
- `-Dhazelcast.mc.httpClient.tls.trustStorePassword=password for your trust store`
- `-Dhazelcast.mc.httpClient.tls.trustStoreType`: Type of the trust store. Its default value is JKS.
- `-Dhazelcast.mc.httpClient.tls.trustManagerAlgorithm`: Name of the algorithm based on which the authentication keys are provided. System default will be used if none provided. You can find out the default by calling `javax.net.ssl.TrustManagerFactory#getDefaultAlgorithm` method.

> You can encrypt the trustStore password and pass it as a command line argument in encrypted form for improved security. See Variable Replacers for more information.

By default, JVM also checks for the validity of the hostname of the certificate. If this test fails, you will see a line similar to the following in the Management Center logs:

```
javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException: No
subject alternative names matching IP address 127.0.0.1 found
```

If you want to disable this check, you will need to start Management Center with the following command line parameter:

```
-Dhazelcast.mc.disableHostnameVerification=true
```

3.16.1. Managing Mutual Authentication Enabled Clusters

If mutual authentication is enabled for the cluster (as described here), Management Center needs to have a keyStore to identify itself. For this, you need to start Management Center with the following command line parameters:

- `-Dhazelcast.mc.httpClient.tls.keyStore=path to your key store`
- `-Dhazelcast.mc.httpClient.tls.keyStorePassword=password for your key store`
- `-Dhazelcast.mc.httpClient.tls.keyStoreType`: Type of the key store. Its default value is JKS.
- `-Dhazelcast.mc.httpClient.tls.keyManagerAlgorithm`: Name of the algorithm based on which the authentication keys are provided. System default will be used if none provided. You can find out the default by calling `javax.net.ssl.KeyManagerFactory#getDefaultAlgorithm` method.

3.17. Configuring Update Interval

You can set a frequency (in seconds) for which Management Center will take information from the Hazelcast cluster, using the element `update-interval` as shown below. `update-interval` is optional and its default value is 3 seconds.

```
<management-center enabled="true" update-interval="3">
    http://localhost:8080/hazelcast-mancenter
</management-center>
```

3.18. Configuring Logging

Management Center uses Logback for its logging. By default, it uses the following configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <layout class="ch.qos.logback.classic.PatternLayout">
            <Pattern>
                %d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} - %msg%n
            </Pattern>
        </layout>
    </appender>

    <root level="INFO">
        <appender-ref ref="STDOUT"/>
    </root>
</configuration>
```

To change the logging configuration, you can create a custom Logback configuration file and start Management Center with the -Dlogback.configurationFile option pointing to your configuration file.

For example, you can create a file named logback-custom.xml with the following content and set logging level to DEBUG. To use this file as the logging configuration, you need to start Management Center with -Dlogback.configurationFile=/path/to/your/logback-custom.xml command line parameter:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>


    <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
        <layout class="ch.qos.logback.classic.PatternLayout">
            <Pattern>
                %d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level %logger{36} - %msg%n
            </Pattern>
        </layout>
    </appender>

    <root level="DEBUG">
        <appender-ref ref="STDOUT"/>
    </root>
</configuration>
```

4. Getting Started

If you have the open source edition of Hazelcast, Management Center can be used for at most 2 members in the cluster. To use it for more members, you need to have either a Management Center license, Hazelcast IMDG Enterprise license or Hazelcast IMDG Enterprise HD license. This license should be entered within the Management Center as described in the following paragraphs.

> ℹ️ Even if you have a Hazelcast IMDG Enterprise or Enterprise HD license key and you set it as explained in the Setting the License Key section, you still need to enter this same license within the Management Center. Please see the following paragraphs to learn how you can enter your license.

Once you browse to `http://localhost:8080/hazelcast-mancenter` and since you are going to use Management Center for the first time, the following dialog box appears.

Configure security

Security Provider:	Default
Username	
Password	
Confirm Password	

Save

> ℹ️ If you already configured security before, a login dialog box appears instead.

It asks you to choose your security provider and create a username and password. Available

security providers are Active Directory, LDAP and JAAS, which are described in the following sections.

Once you press the **Save** button, your administrator account credentials are created and you can log in with your credentials.

If you have more than one cluster that send statistics to Management Center, you can select a cluster to connect by clicking on its name from the list. Otherwise, you will connect to the only cluster that sends statistics automatically upon logging in.

Select cluster
> Cluster A
> Cluster B

Management Center can be used without a license if the cluster that you want to monitor has at most 2 members.

If you have a Management Center license or Hazelcast IMDG Enterprise license, you can enter it by clicking the **Administration** button on the left menu and opening the **Manage License** tab. Here you can enter your license key and press the **Update License** button, as shown below.

Note that a license can likewise be provided using the system property `hazelcast.mc.license` (see Starting with a License for details).

When you try to connect to a cluster that has more than 2 members without entering a license key or if your license key is expired, the following warning message is shown at the top.

Node Limit Exceeded ! Have you entered your license key? If not click here to enter license details or click here to apply for a trial.

If you choose to continue without a license, please remember that Management Center works if your cluster has **at most** two members.

Management Center creates a folder with the name `hazelcast-mancenter` under your `user/home` folder to save data files and above settings/license information. You can change the data folder by

setting the `hazelcast.mancenter.home` system property. Please see the System Properties section to see the description of this property and to learn how to set a system property.

5. Variable Replacers

Variable replacers are used to replace custom strings during loading the configuration, either passed as command line arguments or as part of a configuration file, such as `ldap.properties` or `jaas.properties`. They can be used to mask sensitive information such as usernames and passwords. Of course their usage is not limited to security related information.

Variable replacers implement the interface `com.hazelcast.webmonitor.configreplacer.spi.ConfigReplacer` and they are configured via the following command line arguments:

- `-Dhazelcast.mc.configReplacer.class`: Full class name of the replacer.

- `-Dhazelcast.mc.configReplacer.failIfValueMissing`: Specifies whether the loading configuration process stops when a replacement value is missing. It is an optional attribute and its default value is true.

- Additional command line arguments specific to each replacer implementation. All of the properties for the built-in replacers are explained in the upcoming sections.

The following replacer classes are provided by Hazelcast as example implementations of the `ConfigReplacer` interface. Note that you can also implement your own replacers.

- `EncryptionReplacer`
- `PropertyReplacer`

Each example replacer is explained in the below sections.

5.1. EncryptionReplacer

This example `EncryptionReplacer` replaces encrypted variables by its plain form. The secret key for encryption/decryption is generated from a password which can be a value in a file and/or environment specific values, such as MAC address and actual user data.

Its full class name is `com.hazelcast.webmonitor.configreplacer.EncryptionReplacer` and the replacer prefix is `ENC`. Here are the properties used to configure this example replacer:

- `hazelcast.mc.configReplacer.prop.cipherAlgorithm`: Cipher algorithm used for the encryption/decryption. Its default value is AES.

- `hazelcast.mc.configReplacer.prop.keyLengthBits`: Length (in bits) of the secret key to be generated. Its default value is 128.

- `hazelcast.mc.configReplacer.prop.passwordFile`: Path to a file whose content should be used as a part of the encryption password. When the property is not provided no file is used as a part of the password. Its default value is null.

- `hazelcast.mc.configReplacer.prop.passwordNetworkInterface`: Name of network interface whose MAC address should be used as a part of the encryption password. When the property is not

provided no network interface property is used as a part of the password. Its default value is null.

- `hazelcast.mc.configReplacer.prop.passwordUserProperties`: Specifies whether the current user properties (`user.name` and `user.home`) should be used as a part of the encryption password. Its default value is true.

- `hazelcast.mc.configReplacer.prop.saltLengthBytes`: Length (in bytes) of a random password salt. Its default value is 8.

- `hazelcast.mc.configReplacer.prop.secretKeyAlgorithm`: Name of the secret-key algorithm to be associated with the generated secret key. Its default value is AES.

- `hazelcast.mc.configReplacer.prop.secretKeyFactoryAlgorithm`: Algorithm used to generate a secret key from a password. Its default value is PBKDF2WithHmacSHA256.

- `hazelcast.mc.configReplacer.prop.securityProvider`: Name of a Java Security Provider to be used for retrieving the configured secret key factory and the cipher. Its default value is null.

> Older Java versions may not support all the algorithms used as defaults. Please use the property values supported your Java version.

As a usage example, let's create a password file and generate the encrypted strings out of this file.

1 - Create the password file: `echo '/Za-uG3dDfpd,5.-' > /opt/master-password`

2 - Define the encrypted variables:

```
java -cp hazelcast-mancenter-3.10.3.war \
    -Dhazelcast.mc.configReplacer.prop.passwordFile=/opt/master-password \
    -Dhazelcast.mc.configReplacer.prop.passwordUserProperties=false \
    com.hazelcast.webmonitor.configreplacer.EncryptionReplacer \
    "aPasswordToEncrypt" \

Output:

$ENC{wJxe1vfHTgg=:531:WkAEdSi//YWEbwvVNoU9mUyZ0DE49acJeaJmGalHHfA=}
```

3 - Configure the replacer and provide the encrypted variables as command line arguments while starting Management Center:

```
java \
    -Dhazelcast.mc.configReplacer.class=com.hazelcast.webmonitor.configreplacer
.EncryptionReplacer \
    -Dhazelcast.mc.configReplacer.prop.passwordFile=/opt/master-password \
    -Dhazelcast.mc.configReplacer.prop.passwordUserProperties=false \
    -Dhazelcast.mc.tls.enabled=true \
    -Dhazelcast.mc.tls.keyStore=/opt/mancenter.keystore \
    -Dhazelcast.mc.tls.keyStorePassword=
'$ENC{wJxe1vfHTgg=:531:WkAEdSi//YWEbwvVNoU9mUyZ0DE49acJeaJmGalHHfA=}' \
    -jar hazelcast-mancenter-3.10.3.war
```

5.2. PropertyReplacer

The `PropertyReplacer` replaces variables by properties with the given name. Usually the system properties are used, e.g., `${user.name}`.

Its full class name is `com.hazelcast.webmonitor.configreplacer.PropertyReplacer` and the replacer prefix is empty string ("").

5.3. Implementing Custom Replacers

You can also provide your own replacer implementations. All replacers have to implement the three methods that have the same signatures as the methods of the following interface:

```java
import java.util.Properties;

public interface ConfigReplacer {
    void init(Properties properties);
    String getPrefix();
    String getReplacement(String maskedValue);
}
```

6. Using Management Center with TLS/SSL Only

To encrypt data transmitted over all channels of Management Center using TLS/SSL, make sure you do all of the following:

- Deploy Management Center on a TLS/SSL enabled container or start it from the command line with TLS/SSL enabled. See Installing Management Center.

  ◦ Another option is to place Management Center behind a TLS-enabled reverse proxy. In that case, make sure your reverse proxy sets the necessary HTTP header (`X-Forwarded-Proto`) for resolving the correct protocol.

- Enable TLS/SSL communication to Management Center for your Hazelcast cluster. See
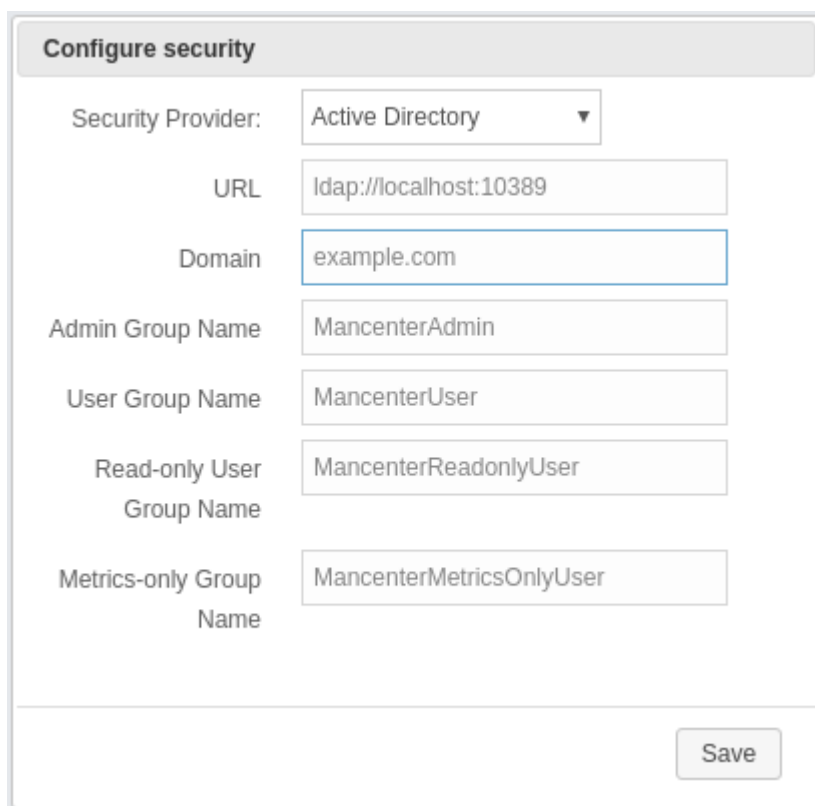
.

- If you're using Clustered JMX on Management center, enable TLS/SSL for it. See Enabling TLS/SSL for Clustered JMX.

- If you're using LDAP authentication, make sure you use LDAPS or enable the "Start TLS" field. See LDAP Authentication.

7. Authentication Options

7.1. Active Directory Authentication

You can use your existing Active Directory server for authentication/authorization on Management Center. In the "Configure Security" page, select **Active Directory** from the "Security Provider" combo, and the following form page appears:

Provide the details in this form for your Active Directory server:

- **URL:** URL of your Active Directory server, including schema (`ldap://` or `ldaps://`) and port.

- **Domain:** Domain of your organization on Active Directory.

- **Admin Group Name:** Members of this group will have admin privileges on the Management Center.

- **User Group Name:** Members of this group will have read and write privileges on the Management Center.

- **Read-only User Group Name:** Members of this group will have only read privilege on the Management Center.

- **Metrics-only Group Name:** Members of this group will have the privilege to see only the metrics on the Management Center.

Once configured, Active Directory settings are saved in a file named `ldap.properties` under the `hazelcast-mancenter3.10.3` folder mentioned in the previous section. If you want to update your settings afterwards, you need to update `ldap.properties` file and click "Reload Security Config" button on the login page.

7.2. JAAS Authentication

You can use your own `javax.security.auth.spi.LoginModule` implementation for authentication/authorization on Management Center. In the "Configure Security" page, select **JAAS** from the "Security Provider" combo box, and the following page appears:

Provide the details in this form for your JAAS `LoginModule` implementation:

- **Login Module Class**: Fully qualified class name of your `javax.security.auth.spi.LoginModule` implementation

- **Admin Group:** Members of this group will have admin privileges on the Management Center.

- **User Group:** Members of this group will have read and write privileges on the Management Center.

- **Read-only User Group:** Members of this group will have only read privilege on the Management Center.

- **Metrics-only Group:** Members of this group will have the privilege to see only the metrics on the Management Center.

Following is an example implementation. Note that we return two `java.security.Principal` instances; one of them is the username and the other one is a group name, which you will use when configuring JAAS security as described above.

```java
import javax.security.auth.Subject;
import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.NameCallback;
import javax.security.auth.callback.PasswordCallback;
import javax.security.auth.login.LoginException;
import javax.security.auth.spi.LoginModule;
import java.security.Principal;
import java.util.Map;

public class SampleLoginModule implements LoginModule {
    private Subject subject;
    private String password;
    private String username;

    @Override
    public void initialize(Subject subject, CallbackHandler callbackHandler, Map<String, ?> sharedState, Map<String, ?> options) {
        this.subject = subject;

        try {
            NameCallback nameCallback = new NameCallback("prompt");
            PasswordCallback passwordCallback = new PasswordCallback("prompt", false);

            callbackHandler.handle(new Callback[] {nameCallback, passwordCallback });

            password = new String(passwordCallback.getPassword());
            username = nameCallback.getName();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    @Override
    public boolean login() throws LoginException {
        if (!username.equals("emre")) {
            throw new LoginException("Bad User");
        }

        if (!password.equals("pass1234")) {
            throw new LoginException("Bad Password");
        }

        subject.getPrincipals().add(new Principal() {
            public String getName() {
                return "emre";
            }
        });

        subject.getPrincipals().add(new Principal() {
            public String getName() {
```

```
                    return "MancenterAdmin";
                }
            });

            return true;
        }

        @Override
        public boolean commit() throws LoginException {
            return true;
        }

        @Override
        public boolean abort() throws LoginException {
            return true;
        }

        @Override
        public boolean logout() throws LoginException {
            return true;
        }
    }
```

7.3. LDAP Authentication

You can use your existing LDAP server for authentication/authorization on Management Center. In the "Configure Security" page, select **LDAP** from the "Security Provider" combo box, and the following form page appears:

Provide the details in this form for your LDAP server:

- **URL:** URL of your LDAP server, including schema (`ldap://` or `ldaps://`) and port.

- **Distinguished name (DN) of user:** DN of a user that has admin privileges on the LDAP server. It is used to connect to the server when authenticating users.

- **Search base DN:** Base DN to use for searching users/groups.

- **Additional user DN:** Appended to "Search base DN" and used for finding users.

- **Additional group DN:** Appended to "Search base DN" and used for finding groups.

- **Admin Group Name:** Members of this group will have admin privileges on the Management Center.

- **User Group Name:** Members of this group will have read and write privileges on the Management Center.

- **Read-only User Group Name:** Members of this group will have only read privilege on the Management Center.

- **Metrics-only Group Name:** Members of this group will have the privilege to see only the metrics on the Management Center.

- **Start TLS:** Enable if your LDAP server uses Start TLS.

- **User Search Filter:** LDAP search filter expression to search for users. For example, `uid={0}` searches for a username that matches with the `uid` attribute.

- **Group Search Filter:** LDAP search filter expression to search for groups. For example, `uniquemember={0}` searches for a group that matches with the `uniquemember` attribute.

> ℹ️ Values for Admin, User, Read-only and Metrics-Only Group Names must be given as plain names. They should not contain any LDAP attributes such as `CN`, `OU` and `DC`.

Once configured, LDAP settings are saved in a file named `ldap.properties` under the `hazelcast-mancenter3.10.3` folder mentioned in the previous section. If you want to update your settings afterwards, you need to update `ldap.properties` file and click "Reload Security Config" button on the login page.

7.3.1. Enabling TLS/SSL for LDAP

If your LDAP server is using `ldaps` (LDAP over SSL) protocol or Start TLS operation, use the following command line parameters for your Management Center deployment:

- `-Dhazelcast.mc.ldap.ssl.trustStore=path to your truststore`: This truststore needs to contain the public key of your LDAP server.

- `-Dhazelcast.mc.ldap.ssl.trustStorePassword=password for your truststore`

- `-Dhazelcast.mc.ldap.ssl.trustStoreType`: Type of the trust store. Its default value is JKS.

- `-Dhazelcast.mc.ldap.ssl.trustManagerAlgorithm`: Name of the algorithm based on which the authentication keys are provided. System default will be used if none provided. You can find out the default by calling `javax.net.ssl.TrustManagerFactory#getDefaultAlgorithm` method.

7.3.2. Password Encryption

By default, the password that you use in LDAP configuration is saved on the `ldap.properties` file in clear text. This might pose a security risk. To store the LDAP password in encrypted form, we offer the following two options:

- **Provide a KeyStore password:** This will create and manage a Java KeyStore under the Management Center home directory. The LDAP password will be stored in this KeyStore in encrypted form.

- **Configure an external Java KeyStore:** This will use an existing Java KeyStore. This option might also be used to store the password in an HSM that provides a Java KeyStore API.

When you do either, the LDAP password you enter on the initial configuration UI dialog will be stored in encrypted form in a Java KeyStore instead of the `ldap.properties` file.

You can also encrypt the password before saving it on `ldap.properties`. See [Variable Replacers](#) for more information.

Providing a Master Key for Encryption

There are two ways to provide a master key for encryption:

- If you deploy Management Center on an application server, you need to set `MC_KEYSTORE_PASS` environment variable before starting Management Center. This option is less secure. You should clear the environment variable once you make sure you can log in with your LDAP credentials to minimize the security risk.

- If you're starting Management Center from the command line, you can start it with `-Dhazelcast.mc.askKeyStorePassword`. Management Center will ask for the KeyStore password upon start and use it as a password for the KeyStore it creates. This option is more secure as it only stores the KeyStore password in the memory.

By default, Management Center will create a Java KeyStore file under the Management Center home directory with the name `mancenter.jceks`. You can change the location of this file by using the `-Dhazelcast.mc.keyStore.path=/path/to/keyStore.jceks` JVM argument.

Configuring an External Java KeyStore

If you don't want Management Center to create a KeyStore for you and use an existing one that you've created before (or an HSM), set the following JVM arguments when starting Management Center:

- `-Dhazelcast.mc.useExistingKeyStore=true`: Enables use of an existing KeyStore.

