RightITnow ECM

Installation Guide for ECM 6.1.x

April 2022

www.RightITnow.com

Copyright Notice

Trademarks

RightITnow
USA
101A Clay Street, #150
San Francisco, CA 94111

www.RightITnow.com

Part Number: instdoc-08-2016

Contents

Chapter 1.
Installing RightITnow ECM

Overview

RightITnow ECM is a real-time, cross-domain event correlation software application that enables enterprises to optimize IT Operations processes, so they can drive down costs, resolve problems faster and assure end user services. It achieves this by automating the event to alert to incident life cycle and bridging the gap between IT Operations center and the Service Desk – driving higher productivity and effectiveness.

Prerequisites

Before attempting to install and configure RightITnow ECM, ensure that you satisfy the following hardware, software, and security requirements. For more details, please see the Hardware and Software Requirements document bundled with ECM.

As an alternative to installing ECM from scratch, we provide a virtual appliance (OVA) and a Docker image for download. Please see the relevant sections in this guide for more details.

Hardware Requirements

Hardware requirements can vary between deployments, depending on the volume of alerts being processed and the amount of data persisted to the database. Generally ECM can run off a single production-grade server. The following are the standard recommended specifications for running ECM in production:

- **CPU:** Dual quad-core processors (8 cores)
- **Memory:** 64GB DDR3 ECC Registered
- **Hardisk:** 128GB+ SSD for the ECM application, 1TB+ SSD for the database (see sections below about database and disk space requirements)
- **RAID:** RAID configurations are recommended where/if applicable
- **Network:** 1Gbit/s Ethernet connection

For smaller-scale deployments or development/UAT servers, these specifications can be scaled down considerably. A dual-core CPU with 16GB of RAM and 256GB HDD would suffice.

Operating Systems

The ECM application is developed in Java and thus a variety of operating systems are supported. Linux-based setups are recommended due to the nature of the product.

Operating Systems (64-bit)

- Red Hat Enterprise Linux 6 or later
- CentOS 6 or later
- Ubuntu 14.04 LTS or later
- Microsoft Windows 10 or Microsoft Windows Server 2016 or later
- Mac OS X 10.6 (Snow Leopard) or later

Software (64-bit where possible)

- Oracle JRE/JDK 8 or Oracle JDK 11 LTS or OpenJDK 11 (including Microsoft Build of OpenJDK and Amazon Corretto)
- MySQL Server 5.6/5.7/8.0 or MariaDB 10.4 (see Installing and Configuring MySQL or MariaDB)
  - Amazon Aurora is supported
  - SSL/TLS connections are supported, please refer to Enabling SSL/TLS connections to MySQL or MariaDB
  - If using MySQL Cluster, RightITnow requires the following server configuration:
    - Gigabit network with low latencies
    - MySQL 5.7 with NDB 7.5.5 or above
    - "NDBCLUSTER" as the cluster engine name
- Microsoft SQL Server 2016/2017/2019 can also be used as a database (see Hardware and Software Requirements)
  - SSL/TLS connections are supported, please refer to Enabling SSL/TLS connections to SQL Server
  - Azure SQL Database is supported. The Standard or Premium tiers are recommended for a production ECM deployment, while Basic can be used for a development/UAT server. Please refer to the Service Tiers documentation.
- Ensure that your firewall is open for Tomcat (TCP port 8080)
- The following LDAP data stores are supported for integration:
  - OpenLDAP
  - Microsoft Active Directory

Installing the packaged RightITnow ECM

Installing and configuring the packaged RightITnow ECM involves the following general steps:

- Installing and Configuring MySQL/MariaDB or SQL Server
- Installing and Configuring RightITnow ECM
- Configuring Connectors

The following subsections describe how to accomplish these steps.

Installing and Configuring MySQL or MariaDB

Installing and configuring MySQL Server or MySQL Cluster depends on your environment and choice of operating system. The official documentation can be found here. MariaDB is a fork of MySQL Server that is now the default package for MySQL on many Linux distributions, and is fully supported by ECM. The official documentation can be found here.

IMPORTANT: In MySQL 5.7, some SQL modes are enabled by default, but are not supported by ECM. Only the following SQL modes should be enabled:

- sql-mode="STRICT_TRANS_TABLES, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER, NO_ENGINE_SUBSTITUTION"

In MySQL 8.0 and MariaDB 10.4, some SQL modes are enabled by default, but are not supported by ECM. Only the following SQL modes should be enabled:

- sql-mode="STRICT_TRANS_TABLES, ERROR_FOR_DIVISION_BY_ZERO, NO_ENGINE_SUBSTITUTION"

The SQL modes can be set in the MySQL configuration file (my.ini or my.cnf).

MySQL or MariaDB can be setup on the same server as the ECM application, or on a separate server on the same local network. It is important to avoid high latencies between separate servers as this will have a noticeable effect on performance and reduce the event-processing rate of the application. The default settings that MySQL and MariaDB ship with should suffice, however the following settings can be changed beforehand which can avoid certain known issues:

- change the **innodb_lock_wait_timeout** setting from 50 to 300
- for any errors related to "Packet for query is too large", change the **max_allowed_packet** setting to a larger size such as 8MB

Further optimizations to the database can be done – it is recommended to follow best practices and to read the relevant optimization guides:

- http://dev.mysql.com/doc/refman/5.7/en/optimization.html
- https://mariadb.com/kb/en/optimization-and-tuning/

Installing and Configuring SQL Server

Installing and configuring SQL Server depends on your environment and choice of operating system. The official documentation can be found here. To replicate your database and distribute it to different locations, please refer to the official documentation here.

Microsoft SQL Server can be setup on the same server as the ECM application, or on a separate server on the same local network. It is important to avoid high latencies between separate servers as this will have a noticeable effect on performance and reduce the event-processing rate of the application.

Migrating RightITnow ECM from MySQL to SQL Server

To migrate ECM from an existing MySQL/MariaDB to a new SQL Server database:

1. Update an existing installation to 6.0 or later.
2. Download the configuration backup file (see Backing up the configuration).
3. Install a new RightITnow ECM 6.0 configured to use a new database in SQL Server (see Installing and Configuring RightITnow ECM).
4. Restore the configuration backup file (see Restoring the configuration) using the option "Provide RightITnow ECM configuration backup" in the License Setup screen.

> *NOTE: The migration process does not preserve the events, alerts and audit records from the original MySQL installation.*

Installing and Configuring RightITnow ECM

To install RightITnow ECM:

1. Download the **rightitnow.zip** file from the RightITnow website (use the link supplied in the registration email you received).
2. Issue the following command from the same directory that contains the file **rightitnow.zip** that you downloaded in the previous step (for Windows, you can extract the archive to a directory of your choosing):

```
unzip rightitnow.zip
```

   This creates a directory named **rightitnow**.

3. MySQL/MariaDB Run the following command to create a database for RightITnow ECM, replacing *dbname* with a database name of your choice:

```
mysql -u root -p -e "create database dbname default character set utf8"
```

   MacOS use **/usr/local/mysq/bin/mysql** instead of **mysql**.

   Microsoft Windows run the command from **c:\Program Files\MySQL\SQL Server *x.x*\bin**, where *x.x* is your version of MySQL Server.

SQL Server Create the database with a name of your choice, using Microsoft SQL Server Management Studio or any other appropriate software.

```
create database dbname
```

4. Start Tomcat as follows:

MacOS/Redhat/CentOS/Ubuntu Run the following command from the same directory from which you ran the unzip command in step 2:

```
./rightitnow/bin/catalina.sh start
```

Microsoft Windows From a command prompt, navigate to the bin directory in the **rightitnow** folder you extracted in step 2, and then run the following command:

```
catalina.bat start
```

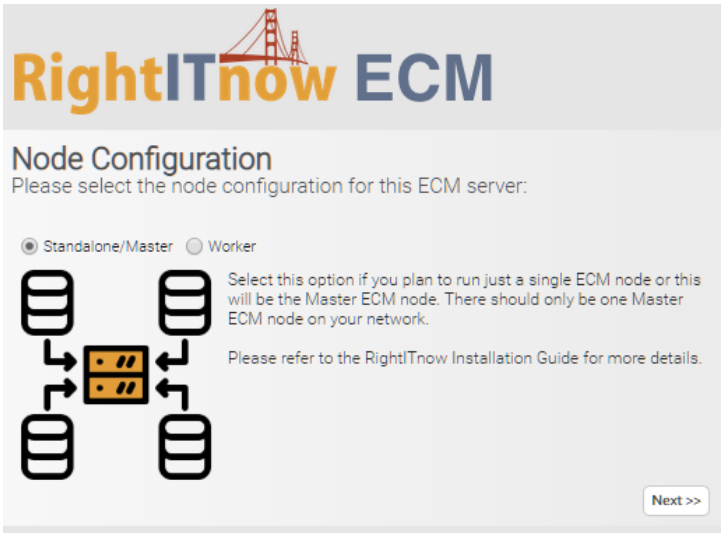
5. Launch the URL http://localhost:8080/rightitnow.

*NOTE: The URL will be different if the system is being accessed remotely; replace **localhost** with the hostname or IP address of the machine running the RightITnow ECM software.*

The End User License Agreement screen appears:

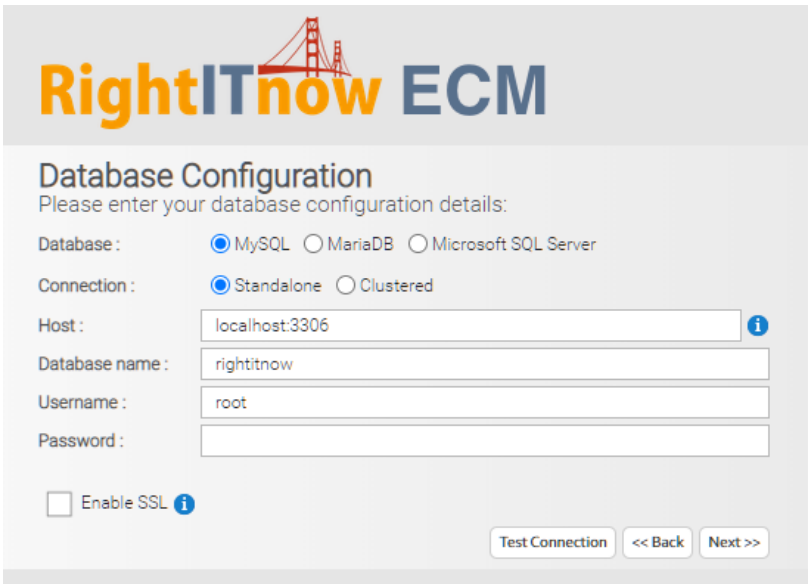
6. Click the checkbox to accept the license terms and then click **Next**.

The Node Configuration screen appears:



7. If you are installing a standalone ECM or you are setting up a Master instance, select **Standalone/Master**. If you are setting up a distributed worker, select the **Worker** option (for more information see Setting up Distributed Workers). Click **Next** to proceed**.**

   The Database Configuration screen appears:



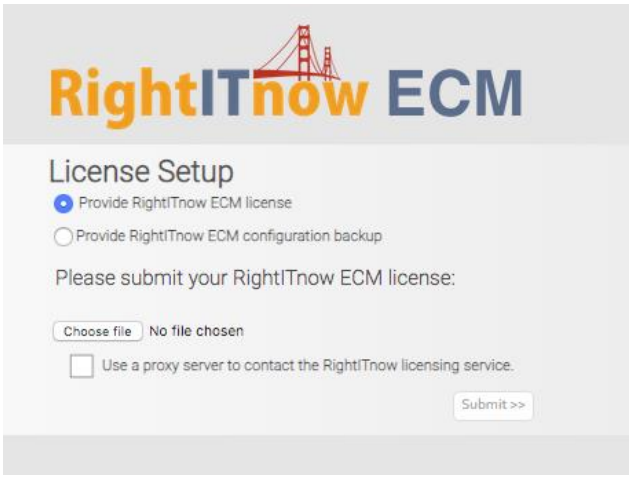
NOTE: You must set up the MySQL cluster before installing RightITnow ECM in a clustered configuration.

NOTE: If connecting to SQL Server, please refer to Using Windows Authentication with SQL Server.

NOTE: If enabling SSL/TLS for MySQL/MariaDB connections, please refer to Enabling SSL/TLS connections to MySQL or MariaDB.

NOTE: If enabling SSL/TLS for SQL Server connections, please refer to Enabling SSL/TLS connections to SQL Server.

8. Enter the values for your database and then click **Next**. The database name will be the name you provided in step 3 above.
9. The License Setup screen appears:



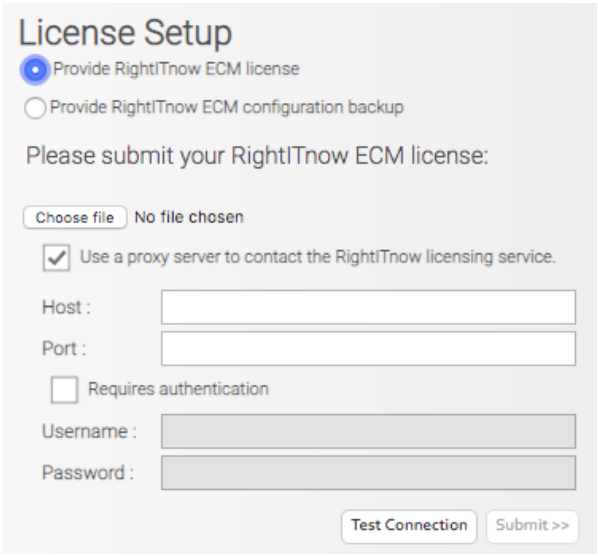
You use the License Setup screen, as described in the following steps, to locate the ECM license that was sent to you, and then to contact the RightITnow Licensing Service to validate the license.

10. Click the **Browse** button to locate the license key.
11. If you do not need to use a proxy server, click **Submit** to contact the RightITnow Licensing Service.

    If you do need to use a proxy server:

    a. Click the proxy server checkbox.

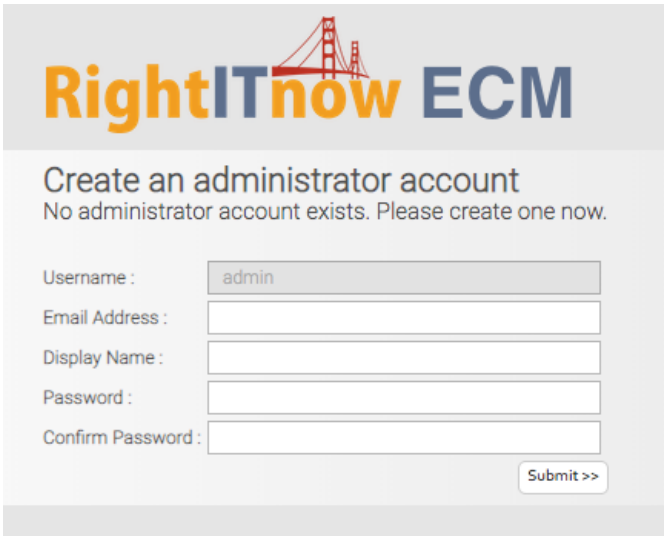
    The Proxy Server fields appear:



b. Complete the **Host** and **Port** fields, and the **Username** and **Password** fields, if required.