- `-Dhazelcast.mc.existingKeyStore.path=/path/to/existing/keyStore.jceks`: Path to the KeyStore. You do not have to set it if you use an HSM.

- `-Dhazelcast.mc.existingKeyStore.pass=somepass`: Password for the KeyStore. You do not have to set it if HSM provides another means to unlock HSM.

- `-Dhazelcast.mc.existingKeyStore.type=JCEKS`: Type of the KeyStore.

- `-Dhazelcast.mc.existingKeyStore.provider=com.yourprovider.MyProvider`: Provider of the KeyStore. Leave empty to use the system provider. Specify the class name of your HSM's `java.security.Provider` implementation if you use an HSM.
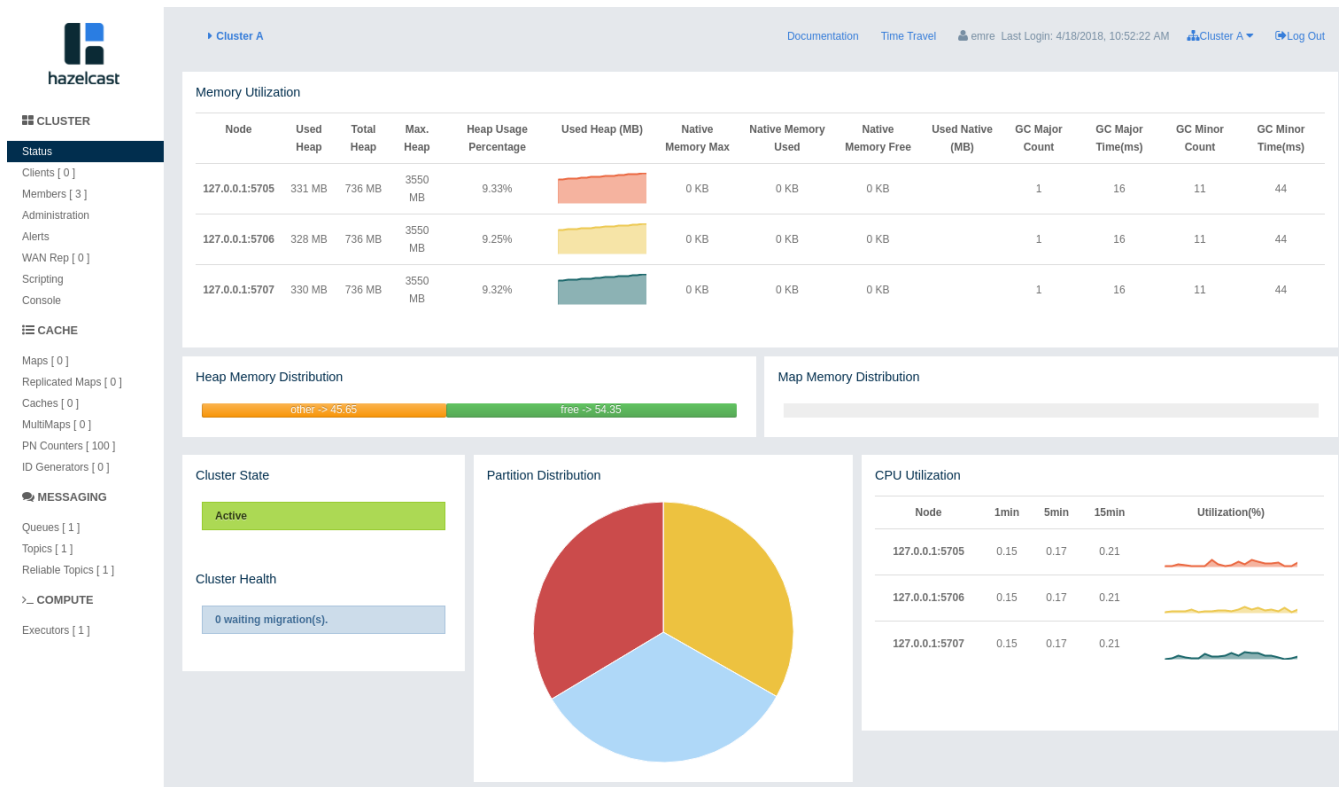
Make sure your KeyStore supports storing `SecretKey`s.

7.3.3. Updating Encrypted Passwords

You can use one of the `updateLdapPassword.sh` or `updateLdapPassword.bat` scripts to update the encrypted LDAP password stored in the KeyStore. It will ask for information about the KeyStore such as its location and password. It will then ask for the new LDAP password that you want to use. After updating the LDAP password, you'll need to click **Reload Security Configuration** button on the main screen.

8. User Interface Overview

Once the page is loaded after selecting a cluster, Status Page appears as shown below.
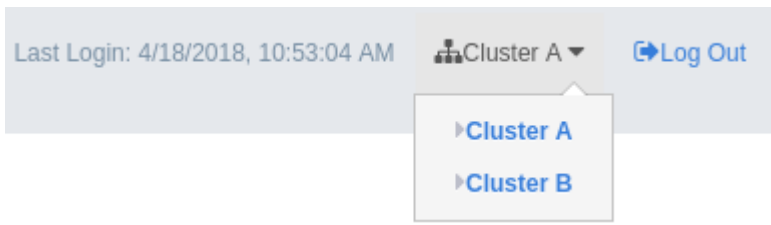
This page provides the fundamental properties of the selected cluster which are explained in the Status Page section. The page has a toolbar on the top and a menu on the left.

8.1. Toolbar
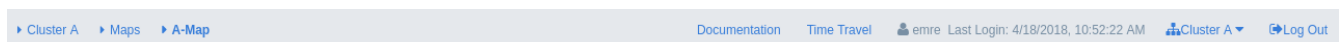
The toolbar has the following elements:

- **Navigation Breadcrumb**: The leftmost element is the navigation breadcrumb that you can use to navigate to the previously opened pages. For example, while you're on the page where you're viewing a Map, you can click the **Maps** link to go back to the page where all Map instances are listed.

- **Documentation**: Opens the Management Center documentation in a new browser tab.

- **Time Travel**: Sees the cluster's situation at a time in the past. Please see the Time Travel section.

- **User name and last login time**: The current user's name and last login time is shown for security purposes.

- **Cluster Selector**: Switches between clusters. When clicked, a drop down list of clusters appears.

The user can select any cluster and once selected, the page immediately loads with the selected cluster's information.

- **Logout**: Closes the current user's session.

8.2. Menu

The Home Page includes a menu on the left which lists the distributed data structures in the cluster, cluster members and clients connected to the cluster (numbers in square brackets show the instance count for each entity), as shown below. You can also see an overview state of your cluster, create alerts, execute codes and perform user/license operations using this menu:

CLUSTER

Status
Clients [0]
Members [2]
Administration
Alerts
WAN Replication [2]
Scripting
Console

CACHE

Maps [3]

Replicated Maps [0]

Caches [0]

MultiMaps [0]

PN Counters [0]

ID Generators [0]

MESSAGING

Queues [0]

Topics [0]

Reliable Topics [0]

>_ COMPUTE

Executors [0]

> 🛈 Distributed data structures will be shown there when the proxies are created for them.

> 🛈 WAN Replication button is only visible with Hazelcast IMDG Enterprise license.

Below is the list of menu items with links to their explanations.

- Status

- Clients

- Members

- Administration

* [Alerts](#)

* [WAN Replication](#)

* [Scripting](#)

* [Console](#)

* [Maps](#)

* [Replicated Maps](#)

* [Caches](#)

* [MultiMaps](#)

* [PN Counters](#)

* [ID Generators](#)

* [Queues](#)

* [Topics](#)

* [Reliable Topics](#)

* [Executors](#)

9. Status Page

This is the first page appearing after logging in. It gives an overview of the connected cluster. The following subsections describe each portion of the page.

9.1. Memory Utilization

This part of the page provides information related to memory usages for each member, as shown below.

Memory Utilization

Node	Used Heap	Total Heap	Max. Heap	Heap Usage Percentage	Used Heap	Native Memory Max	Native Memory Used	Native Memory Free	Used Native (MB)	GC Major Count	GC Major Time(ms)	GC Minor Count	GC Minor Time(ms)
127.0.0.1:5701	250 MB	374 MB	3550 MB	7.05%		0 KB	0 KB	0 KB		1	41	17	87
127.0.0.1:5702	250 MB	374 MB	3550 MB	7.05%		0 KB	0 KB	0 KB		1	41	17	87

The first column lists the members with their IPs and ports. The next columns show the used and free memories out of the total memory reserved for Hazelcast usage, in real-time. The **Max. Heap** column lists the maximum memory capacity of each member and the **Heap Usage Percentage** column lists the percentage value of used memory out of the maximum memory. The **Used Heap** column shows the memory usage of members graphically. When you move the mouse cursor on a desired graph, you can see the memory usage at the time where the cursor is placed. Graphs under this column shows the memory usages approximately for the last 2 minutes.

9.2. Heap Memory Distribution

This part of the page graphically provides the cluster wise breakdown of heap memory, as shown below. The blue area is the heap memory used by maps (including all owned/backup entires and any near cache usage). The dark yellow area is the heap memory used by both non-Hazelcast entities and all Hazelcast entities except the map (i.e. the heap memory used by all entities subtracted by the heap memory used by map). The green area is the free heap memory out of the whole cluster's total committed heap memory.
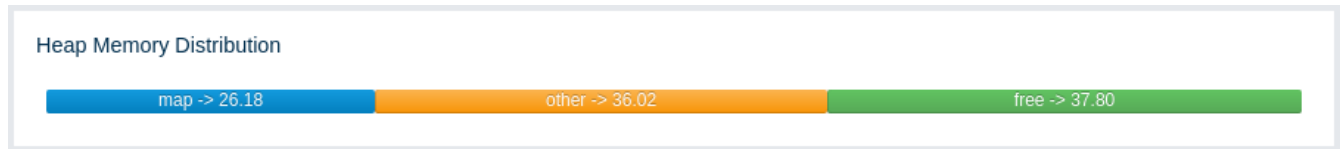
In the above example, you can see 26.18% of the total heap memory is used by Hazelcast maps, 36.02% is used by both non-Hazelcast entities and all Hazelcast entities except the map and 37.80% of the total heap memory is free.

9.3. Map Memory Distribution

This part provides the percentage values of the memories used by each map, out of the total cluster memory reserved for all Hazelcast maps.
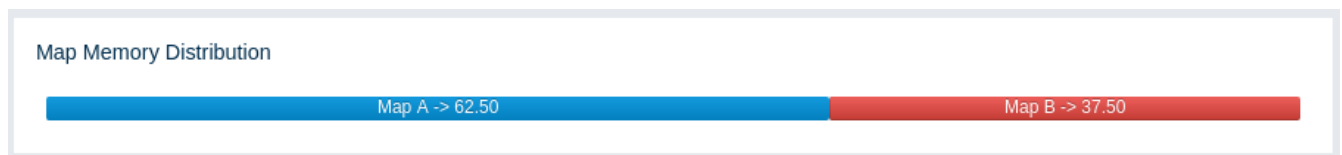
In the above example, you can see 62.50% of the total map memory is used by **Map A** and 37.50% is used by **Map B**.

9.4. Cluster State/Health

This part shows the current cluster state and the cluster's health. For more information on cluster states, see Cluster State. Cluster health shows how many migrations are taking place currently.
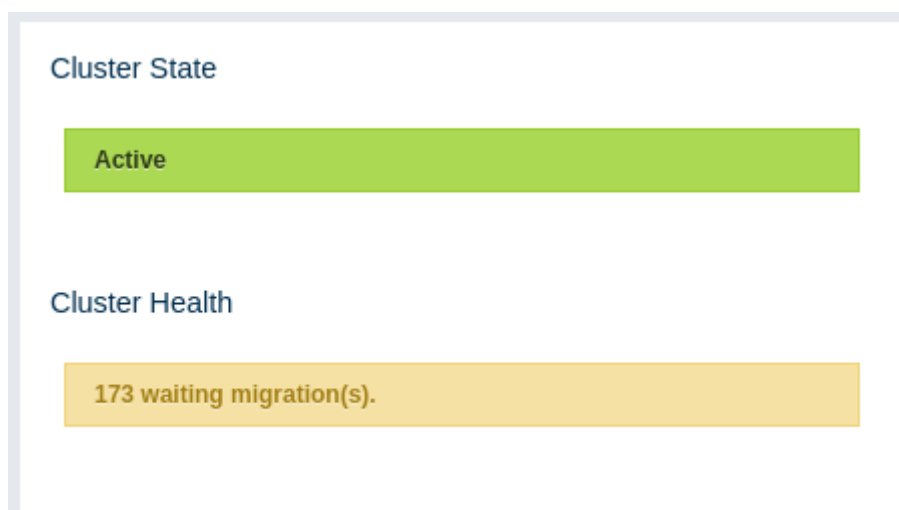
9.5. Partition Distribution

This pie chart shows what percentage of partitions each cluster member has, as shown below.
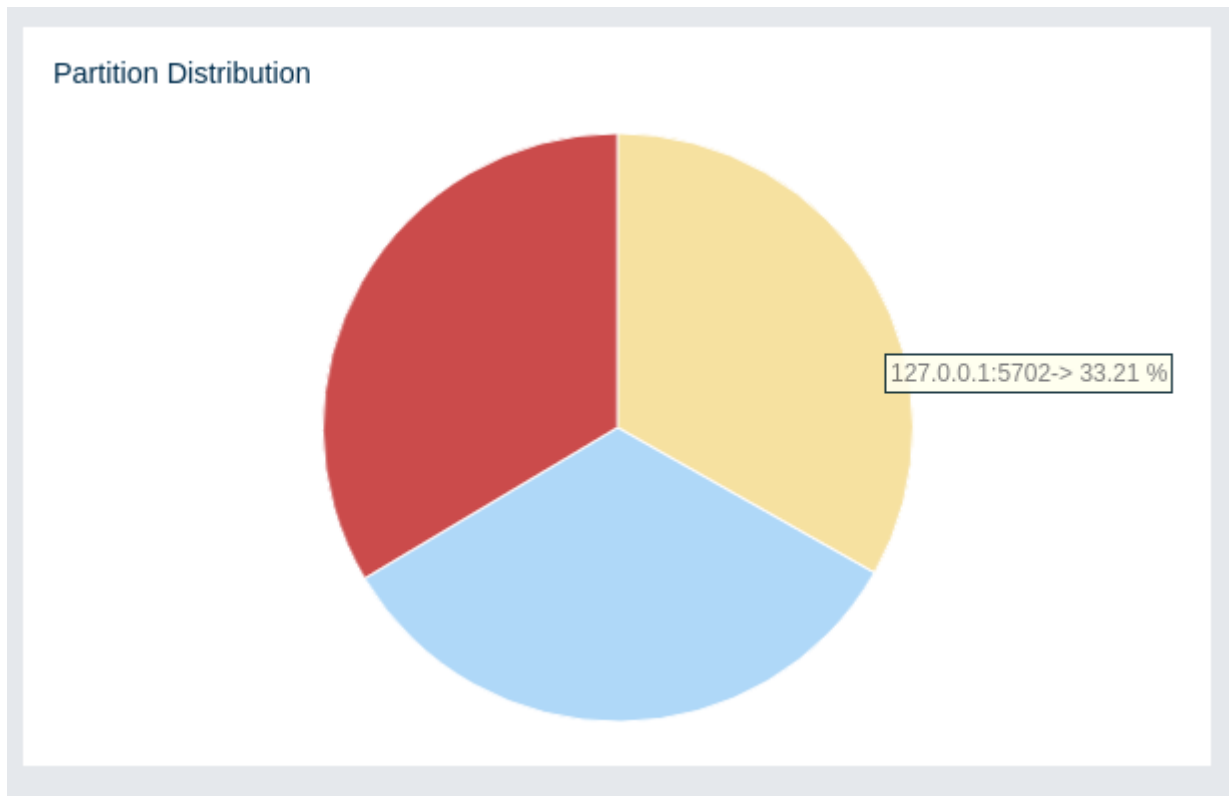
You can see each member's partition percentages by placing the mouse cursor on the chart. In the above example, you can see the member "127.0.0.1:5702" has 33.21% of the total partition count (which is 271 by default and configurable, please see the `hazelcast.partition.count` property explained in the System Properties section.

> The partition distribution pie chart will show no information until you create your distributed objects. When you add new members to your cluster, there will be no partition migration since partitions do not exist yet. Once you connect to your cluster and, for example, create a map (using `hazelcastInstance.getMap()`), only then this pie chart starts to show partition distribution information.

9.6. CPU Utilization

This part of the page provides load and utilization information for the CPUs for each cluster member, as shown below.

CPU Utilization

Node	1min	5min	15min	Utilization(%)
127.0.0.1:5701	0.25	0.24	0.18	
127.0.0.1:5702	0.25	0.24	0.18	

The first column lists the members with their IPs and ports. The next columns list the system load averages on each member for the last 1, 5 and 15 minutes. These average values are calculated as the sum of the count of runnable entities running on and queued to the available CPUs averaged over the last 1, 5 and 15 minutes. This calculation is operating system specific, typically a damped time-dependent average. If system load average is not available, these columns show negative values.

The last column (**Utilization(%)**) graphically shows the recent load on the CPUs. When you move the mouse cursor on a chart, you can see the CPU load at the time where the cursor is placed. Charts under this column shows the CPU loads approximately for the last 2 minutes. If recent CPU load is not available, you will see a negative value.

10. Monitoring Caches

You can see a list of all the caches in your cluster by clicking on the **Caches** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⇕	Entries ⇕	Hits ⇕	Misses ⇕
⧉ cache-42	19	0	0
⧉ cache-43	17	0	0
⧉ cache-44	7	0	0
⧉ cache-45	18	0	0
⧉ cache-46	14	0	0
⧉ cache-47	12	0	0
⧉ cache-48	8	0	0
⧉ cache-49	31	0	0
⧉ cache-5	6	0	0
⧉ cache-50	15	0	0
⧉ cache-51	13	0	0
⧉ cache-52	24	0	0
⧉ cache-53	12	0	0
⧉ cache-54	8	0	0
⧉ cache-55	15	0	0
⧉ cache-56	29	0	0
⧉ cache-57	12	0	0
⧉ cache-58	6	0	0
⧉ cache-59	13	0	0
⧉ cache-6	7	0	0

« ‹ 1 2 **3** 4 5 6 › »

You can filter the caches shown and you can also sort the table by clicking on the column headers. Clicking on the cache name will open a new page for monitoring that cache instance on the right, as shown below.

Cache Statistics Data Table

# ▲	Members ⇕	Entries ⇕	Gets ⇕	Puts ⇕	Removals ⇕	Evictions ⇕	Hits ⇕	Misses ⇕	Avg Get Time (μs) ⇕	Avg Put Time (μs) ⇕
1	127.0.0.1:5705	6	0	6	0	0	0	0	0.00	13.76
2	127.0.0.1:5706	9	0	9	0	0	0	0	0.00	11.63
#	TOTAL	15	0	15	0	0	0	0	-	-

First Previous 1 Next Last

Cache Throughput Data Table   ⚙Last Minute▾

# ▲	Members ⇕	Puts/s ⇕	Gets/s ⇕	Removes/s ⇕
1	127.0.0.1:5705	0	0	0
2	127.0.0.1:5706	0	0	0

First Previous 1 Next Last

On top of the page, four charts monitor the **Gets**, **Puts**, **Removals** and **Evictions** in real-time. The X-axis of all the charts show the current system time. To open a chart as a separate dialog, click on the ⤢ button placed at the top right of each chart.

Under these charts is the Cache Statistics Data Table. From left to right, this table lists the IP

addresses and ports of each member, and the entry, get, put, removal, eviction, and hit and miss counts per second in real-time.

You can navigate through the pages using the buttons at the bottom right of the table (**First, Previous, Next, Last**). You can ascend or descend the order of the listings in each column by clicking on column headings.

Under the Cache Statistics Data Table, there is Cache Throughput Data Table.

From left to right, this table lists:

- the IP address and port of each member,
- the put/s, get/s and remove/s operation rates on each member.

You can select the period in the combo box placed at the top right corner of the window, for which the table data will be shown. Available values are **Since Beginning**, **Last Minute**, **Last 10 Minutes** and **Last 1 Hour**.

> ℹ️ You need to enable the statistics for caches to monitor them in the Management Center. Use the `<statistics-enabled>` element or `setStatisticsEnabled()` method in declarative or programmatic configuration, respectively, to enable the statistics. Please refer to the JCache Declarative Configuration section for more information.

11. Managing Maps

You can see a list of all the maps in your cluster by clicking on the **Maps** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⇕	Entries ⇕	EntryMemory ⇕	BackupMemory ⇕	Events ⇕	Hits ⇕	Locks ⇕	DirtyEntries ⇕
map-45	50	7.62 kB	7.62 kB	0	0	0	0
map-46	50	7.62 kB	7.62 kB	0	0	0	0
map-47	50	7.62 kB	7.62 kB	0	0	0	0
map-48	50	7.62 kB	7.62 kB	0	0	0	0
map-49	50	7.62 kB	7.62 kB	0	0	0	0
map-5	50	7.62 kB	7.62 kB	0	0	0	0
map-50	50	7.62 kB	7.62 kB	0	0	0	0
map-51	50	7.62 kB	7.62 kB	0	0	0	0
map-52	50	7.62 kB	7.62 kB	0	0	0	0
map-53	50	7.62 kB	7.62 kB	0	0	0	0
map-54	50	7.62 kB	7.62 kB	0	0	0	0
map-55	50	7.62 kB	7.62 kB	0	0	0	0
map-56	50	7.62 kB	7.62 kB	0	0	0	0
map-57	50	7.62 kB	7.62 kB	0	0	0	0
map-58	50	7.62 kB	7.62 kB	0	0	0	0
map-59	50	7.62 kB	7.62 kB	0	0	0	0
map-6	50	7.62 kB	7.62 kB	0	0	0	0
map-60	50	7.62 kB	7.62 kB	0	0	0	0
map-61	50	7.62 kB	7.62 kB	0	0	0	0
map-62	50	7.62 kB	7.62 kB	0	0	0	0

« ‹ 1 2 **3** 4 5 6 › »

You can filter the maps shown and you can also sort the table by clicking on the column headers. Clicking on a map name will open a new page for monitoring that map instance on the right, as shown below.

Size		Throughput		Memory		Map Browser	Map Config

Map Memory Data Table (In-Memory Format: BINARY)

# ▲	Members	Entries ⇕	Entry Memory ⇕	Backups ⇕	Backup Memory ⇕	Events ⇕	Hits ⇕	Locks ⇕	Dirty Entries ⇕
1	127.0.0.1:5705	19	2.89 KB	31	4.72 KB	0	0	0	0
2	127.0.0.1:5706	31	4.72 KB	19	2.89 KB	0	0	0	0
#	TOTAL	50	7.62 KB	50	7.62 KB	0	0	0	0

First Previous 1 Next Last

Map Throughput Data Table   ⚙Last Minute▾

#▲	Members ⇕	Puts/s ⇕	Gets/s ⇕	Removes/s⇕	Avg Put Lat.⇕	Avg Get Lat.⇕	Avg Remove Lat. ⇕	Max Put Lat. ⇕	Max Get Lat. ⇕	Max Remove Lat. ⇕
1	127.0.0.1:5705	0	0	0	0	0	0	0	0	0
2	127.0.0.1:5706	0	0	0	0	0	0	0	0	0

First Previous 1 Next Last

Near Cache Data Table

# ▲	Members ⇕	Entries ⇕	Entry Memory ⇕	Hits ⇕	Misses ⇕	Ratio ⇕
No data available in table						

First Previous Next Last

The below subsections explain the portions of this window.

11.1. Map Browser

Use the Map Browser tool to retrieve properties of the entries stored in the selected map. To open the Map Browser tool, click on the **Map Browser** button, located at the top right of the window. Once opened, the tool appears as a dialog, as shown below.

Map Browser			✖

2	Integer	⬍	Browse

Value:	2	Class:	
Cost:	0.11 KB	Creation Time:	Thu Apr 19 16:56:33 MSK 2018
Expiration Time:	NOT_AVAILABLE	Hits:	0
Access Time:	NOT_AVAILABLE	Update Time:	Thu Apr 19 16:56:33 MSK 2018
Version:	0		

Once the key and the key's type are specified and the **Browse** button is clicked, the key's properties along with its value are listed.

11.2. Map Config

Use the Map Config tool to set the selected map's attributes, such as the backup count, TTL, and eviction policy. To open the Map Config tool, click on the **Map Config** button, located at the top right of the window. Once opened, the tool appears as a dialog, as shown below.

You can change any attribute and click the **Update** button to save your changes.

11.3. Map Monitoring

Besides the Map Browser and Map Config tools, the map monitoring page has monitoring options that are explained below. All of these options perform real-time monitoring.

On top of the page, small charts monitor the size, throughput, memory usage, backup size, etc. of the selected map in real-time. The X-axis of all the charts show the current system time. You can select other small monitoring charts using the ⚙ button at the top right of each chart. When you click the button, the monitoring options are listed, as shown below.
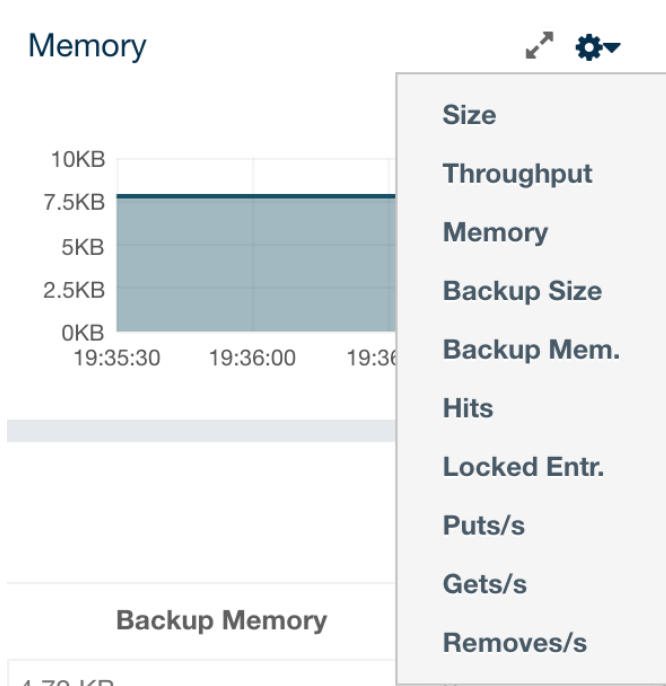
When you click on a desired monitoring, the chart is loaded with the selected option. To open a chart as a separate dialog, click on the ⤢ button placed at the top right of each chart. The monitoring charts below are available:

- **Size**: Monitors the size of the map. Y-axis is the entry count (should be multiplied by 1000).

- **Throughput**: Monitors get, put and remove operations performed on the map. Y-axis is the operation count.

- **Memory**: Monitors the memory usage on the map. Y-axis is the memory count.

- **Backups**: Chart loaded when "Backup Size" is selected. Monitors the size of the backups in the map. Y-axis is the backup entry count (should be multiplied by 1000).

- **Backup Memory**: Chart loaded when "Backup Mem." is selected. Monitors the memory usage of the backups. Y-axis is the memory count.

- **Hits**: Monitors the hit count of the map.

- **Puts/s, Gets/s, Removes/s**: These three charts monitor the put, get and remove operations (per second) performed on the selected map.

Under these charts are **Map Memory** and **Map Throughput** data tables. The Map Memory data table provides memory metrics distributed over members, as shown below.

Map Memory Data Table (In-Memory Format: BINARY)

# ▲	Members	Entries ⬍	Entry Memory ⬍	Backups ⬍	Backup Memory ⬍	Events ⬍	Hits ⬍	Locks ⬍	Dirty Entries ⬍
1	127.0.0.1:5705	19	2.89 KB	31	4.72 KB	0	0	0	0
2	127.0.0.1:5706	31	4.72 KB	19	2.89 KB	0	0	0	0
#	TOTAL	50	7.62 KB	50	7.62 KB	0	0	0	0

First Previous 1 Next Last

From left to right, this table lists the IP address and port, entry counts, memory used by entries, backup entry counts, memory used by backup entries, events, hits, locks and dirty entries (in the cases where **MapStore** is enabled, these are the entries that are put to/removed from the map but

not written to/removed from a database yet) of each entry in the map. You can navigate through the pages using the buttons at the bottom right of the table (**First, Previous, Next, Last**). You can ascend or descend the order of the listings by clicking on the column headings.

Map Throughput data table provides information about the operations (get, put, remove) performed on each member in the map, as shown below.

From left to right, this table lists:

- the IP address and port of each member,

- the put, get and remove operations on each member,

- the average put, get, remove latencies,

- and the maximum put, get, remove latencies on each member.

You can select the period in the combo box placed at the top right corner of the window, for which the table data will be shown. Available values are **Since Beginning**, **Last Minute**, **Last 10 Minutes** and **Last 1 Hour**.

You can navigate through the pages using the buttons placed at the bottom right of the table (**First, Previous, Next, Last**). To ascend or descent the order of the listings, click on the column headings.

12. Monitoring Replicated Maps

You can see a list of all the Replicated Maps in your cluster by clicking on the **Replicated Maps** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⇕	Entries ⇕	Entry Memory ⇕	Events ⇕	Hits ⇕
⬈ Replicated Map 28	16	0	0	0
⬈ Replicated Map 29	16	0	0	0
⬈ Replicated Map 3	18	0	0	0
⬈ Replicated Map 30	32	0	0	0
⬈ Replicated Map 31	48	0	0	0
⬈ Replicated Map 32	12	0	0	0
⬈ Replicated Map 33	14	0	0	0
⬈ Replicated Map 34	40	0	0	0
⬈ Replicated Map 35	4	0	0	0
⬈ Replicated Map 36	20	0	0	0
⬈ Replicated Map 37	6	0	0	0
⬈ Replicated Map 38	16	0	0	0
⬈ Replicated Map 39	16	0	0	0
⬈ Replicated Map 4	50	0	0	0
⬈ Replicated Map 40	44	0	0	0
⬈ Replicated Map 41	42	0	0	0
⬈ Replicated Map 42	4	0	0	0
⬈ Replicated Map 43	16	0	0	0
⬈ Replicated Map 44	26	0	0	0
⬈ Replicated Map 45	34	0	0	0

« ‹ 1 **2** 3 4 5 › »

You can filter the Replicated Maps shown and you can also sort the table by clicking on the column headers. Clicking on a Replicated Map name will open a new page for monitoring that Replicate Map instance on the right, as shown below.

Replicated Map Memory Data Table

# ▲	Members ⇕	Entries ⇕	Entry Memory ⇕	Events ⇕	Hits ⇕
1	127.0.0.1:5705	20	0 KB	0	0
2	127.0.0.1:5706	20	0 KB	0	0
#	**TOTAL**	**40**	**0 KB**	**0**	**0**

First Previous 1 Next Last

Replicated Map Throughput Data Table   ⚙Last Minute▾

#▲	Members ⇕	Puts/s ⇕	Gets/s ⇕	Removes/s⇕	Avg Put Lat. ⇕	Avg Get Lat. ⇕	Avg Remove Lat. ⇕	Max Put Lat. ⇕	Max Get Lat. ⇕	Max Remove Lat. ⇕
1	127.0.0.1:5705	0	0	0	0	0	0	0	0	0
2	127.0.0.1:5706	0	0	0	0	0	0	0	0	0

First Previous 1 Next Last

In this page, you can monitor metrics and also re-configure the selected Replicated Map. All of the statistics are real-time monitoring statistics.

When you click on a desired monitoring, the chart is loaded with the selected option. Also you can open the chart in new window.

- **Size**: Monitors the size of the Replicated Map. Y-axis is the entry count (should be multiplied by 1000).

- **Throughput**: Monitors get, put and remove operations performed on the Replicated Map. Y-axis is the operation count.

- **Memory**: Monitors the memory usage on the Replicated Map. Y-axis is the memory count.

- **Hits**: Monitors the hit count of the Replicated Map.

- **Puts/s, Gets/s, Removes/s**: These three charts monitor the put, get and remove operations (per second) performed on the selected Replicated Map, the average put, get, remove latencies, and the maximum put, get, remove latencies on each member.

The Replicated Map Throughput Data Table provides information about operations (get, put, remove) performed on each member in the selected Replicated Map.

#▲	Members ⬍	Puts/s ⬍	Gets/s ⬍	Removes/s⬍	Avg Put Lat.⬍	Avg Get Lat.⬍	Avg Remove Lat. ⬍	Max Put Lat.⬍	Max Get Lat.⬍	Max Remove Lat. ⬍
1	127.0.0.1:5705	0.10	0	0	0 ms	0	0	0 ms	0	0
2	127.0.0.1:5706	0.15	0	0	0 ms	0	0	0 ms	0	0

Replicated Map Throughput Data Table — ⚙Last Minute▾

First Previous 1 Next Last

From left to right, this table lists:

- the IP address and port of each member,

- the put, get, and remove operations on each member,

- the average put, get, and remove latencies,

- and the maximum put, get, and remove latencies on each member.

You can select the period from the combo box placed at the top right corner of the window, in which the table data is shown. Available values are **Since Beginning**, **Last Minute**, **Last 10 Minutes** and **Last 1 Hour**.
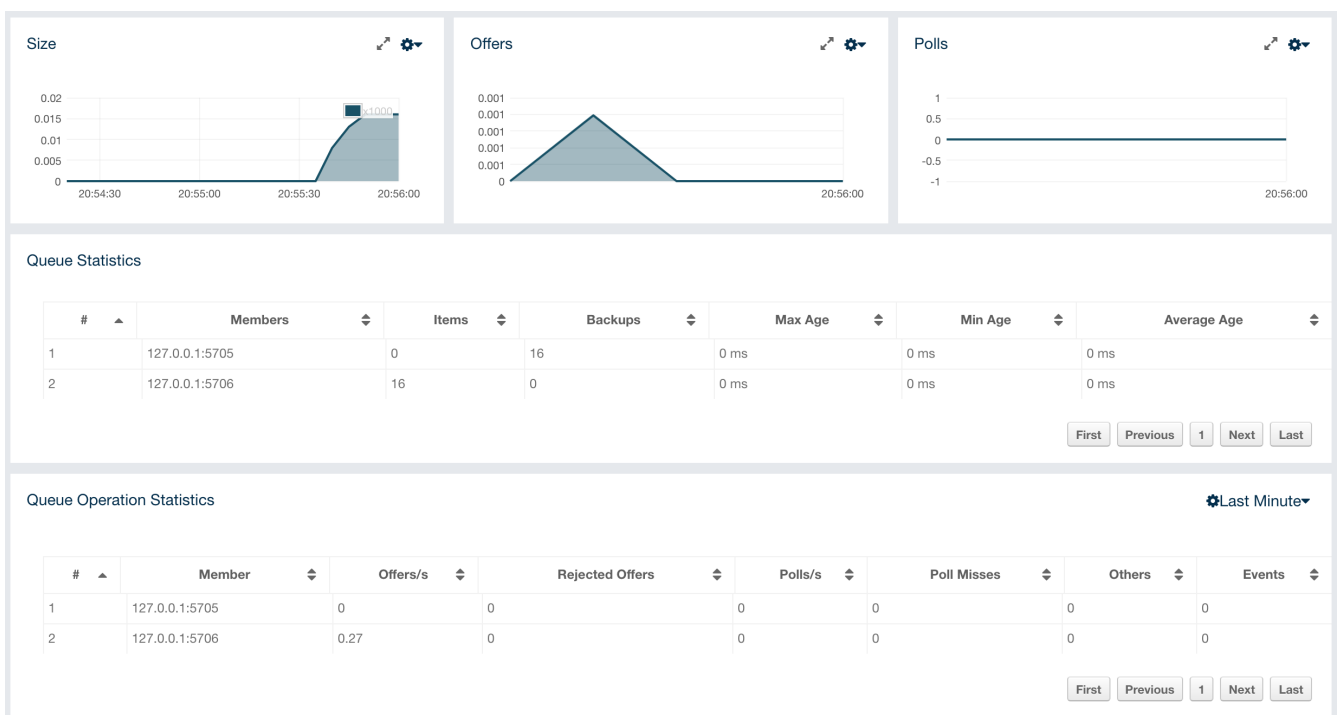
You can navigate through the pages using the buttons placed at the bottom right of the table (**First, Previous, Next, Last**). To ascend or descent the order of the listings, click on the column headings.

13. Monitoring Queues

You can see a list of all the queues in your cluster by clicking on the **Queues** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⇕	Items ⇕	Backups ⇕	Max age ⇕	Min age ⇕	Average age ⇕
⤴ Queue - 0	17	17	0s	0s	0s
⤴ Queue - 1	5	5	0s	0s	0s
⤴ Queue - 2	1	1	0s	0s	0s
⤴ Queue - 3	16	16	0s	0s	0s
⤴ Queue - 4	16	16	0s	0s	0s
⤴ Queue - 5	11	11	0s	0s	0s
⤴ Queue - 6	10	10	0s	0s	0s
⤴ Queue - 7	9	9	0s	0s	0s
⤴ Queue - 8	14	13	0s	0s	0s
⤴ Queue - 9	5	0	0s	0s	0s

You can filter the queues shown and you can also sort the table by clicking on the column headers. Clicking on a queue name will open a new page for monitoring that queue instance on the right, as shown below.

Queue Statistics

#	▲	Members ⇕	Items ⇕	Backups ⇕	Max Age ⇕	Min Age ⇕	Average Age ⇕
1		127.0.0.1:5705	0	16	0 ms	0 ms	0 ms
2		127.0.0.1:5706	16	0	0 ms	0 ms	0 ms

First Previous 1 Next Last

Queue Operation Statistics ⚙Last Minute▾

#	▲	Member ⇕	Offers/s ⇕	Rejected Offers ⇕	Polls/s ⇕	Poll Misses ⇕	Others ⇕	Events ⇕
1		127.0.0.1:5705	0	0	0	0	0	0
2		127.0.0.1:5706	0.27	0	0	0	0	0

First Previous 1 Next Last

On top of the page, small charts monitor the size, offers and polls of the selected queue in real-time. The X-axis of all the charts shows the current system time. To open a chart as a separate dialog, click on the ⤢ button placed at the top right of each chart. The monitoring charts below are available:

- **Size**: Monitors the size of the queue. Y-axis is the entry count (should be multiplied by 1000).

- **Offers**: Monitors the offers sent to the selected queue. Y-axis is the offer count.

- **Polls**: Monitors the polls sent to the selected queue. Y-axis is the poll count.

Under these charts are **Queue Statistics** and **Queue Operation Statistics** tables. The Queue Statistics table provides item and backup item counts in the queue and age statistics of items and backup items at each member, as shown below.

Queue Statistics

# ▲	Members	Items	Backups	Max Age	Min Age	Average Age
1	127.0.0.1:5705	0	16	0 ms	0 ms	0 ms
2	127.0.0.1:5706	16	0	0 ms	0 ms	0 ms

First Previous 1 Next Last

From left to right, this table lists the IP address and port, items and backup items on the queue of each member, and maximum, minimum and average age of items in the queue. You can navigate through the pages using the buttons placed at the bottom right of the table (**First, Previous, Next, Last**). The order of the listings in each column can be ascended or descended by clicking on column headings.

Queue Operations Statistics table provides information about the operations (offers, polls, events) performed on the queues, as shown below.

Queue Operation Statistics   ⚙Last Minute▾

# ▲	Member	Offers/s	Rejected Offers	Polls/s	Poll Misses	Others	Events
1	127.0.0.1:5701	0	0	0	0	0	0
2	127.0.0.1:5703	0	0	0	0	0	0
3	127.0.0.1:5702	0	0	0	0	0	0
4	127.0.0.1:5708	0	0	0	0	0	0
5	127.0.0.1:5707	0	0	0	0	0	0
6	127.0.0.1:5706	0	0	0	0	0	0
7	127.0.0.1:5709	37.67	0	37.67	0	0	0
8	127.0.0.1:5704	0	0	0	0	0	0
9	127.0.0.1:5713	0	0	0	0	0	0
10	127.0.0.1:5716	0	0	0	0	0	0

First Previous 1 2 Next Last

From left to right, this table lists the IP address and port of each member, and counts of offers, rejected offers, polls, poll misses and events.

You can select the period in the combo box placed at the top right corner of the window to show the table data. Available values are **Since Beginning**, **Last Minute**, **Last 10 Minutes** and **Last 1 Hour**.

You can navigate through the pages using the buttons placed at the bottom right of the table (**First, Previous, Next, Last**). Click on the column headings to ascend or descend the order of the listings.