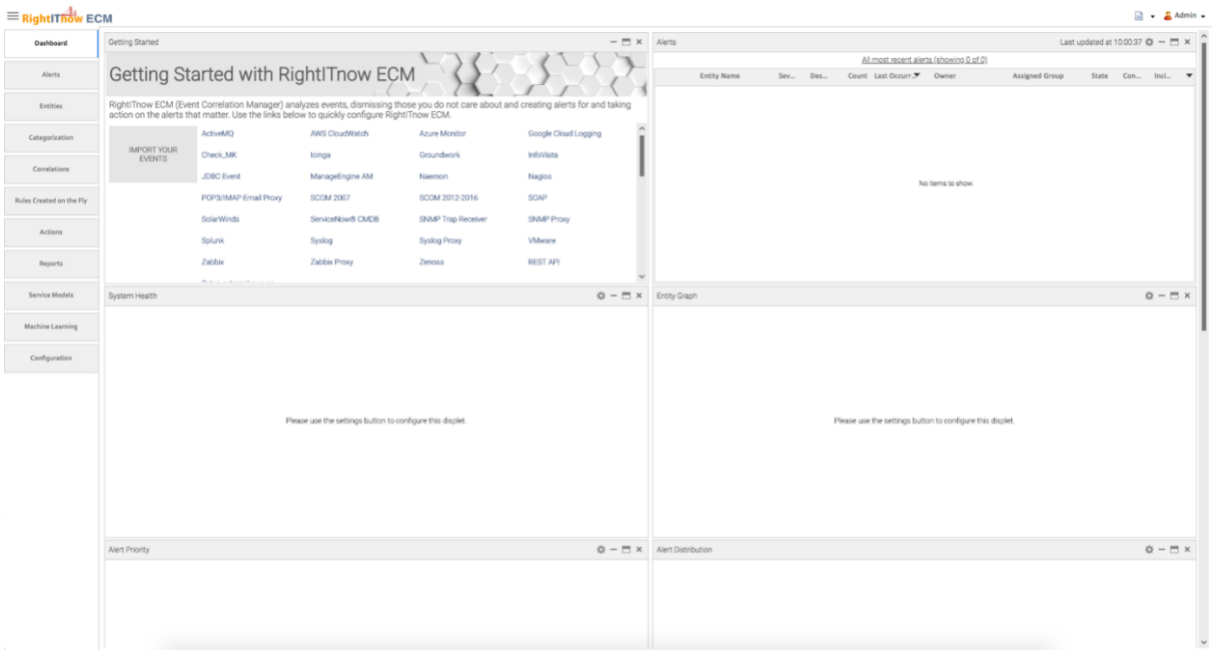
c. Click **Submit**.

The setup process begins and then the Administrator setup screen appears:

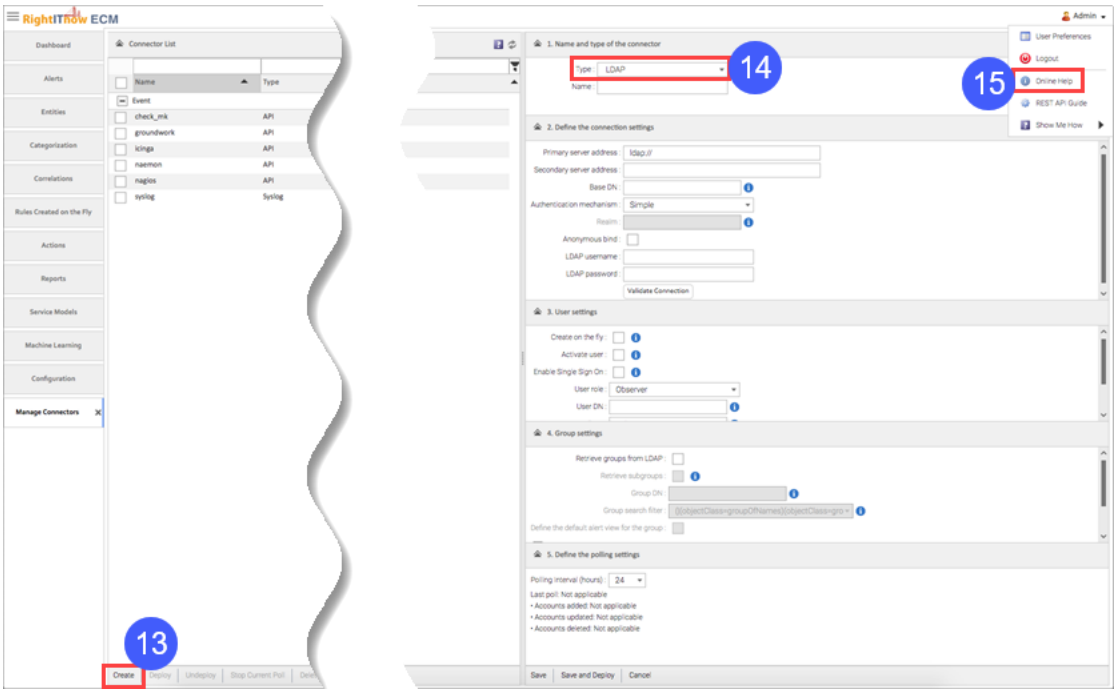


12. Complete the administrator account fields, and then click **Submit**.

After clicking **Submit**, the **Dashboard** appears:



13. If you would like to setup LDAP or Active Directory to import users, go to the **Configuration** tab and click on **Manage Connectors** under External Systems. The Manage Connectors tab appears.



14. Click **Create**.
15. Select the LDAP connector type.
16. See **Configuring Connectors** in the online help for more details.

Using ECM as a Service

Starting ECM as a service depends on your operating system. To start ECM as a service:

Linux with systemd

1. Issue the command replacing the params *ExecStart, ExecStop, user* and *group* with the correct info of your choice. The script below can also be found under the *utility* directory of the ECM installation.

```
sudo vi /etc/systemd/system/rightitnow.service


[Unit]

Description=RightITnow ECM

After=network.target


[Service]

Type=forking
```

```
ExecStart=/rightitnow/bin/startup.sh

ExecStop=/rightitnow/bin/shutdown.sh



User=<user>

Group=<group>

UMask=0007

RestartSec=10

Restart=always



[Install]

WantedBy=multi-user.target
```

2. Restart the daemon:

```
sudo systemctl daemon-reload
```

3. Enable RightITnow ECM service:

```
sudo systemctl enable rightitnow.service
```

4. Start RightITnow ECM service:

```
sudo systemctl start rightitnow
```

Microsoft Windows

There are two ways to create the service, choose the appropriate one:

1. Create a service in the Windows Services app. The official documentation covers this in detail.
   a. Issue the following command from the *rightitnow/bin* directory, where <*service_name*> is the name of the service. This needs to be done under a user with Administrator privileges. Note that the service will be called “Apache Tomcat 8.5 <service_name>” and that no spaces can be used in the custom service name:

```
C:\rightitnow\bin>service install <service_name>
```

   b. Go to Services, select the service, select Properties, set the Startup Type to Automatic and apply the correct Log On account that you want ECM to run under.

c. Start the Service and wait for Tomcat to start up. The *catalina* log file under *rightitnow/logs* can be used to monitor the deployment of ECM.

d. To remove the service, use the same command above with the *remove* parameter:

```
C:\rightitnow\bin>service remove <service_name>
```

2. Create a shortcut in the startup folder with the following command executed via command line, replacing the <*rightitnow*> path where applicable:

```
C:\rightitnow\bin>powershell "$s=(New-Object -COM
WScript.Shell).CreateShortcut('%userprofile%\Start
Menu\Programs\Startup\\RightITnow.lnk');$s.TargetPath='<rightitnow>\bi
n\startup.bat';$s.WorkingDirectory='<rightitnow>\bin';$s.Save()"
```

Using Windows Authentication with SQL Server

Authenticating against SQL Server can be done either with a username and password, which requires Mixed Mode Authentication to be enabled on SQL Server, or by using Windows Authentication. For the latter, the following prerequisites and configuration is required before setting up ECM, depending on your host and environment.

- Except for NTLMv2, the ECM and SQL Server hosts must be part of the same Windows domain or in trusted domains.
- The user that will be used for authentication needs to be added in SQL Server for authentication to work, and granted *db_owner* permission to the ECM database.
- The MSSQL JDBC driver used by ECM does not support Extended Protection, this should be set to Off in the SQL Server properties.

Native Authentication (Windows hosts)

If ECM is hosted on a Windows server, an additional DLL needs to be downloaded on the host, depending on the architecture:

- Download for 64-bit hosts
- Download for 32-bit hosts

The file needs to be placed in a folder that is on the PATH environmental variable, for example *C:\Windows*. Once this is done, start ECM and perform the setup. No username and password need to be provided and they will not be stored by ECM since the credentials of the currently logged-in Windows user (or if running ECM as a service, the Log On user defined for the service) will be used.

NTLMv2 (Windows or Linux hosts)

NTLM does not require the ECM host to be on the same domain as the SQL Server. All that is required is the username (supplied in *DOMAIN\username* format) and the password. Note that these are stored by ECM locally in a properties file.

Kerberos (Windows or Linux hosts)

Kerberos is the preferred authentication method if the ECM server is hosted on an OS other than Windows. This guide assumes that Kerberos is already setup and working properly in your environment, and only describes the configuration required for ECM to connect to SQL Server.

The instructions below assume the following machine names:

- **win-dc01.dev.local**- the domain controller (Active Directory)
- **win-sql01.dev.local** - the SQL server instance
- **linux-ecm01.dev.local** – the ECM server instance (Linux)

All the machines above are members of the **DEV.LOCAL** domain.

Register a Service Principal Name

The first step is to register a Service Principal Name (SPN) with Active Directory, which assumes the role of the Key Distribution Center in a Windows domain. The SPN, after it is registered, maps to the Windows account OR computer that started the SQL Server instance service. If the SPN registration has not been performed or fails, the Windows security layer cannot determine the account associated with the SPN, and Kerberos authentication is not used. For this example, the SPN would be **MSSQLSvc/win-sql01.dev.local:1433@DEV.LOCAL**

More information on this topic can be found here, but the basic steps are as follows:

- If the SQL Server instance is running under a Windows account that has permissions to set SPNs, then the SPN is probably already set correctly. To verify this, run the command below on the domain controller to list the SPNs for the relevant account or computer name, which should show the correct SPN for the SQL Server.

```
setspn -L DEV.LOCAL\accountname

OR
setspn -L DEV.LOCAL\computername
```

- If the SQL Server instance is running under a Windows account that does not have permissions to set SPNs (such as local or network services accounts), then the SPN must be registered manually with the following command:

```
setspn -A MSSQLSvc/win-sql01.dev.local:1433@DEV.LOCAL
DEV.LOCAL\accountname
OR
setspn -A MSSQLSvc/win-sql01.dev.local:1433@DEV.LOCAL
DEV.LOCAL\computername
```

- It is important to make sure that the SPN is only registered to one user account or one computer. To verify this, use the `setspn -X` command and if duplicates are found, use the `setspn -D` command to unregister the duplicates.

Create a domain user account for ECM

Once the SPN is setup correctly, create a new domain user account (or use an existing account) that will be used by ECM to login to SQL Server. The account's properties, such as password expiration and Kerberos encryption levels, will depend on the security and group policies of your organization.

The domain user should then be added to the list of users with permission to connect to SQL Server and needs to be granted *db_owner* permission to the database that will be used by ECM.

At this stage it is recommended to verify if Kerberos authentication is working correctly with SQL Server. Login to a domain computer using the newly created user account and using Microsoft SQL Server Management Studio (or any other appropriate software) login to the SQL Server using Windows authentication. Run the SQL query below to determine if connections from the client computer are using Kerberos. If the *auth_scheme* is 'NTLM', then Kerberos is not working correctly and the configuration needs to revisited. Note that NTLM is always used for local connections so make sure to test from a remote client.

```
select client_net_address, auth_scheme from sys.dm_exec_connections
```

Results | Messages

	client_net_address	auth_scheme
1	192.168.12.119	KERBEROS

Export a keytab for the domain user

ECM uses a keytab file to authenticate itself to the domain controller and SQL server. This file contains the private key for the account and should be protected accordingly. This example assumes that the account created previously is called *ecmuser*. To generate the file, run the command below (all on a single line) on the domain controller, and then move the generated keytab to the ECM server. After running the ktpass command, check for duplicate SPNs as explained previously to make sure that the SPN was not registered on the *ecmuser* account as well.

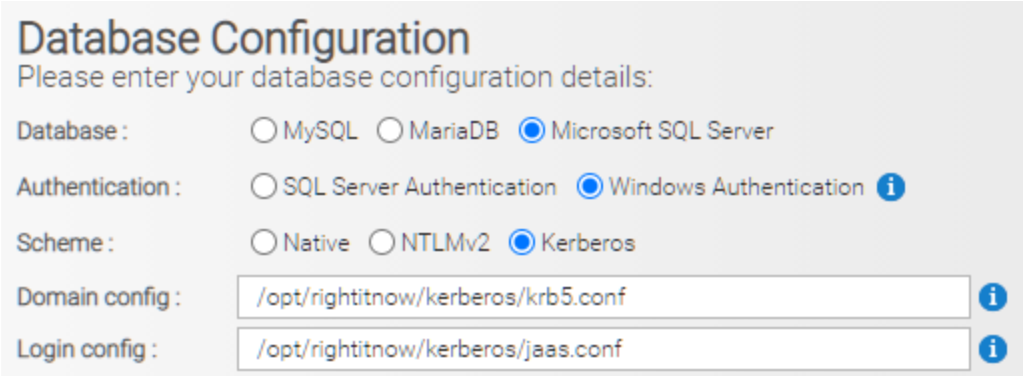
```
ktpass /out c:\ecmuser.keytab /mapuser ecmuser@DEV.LOCAL /princ MSSQLSvc/win-
sql01.dev.local:1433@DEV.LOCAL /pass ecmuserpassword /ptype KRB5_NT_PRINCIPAL
/crypto All /kvno 0
```

NOTE: the *kvno* in the above command is only correct if you have a newly created account 'ecmuser'. If the account already existed and ktpass was already issued before, AD has incremented the key version number stored within AD (operational/dynamic attribute 'msDS-KeyVersionNumber'). When you use ktpass consecutively you first must check the mentioned attribute and use that value +1 as the value for '/kvno'. More information on the ktpass command can be found here. A quick way to see the attribute for a user is the following PowerShell command:

```
dsquery * -filter sAMAccountName=ecmuser -attr msDS-KeyVersionNumber
```

Create the Kerberos configuration for ECM

The Kerberos configuration needs to be provided to ECM via two configuration files, *krb5.conf* and *jaas.conf*. These files should be created on the ECM server and placed in a location accessible to ECM. Their paths need to be provided to ECM during the database configuration phase of the ECM setup:



1. Create **krb5.conf**: In the ECM installation directory structure create a **krb5.conf** file with the following content:

```
[libdefaults]

	default_realm = DEV.LOCAL

	default_tkt_enctypes = rc4-hmac aes256-cts-hmac-sha1-96 aes128-cts-hmac-
sha1-96

	default_tgs_enctypes = rc4-hmac aes256-cts-hmac-sha1-96 aes128-cts-hmac-
sha1-96

	permitted_enctypes = rc4-hmac aes256-cts-hmac-sha1-96 aes128-cts-hmac-
sha1-96

	forwardable=true

[realms]

	DEV.LOCAL = {

		kdc = win-dc01.dev.local:88

		default_domain = DEV.LOCAL

	}

[domain_realm]

	dev.local= DEV.LOCAL

	.dev.local= DEV.LOCAL
```

2. In the **krb5.conf** file:

   a. Replace the occurrences of **dev.local** with your domain, respecting the upper/lower case in the example.

b. Replace **kdc** with your AD server.

NOTE: Multiple AD servers are supported by adding additional **kdc** entries under the [realms] section. If this is done, then the following should be added under the [libdefaults] section:

```
        dns_lookup_kdc = true

        dns_lookup_realm = true
```

3. Create **jaas.conf**: In the ECM installation directory structure create a **jaas.conf** file with the following content:

```
SQLJDBCDriver {

    com.sun.security.auth.module.Krb5LoginModule required

    principal="MSSQLSvc/win-sql01.dev.local:1433@DEV.LOCAL"

    useTicketCache=false

    renewTGT=false

    useKeyTab=true

    storeKey=true

    keyTab="/path/to/ecmuser.keytab"

    doNotPrompt=true;

};
```

4. In the **jaas.conf** file:

   a. Replace **principal** with the SPN you registered previously.

   b. Set **keytab** to the location of your **ecmuser.keytab** file in the local file system.