14. Monitoring Topics

You can see a list of all the topics in your cluster by clicking on the **Topics** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⇕	Member ⇕	Publishes ⇕	Receives ⇕
▶ Topic - 0 (2)	127.0.0.1:5706, 127.0.0.1:5705	5	0
▶ Topic - 1 (2)	127.0.0.1:5706, 127.0.0.1:5705	8	0
▼ Topic - 2 (2)	127.0.0.1:5706, 127.0.0.1:5705	7	0
⬀ 127.0.0.1:5706		0	0
⬀ 127.0.0.1:5705		7	0
▶ Topic - 3 (2)	127.0.0.1:5706, 127.0.0.1:5705	23	0
▶ Topic - 4 (2)	127.0.0.1:5706, 127.0.0.1:5705	20	0

You can filter the topics shown and you can also sort the table by clicking on the column headers. Clicking on a topic name will open a new page for monitoring that topic instance on the right, as shown below.
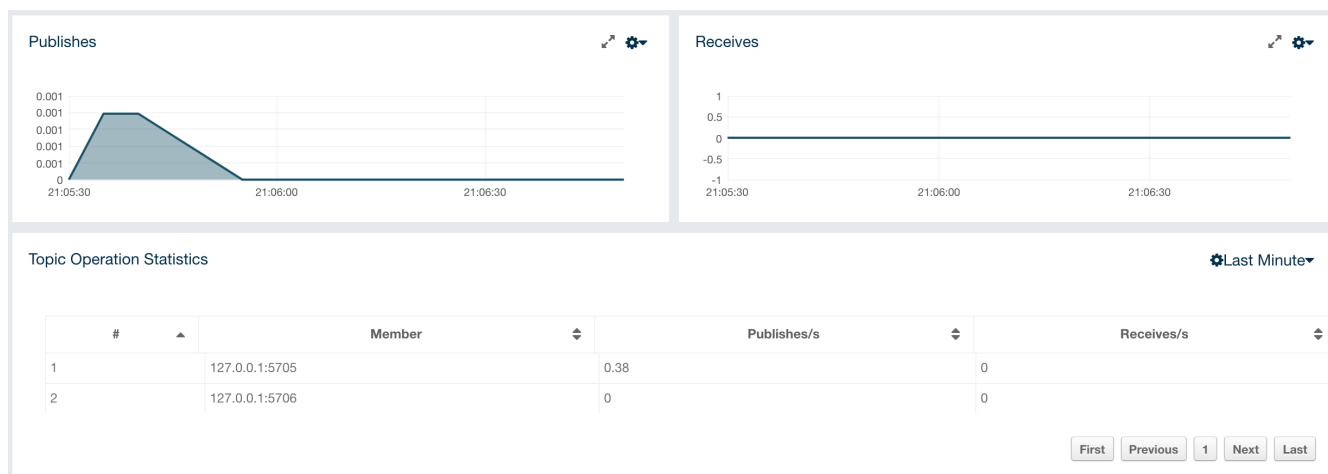
On top of the page, two charts monitor the **Publishes** and **Receives** in real-time. They show the published and received message counts of the cluster, the members of which are subscribed to the selected topic. The X-axis of both charts show the current system time. To open a chart as a separate dialog, click on the ⤢ button placed at the top right of each chart.

Under these charts is the Topic Operation Statistics table. From left to right, this table lists the IP addresses and ports of each member, and counts of the messages published and received per second in real-time. You can select the period in the combo box placed at top right corner of the table to show the table data. The available values are **Since Beginning**, **Last Minute**, **Last 10 Minutes** and **Last 1 Hour**.

You can navigate through the pages using the buttons placed at the bottom right of the table (**First, Previous, Next, Last**). Click on the column heading to ascend or descend the order of the listings.

15. Monitoring Reliable Topics

You can see a list of all the Reliable Topics in your cluster by clicking on the **Reliable Topics** menu item on the left panel. A new page is opened on the right, as shown below.

You can filter the Reliable Topics shown and you can also sort the table by clicking on the column headers. Clicking on a Reliable Topic name will open a new page for monitoring that Reliable Topic instance on the right, as shown below.
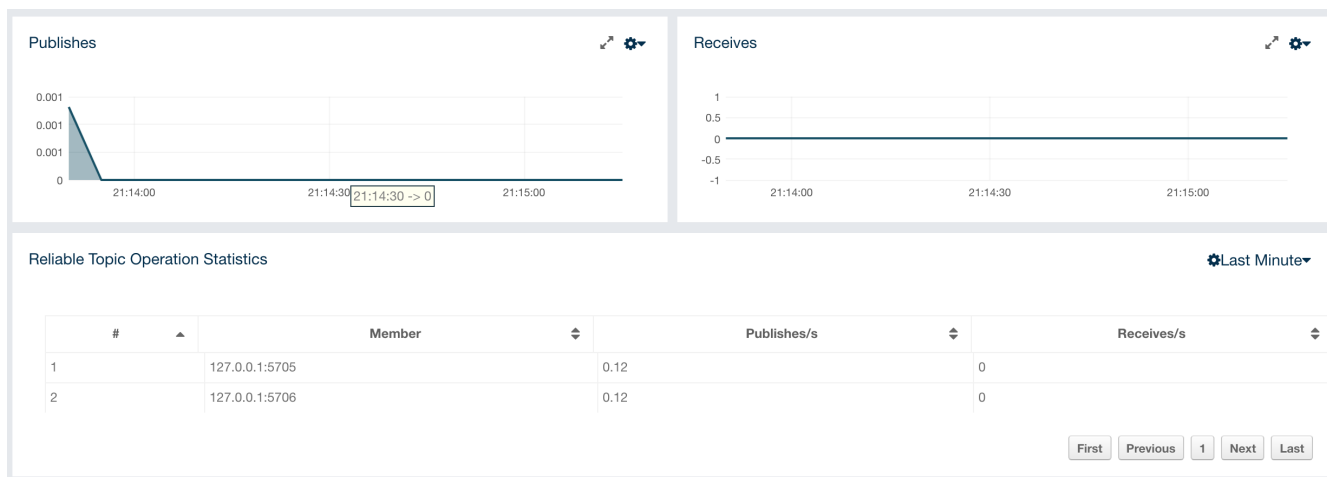
On top of the page, two charts monitor the **Publishes** and **Receives** in real-time. They show the published and received message counts of the cluster, the members of which are subscribed to the selected reliable topic. The X-axis of both charts show the current system time. To open a chart as a separate dialog, click on the ⬈ button placed at the top right of each chart.

Under these charts is the Reliable Topic Operation Statistics table. From left to right, this table lists the IP addresses and ports of each member, and counts of the messages published and received per second in real-time. You can select the period in the combo box placed at top right corner of the table to show the table data. The available values are **Since Beginning**, **Last Minute**, **Last 10 Minutes** and **Last 1 Hour**.

You can navigate through the pages using the buttons placed at the bottom right of the table (**First, Previous, Next, Last**). Click on the column heading to ascend or descend the order of the listings.

16. Monitoring Multimaps

You can see a list of all the MultiMaps in your cluster by clicking on the **MultiMaps** menu item on the left panel. A new page is opened on the right, as shown below.

You can filter the MultiMaps shown and you can also sort the table by clicking on the column headers. Clicking on a MultiMap name will open a new page for monitoring that MultiMap instance on the right.

MultiMap is a specialized map where you can associate a key with multiple values. This monitoring option is similar to the **Maps** option: the same monitoring charts and data tables monitor MultiMaps. The differences are that you cannot browse the MultiMaps and re-configure it. Please see the Managing Maps.
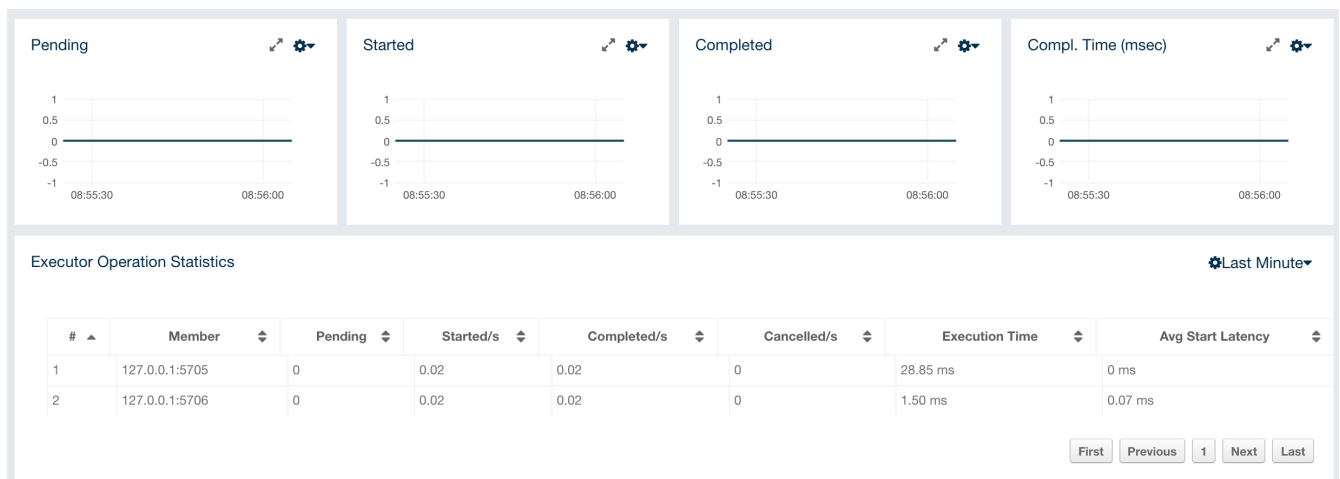
17. Monitoring Executors

You can see a list of all the Executors in your cluster by clicking on the **Executors** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⇕	Member ⇕	Pending ⇕	Started ⇕	Completed ⇕	Cancelled ⇕	Execution Time ⇕	Avg Start Latency ⇕
▶ Executor - 0 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		0, 0
▶ Executor - 1 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		1, 0
▶ Executor - 10 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		1, 2
▶ Executor - 11 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		1, 1
▶ Executor - 12 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		1, 1
▶ Executor - 13 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		0, 0
▶ Executor - 14 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		5, 5
▶ Executor - 15 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		5, 5
▼ Executor - 16 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		4, 0
⬀ 127.0.0.1:5706		0	1	1	0	1m 30s	4
⬀ 127.0.0.1:5705		0	1	1	0	28m 51s	0
▶ Executor - 17 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		4, 4
▶ Executor - 18 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		3, 4
▶ Executor - 19 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		3, 3
▶ Executor - 2 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		0, 1
▶ Executor - 20 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		3, 3
▶ Executor - 21 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		2, 2
▶ Executor - 22 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		1, 1
▶ Executor - 23 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		0, 1
▶ Executor - 24 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		1, 0
▶ Executor - 25 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		5, 6
▶ Executor - 26 (2)	127.0.0.1:5706, 127.0….	0, 0	1, 1	1, 1	0, 0		5, 5

« ‹ **1** 2 3 4 5 › »

You can filter the Executors shown and you can also sort the table by clicking on the column headers. Clicking on an Executor name will open a new page for monitoring that Executor instance on the right, as shown below.

Pending	Started	Completed	Compl. Time (msec)

Executor Operation Statistics   ⚙ Last Minute ▾

# ▲	Member ⇕	Pending ⇕	Started/s ⇕	Completed/s ⇕	Cancelled/s ⇕	Execution Time ⇕	Avg Start Latency ⇕
1	127.0.0.1:5705	0	0.02	0.02	0	28.85 ms	0 ms
2	127.0.0.1:5706	0	0.02	0.02	0	1.50 ms	0.07 ms

First Previous 1 Next Last

On top of the page, small charts monitor the pending, started, completed, etc. executors in real-time. The X-axis of all the charts shows the current system time. You can select other small monitoring charts using the ⚙▾ button placed at the top right of each chart. Click the button to list the monitoring options, as shown below.
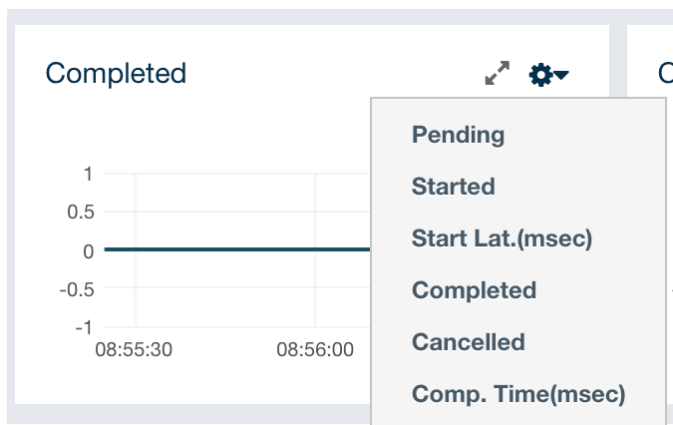
When you click on a desired monitoring, the chart loads with the selected option. To open a chart as a separate dialog, click on the button placed at top right of each chart. The below monitoring charts are available:

- **Pending**: Monitors the pending executors. Y-axis is the executor count.

- **Started**: Monitors the started executors. Y-axis is the executor count.

- **Start Lat. (msec.)**: Shows the latency when executors are started. Y-axis is the duration in milliseconds.

- **Completed**: Monitors the completed executors. Y-axis is the executor count.

- **Comp. Time (msec.)**: Shows the completion period of executors. Y-axis is the duration in milliseconds.

Under these charts is the **Executor Operation Statistics** table, as shown below.

From left to right, this table lists the IP address and port of members, the counts of pending, started and completed executors per second, and the execution time and average start latency of executors on each member. You can navigate through the pages using the buttons placed at the bottom right of the table (**First, Previous, Next, Last**). Click on the column heading to ascend or descend the order of the listings.

18. Monitoring WAN Replication

WAN Replication schemes are listed under the **WAN Replication** menu item on the left. When you click on a scheme, a new page for monitoring the targets which that scheme has appears on the right, as shown below.

In this page, you see **WAN Replication Operations Table** for each target which belongs to this scheme. One of the example tables is shown below.

- **Connected**: Status of the member connection to the target.

- **Outbound Recs (sec)**: Average of event count per second. Please see the paragraph below.

- **Outbound Lat (ms)**: Average latency of sending a record to the target from this member. Please see the paragraph below.

- **Outbound Queue**: Number of records waiting in the queue to be sent to the target.

- **Action**: Pause/resume replication of this member's records.

- **State**: Shows current state of the WAN publisher on a member. See Changing WAN Publisher State for the list of possible WAN publisher states.

Outbound Recs and **Outbound Lat** are based on the following internal statistics:

- Total published event count (TBEC): Total number of events that are successfully sent to the target cluster since the start-up of the member.

- Total latency (TL): Grand total of each event's waiting time in the queue, including network transmit and receiving ACK from the target.

Each member sends these two statistics to the Management Center at intervals of 3 seconds (update

interval). Management Center derives **Outbound Recs/s** and **Outbound Lat** from these statistics as formulated below:

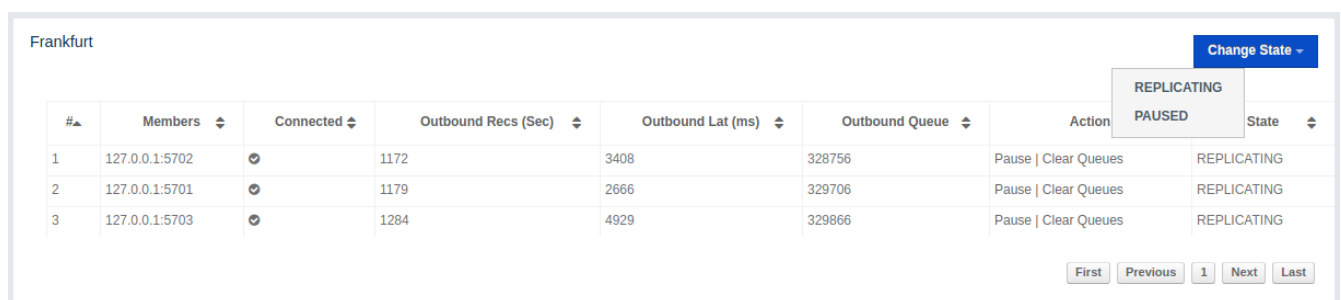Outbound Recs/s = (Current TBEC - Previous TBEC) / Update Interval

Outbound Latency = (Current TL - Previous TL) / (Current TBEC - Previous TBEC)

18.1. Changing WAN Publisher State

A WAN publisher can be in one of the following states:

- `REPLICATING` (default): State where both enqueuing new events is allowed, enqueued events are replicated to the target cluster and WAN sync is enabled.
- `PAUSED`: State where new events are enqueued but they are dequeued. Some events which have been dequeued before the state was switched may still be replicated to the target cluster but further events will not be replicated. WAN sync is enabled.

You can change a WAN publisher's state by clicking the `Change State` dropdown button on top right hand corner of the WAN Replication Operations Table.

Frankfurt								Change State ▾
							REPLICATING	
#▲	Members ⇅	Connected ⇅	Outbound Recs (Sec) ⇅	Outbound Lat (ms) ⇅	Outbound Queue ⇅	Action	PAUSED State ⇅	
1	127.0.0.1:5702	✔	1172	3408	328756	Pause | Clear Queues	REPLICATING	
2	127.0.0.1:5701	✔	1179	2666	329706	Pause | Clear Queues	REPLICATING	
3	127.0.0.1:5703	✔	1284	4929	329866	Pause | Clear Queues	REPLICATING	
						First Previous 1 Next Last		

18.2. WAN Sync

You can initiate a synchronization operation on an IMap for a specific target cluster. Synchronization operation sends all the data of an IMap to a target cluster to align the state of target IMap with source IMap. Synchronization is useful if two remote clusters lost their synchronization due to WAN queue overflow or in restart scenarios.

To initiate WAN Sync, open the **WAN Replication** menu item on the left and navigate to the **Sync** tab.

Click **Start WAN Sync** button to open the dialog, enter the target details for the sync operation and click **Sync** to start the operation.

You can also use the "All Maps" option in the above dialog if you want to synchronize all the maps in source and target cluster.

You can see the progress of the operation once you initiate it.

18.3. Add WAN Replication Configuration

You can add a WAN replication configuration dynamically to a cluster. It is useful for having one-off

WAN sync operations. The added configuration has two caveats:

- It is not persistent, so it will not survive a member restart.

- It cannot be used as a target for regular WAN replication. It can only be used for WAN sync.

See the Hazelcast IMDG Reference Manual section on WAN Replication for details about the fields and their possible values.

After clicking **Add Configuration** button, the new WAN replication configuration is added to the cluster. You can see the new configuration when you try to initiate a WAN sync operation as described in the previous section.

19. Monitoring Members

Use this menu item to monitor each cluster member and perform operations like running garbage collection (GC) and taking a thread dump.

You can see a list of all the members in your cluster by clicking on the **Members** menu item on the left panel. A new page is opened on the right, as shown below.

Member ⇕	Slow Operations ⇕	Owned Partition...	Version ⇕	OS Total Physic...	OS Comitted Vir...	OS Free Physica...	OS System CPU ...	OS Max File Des...	OS Open File De...
127.0.0.1:5706	No	Yes	3.10.0	16.00 GB	8.14 GB	3.31 GB	4%	10240	100
127.0.0.1:5705	No	Yes	3.10.0	16.00 GB	8.14 GB	3.31 GB	5%	10240	100

You can filter the members shown and you can also sort the table by clicking on the column headers. Clicking on a member name will open a new page for monitoring that member on the right, as shown below.
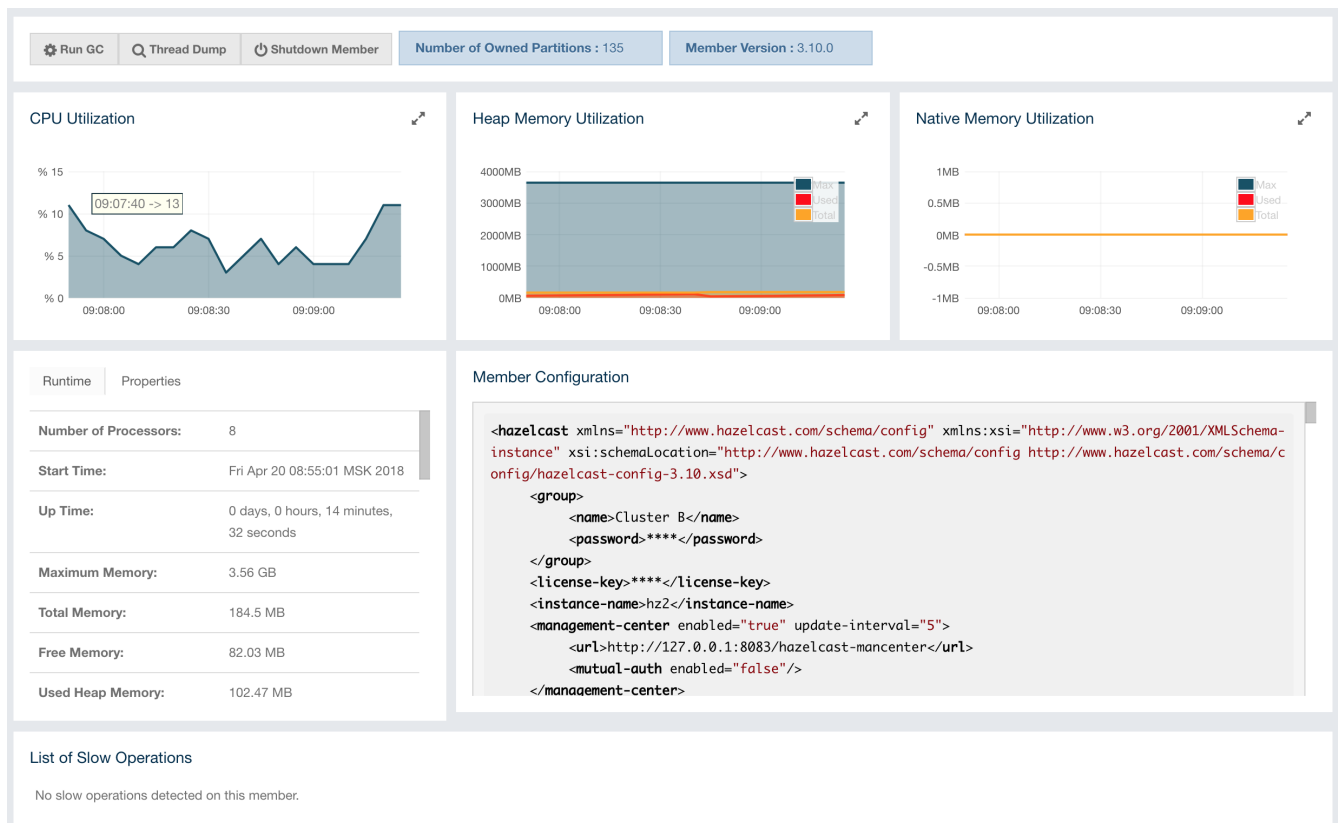
The **CPU Utilization** chart shows the percentage of CPU usage on the selected member. The **Memory Utilization** chart shows the memory usage on the selected member with three different metrics (maximum, used and total memory). You can open both of these charts as separate windows using the ⚙ button placed at top right of each chart; this gives you a clearer view of the chart.