Configure ECM to use Kerberos authentication

When installing ECM, select Windows authentication and Kerberos on the Database setup screen, and provide the required information. For troubleshooting please refer to Troubleshooting Kerberos.

Enabling SSL/TLS connections to MySQL or MariaDB

ECM supports encrypted connections to the MySQL or MariaDB server. If the database server has the *require_secure_transport* setting enabled, then SSL/TLS has to be enabled or ECM will not be able to communicate with the database at all. For more details on enabling encrypted connections please refer to the official documentation for MySQL or for MariaDB. This guide assumes that you have enabled support for at least one of the TLS protocols (v1, v1.1 or v1.2) in your database server.

If you wish to automatically trust the database server's certificate, select the *Trust server certificate* option during the ECM database setup. This means that the server is

automatically trusted and you do not need to provide the certificates to ECM. The connection will still be encrypted.

If you do not want to automatically trust the certificates, you need to have access to the following files:

- **ca.pem**: the Certificate Authority (CA) certificate file used to sign the client and server certificates
- **client-cert.pem**: the client public key certificate file which is used during client authentication
- **client-key.pem**: the client private key file which is used during client authentication

If MySQL generated the certificates and keys automatically or you used the *mysql_ssl_rsa_setup* utility, these files are usually located in MySQL's data directory. Otherwise use the certificate and key files that were used to setup SSL/TLS in MySQL.

These files will need to be stored in a Java trust store and keystore accessible by ECM. Use Java's keytool (typically located in the bin subdirectory of your JDK or JRE installation) to import the server certificates. For more information please refer to the Connector/J JDBC documentation.

1. Import the CA certificate to a Java trust store and set a password:

```
keytool -importcert -alias MySQLCACert -file ca.pem -keystore truststore.jks
-storepass changeme
```

2. Convert the client key and certificate files to a PKCS #12 archive and set a password:

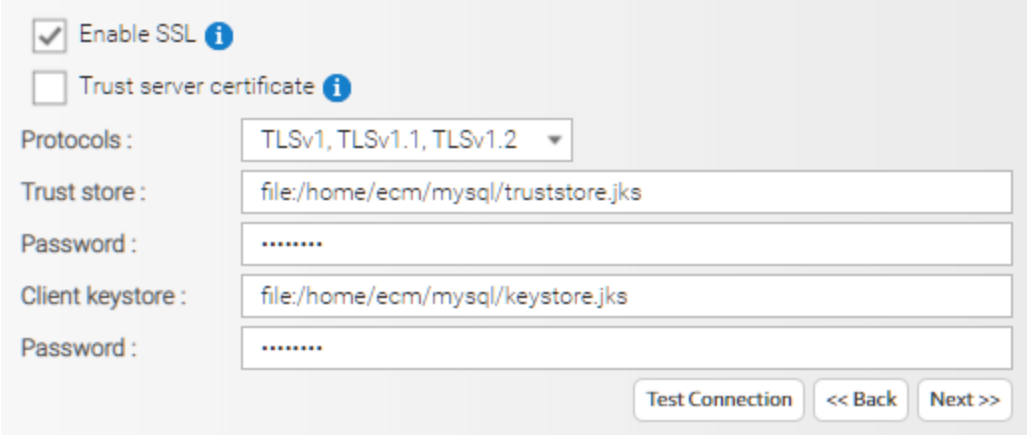
```
openssl pkcs12 -export -in client-cert.pem -inkey client-key.pem -name
"mysqlclient" -passout pass:changeme -out client-keystore.p12
```

3. Import the PKCS #12 archive into a Java keystore and set a password:

```
keytool -importkeystore -srckeystore client-keystore.p12 -srcstoretype
pkcs12 -srcstorepass changeme -destkeystore keystore.jks -deststoretype JKS
-deststorepass changeme
```

After the last step you can delete the PKCS #12 archive. Place these two files (*truststore.jks* and *keystore.jks*) on the same server as ECM in a location that is accessible by ECM.

During the ECM setup, you will need to select the *Enable SSL/TLS* option during the database configuration phase. This will show several fields:



- *Protocols*: here you can select the protocols supported by your database server
- *Trust store and password*: the location (on the ECM server) and password of the Java trust store containing the database server's CA certificate
- *Client keystore and password*: the location (on the ECM server) and password of the Java keystore containing the client key and certificate provided by the database server

Once these fields are filled in, test the connection and if successful, resume the normal setup process.

To debug issues with SSL/TLS, you can add the system property *-Djavax.net.debug=all* to the CATALINA_OPTS property in rightitnow/bin/catalina.sh (or catalina.bat on Windows) so that you can see what keystores and truststores are being used, as well as what is going on during the SSL handshake and certificate exchange. This information will be shown in the rightitnow/logs/catalina.out log.

Enabling SSL/TLS connections to SQL Server

ECM supports encrypted connections to SQL Server. For more details on enabling encrypted connections please refer to the official documentation. This guide assumes that you have enabled TLS support and assigned a certificate to your SQL Server instance.

If you wish to automatically trust the database server's certificate, select the *Trust server certificate* option during the ECM database setup. This means that the server is automatically trusted and you do not need to provide the certificates to ECM. The connection will still be encrypted.

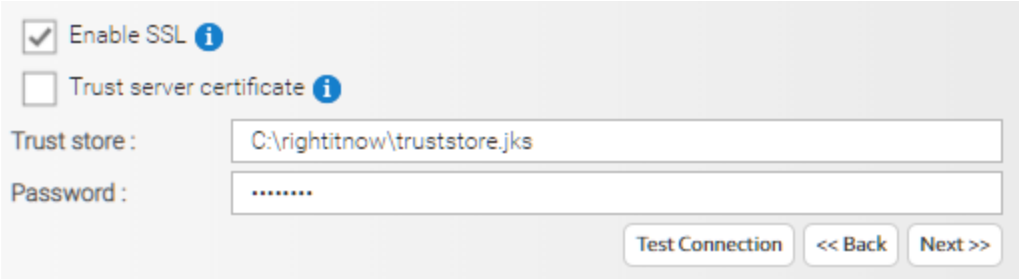
If you do not want to automatically trust the certificate, you need to have access to the certificate being used by the server instance. If not already done, the certificate can be exported as a DER-encoded binary X.509 (.cer) file using the Microsoft Management Console, as explained here. It might also be necessary to export the Certificate Authority's (CA) certificate that was used to sign the server certificate, but this is usually not required if the CA is well-known.

The certificate(s) will need to be stored in a Java trust store accessible by ECM. Use Java's keytool (typically located in the bin subdirectory of your JDK or JRE installation) to import the server certificate(s):

```
keytool -import -v -trustcacerts -alias sqlserver -file
C:\path\to\sqlserver.cer -keystore C:\rightitnow\truststore.jks -storepass
changeme
```

Place the generated *truststore.jks* on the same server as ECM in a location that is accessible by ECM.

During the ECM setup, you will need to select the *Enable SSL/TLS* option during the database configuration phase, and provide the path to the truststore and the password, as shown below:



Once these fields are filled in, test the connection and if successful, resume the normal setup process.

> *NOTE: When entering the database configuration details, the hostname of the server needs to match exactly with the CN field in the imported certificate.*

To debug issues with SSL/TLS, you can add the system property *-Djavax.net.debug=all* to the CATALINA_OPTS property in rightitnow/bin/catalina.sh (or catalina.bat on Windows) so that you can see what keystores and truststores are being used, as well as what is going on during the SSL handshake and certificate exchange. This information will be shown in the rightitnow/logs/catalina.out log.

Enabling HTTPS access to ECM

ECM is deployed inside the Tomcat application container; therefore enabling HTTPS (SSL/TLS) access to the ECM web application involves enabling SSL/TLS within Tomcat. The official documentation covers this in detail, however the general steps are outlined below.

Setting up the keystore

1. Create a new keystore with a keypair, making note of the password and ensuring the alias field and the "first and last name" field matches the hostname of the system, in this example it is *example.rightitnow.com*

```
$ keytool -genkeypair -alias example.rightitnow.com -keyalg RSA -
keysize 2048 -keystore rightitnow.keystore
```

```
Enter keystore password:

Re-enter new password:

What is your first and last name?

  [Unknown]:  example.rightitnow.com

What is the name of your organizational unit?

  [Unknown]:  Internal IT

What is the name of your organization?

  [Unknown]:  RightITnow

What is the name of your City or Locality?

  [Unknown]:  Islington

What is the name of your State or Province?

  [Unknown]:  London

What is the two-letter country code for this unit?

  [Unknown]:  GB

Is CN=example.rightitnow.com, OU=Internal IT, O=RightITnow,
L=Islington, ST=London, C=GB correct?

  [no]:  yes


Enter key password for <example.rightitnow.com>

    (RETURN if same as keystore password):
```

2. Extract a certificate signing request (CSR) - this should be signed by a certificate authority to obtain the signed certificate for your domain. This step may be different depending on the requirements of your CA.

```
keytool -certreq -keyalg RSA -alias example.rightitnow.com -file
example.csr -keystore rightitnow.keystore
```

3. Once your certificate has been issued by the CA, download it from them specifying that you will be installing it in Tomcat. Usually they provide 2 certificates, the one for your domain and another which is the trusted root CA certificate. You need to import these into the java keystore that you created in step 1:

```
keytool -import -trustcacerts -file ca.crt -keystore
rightitnow.keystore
```

```
keytool -importcert -file example.crt -alias example.rightitnow.com -
keystore rightitnow.keystore
```

Modify the Tomcat configuration

4. Edit rightitnow/conf/server.xml and add the following under the 8080 HTTP Connector definition, ensuring the keystoreFile is pointing to the keystore and the password is correct:

```
<Connector port="8443"
protocol="org.apache.coyote.http11.Http11Protocol"
maxHttpHeaderSize="262144" SSLEnabled="true" maxThreads="500"
scheme="https" secure="true" keystoreFile="rightitnow.keystore"
keystorePass="PASSWORD_HERE" clientAuth="false" sslProtocol="TLS"
sslEnabledProtocols="TLSv1.2,TLSv1.1,TLSv1" />
```

5. Run the following command and ensure there are no errors:

```
rightitnow/bin/catalina.sh configtest
```

6. Restart ECM and navigate to the web interface on port 8443, you can use the browser certificate inspector to validate that the certificate is being served correctly.
7. If you wish to use the standard HTTP and HTTPS ports (80 and 443), we recommend setting up redirection in your server's firewall (80 to 8080 and 443 to 8443). The example below shows how to do this using iptables:

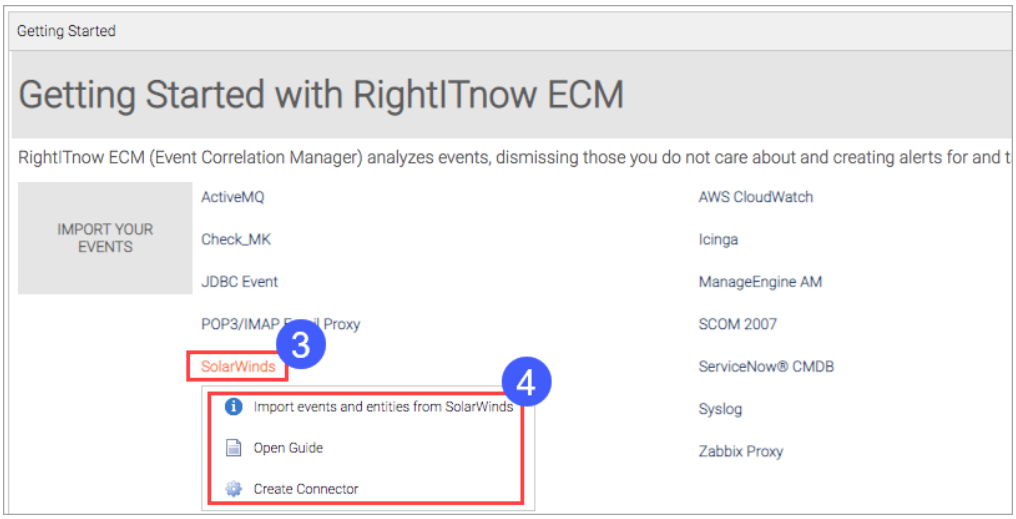
```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port
8080

iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port
8443
```

Configuring Connectors

If you would like to use a connector with RightITnow ECM, then the Getting Started displet is where you can find the documentation, resources and quick links to configure the connectors.

To use the Getting Started displet:

1. Login to ECM, and then click the Dashboard tab on the left.
2. Select the Getting Started displet.



3. Select a connector.
4. Use the corresponding inks to download the associated software and artifacts.

Deploying the RightITnow ECM Virtual Appliance

While deploying ECM in production is typically done via the packaged ECM, RightITnow provides a virtual appliance in OVA format which can be quickly deployed on any virtualization platform that supports OVA, or locally if using software such as VMware Workstation Player. The appliance contains all dependencies required to run ECM, including Java and MySQL. This reduces the time and effort required to deploy ECM and is especially useful to try out our product or for setting up development and UAT servers.

To download the appliance, use the download link supplied in the email you received when signing up for a trial of ECM. The file is called *rightitnow-ecm.ova*.

Virtual Appliance Details

The appliance is configured with the following virtual hardware and is created and exported from VMware vSphere 6.5 (virtual hardware version 13):

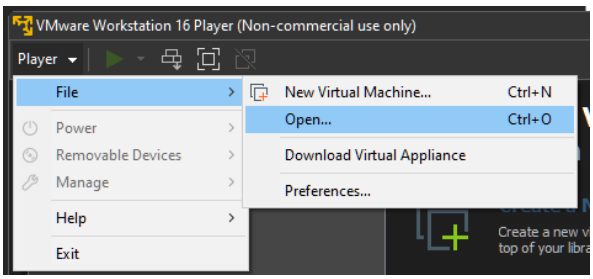
- **vCPUs:** 4 (1 core per socket)
- **Memory:** 16GB
- **Hardisk:** 40GB SCSI-lsilogic
- **Network:** VmxNet3
- **Guest OS:** Ubuntu 20.04.3 LTS

It is possible to reduce the number of vCPUs and the memory allocation although it is recommended not to go below 2 vCPUs and 12 GB of memory. By default, ECM is setup to use up to 8 GB of memory and under heavy load it might result in an out of memory error if this cannot be allocated.

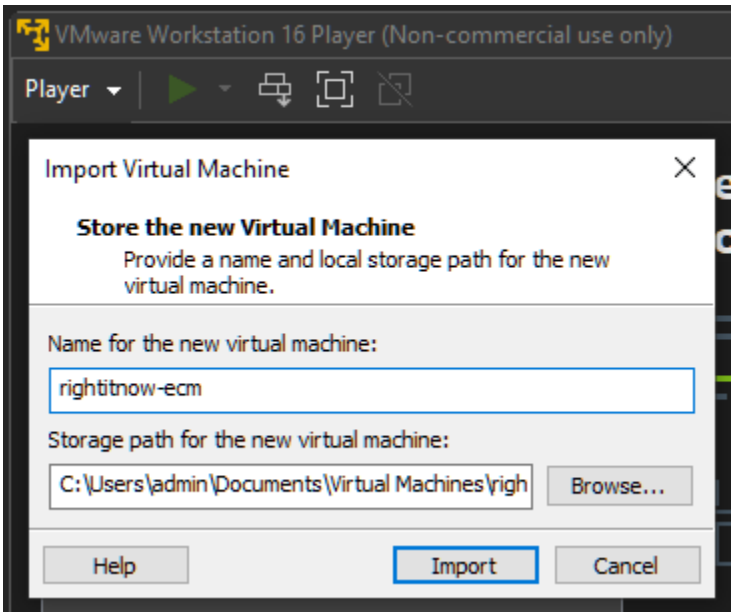
Deploying the Virtual Appliance

Once downloaded, the OVA file can be deployed on any platform that supports it, such as VMware vSphere, KVM and Proxmox VE. It can also be deployed locally through software such as VirtualBox and VMware Workstation Player, with the latter being the easiest to work with. The following instructions for VMware Workstation Player apply to version 16 and above:

1. Open VMware Workstation Player and go to Player → File → Open



2. Select the *rightitnow-ecm.ova* file and click Open.
3. Name the virtual machine and select a location to import it to.