The window titled **Partitions** shows which partitions are assigned to the selected member. **Runtime** is a dynamically updated window tab showing the processor number, the start and up times, and the maximum, total and free memory sizes of the selected member. These values are collected from the default MXBeans provided by the Java Virtual Machine (JVM). Descriptions from the Javadocs and some explanations are below:

- **Number of Processors**: Number of processors available to the member (JVM).
- **Start Time**: Start time of the member (JVM) in milliseconds.
- **Up Time**: Uptime of the member (JVM) in milliseconds
- **Maximum Memory**: Maximum amount of memory that the member (JVM) will attempt to use.
- **Free Memory**: Amount of free memory in the member (JVM).
- **Used Heap Memory**: Amount of used memory in bytes.
- **Max Heap Memory**: Maximum amount of memory in bytes that can be used for memory management.
- **Used Non-Heap Memory**: Amount of used memory in bytes.

- **Max Non-Heap Memory**: Maximum amount of memory in bytes that can be used for memory management.

- **Total Loaded Classes**: Total number of classes that have been loaded since the member (JVM) has started execution.

- **Current Loaded Classes**: Number of classes that are currently loaded in the member (JVM).

- **Total Unloaded Classes**: Total number of classes unloaded since the member (JVM) has started execution.

- **Total Thread Count**: Total number of threads created and also started since the member (JVM) started.

- **Active Thread Count**: Current number of live threads including both daemon and non-daemon threads.

- **Peak Thread Count**: Peak live thread count since the member (JVM) started or peak was reset.

- **Daemon Thread Count**: Current number of live daemon threads.

- **OS: Free Physical Memory**: Amount of free physical memory in bytes.

- **OS: Committed Virtual Memory**: Amount of virtual memory that is guaranteed to be available to the running process in bytes.

- **OS: Total Physical Memory**: Total amount of physical memory in bytes.

- **OS: Free Swap Space**: Amount of free swap space in bytes. Swap space is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space.

- **OS: Total Swap Space**: Total amount of swap space in bytes.

- **OS: Maximum File Descriptor Count**: Maximum number of file descriptors. File descriptor is an integer number that uniquely represents an opened file in the operating system.

- **OS: Open File Descriptor Count**: Number of open file descriptors.

- **OS: Process CPU Time**: CPU time used by the process on which the member (JVM) is running in nanoseconds.

- **OS: Process CPU Load**: Recent CPU usage for the member (JVM) process. This is a double with a value from 0.0 to 1.0. A value of 0.0 means that none of the CPUs were running threads from the member (JVM) process during the recent period of time observed, while a value of 1.0 means that all CPUs were actively running threads from the member (JVM) 100% of the time during the recent period being observed. Threads from the member (JVM) include the application threads as well as the member (JVM) internal threads.

- **OS: System Load Average**: System load average for the last minute. The system load average is the average over a period of time of this sum: (the number of runnable entities queued to the available processors) + (the number of runnable entities running on the available processors). The way in which the load average is calculated is operating system specific but it is typically a damped time-dependent average.

- **OS: System CPU Load**: Recent CPU usage for the whole system. This is a double with a value from 0.0 to 1.0. A value of 0.0 means that all CPUs were idle during the recent period of time observed, while a value of 1.0 means that all CPUs were actively running 100% of the time during the recent period being observed.

Next to the **Runtime** tab, the **Properties** tab shows the system properties. The **Member Configuration** window shows the XML configuration of the connected Hazelcast cluster.

The **List of Slow Operations** gives an overview of detected slow operations which occurred on that member. The data is collected by the SlowOperationDetector.

Click on an entry to open a dialog which shows the stacktrace and detailed information about each slow invocation of this operation.

Besides the aforementioned monitoring charts and windows, you can also perform operations on the selected member through this page. The operation buttons are located at the top right of the page, as explained below:

- **Run GC**: Press this button to execute garbage collection on the selected member. A notification stating that the GC execution was successful will be shown.

- **Thread Dump**: Press this button to take a thread dump of the selected member and show it as a separate dialog to the user.

- **Shutdown Node**: Press this button to shutdown the selected member.

- **Promote Member**: Only shown for lite members. Press this button to promote a lite member to a data member.
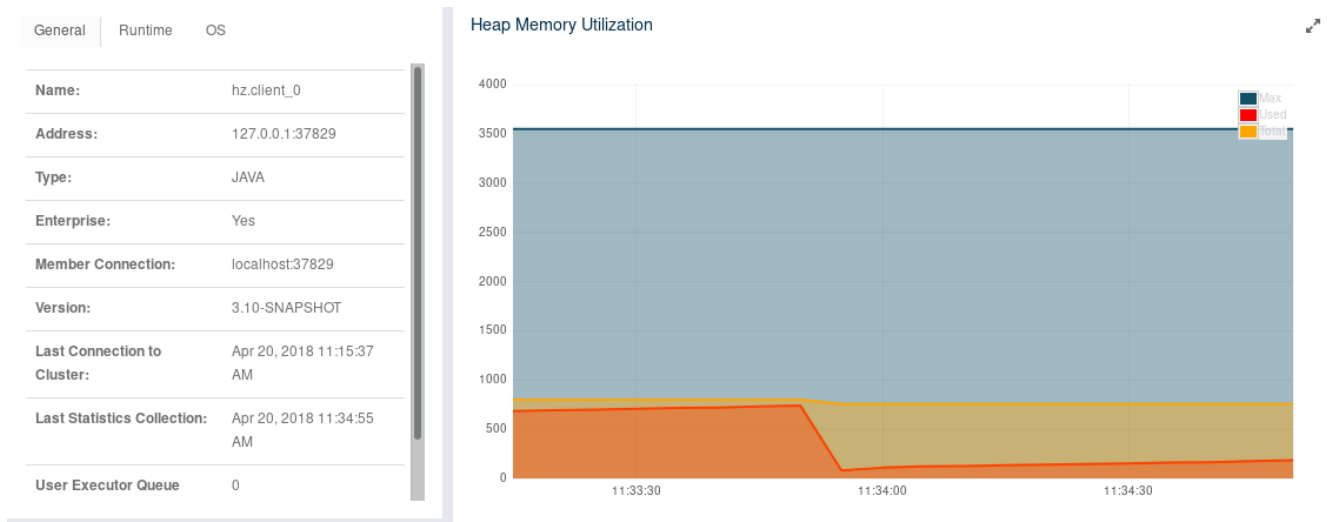
20. Monitoring Clients

You can use the **Clients** menu item to monitor all the clients that are connected to your Hazelcast cluster. Currently only the Java clients can be monitored.

As a prerequisite, you need to enable the client statistics before starting your clients. This can be done by setting the `hazelcast.client.statistics.enabled` system property to `true`. Please see the System Properties section in the Hazelcast IMDG Reference Manual for more information. After you enable the client statistics, you can monitor your clients using Hazelcast Management Center.

You can see a list of all the clients in your cluster by clicking on the **Clients** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⬍	Address ⬍	Enterprise ⬍	Type ⬍	Member Connection ⬍	Version ⬍	UUID ⬍
hz.client_4	127.0.0.1:33903	Yes	JAVA	localhost:33903	3.10	f38e94e3-460a-4133-abd...
hz.client_6	127.0.0.1:40063	Yes	JAVA	localhost:40063	3.10	28994ba5-fce1-42d5-914...
hz.client_7	127.0.0.1:43941	Yes	JAVA	localhost:43941	3.10	1cb74477-420d-4ea8-82f...
hz.client_2	127.0.0.1:43065	Yes	JAVA	localhost:43065	3.10	5c35b691-8ff3-4822-ad7f...
hz.client_8	127.0.0.1:36709	Yes	JAVA	localhost:36709	3.10	b4e674d0-68f9-43f6-b9fb...
hz.client_0	127.0.0.1:37829	Yes	JAVA	localhost:37829	3.10	19480387-026d-41e9-b7...
hz.client_9	127.0.0.1:36677	Yes	JAVA	localhost:36677	3.10	27b9f8c6-3b25-4f8c-b96e...
hz.client_5	127.0.0.1:36489	Yes	JAVA	localhost:36489	3.10	43b87d98-8142-4b9f-9a3...
hz.client_1	127.0.0.1:39431	Yes	JAVA	localhost:39431	3.10	8bb7e9fe-4b01-44a3-a14...
hz.client_3	127.0.0.1:40111	Yes	JAVA	localhost:40111	3.10	9589a5e2-5357-47d1-82...

You can filter the clients shown and you can also sort the table by clicking on the column headers. Clicking on a client name will open a new page for monitoring that client on the right, as shown below.

General | Runtime | OS

Name:	hz.client_0
Address:	127.0.0.1:37829
Type:	JAVA
Enterprise:	Yes
Member Connection:	localhost:37829
Version:	3.10-SNAPSHOT
Last Connection to Cluster:	Apr 20, 2018 11:15:37 AM
Last Statistics Collection:	Apr 20, 2018 11:34:55 AM
User Executor Queue	0

Heap Memory Utilization

Client Near Cache Statistics (for Maps)

#	Map Name	Creation Time	Evictions	Expirations	Hits	Misses	Owned Entry Count	Owned Entry Memory Cost	LP Duration	LP Key Count	LP Time	LP Written Bytes	LP Failure
1	other,map	Apr 20, 2018 11:15:39 AM	0	0	100	50	50	5.65 KB	0	0	0	0 KB	
2	employee-map	Apr 20, 2018 11:15:37 AM	0	0	100	50	50	5.65 KB	0	0	0	0 KB	
3	some=map	Apr 20, 2018 11:15:38 AM	0	0	100	50	50	5.65 KB	0	0	0	0 KB	
4	samename	Apr 20, 2018 11:15:39 AM	0	0	100	50	50	5.65 KB	0	0	0	0 KB	
5	book.map	Apr 20, 2018 11:15:38 AM	0	0	100	50	50	5.65 KB	0	0	0	0 KB	
6	another\map	Apr 20, 2018 11:15:39 AM	0	0	100	50	50	5.65 KB	0	0	0	0 KB	

First | Previous | 1 | Next | Last

Client Near Cache Statistics (for Caches)

#	Cache Name	Creation Time	Evictions	Expirations	Hits	Misses	Owned Entry Count	Owned Entry Memory Cost	LP Duration	LP Key Count	LP Time	LP Written Bytes	LP Failure
1	a-cache	Apr 20, 2018 11:15:40 AM	0	0	0	50	50	5.65 KB	0	0	0	0 KB	
2	b.cache	Apr 20, 2018 11:15:40 AM	0	0	0	0	0	0 KB	0	0	0	0 KB	
3	samename	Apr 20, 2018 11:15:40 AM	0	0	0	0	0	0 KB	0	0	0	0 KB	
4	e\cache	Apr 20, 2018 11:15:40 AM	0	0	0	0	0	0 KB	0	0	0	0 KB	
5	c=cache	Apr 20, 2018 11:15:40 AM	0	0	0	0	0	0 KB	0	0	0	0 KB	
6	d,cache	Apr 20, 2018 11:15:40 AM	0	0	0	0	0	0 KB	0	0	0	0 KB	

First | Previous | 1 | Next | Last

The **Heap Memory Utilization** chart shows the memory usage on the selected client with three different metrics (maximum, used and total memory). You can open both of these charts as separate windows using the ⚙ button placed at top right of each chart; this gives you a clearer view of the chart.

General is a dynamically updated window tab showing general information about the client. Below are brief explanations for each piece of information:

- **Name**: Name of the client instance.
- **Address**: Address of the client, shown as `<IP>:<port>`.
- **Type**: Type of the client. Java client is the only supported client type at the moment.

- **Enterprise**: Yes, if the client is an Hazelcast IMDG Enterprise client.

- **Member Connection**: Shows to which member a client is currently connected to. Please note that **ALL** means a client is configured so that it **might** connect to all members of a cluster, i.e., it might not have a connection to all members all the time.

- **Version**: Version of the client.

- **Last Connection to Cluster**: Time that the client connected to the cluster. It is reset on each reconnection.

- **Last Statistics Collection**: Time when the latest update for the statistics is collected from the client.

- **User Executor Queue Size**: Number of waiting tasks in the client user executor.

Next to the **General** tab, the **Runtime** tab shows the processor number, uptime, and maximum, total and free memory sizes of the selected client. These values are collected from the default MXBeans provided by the Java Virtual Machine (JVM). Descriptions from the Javadocs and some explanations are below:

- **Number of Processors**: Number of processors available to the client (JVM).

- **Up Time**: Uptime of the client (JVM).

- **Maximum Memory**: Maximum amount of memory that the client (JVM) will attempt to use.

- **Total Memory**: Amount of total heap memory currently available for current and future objects in the client (JVM).

- **Free Memory**: Amount of free heap memory in the client (JVM).

- **Used Memory**: Amount of used heap memory in the client (JVM).

Next to the **Runtime** tab, the **OS** tab shows statistics about the operating system of the client. These values are collected from the default MXBeans provided by the Java Virtual Machine (JVM). Descriptions from the Javadocs and some explanations are below:

- **Free Physical Memory**: Amount of free physical memory.

- **Committed Virtual Memory**: Amount of virtual memory that is guaranteed to be available to the running process.

- **Total Physical Memory**: Total amount of physical memory.

- **Free Swap Space**: Amount of free swap space. Swap space is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space.

- **Total Swap Space**: Total amount of swap space.

- **Maximum File Descriptor Count**: Maximum number of file descriptors. File descriptor is an integer number that uniquely represents an opened file in the operating system.

- **Open File Descriptor Count**: Number of open file descriptors.

- **Process CPU Time**: CPU time used by the process on which the member (JVM) is running.

- **System Load Average**: System load average for the last minute. The system load average is the average over a period of time of this sum: (the number of runnable entities queued to the

available processors) + (the number of runnable entities running on the available processors). The way in which the load average is calculated is operating system specific but it is typically a damped time-dependent average.

> ℹ️ Some of the Runtime/OS statistics may not be available for your client's JVM implementation/operating system. **UNKNOWN** is shown for these types of statistics. Please refer to your JVM/operating system documentation for further details.

The **Client Near Cache Statistics** table shows statistics related to Near Cache of a client. There are two separate tables; one for maps and one for caches.

- **Map/Cache Name**: Name of the map or cache.
- **Creation Time**: Creation time of this Near Cache on the client.
- **Evictions**: Number of evictions of Near Cache entries owned by the client.
- **Expirations**: Number of TTL and max-idle expirations of Near Cache entries owned by the client.
- **Hits**: Number of hits (reads) of Near Cache entries owned by the client.
- **Misses**: Number of misses of Near Cache entries owned by the client.
- **Owned Entry Count**: Number of Near Cache entries owned by the client.
- **Owned Entry Memory Cost**: Memory cost of Near Cache entries owned by the client.
- **LP Duration**: Duration of the last Near Cache key persistence (when the pre-load feature is enabled).
- **LP Key Count**: Number of Near Cache key persistences (when the pre-load feature is enabled).
- **LP Time**: Time of the last Near Cache key persistence (when the pre-load feature is enabled).
- **LP Written Bytes**: Written number of bytes of the last Near Cache key persistence (when the pre-load feature is enabled).
- **LP Failure**: Failure reason of the last Near Cache persistence (when the pre-load feature is enabled).

> ℹ️ Please note that you can configure the time interval for which the client statistics are collected and sent to the cluster, using the system property `hazelcast.client.statistics.period.seconds`. Please see the [System Properties section](http://docs.hazelcast.org/docs/latest/manual/html-single/index.html#client-system-properties) in the Hazelcast IMDG Reference Manual for more information.

21. Monitoring PN Counters

You can see a list of all the PN counters in your cluster by clicking on the **Counters** menu item on the left panel. A new page is opened on the right, as shown below.

Name ⇅	Operations per second INC ⇅	Operations per second DEC ⇅	Number of Replicas ⇅
⬀ PN Counter - 0	6.549	3.699	2
⬀ PN Counter - 1	7.399	0	1
⬀ PN Counter - 2	6.549	0	2
⬀ PN Counter - 3	7.399	7.399	1
⬀ PN Counter - 4	6.549	0	2
⬀ PN Counter - 5	7.399	0	1
⬀ PN Counter - 6	6.549	3.699	2
⬀ PN Counter - 7	7.399	0	1
⬀ PN Counter - 8	6.549	0	2
⬀ PN Counter - 9	7.399	7.399	1

You can filter the counters shown and you can also sort the table by clicking on the column headers. The monitoring data available are:

- **Operations per second INC**: Average number of times the counter was incremented per second during the last timeslice

- **Operations per second DEC**: Average number of times the counter was decremented per second during the last timeslice

- **Number of Replicas**: The number of member instances that have a state for the counter

Clicking on a counter name will open a new page for monitoring that specific counter instance, as shown below.

Member ⇅	Operations per second INC ⇅	Operations per second DEC ⇅	Value ⇅
127.0.0.1:5706	0	0	200
127.0.0.1:5705	0	0	200

The table can likewise be sorted by clicking the column headers. It shows IP and port of the members that have a state for the specific counter named in the page's title. The monitoring data available are:

- **Operations per second INC**: Average number of times the counter was incremented on that member per second during the last timeslice

- **Operations per second DEC**: Average number of times the counter was decremented on that member per second during the last timeslice

- **Value**: The current value of the counter on that member

22. Monitoring Flake ID Generators

You can see a list of all Flake ID Generators in your cluster by clicking on the **ID Generators** menu

item on the left panel. A new page is opened on the right, as shown below.

Name ⇕	Avg. Batch Requests ⇕	Avg. Batch Size ⇕
⬈ Flake ID Generator - 0	1	100
⬈ Flake ID Generator - 1	1	100
⬈ Flake ID Generator - 2	1	100
⬈ Flake ID Generator - 3	1	100
⬈ Flake ID Generator - 4	1	100
⬈ Flake ID Generator - 5	1	100
⬈ Flake ID Generator - 6	1	100
⬈ Flake ID Generator - 7	1	100
⬈ Flake ID Generator - 8	1	100
⬈ Flake ID Generator - 9	1	100

You can filter the generators shown and you can also sort the table by clicking on the column headers. The monitoring data available are:

- **Operations per second**: Average number of times per second the generator created a batch of new IDs during the last timeslice

Clicking on a generator name will open a new page for monitoring that specific generator instance, as shown below.

Member ⇕	Batch Requests ⇕	Avg. Batch Size ⇕
127.0.0.1:5710	2	100
127.0.0.1:5709	2	100
127.0.0.1:5706	2	100
127.0.0.1:5708	2	100
127.0.0.1:5707	2	100

The table can likewise be sorted by clicking the column headers. It shows IP and port of the members that have a state for the specific generator named in the page's title. The monitoring data available are:

- **Operations per second**: Average number of times per second the generator created a batch of new IDs on that member during the last timeslice

> ℹ The operations per second is not the number of new IDs generated or used but the number of ID batches. The batch size is configurable, usually it contains hundreds or thousands of IDs. A client uses all IDs from a batch before a new batch is requested.