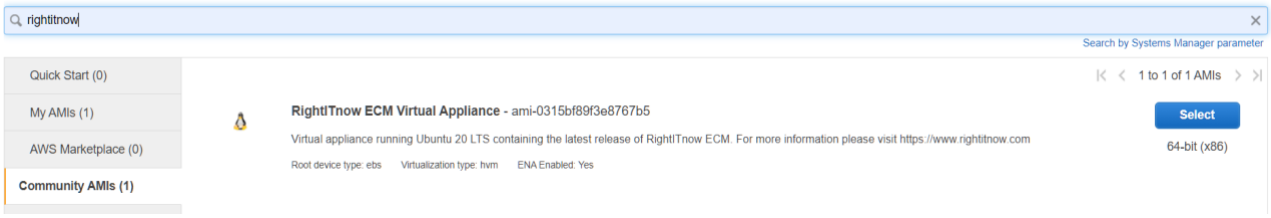


4. Click Import and wait for the OVA to be imported. It will then appear in the list of VMs.
5. To edit the virtual hardware, select the VM and click on Edit virtual machine settings, if required.
6. To power on the VM click the Power on button. Wait for Ubuntu to boot up and you should see the Login screen.

Deploying on Amazon AWS

To deploy the appliance on AWS, RightITnow provides an Amazon Machine Image (AMI) that can be easily deployed as an instance in EC2. To deploy the AMI:

1. Access your AWS console and switch to one of the following regions: us-east-1 (N. Virginia), eu-west-1 (Ireland) or ap-northeast-1 (Tokyo).
2. Navigate to the EC2 dashboard and click Launch Instances.
3. Select “Community AMIs” on the left-hand side filter and search for “rightitnow”. Locate the RightITnow ECM Virtual Appliance AMI and click Select.



4. Pick an instance type – we recommend an instance with at least 4 vCPUs and 16 GB of memory, such as t2.xlarge.
5. Proceed with the setup of the instance configuring it to your needs. There are no special settings required but the appliance will need to obtain an IP address and the security group needs to allow inbound requests on port 8080 to access ECM.

Deploying on Microsoft Azure

Azure does not support direct importation of OVA appliances but it can be converted into a managed disk from which a Virtual Machine can be created in Azure. To convert and upload the OVA:

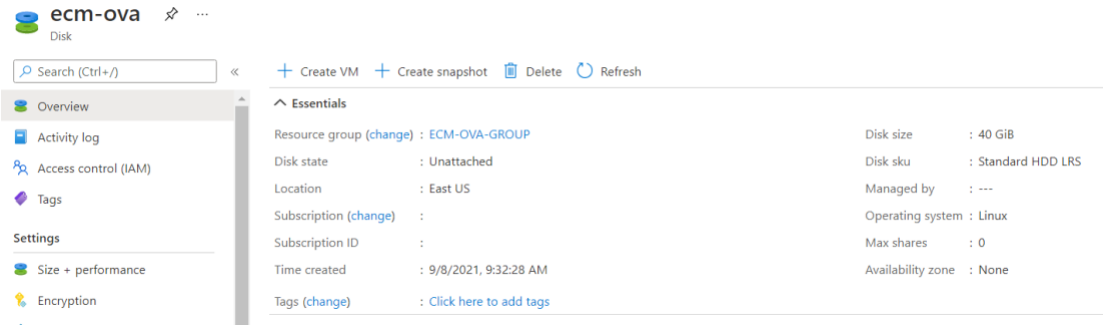
1. Extract the contents of the *rightitnow-ecm.ova* appliance using 7zip or any other extraction tool (the OVA is in tar format). One of the files extracted should be *rightitnow-ecm-disk1.vmdk* which is the appliance's virtual disk.
2. Convert the VMDK to a VHD file which is supported by Azure. This requires a tool such as VBoxManage.exe which is part of VirtualBox (the extension pack needs to be installed as well). The VHD needs to be in Fixed sized which will result in a 40GB VHD file:

```
VBoxManage.exe clonehd --format vhd --variant Fixed
C:\path\to\rightitnow-ecm-disk1.vmdk C:\path\to\rightitnow-ecm.vhd
```

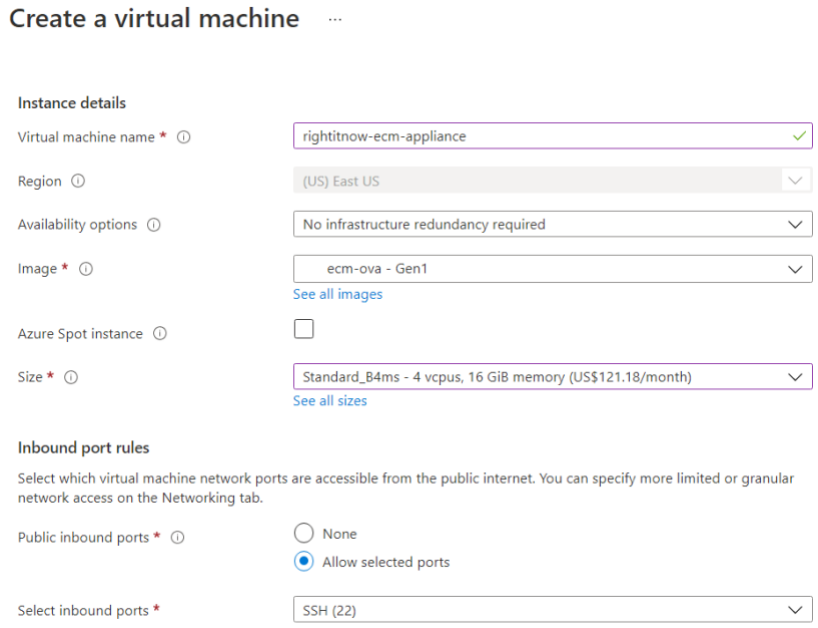
3. Note the size in bytes of the new *rightitnow-ecm.vhd* file, it will be used in the next steps.
4. Install azcopy and the Azure CLI and configure the CLI to login with the subscription under which you want to deploy the ECM appliance.
5. Use the following commands to create an empty managed disk in Azure and to upload the *rightitnow-ecm.vhd* file to it (replace the options in <> with your values):

```
az disk create -n <yourdiskname> -g <yourresourcegroupname> -l
<yourregion> --for-upload --upload-size-bytes <VHD size in bytes> --sku
standard_lrs --os-type Linux

az disk grant-access -n <yourdiskname> -g <yourresourcegroupname> --
access-level Write --duration-in-seconds 86400

AzCopy.exe copy "C:\path\to\rightitnow-ecm.vhd" "<SAS URI returned from
grant-access>" --blob-type PageBlob

az disk revoke-access -n <yourdiskname> -g <yourresourcegroupname>
```

6. Access the Azure portal and open the Disk resource that should have been created (the disk is called *ecm-ova* in the example below).



7. Click the Create VM button and enter the required details. We recommend a VM with at least 4 vCPUs and 16 GB of memory, such as B4ms.



8. Proceed with the setup of the VM configuring it to your needs. There are no special settings required but the appliance will need to obtain an IP address and the security group needs to allow inbound requests on port 8080 to access ECM.

9. Sometimes on the first bootup, the VM does not seem to be accessible over the network. Use the serial console and note any errors related to network interfaces or cloud-init. If this is the case, restart the VM and it should successfully obtain an IP address. Please see the Troubleshooting section for more information.

10. When using SQL Server on Azure if the procedure '**sp_msforeachtable**' is shown to be missing, please run the stored procedure listed here, when ECM is offline. Once the procedures are created ECM can be started up.

Accessing RightITnow ECM

If the network was successfully configured and the VM has a valid IP address, then ECM can

be accessed at http://ecm-ova:8080 or http://<ip>:8080. It may take a few minutes for ECM to start up the first time, during which the browser might appear to load indefinitely. Unless an error is reported, it is advisable to wait a few minutes before troubleshooting further.