23. Scripting

You can use the scripting feature of this tool to execute codes on the cluster. To use this feature, click the **Scripting** menu item on the left panel. Once selected, the scripting feature opens as shown below.

In this window, the **Scripting** part is the actual coding editor. You can select the members on which the code will execute from the **Members** list shown at the right side of the window. Below the members list, a combo box enables you to select a scripting language: currently, JavaScript, Ruby, Groovy and Python languages are supported. After you write your script and press the **Execute** button, you can see the execution result in the **Result** part of the window.

> To use the scripting languages other than JavaScript on a member, the libraries for those languages should be placed in the classpath of that member.

There are **Save** and **Delete** buttons on the top right of the scripting editor. To save your scripts, press the **Save** button after you type a name for your script into the field next to this button. The scripts you saved are listed in the **Saved Scripts** part of the window, located at the bottom right of the page. Click on a saved script from this list to execute or edit it. If you want to remove a script that you wrote and saved before, select it from this list and press the **Delete** button.

In the scripting engine you have a `HazelcastInstance` bonded to a variable named `hazelcast`. You can invoke any method that `HazelcastInstance` has via the `hazelcast` variable. You can see example usage for JavaScript below.
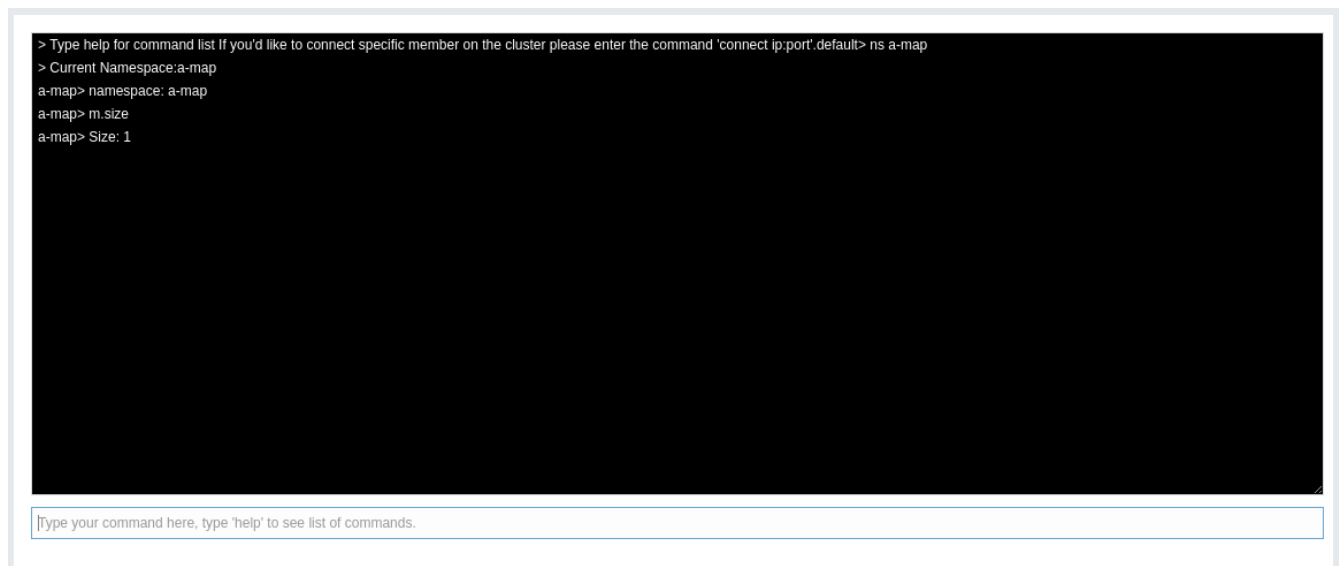
```
var name = hazelcast.getName();
var node = hazelcast.getCluster().getLocalMember();
var employees = hazelcast.getMap("employees");
employees.put("1","John Doe");
employees.get("1"); // will return "John Doe"
```

24. Executing Console Commands

The Management Center has a console feature that enables you to execute commands on the cluster. For example, you can perform put`s and `get`s on a map, after you set the namespace with the command `ns <name of your map>`. The same is valid for queues, topics, etc. To execute your command, type it into the field below the console and press **Enter**. Type help to see all the commands that you can use.

Open a console window by clicking on the **Console** button located on the left panel. Below is a sample view with some executed commands.

```
> Type help for command list If you'd like to connect specific member on the cluster please enter the command 'connect ip:port'.default> ns a-map
> Current Namespace:a-map
a-map> namespace: a-map
a-map> m.size
a-map> Size: 1
```

```
Type your command here, type 'help' to see list of commands.
```

25. Creating Alerts

You can use the alerts feature of this tool to receive alerts and/or e-mail notifications by creating filters. In these filters, you can specify criteria for cluster members or data structures. When the specified criteria are met for a filter, the related alert is shown as a pop-up message on the top right of the page or sent as an e-mail.

Once you click the **Alerts** button located on the left panel, the page shown below appears.

If you want to enable the Management Center to send e-mail notifications to the Management Center Admin users, you need to configure the SMTP server. To do this, click on the **Create STMP Config** shown above. The form shown below appears.
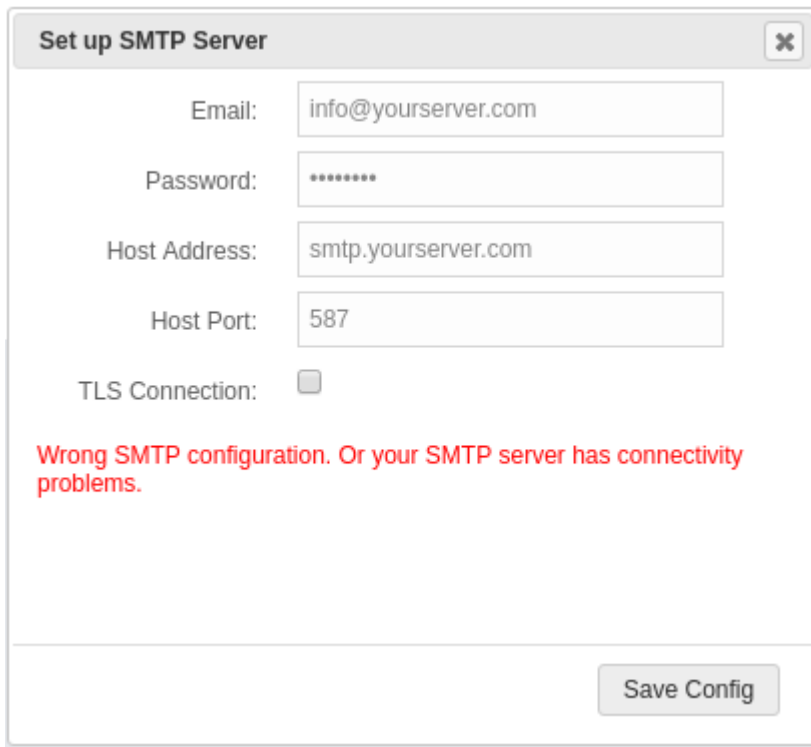
In this form, specify the e-mail address from which the notifications will be sent and also its password. Then, provide the SMTP server host address and port. Finally, check the **TLS Connection** checkbox if the connection is secured by TLS (Transport Layer Security).

After you provide the required information, click on the **Save Config** button. After a processing period (for a couple of seconds), the form will be closed if the configuration is created successfully. In this case, an e-mail will be sent to the e-mail address you provided in the form stating that the SMTP configuration is successful and e-mail alert system is created.

If not, you will see an error message at the bottom of this form as shown below.
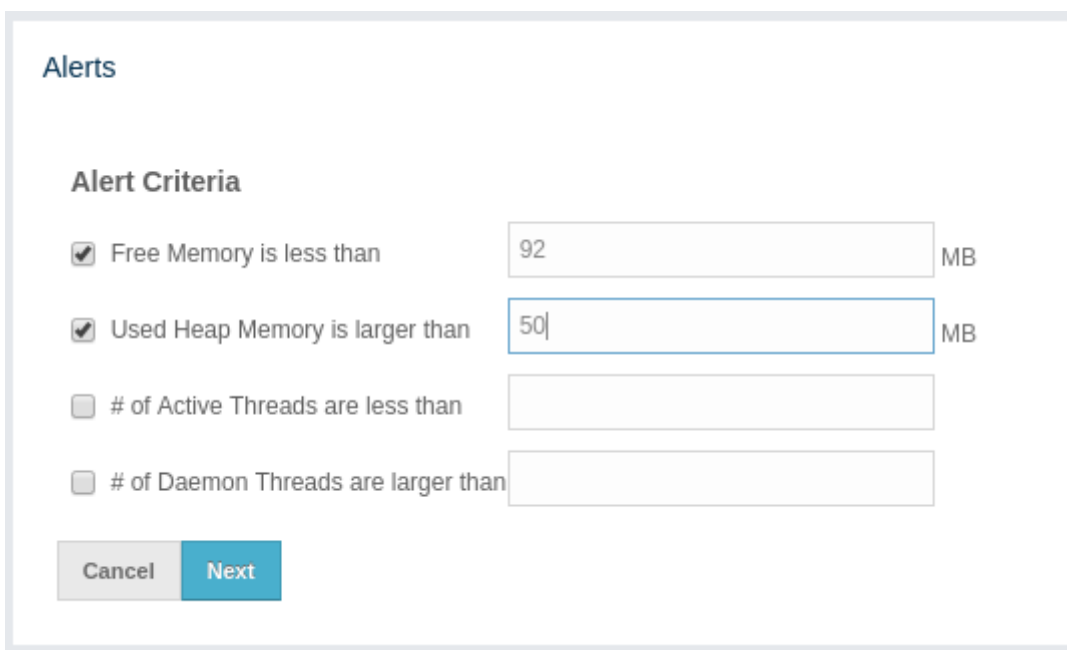
As you can see, the reasons can be wrong SMTP configuration or connectivity problems. In this case, please check the form fields and check for any causes for the connections issues with your server.

25.1. Creating Filters for Cluster Members

Select **Member Alerts** check box to create filters for some or all members in the cluster. Once selected, the next screen asks for which members the alert will be created. Select the desired members and click on the **Next** button. On the next page (shown below), specify the criteria.

You can create alerts when:

- free memory on the selected members is less than the specified number.

- used heap memory is larger than the specified number.

- the number of active threads are less than the specified count.

- the number of daemon threads are larger than the specified count.

When two or more criteria is specified they will be bound with the logical operator **AND**.

On the next page, give a name for the filter. Then, select whether notification e-mails will be sent to the Management Center Admins using the **Send Email Alert** checkbox. Then, provide a time interval (in seconds) for which the e-mails with the **same notification content** will be sent using the **Email Interval (secs)** field. Finally, select whether the alert data will be written to the disk (if checked, you can see the alert log at the folder **/users/<your user>/hazelcast-mancenter3.10.3**).

Click on the **Save** button; your filter will be saved and put into the **Filters** part of the page. To edit the filter, click on the 🔧 icon. To delete it, click on the ✖ icon.

25.2. Creating Filters for Data Types

Select the **Data Type Alerts** check box to create filters for data structures. The next screen asks for which data structure (maps, queues, multimaps, executors) the alert will be created. Once a structure is selected, the next screen immediately loads and you then select the data structure instances (i.e. if you selected **Maps**, it will list all the maps defined in the cluster, you can select one map or more). Select as desired, click on the **Next** button, and select the members on which the selected data structure instances will run.

The next screen, as shown below, is the one where you specify the criteria for the selected data structure.

As the screen shown above shows, you will select an item from the left combo box, select the operator in the middle one, specify a value in the input field, and click on the **Add** button. You can create more than one criteria in this page; those will be bound by the logical operator **AND**.

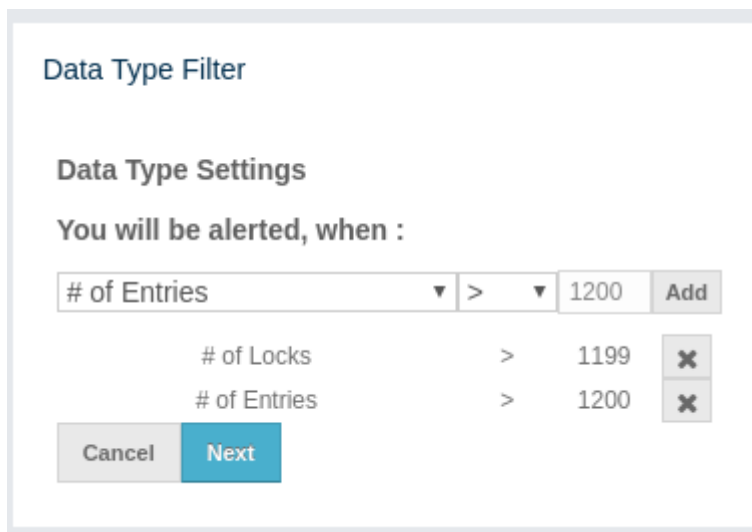After you specify the criteria, click the **Next** button. On the next page, give a name for the filter. Then, select whether notification e-mails will be sent to the Management Center Admins using the **Send Email Alert** checkbox. Then, provide a time interval (in seconds) for which the e-mails with the **same notification content** will be sent using the **Email Interval (secs)** field. Finally, select

whether the alert data will be written to the disk (if checked, you can see the alert log at the folder **/users/<your user>/hazelcast-mancenter3.10.3**).

Click on the **Save** button; your filter will be saved and put into the **Filters** part of the page. To edit the filter, click on the ⚙ icon. To delete it, click on the ✖ icon.

26. Administering Management Center

ℹ️   This menu item is available only to admin users.

26.1. Cluster State

The admin user can see and change the cluster state and shut down the cluster using the operations listed in this screen as shown below.

Cluster States:

- **Active**: Cluster will continue to operate without any restriction. All operations are allowed. This is the default state of a cluster.

- **No Migration**: Migrations (partition rebalancing) and backup replications are not allowed. Cluster will continue to operate without any restriction. All other operations are allowed.

- **Frozen**: New members are not allowed to join, except the members left in **this** state or **Passive** state. All other operations except migrations are allowed and will operate without any restriction.

- **Passive**: New members are not allowed to join, except the members left in **this** state or **Frozen** state. All operations, except the ones marked with `AllowedDuringPassiveState`, will be rejected immediately.

- **In Transition**: Shows that the cluster state is in transition. This is a temporary and intermediate state. It is not allowed to set it explicitly.

Changing Cluster State
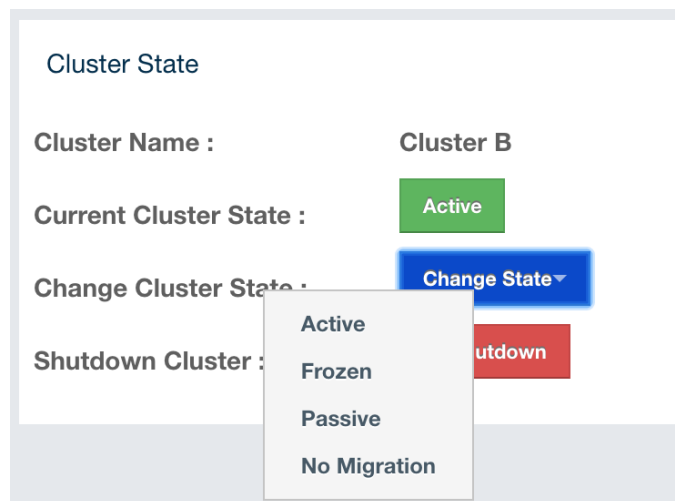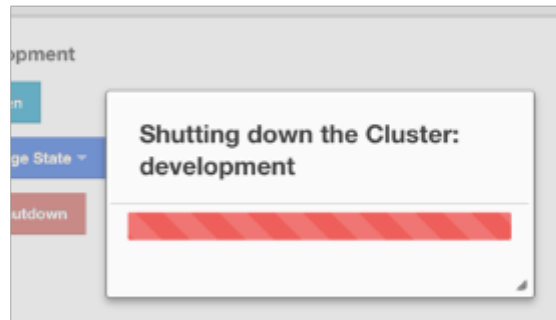
- Click the dropdown menu and choose the state to which you want your cluster to change. A pop-up will appear and stay on the screen until the state is successfully changed.
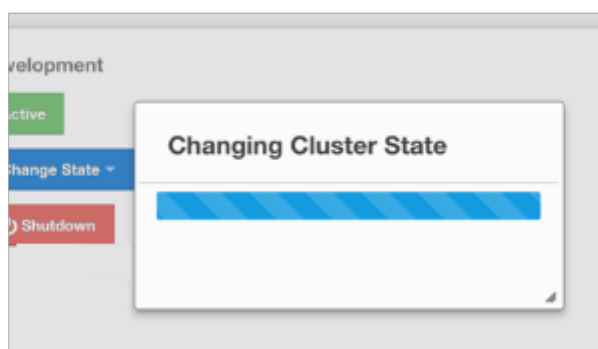
Shutting Down the Cluster

- Click the **Shutdown** button. A pop-up will appear and stay on screen until the cluster is successfully shutdown.

If an exception occurs during the state change or shutdown operation on the cluster, this exception message will be shown on the screen as a notification.

26.2. Manage License

To update the Management Center license, you can open the **Manage License** tab and click **Update License** button and enter the new license code.

Alternatively a license can be provided using the system property `hazelcast.mc.license` (see Starting with a License for details).

26.3. Socket Interceptor

If the Hazelcast cluster is configured to use a socket interceptor, you need to configure a socket interceptor for Management Center as well. You can click on the **Configure Socket Interceptor** button, enter the name of your socket interceptor class and the configuration parameters. Click the **Configure** button to save your configuration and enable socket interceptor. This class needs to be on your classpath when you're starting Management Center. The configuration parameters you provide will be used to invoke the `init` method of your socket interceptor implementation if it has such a method.

Socket Interceptor

Please enter the class name of your socket interceptor (make sure that you have the class on the classpath):

Class Name

List of configuration parameters.
Each line should have a key-value pair separated by an equals sign (=).
Example:
param1=value1
param2=value2

Parameters

Configure Socket Interceptor

Following is a sample socket interceptor class implementation:

```
package com.example;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;

public class SampleSocketInterceptor {
    // this method is optional
    public void init(Map<String, String> parameters) {
        // here goes the initialization logic for your socket interceptor
    }

    public void onConnect(Socket connectedSocket) throws IOException {
        // socket interceptor logic
        try {
            OutputStream out = connectedSocket.getOutputStream();
            InputStream in = connectedSocket.getInputStream();
            int multiplyBy = 2;
            while (true) {
                int read = in.read();
                if (read == 0) {
                    break;
                }
                out.write(read * multiplyBy);
                out.flush();
            }
        } catch (IOException e) {
            throw e;
        }
    }
}
```

A socket interceptor implementation needs to satisfy the following two conditions:

1. Have a no-argument constructor

2. Have a public onConnect method with the following signature:

    ```
    void onConnect(Socket connectedSocket) throws IOException
    ```

26.3.1. Disabling Socket Interceptor

To disable the socket interceptor, you need to click the **Configure Socket Interceptor** button first and then click the **Disable** button on the dialog.

26.4. Change Url

Hazelcast cluster members need to be configured with Management Center's URL before they're started. If Management Center's URL is changed for some reason, you can use this page to make Hazelcast members send their statistics to the new Management Center URL. However, this has the following caveats:

1. This configuration change is not persistent. If a member is restarted without any updates to its configuration, it will go back to sending its statistics to the original URL

2. If a new member joins the cluster, it will not know of the URL change, and will send its statistics to the URL that it's configured with.

Change Server URL

Cluster Name:

```
dev
```

Password:

```
••••••••
```

Member IP:

```
127.0.0.1
```

Member Port:

```
5701
```

Server URL:

```

```

SSL:

☐
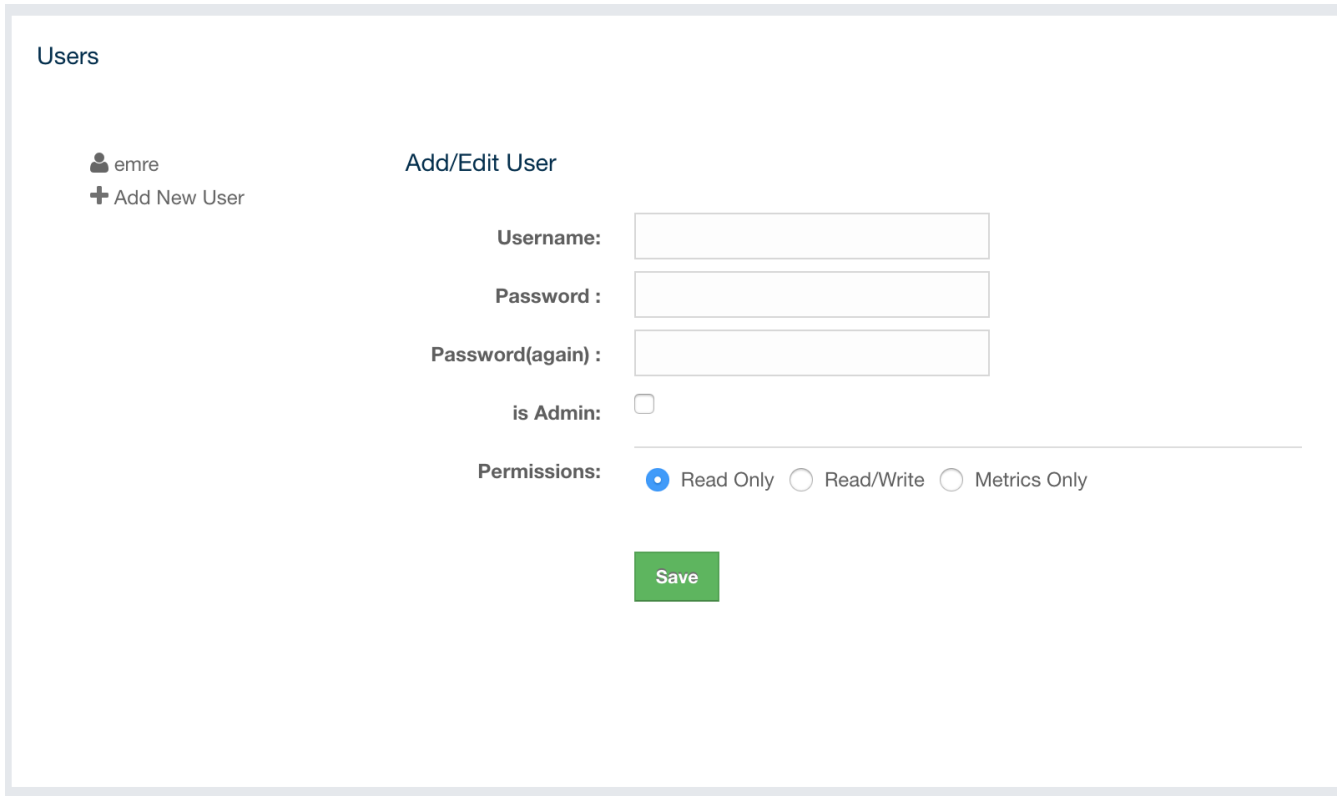
Set URL

To change the URL, enter the **Cluster Name** and **Password**, provide the IP address and port for one of the members, and specify the new Management Center URL in the **Server URL** field. If the cluster members are configured to use TLS/SSL for communicating between themselves, check the

SSL box. Clicking the **Set URL** button will update the Management Center URL.

26.5. Users

To add a user to the system, specify the username, e-mail and password in the **Add/Edit User** part of the page. If the user to be added will have administrator privileges, select **isAdmin** checkbox. **Permissions** field has the following checkboxes:

- **Metrics Only**: If this permission is given to the user, only **Home**, **Documentation** and **Time Travel** items will be visible at the toolbar on that user's session. Also, the users with this permission cannot browse a map or a cache to see their contents, cannot update a map configuration, run a garbage collection and take a thread dump on a cluster member, or shutdown a member (please see Monitoring Members).

- **Read Only**: If this permission is given to the user, only **Home**, **Documentation** and **Time Travel** items will be visible at the toolbar at that user's session. Also, users with this permission cannot update a map configuration, run a garbage collection and take a thread dump on a cluster member, or shutdown a member (please see Monitoring Members).

- **Read/Write**: If this permission is given to the user, **Home**, **Scripting**, **Console**, **Documentation** and **Time Travel** items will be visible. The users with this permission can update a map configuration and perform operations on the members.

After you enter/select all fields, click **Save** button to create the user. You will see the newly created user's username on the left side, in the **Users** part of the page.
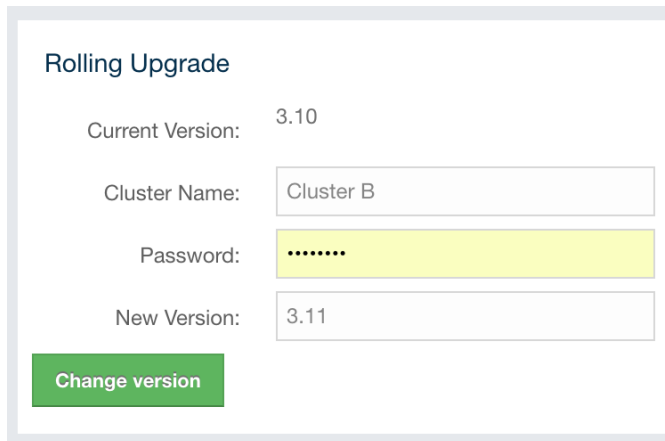
To edit or delete a user, select a username listed in the **Users**. Selected user information appears on the right side of the page. To update the user information, change the fields as desired and click the **Save** button. You can also change a user's password or delete the user account. To change the user's password, click the **Change Password** button. To delete the user from the system, click the **Delete**

button. Note that changing the password of a user and deleting the user account both require you to enter your own password.

26.6. Rolling Upgrade

The admin user can upgrade the cluster version once all members of the cluster have been upgraded to the intended codebase version as described in the Rolling Upgrade Procedure section of the Hazelcast IMDG Reference Manual.

Open the **Rolling Upgrade** tab to perform a rolling upgrade and change the cluster's version.

Enter the group name/password of the cluster and the version you want to upgrade the cluster to, and click on the **Change Version** button.
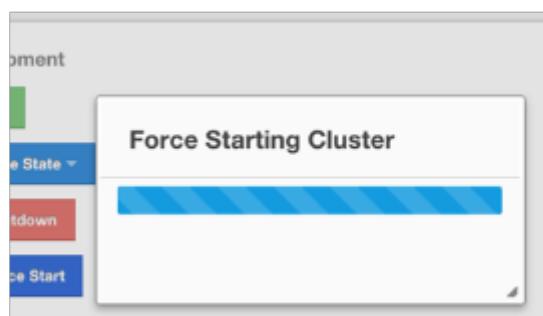
Once the operation succeeds, you will see the following notification:
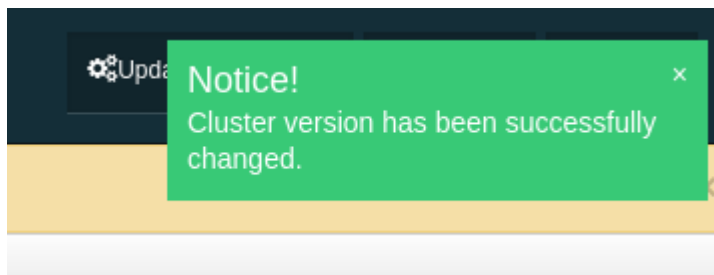
26.7. Hot Restart

Force Start the Cluster

Restart process cannot be completed if a member crashes permanently and cannot recover from the failure since it cannot start or it fails to load its own data. In that case, you can force the cluster to clean its persisted data and make a fresh start. This process is called **force start**.

Click the **Force Start** button. A pop-up will appear and stay on screen until the operation is triggered.

If an exception occurs, this exception message will be showed on the screen as a notification.

> The operations explained in this section (Hot Restart) can also be performed using REST API and the script `cluster.sh`. Please refer to the Using REST API for Cluster Management section and Using the Script cluster.sh section.

27. Checking Past Status with Time Travel

Use the Time Travel toolbar item to check the status of the cluster at a time in the past. When you select it on the toolbar, a small window appears on top of the page, as shown below.
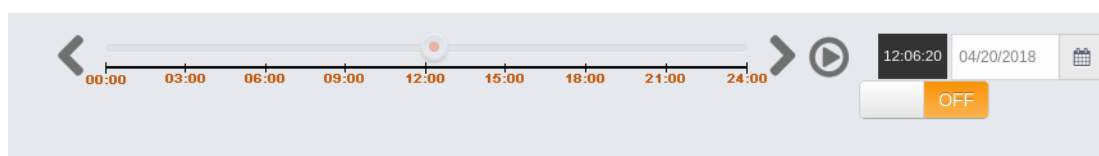
To see the cluster status in a past time, you should first enable the Time Travel. Click on the area where it says **OFF** (on the right of Time Travel window). It will turn to **ON** after it asks whether to enable the Time Travel with a dialog: click on **Enable** in the dialog to enable the Time Travel.

Once it is **ON**, the status of your cluster will be stored on your disk as long as your web server is alive.

You can go back in time using the slider and/or calendar and check your cluster's situation at the selected time. All data structures and members can be monitored as if you are using the Management Center normally (charts and data tables for each data structure and members). Using the arrow buttons placed at both sides of the slider, you can go back or further with steps of 5 seconds. It will show status if Time Travel has been **ON** at the selected time in past; otherwise, all the charts and tables will be shown as empty.

The historical data collected with Time Travel feature are stored in a file database on the disk. These files can be found in the folder `<User's Home Directory>/hazelcast-mancenter3.10.3`, e.g., `/home/someuser/hazelcast-mancenter3.10.3`. This folder can be changed using the

`hazelcast.mancenter.home` property on the server where Management Center is running.

Time travel data files are created monthly. Their file name format is `[group-name]-[year][month].db` and `[group-name]-[year][month].lg`. Time travel data is kept in the `*.db` files. The files with the extension `lg` are temporary files created internally and you do not have to worry about them.

> ℹ️ Due to security concerns, time travel can only be used if the cluster name consists of alphanumeric characters, underscores and dashes.

28. Clustered REST via Management Center

Hazelcast IMDG Enterprise

The Clustered REST API is exposed from Management Center to allow you to monitor clustered statistics of distributed objects.

28.1. Enabling Clustered REST

To enable Clustered REST on your Management Center, pass the following system property at startup. This property is disabled by default.

```
-Dhazelcast.mc.rest.enabled=true
```

28.2. Clustered REST API Root

The entry point for Clustered REST API is `/rest/`.

This resource does not have any attributes.

28.3. Clusters Resource

This resource returns a list of clusters that are connected to the Management Center.

28.3.1. Retrieve Clusters

- **Request Type:** GET
- **URL:** `/rest/clusters`
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters
```

- **Response:** `200` (application/json)
- **Body:**

```
["dev","qa"]
```

28.4. Cluster Resource

This resource returns information related to the provided cluster name.

28.4.1. Retrieve Cluster Information

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/
```

- **Response:** 200 (application/json)
- **Body:**

```
{"masterAddress":"192.168.2.78:5701"}
```

28.5. Members Resource

This resource returns a list of members belonging to the provided clusters.

28.5.1. Retrieve Members [GET] [/rest/clusters/{clustername}/members]

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/members
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/members
```

- **Response:** 200 (application/json)
- **Body:**

```
["192.168.2.78:5701","192.168.2.78:5702","192.168.2.78:5703","192.168.2.78:5704"]
```

28.6. Member Resource

This resource returns information related to the provided member.

28.6.1. Retrieve Member Information

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/members/{member}

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701
```

- **Response:** 200 (application/json)

- **Body:**

```
{
  "cluster":"dev",
  "name":"192.168.2.78:5701",
  "maxMemory":129957888,
  "ownedPartitionCount":68,
  "usedMemory":60688784,
  "freeMemory":24311408,
  "totalMemory":85000192,
  "connectedClientCount":1,
  "master":true
}
```

28.6.2. Retrieve Connection Manager Information

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/members/{member}/connectionManager

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701/connectionManager
```

- **Response:** 200 (application/json)

- **Body:**

```
{
    "clientConnectionCount":2,
    "activeConnectionCount":5,
    "connectionCount":5
}
```

28.6.3. Retrieve Operation Service Information

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/members/{member}/operationService

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701/operationService
```

- **Response:** 200 (application/json)

- **Body:**

```
{
    "responseQueueSize":0,
    "operationExecutorQueueSize":0,
    "runningOperationsCount":0,
    "remoteOperationCount":1,
    "executedOperationCount":461139,
    "operationThreadCount":8
}
```

28.6.4. Retrieve Event Service Information

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/members/{member}/eventService

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701/eventService
```

- **Response:** 200 (application/json)

- **Body:**

```
{
  "eventThreadCount":5,
  "eventQueueCapacity":1000000,
  "eventQueueSize":0
}
```

28.6.5. Retrieve Partition Service Information

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/members/{member}/partitionService
- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701/partitionService
```

- **Response:** 200 (application/json)
- **Body:**

```
{
  "partitionCount":271,
  "activePartitionCount":68
}
```

28.6.6. Retrieve Proxy Service Information

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/members/{member}/proxyService
- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701/proxyService
```

- **Response:** 200 (application/json)
- **Body:**

```
{
  "proxyCount":8
}
```

28.6.7. Retrieve All Managed Executors

- **Request Type:** GET

- **URL:** `/rest/clusters/{clustername}/members/{member}/managedExecutors`

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701/managedExecutors
```

- **Response:** 200 (application/json)

- **Body:**

```
["hz:system","hz:scheduled","hz:client","hz:query","hz:io","hz:async"]
```

28.6.8. Retrieve a Managed Executor

- **Request Type:** GET

- **URL:** `/rest/clusters/{clustername}/members/{member}/managedExecutors/{managedExecutor}`

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/members/192.168.2.78:5701
        /managedExecutors/hz:system
```

- **Response:** 200 (application/json)

- **Body:**

```
{
   "name":"hz:system",
   "queueSize":0,
   "poolSize":0,
   "remainingQueueCapacity":2147483647,
   "maximumPoolSize":4,
   "completedTaskCount":12,
   "terminated":false
}
```

28.7. Client Endpoints Resource

This resource returns a list of client endpoints belonging to the provided cluster. Please consider using the newly added Client Statistics Resource as it contains more detailed information about

clients.

28.7.1. Retrieve List of Client Endpoints

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/clients
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/clients
```

- **Response:** 200 (application/json)
- **Body:**

```
["192.168.2.78:61708"]
```

28.7.2. Retrieve Client Endpoint Information

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/clients/{client}
- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/clients/192.168.2.78:61708
```

- **Response:** 200 (application/json)
- **Body:**

```
{
  "uuid":"6fae7af6-7a7c-4fa5-b165-cde24cf070f5",
  "address":"192.168.2.78:61708",
  "clientType":"JAVA"
}
```

28.8. Maps Resource

This resource returns a list of maps belonging to the provided cluster.

28.8.1. Retrieve List of Maps

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/maps

- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/maps
```

- **Response:** 200 (application/json)
- **Body:**

```
["customers","orders"]
```

28.8.2. Retrieve Map Information

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/maps/{mapName}
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/maps/customers
```

- **Response:** 200 (application/json)
- **Body:**

```
{
    "cluster": "dev",
    "name": "customers",
    "ownedEntryCount": 5085,
    "backupEntryCount": 5076,
    "ownedEntryMemoryCost": 833940,
    "backupEntryMemoryCost": 832464,
    "heapCost": 1666668,
    "lockedEntryCount": 2,
    "dirtyEntryCount": 0,
    "hits": 602,
    "lastAccessTime": 1532689094579,
    "lastUpdateTime": 1532689094576,
    "creationTime": 1532688789256,
    "putOperationCount": 5229,
    "getOperationCount": 2162,
    "removeOperationCount": 150,
    "otherOperationCount": 3687,
    "events": 10661,
    "maxPutLatency": 48,
    "maxGetLatency": 35,
    "maxRemoveLatency": 18034,
    "avgPutLatency": 0.5674125071715433,
    "avgGetLatency": 0.2479185938945421,
    "avgRemoveLatency": 5877.986666666667
}
```

28.9. MultiMaps Resource

This resource returns a list of multimaps belonging to the provided cluster.

28.9.1. Retrieve List of MultiMaps

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/multimaps

- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/multimaps
```

- **Response:** 200 (application/json)

- **Body:**

```
["customerAddresses"]
```

28.9.2. Retrieve MultiMap Information

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/multimaps/{multimapname}

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/multimaps/customerAddresses
```

- **Response:** 200 (application/json)

- **Body:**

```
{
     "cluster": "dev",
     "name": "customerAddresses",
     "ownedEntryCount": 4862,
     "backupEntryCount": 4860,
     "ownedEntryMemoryCost": 0,
     "backupEntryMemoryCost": 0,
     "heapCost": 0,
     "lockedEntryCount": 1,
     "dirtyEntryCount": 0,
     "hits": 22,
     "lastAccessTime": 1532689253314,
     "lastUpdateTime": 1532689252591,
     "creationTime": 1532688790593,
     "putOperationCount": 5125,
     "getOperationCount": 931,
     "removeOperationCount": 216,
     "otherOperationCount": 373570,
     "events": 0,
     "maxPutLatency": 8,
     "maxGetLatency": 1,
     "maxRemoveLatency": 18001,
     "avgPutLatency": 0.3758048780487805,
     "avgGetLatency": 0.11170784103114931,
     "avgRemoveLatency": 1638.8472222222222
}
```

28.10. ReplicatedMaps Resource

This resource returns a list of replicated maps belonging to the provided cluster.

28.10.1. Retrieve List of ReplicatedMaps

- **Request Type:** GET

- **URL:** `/rest/clusters/{clustername}/replicatedmaps`

- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/replicatedmaps
```

- **Response:** `200` (application/json)

- **Body:**

```
["replicated-map-1"]
```

28.10.2. Retrieve ReplicatedMap Information

- **Request Type:** GET

- **URL:** `/rest/clusters/{clustername}/replicatedmaps/{replicatedmapname}`

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/replicatedmaps/replicated-map-1
```

- **Response:** `200` (application/json)

- **Body:**

```
{
    "cluster": "dev",
    "name": "replicated-map-1",
    "ownedEntryCount": 10955,
    "ownedEntryMemoryCost": 394380,
    "hits": 15,
    "lastAccessTime": 1532689312581,
    "lastUpdateTime": 1532689312581,
    "creationTime": 1532688789493,
    "putOperationCount": 11561,
    "getOperationCount": 1051,
    "removeOperationCount": 522,
    "otherOperationCount": 355552,
    "events": 6024,
    "maxPutLatency": 1,
    "maxGetLatency": 1,
    "maxRemoveLatency": 1,
    "avgPutLatency": 0.006400830377994983,
    "avgGetLatency": 0.012369172216936251,
    "avgRemoveLatency": 0.011494252873563218
}
```

28.11. Queues Resource

This resource returns a list of queues belonging to the provided cluster.

28.11.1. Retrieve List of Queues

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/queues
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/queues
```

- **Response:** 200 (application/json)
- **Body:**

```
["messages"]
```

28.11.2. Retrieve Queue Information

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/queues/{queueName}
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/queues/messages
```

- **Response:** 200 (application/json)
- **Body:**

```
{
  "cluster":"dev",
  "name":"messages",
  "ownedItemCount":55408,
  "backupItemCount":55408,
  "minAge":0,
  "maxAge":0,
  "aveAge":0,
  "numberOfOffers":55408,
  "numberOfRejectedOffers":0,
  "numberOfPolls":0,
  "numberOfEmptyPolls":0,
  "numberOfOtherOperations":0,
  "numberOfEvents":0,
  "creationTime":1403602694196
}
```

28.12. Topics Resource

This resource returns a list of topics belonging to the provided cluster.

28.12.1. Retrieve List of Topics

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/topics
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/topics
```

- **Response:** 200 (application/json)
- **Body:**

```
["news"]
```

28.12.2. Retrieve Topic Information

- **Request Type:** GET
- **URL:** /rest/clusters/{clustername}/topics/{topicName}
- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/topics/news
```

- **Response:** 200 (application/json)

- **Body:**

```json
{
    "cluster":"dev",
    "name":"news",
    "numberOfPublishes":56370,
    "totalReceivedMessages":56370,
    "creationTime":1403602693411
}
```

28.13. Executors Resource

This resource returns a list of executors belonging to the provided cluster.

28.13.1. Retrieve List of Executors

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/executors

- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/executors
```

- **Response:** 200 (application/json)

- **Body:**

```
["order-executor"]
```

28.13.2. Retrieve Executor Information [GET] [/rest/clusters/{clustername}/executors/{executorName}]

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/executors/{executorName}

- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/executors/order-executor
```

- **Response:** 200 (application/json)

- **Body:**

```
{
    "cluster":"dev",
    "name":"order-executor",
    "creationTime":1403602694196,
    "pendingTaskCount":0,
    "startedTaskCount":1241,
    "completedTaskCount":1241,
    "cancelledTaskCount":0
}
```

28.14. Client Statistics Resource

This resource returns a list of clients belonging to the provided cluster.

28.14.1. Retrieve List of Client UUIDs

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/clientStats

- **Request:**

```
curl http://localhost:8083/hazelcast-mancenter/rest/clusters/dev/clientStats
```

- **Response:** 200 (application/json)

- **Body:**

```
[
    "f3b1e0e9-ea67-41b2-aba5-ea7480f02a93",
    "cebf4dc9-852c-4605-a181-ffe1cca371a4",
    "2371eed5-26e0-4470-92c1-41ea17110ef6",
    "139990b3-fbc0-43a8-9c12-be53913333f7",
    "d0364a1e-8665-46a8-af1d-be1af5580d07",
    "7f337f8a-3538-4b5c-8ffc-9d4ae459e956",
    "6ef9b6e5-5add-40d9-9319-ce502f55b5fc",
    "fead3a99-19de-431c-9dd0-d6ecc4a4b9c8",
    "e788e04e-2ded-4992-9d76-52c1973216e5",
    "654fc9fb-c5c1-48a0-9b69-0c129fce860f"
]
```

28.14.2. Retrieve Detailed Client Statistics [GET] [/rest/clusters/{clustername}/clientStats/{clientUuid}]

- **Request Type:** GET

- **URL:** /rest/clusters/{clustername}/clientStats/{clientUuid}

- **Request:**

```
curl http://localhost:8083/hazelcast-
mancenter/rest/clusters/dev/clientStats/2371eed5-26e0-4470-92c1-41ea17110ef6
```

- **Response:** 200 (application/json)

- **Body:**

```
{
    "type": "JAVA",
    "name": "hz.client_7",
    "address": "127.0.0.1:42733",
    "clusterConnectionTimestamp": 1507874427419,
    "enterprise": true,
    "lastStatisticsCollectionTime": 1507881309434,
    "osStats": {
      "committedVirtualMemorySize": 12976173056,
      "freePhysicalMemorySize": 3615662080,
      "freeSwapSpaceSize": 8447324160,
      "maxFileDescriptorCount": 1000000,
      "openFileDescriptorCount": 191,
      "processCpuTime": 252980000000,
      "systemLoadAverage": 83.0,
      "totalPhysicalMemorySize": 16756101120,
      "totalSwapSpaceSize": 8447324160
    },
    "runtimeStats": {
      "availableProcessors": 12,
      "freeMemory": 135665432,
      "maxMemory": 3724541952,
      "totalMemory": 361234432,
      "uptime": 6894992,
      "usedMemory": 225569000
    },
    "nearCacheStats": {
      "CACHE": {
        "a-cache": {
          "creationTime": 1507874429719,
          "evictions": 0,
          "hits": 0,
          "misses": 50,
          "ownedEntryCount": 0,
          "expirations": 0,
          "ownedEntryMemoryCost": 0,
          "lastPersistenceDuration": 0,
          "lastPersistenceKeyCount": 0,
          "lastPersistenceTime": 0,
          "lastPersistenceWrittenBytes": 0,
          "lastPersistenceFailure": ""
```

```json
        },
        "b.cache": {
          "creationTime": 1507874429973,
          "evictions": 0,
          "hits": 0,
          "misses": 50,
          "ownedEntryCount": 0,
          "expirations": 0,
          "ownedEntryMemoryCost": 0,
          "lastPersistenceDuration": 0,
          "lastPersistenceKeyCount": 0,
          "lastPersistenceTime": 0,
          "lastPersistenceWrittenBytes": 0,
          "lastPersistenceFailure": ""
        }
      },
      "MAP": {
        "other,map": {
          "creationTime": 1507874428638,
          "evictions": 0,
          "hits": 100,
          "misses": 50,
          "ownedEntryCount": 0,
          "expirations": 0,
          "ownedEntryMemoryCost": 0,
          "lastPersistenceDuration": 0,
          "lastPersistenceKeyCount": 0,
          "lastPersistenceTime": 0,
          "lastPersistenceWrittenBytes": 0,
          "lastPersistenceFailure": ""
        },
        "employee-map": {
          "creationTime": 1507874427959,
          "evictions": 0,
          "hits": 100,
          "misses": 50,
          "ownedEntryCount": 0,
          "expirations": 0,
          "ownedEntryMemoryCost": 0,
          "lastPersistenceDuration": 0,
          "lastPersistenceKeyCount": 0,
          "lastPersistenceTime": 0,
          "lastPersistenceWrittenBytes": 0,
          "lastPersistenceFailure": ""
        }
      }
    },
    "userExecutorQueueSize": 0,
    "memberConnection": "ALL",
    "version": "UNKNOWN"
  }
```

29. Clustered JMX via Management Center

Hazelcast IMDG Enterprise

Clustered JMX via Management Center allows you to monitor clustered statistics of distributed objects from a JMX interface.

29.1. Configuring Clustered JMX

In order to configure Clustered JMX, use the following command line parameters for your Management Center deployment.

- `-Dhazelcast.mc.jmx.enabled=true` (default is false)
- `-Dhazelcast.mc.jmx.port=9000` (optional, default is 9999)
- `-Dcom.sun.management.jmxremote.ssl=false`

Starting with Hazelcast Management Center 3.8.4, you can also use the following parameters:

- `-Dhazelcast.mc.jmx.rmi.port=9001` (optional, default is 9998)
- `-Dhazelcast.mc.jmx.host=localhost` (optional, default is server's host name)

With embedded Jetty, you do not need to deploy your Management Center application to any container or application server.

You can start Management Center application with Clustered JMX enabled as shown below.

```
java -Dhazelcast.mc.jmx.enabled=true -Dhazelcast.mc.jmx.port=9999 -Dcom.sun.
management.jmxremote.ssl=false -jar hazelcast-mancenter-3.10.3.war
```

Once Management Center starts, you should see a log similar to below.

```
INFO: Management Center 3.3
Jun 05, 2014 11:55:32 AM com.hazelcast.webmonitor.service.jmx.impl.JMXService
INFO: Starting Management Center JMX Service on port :9999
```

You should be able to connect to Clustered JMX interface from the address `localhost:9999`.

You can use `jconsole` or any other JMX client to monitor your Hazelcast Cluster. As a sample, below is the `jconsole` screenshot of the Clustered JMX hierarchy.

```
▼ 📁 ManagementCenter[dev]
   ▼ 📁 Clients
      ▶ 🌐 "192.168.2.79:56678"
   ▼ 📁 Executors
      ▶ 🌐 testExecutor
   ▼ 📁 Maps
      ▶ 🌐 a
      ▶ 🌐 b
      ▶ 🌐 testMap
      ▶ 🌐 testMap3
   ▼ 📁 Members
      ▶ 🌐 "192.168.2.79:5701"
      ▶ 🌐 "192.168.2.79:5702"
      ▶ 🌐 "192.168.2.79:5703"
      ▶ 🌐 "192.168.2.79:5704"
   ▼ 📁 MultiMaps
      ▶ 🌐 testMultiMap
   ▼ 📁 Queues
      ▶ 🌐 testQueue
   ▼ 📁 Services
      ▶ 📁 "ManagedExecutor[hz:async]"
      ▶ 📁 "ManagedExecutor[hz:client]"
      ▶ 📁 "ManagedExecutor[hz:global-operation]"
      ▶ 📁 "ManagedExecutor[hz:io]"
      ▶ 📁 "ManagedExecutor[hz:query]"
      ▶ 📁 "ManagedExecutor[hz:scheduled]"
      ▶ 📁 "ManagedExecutor[hz:system]"
      ▶ 📁 ConnectionManager
      ▶ 📁 EventService
      ▶ 📁 OperationService
      ▶ 📁 PartitionService
      ▶ 📁 ProxyService
   ▼ 📁 Topics
      ▶ 🌐 testTopic
   ▶ 🌐 dev
```

29.1.1. Enabling TLS/SSL for Clustered JMX

By default, Clustered JMX is served unencrypted. To enable TLS/SSL for Clustered JMX, use the following command line parameters for your Management Center deployment.

- `-Dhazelcast.mc.jmx.ssl=true` (default is false)
- `-Dhazelcast.mc.jmx.ssl.keyStore=path to your keyStore`
- `-Dhazelcast.mc.jmx.ssl.keyStorePassword=password for your keyStore`

Following is an example on how to start Management Center with a TLS/SSL enabled Clustered JMX service on port 65432:

```
java -Dhazelcast.mc.jmx.enabled=true -Dhazelcast.mc.jmx.port=65432 -Dhazelcast.mc.jmx
.ssl=true -Dhazelcast.mc.jmx.ssl.keyStore=/some/dir/selfsigned.jks -Dhazelcast.mc.jmx
.ssl.keyStorePassword=yourpassword -jar hazelcast-mancenter-3.10.3.war
```

> ℹ️ You can encrypt the keyStore password and pass it as a command line argument in encrypted form for improved security. See Variable Replacers for more information.

Then you can use the following command to connect to the Clustered JMX service using JConsole with address `localhost:65432`:

```
jconsole -J-Djavax.net.ssl.trustStore=/some/dir/selftrusted.ts -J-Djavax.net.ssl
.trustStorePassword=trustpass
```

Additional TLS/SSL Configuration Options

Following are some additional command line arguments that you can use to configure TLS/SSL for clustered JMX:

- `-Dhazelcast.mc.jmx.ssl.keyStoreType`: Type of the keystore. Its default value is JKS.
- `-Dhazelcast.mc.jmx.ssl.keyManagerAlgorithm`: Name of the algorithm based on which the authentication keys are provided. System default will be used if none provided. You can find out the default by calling `javax.net.ssl.KeyManagerFactory#getDefaultAlgorithm` method.

29.2. Clustered JMX API

The management beans are exposed with the following object name format.

```
ManagementCenter[`*cluster name*`]:type=<`*object type*`>,name=<`*object name*`>
,member="<`*cluster member IP address*`>"
```

Object name starts with `ManagementCenter` prefix. Then it has the cluster name in brackets followed by a colon. After that, `type`,`name` and `member` attributes follows, each separated with a comma.

- `type` is the type of object. Values are `Clients`, `Executors`, `Maps`, `Members`, `MultiMaps`, `Queues`, `Counters`, `Services`, and `Topics`.
- `name` is the name of object.
- `member` is the member address of object (only required if the statistics are local to the member).

A sample bean is shown below.

```
ManagementCenter[dev]:type=Services,name=OperationService,member="192.168.2.79:5701"
```

Here is the list of attributes that are exposed from the Clustered JMX interface.

- **ManagementCenter[`<ClusterName>`]**
- ClientStats
  - `<Client UUID>`
    - HeapUsedMemory
    - HeapFreeMemory
    - HeapMaxMemory
    - HeapTotalMemory
    - ClientName
    - AvailableProcessors
    - Uptime
    - Enterprise

- MemberConnection
- ClusterConnectionTimestamp
- LastStatisticsCollectionTime
- UserExecutorQueueSize
- CommittedVirtualMemorySize
- FreePhysicalMemorySize
- FreeSwapSpaceSize
- MaxFileDescriptorCount
- OpenFileDescriptorCount
- ProcessCpuTime
- SystemLoadAverage
- TotalPhysicalMemorySize
- TotalSwapSpaceSize
- Version
- Address
- Type
- CACHE
  - `<Cache Name>`
    - Evictions
    - Expirations
    - Hits
    - Misses
    - OwnedEntryCount
    - OwnedEntryMemoryCost
    - LastPersistenceDuration
    - LastPersistenceKeyCount
    - LastPersistenceTime
    - LastPersistenceWrittenBytes
    - LastPersistenceFailure
    - CreationTime
- MAP
  - `<Map Name>`
    - Evictions
    - Expirations
    - Hits

- Misses
- OwnedEntryCount
- OwnedEntryMemoryCost
- LastPersistenceDuration
- LastPersistenceKeyCount
- LastPersistenceTime
- LastPersistenceWrittenBytes
- LastPersistenceFailure
- CreationTime

- Clients
  - `<Client Address>`
    - Address
    - ClientType
    - Uuid
- Executors
  - `<Executor Name>`
    - Cluster
    - Name
    - StartedTaskCount
    - CompletedTaskCount
    - CancelledTaskCount
    - PendingTaskCount
- Maps
  - `<Map Name>`
    - Cluster
    - Name
    - BackupEntryCount
    - BackupEntryMemoryCost
    - CreationTime
    - DirtyEntryCount
    - Events
    - GetOperationCount
    - HeapCost
    - Hits
    - LastAccessTime

- LastUpdateTime

- LockedEntryCount

- MaxGetLatency

- MaxPutLatency

- MaxRemoveLatency

- OtherOperationCount

- OwnedEntryCount

- PutOperationCount

- RemoveOperationCount

- AvgGetLatency

- AvgPutLatency

- AvgRemoveLatency

- ReplicatedMaps
  - <Replicated Map Name>
    - Cluster

    - Name

    - BackupEntryCount

    - BackupEntryMemoryCost

    - CreationTime

    - DirtyEntryCount

    - Events

    - GetOperationCount

    - HeapCost

    - Hits

    - LastAccessTime

    - LastUpdateTime

    - LockedEntryCount

    - MaxGetLatency

    - MaxPutLatency

    - MaxRemoveLatency

    - OtherOperationCount

    - OwnedEntryCount

    - PutOperationCount

    - RemoveOperationCount

    - AvgGetLatency

- AvgPutLatency
- AvgRemoveLatency
- Members
  - <Member Address>
    - ConnectedClientCount
    - HeapFreeMemory
    - HeapMaxMemory
    - HeapTotalMemory
    - HeapUsedMemory
    - IsMaster
    - OwnedPartitionCount
- MultiMaps
  - <MultiMap Name>
    - Cluster
    - Name
    - BackupEntryCount
    - BackupEntryMemoryCost
    - CreationTime
    - DirtyEntryCount
    - Events
    - GetOperationCount
    - HeapCost
    - Hits
    - LastAccessTime
    - LastUpdateTime
    - LockedEntryCount
    - MaxGetLatency
    - MaxPutLatency
    - MaxRemoveLatency
    - OtherOperationCount
    - OwnedEntryCount
    - PutOperationCount
    - RemoveOperationCount
    - AvgGetLatency
    - AvgPutLatency

- AvgRemoveLatency
- Queues
  - <Queue Name>
    - Cluster
    - Name
    - MinAge
    - MaxAge
    - AvgAge
    - OwnedItemCount
    - BackupItemCount
    - OfferOperationCount
    - OtherOperationsCount
    - PollOperationCount
    - RejectedOfferOperationCount
    - EmptyPollOperationCount
    - EventOperationCount
    - CreationTime
- Counters
  - <Counter Name>
    - Cluster
    - Name
    - ReplicaCount
    - Time
    - OpsPerSecInc (for each member)
    - OpsPerSecDec (for each member)
    - Value (for each member)
- Services
  - ConnectionManager
    - ActiveConnectionCount
    - ClientConnectionCount
    - ConnectionCount
  - EventService
    - EventQueueCapacity
    - EventQueueSize
    - EventThreadCount

- OperationService

  - ExecutedOperationCount

  - OperationExecutorQueueSize

  - OperationThreadCount

  - RemoteOperationCount

  - ResponseQueueSize

  - RunningOperationsCount

- PartitionService

  - ActivePartitionCount

  - PartitionCount

- ProxyService

  - ProxyCount

- ManagedExecutor[hz::async]

  - Name

  - CompletedTaskCount

  - MaximumPoolSize

  - PoolSize

  - QueueSize

  - RemainingQueueCapacity

  - Terminated

- ManagedExecutor[hz::client]

  - Name

  - CompletedTaskCount

  - MaximumPoolSize

  - PoolSize

  - QueueSize

  - RemainingQueueCapacity

  - Terminated

- ManagedExecutor[hz::global-operation]

  - Name

  - CompletedTaskCount

  - MaximumPoolSize

  - PoolSize

  - QueueSize

  - RemainingQueueCapacity

- Terminated
  - ManagedExecutor[hz::io]
    - Name
    - CompletedTaskCount
    - MaximumPoolSize
    - PoolSize
    - QueueSize
    - RemainingQueueCapacity
    - Terminated
  - ManagedExecutor[hz::query]
    - Name
    - CompletedTaskCount
    - MaximumPoolSize
    - PoolSize
    - QueueSize
    - RemainingQueueCapacity
    - Terminated
  - ManagedExecutor[hz::scheduled]
    - Name
    - CompletedTaskCount
    - MaximumPoolSize
    - PoolSize
    - QueueSize
    - RemainingQueueCapacity
    - Terminated
  - ManagedExecutor[hz::system]
    - Name
    - CompletedTaskCount
    - MaximumPoolSize
    - PoolSize
    - QueueSize
    - RemainingQueueCapacity
    - Terminated
- Topics
  - `<Topic Name>`

- Cluster
- Name
- CreationTime
- PublishOperationCount
- ReceiveOperationCount
- Flake ID Generators
  - `<Generator Name>`
    - Cluster
    - Name
    - Time
    - OpsPerSec (per member)

29.3. Integrating with New Relic

Use the Clustered JMX interface to integrate Hazelcast Management Center with **New Relic**. To perform this integration, attach New Relic Java agent and provide an extension file that describes which metrics will be sent to New Relic.

Please see Custom JMX instrumentation by YAML on the New Relic webpage.

Below is an example Map monitoring `.yml` file for New Relic.

```
name: Clustered JMX
version: 1.0
enabled: true

jmx:
  - object_name: ManagementCenter[clustername]:type=Maps,name=mapname
    metrics:
      - attributes: PutOperationCount, GetOperationCount, RemoveOperationCount, Hits,\
            BackupEntryCount, OwnedEntryCount, LastAccessTime, LastUpdateTime
        type: simple
  - object_name: ManagementCenter[clustername]:type=Members,name="member address in\
        double quotes"
    metrics:
      - attributes: OwnedPartitionCount
        type: simple
```

Put the `.yml` file in the `extensions` folder in your New Relic installation. If an `extensions` folder does not exist there, create one.

After you set your extension, attach the New Relic Java agent and start Management Center as shown below.

```
java -javaagent:/path/to/newrelic.jar -Dhazelcast.mc.jmx.enabled=true\
    -Dhazelcast.mc.jmx.port=9999 -jar hazelcast-mancenter-3.10.3.war
```

If your logging level is set as FINER, you should see the log listing in the file `newrelic_agent.log`, which is located in the `logs` folder in your New Relic installation. Below is an example log listing.

```
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINE:
    JMX Service : querying MBeans (1)
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    JMX Service : MBeans query ManagementCenter[dev]:type=Members,
    name="192.168.2.79:5701", matches 1
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric OwnedPartitionCount : 68
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    JMX Service : MBeans query ManagementCenter[dev]:type=Maps,name=orders,
    matches 1
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric Hits : 46,593
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric BackupEntryCount : 1,100
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric OwnedEntryCount : 1,100
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric RemoveOperationCount : 0
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric PutOperationCount : 118,962
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric GetOperationCount : 0
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric LastUpdateTime : 1,401,962,426,811
Jun 5, 2014 14:18:43 +0300 [72696 62] com.newrelic.agent.jmx.JmxService FINER:
    Recording JMX metric LastAccessTime : 1,401,962,426,811
```

Then you can navigate to your New Relic account and create Custom Dashboards. Please see Creating custom dashboards.

While you are creating the dashboard, you should see the metrics that you are sending to New Relic from Management Center in the **Metrics** section under the JMX folder.

29.4. Integrating with AppDynamics

Use the Clustered JMX interface to integrate Hazelcast Management Center with **AppDynamics**. To perform this integration, attach AppDynamics Java agent to the Management Center.

For agent installation, refer to Install the App Agent for Java page.

For monitoring on AppDynamics, refer to Using AppDynamics for JMX Monitoring page.

After installing AppDynamics agent, you can start Management Center as shown below.

```
java -javaagent:/path/to/javaagent.jar -Dhazelcast.mc.jmx.enabled=true\
    -Dhazelcast.mc.jmx.port=9999 -jar hazelcast-mancenter-3.10.3.war
```

When Management Center starts, you should see the logs below.

```
Started AppDynamics Java Agent Successfully.
Hazelcast Management Center starting on port 8080 at path : /hazelcast-mancenter
```

30. Management Center Documentation

To see the Management Center documentation (this Reference Manual), click on the **Documentation** button located at the toolbar. This Management Center manual will appear as a tab.

31. Suggested Heap Size

Table 1. For 2 Cluster Members

Mancenter Heap Size	# of Maps	# of Queues	# of Topics
256m	3k	1k	1k
1024m	10k	1k	1k

Table 2. For 10 Members

Mancenter Heap Size	# of Maps	# of Queues	# of Topics
256m	50	30	30
1024m	2k	1k	1k

Table 3. For 20 Members

Mancenter Heap Size	# of Maps	# of Queues	# of Topics
256m*	N/A	N/A	N/A
1024m	1k	1k	1k

* With 256m heap, Management Center is unable to collect statistics.