An appliance administration tool is also configured and can be accessed at https://ecm-ova:9090 or http://<ip>:9090. This tool allows for monitoring and management of the OS and running services.

To login to the appliance from the virtual console or via SSH, use the following username and password: **ecm-admin** / **ecm-admin**. This is also the password to use when issuing *sudo* commands.

Configuring RightITnow ECM

If the ECM UI can be successfully reached, the initial setup process will be started, which is documented in the Installing and Configuring RightITnow ECM section from step 5 onwards. Please note:

- When asked for the database details in step 7, the default hostname and username can be used together with the password **ecm-admin**. This will configure ECM to use the MySQL server already installed and configured in the appliance. Alternatively, you are free to use your own database server, but make sure to configure it accordingly as described in Installing the packaged RightITnow ECM.
- When asked to submit the license file in step 9, this can be downloaded through the link sent in the email after you have registered for a trial of RightITnow ECM.

The ECM installation itself is located under */home/ecm-admin*. A systemd service called *rightitnow-ecm* is setup to control the ECM application through the *systemctl* command and to automatically start ECM on boot up:

```
sudo systemctl start|stop|restart|status rightitnow-ecm
```

If at any point a reset of the application is required, the *reset.sh* script can be executed from the terminal (via virtual console or ssh). This will wipe out all working files and redeploy the application to its original state. The database itself is not recreated, but this can be done by accessing the MySQL command line tool (the password when prompted is **ecm-admin**):

```
mysql -u root -p

drop database rightitnow;

create database rightitnow default character set utf8;

exit
```

Upgrading RightITnow ECM

To upgrade the version of ECM within an existing virtual appliance, download the packaged version of ECM (rightitnow.zip), replace the existing one under /home/ecm-admin, and run the *reset.sh* script as explained above. Navigate to the ECM web UI where the initial setup process will be started, and when the database configuration is requested, enter the details of the existing database and ECM will automatically perform the upgrade.

Configuring network settings

By default, Ubuntu is configured to use *cloud-init*, which is an industry standard multi-distribution method for cross-platform cloud instance initialization. If your virtualization or cloud platform supports this, then network settings will be obtained through *cloud-init*. If not, then a standard DHCP request is issued which should obtain the IP address and other network settings from your DHCP server.

If you need to change the DHCP settings or want to set up a static IP address, open the file */etc/netplan/00-installer-config.yaml* with vim or nano and change accordingly. For more information about configuring networks in Ubuntu see the official documentation.

To change the default hostname (ecm-ova), use the *hostnamectl* command:

```
sudo hostnamectl set-hostname newHostname
```

Then open */etc/hosts* and update any references to ecm-ova to the new hostname.

Troubleshooting

The following table suggests some steps you can take to troubleshoot the virtual appliance deployment.

Issue	Suggested Actions
Unable to assign an IP address to the VM, including static	Use the virtual or serial console and issue the command “ip a” to view all interfaces and their IPs. If an interface is marked as down, note the name of interface e.g. eth0 and modify */etc/netplan/00-installer-config.yaml* with vim or nano so that the name of the interface matches, and reboot the appliance. If the above does not fix the issue, this needs to be addressed depending on the virtualization platform the OVA is deployed on, such as by changing the type of the network interface and installing the correct drivers in Ubuntu.
DHCP related issues	Use the virtual or serial console and issue the command “ip a” to view all interfaces and their IPs. Note the name of the interface e.g. eth0 and issue the following command to manually request a DHCP address: `sudo dhclient eth0`

Issue	Suggested Actions

	If not using cloud-init, we recommend disabling and remove it from Ubuntu, this tends to fix DHCP related problems:
	
	`sudo touch /etc/cloud/cloud-init.disabled`
	`sudo dpkg-reconfigure cloud-init`
	`sudo apt-get purge cloud-init`
	`sudo rm -rf /etc/cloud/`
	`sudo rm -rf /var/lib/cloud/`
	`sudo reboot`

reset.sh throws "PID file found but either no matching process was found or the current user does not have permission to stop the process. Stop aborted."	Manually delete the */home/ecm-admin/rightitnow/catalina.pid* file and try again.
"Setup was interrupted.." error message on the ECM web UI	Manually restart the ECM service with *sudo systemctl restart rightitnow-ecm*
Other errors relating to ECM or ECM not starting up correctly	View the logs under */home/ecm-admin/rightitnow/logs* and contact RightITnow support

Deploying the RightITnow ECM Docker Container

RightITnow provides a Docker container which is a standard and quick way to deploy on cloud or on-premise. The image contains RightITnow ECM and can be easily downloaded from the DockerHub registry. This reduces the time and effort required to deploy ECM and is especially useful to try out our product or for setting up development and UAT servers. ECM requires the MySQL database which can be downloaded from the registry as well.

Installing ECM via Docker

Docker is required to download images and run containers. Make sure you have it installed and running on the machine where you are planning to run ECM. The following instructions for Docker apply to version 20.10.7 or later.

1. Download ECM image from DockerHub and start a new container:

```
docker run -t -p 8080:8080 rightitnow/ecm:<tag>
```

2. Download MySQL database from DockerHub and start a new container:

```
docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d
mysql:8
```

3. Find the docker ID and connect to the MySQL server:

```
docker ps

docker exec -it <docker_id> bash

mysql -u root -p
```

4. Create the database:

```
create database rightitnow default character set utf8;

SET GLOBAL sql_mode =
'STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION'
;

SET SESSION sql_mode =
'STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION'
;
```

5. Open ECM in your browser by accessing http://<SERVER IP>:8080/rightitnow. If the ECM UI can be successfully reached, the initial setup process will be started, which is documented in the Installing and Configuring RightITnow ECM section from step 5 onwards.
6. Commit the changes into a new image and note the ID of the new image:

```
docker commit <docker_id> rightitnow/ecm

docker image ls -a
```

7. Stop the current container and run the new image generated:

```
docker stop <docker_id>

docker run -t -p 8080:8080 <new_docker_id>
```

Upgrading ECM via Docker

To upgrade the Docker image follow these instructions:

1. Download ECM image from DockerHub and start a new container:

```
docker run -t -p 8080:8080 rightitnow/ecm:<tag>
```

2. Login to the new container started:

```
docker ps

docker exec -it <docker_id> bash
```

3. Restore any custom settings from the previous installation:

```
/usr/local/tomcat/webapps/rightitnow/WEB-INF/db.properties contains
Database settings

/usr/local/tomcat/webapps/rightitnow/bin/catalina.sh contains Java
memory settings

/usr/local/tomcat/webapps/rightitnow/conf/server.xml contains ports and
SSL settings
```

4. Commit the changes into a new image and note the ID of the new image:

```
docker commit <docker_id> rightitnow/ecm

docker image ls -a
```

5. Stop the container and login to the new image generated:

```
docker stop <docker_id>

docker run -t -p 8080:8080 <new_docker_id>
```

Chapter 2. Setting up Distributed Workers

Overview

Distributed workers are additional ECM nodes deployed on your network. All nodes form a cluster that allows ECM to process a larger volume of events, increases high availability and fault tolerance, and allows for limited read-only access to the ECM UI. The cluster uses the Hazelcast In-Memory Data Grid to share data and synchronize tasks across the ECM nodes.

Prerequisites

The hardware and software requirements for a distributed worker are the same as the Master ECM server, which are outlined above. Additionally, the following items need to be checked to make sure that the ECM cluster works as expected:

- The Master ECM node should already be installed and running before setting up additional worker nodes. Worker nodes should be installed on separate servers and should not share the same environment as the Master.
- Worker nodes need to have access to the same database that the Master node has access to, which could either be running on the same server as the Master ECM or on its own server. Low latencies between nodes and the database is essential for good performance.
- All nodes need to be able to communicate together via TCP/IP on a specified port (default 8081). Please ensure that this port is available and that you add it to your firewall. Note that this port is different from the Tomcat port (default 8080) which allows web access to ECM.
- All nodes should be on the same time zone and synchronized with each other or with a trusted pool via NTP eg. https://www.ntppool.org/en/use.html. This is important as it ensures that periodic tasks run by ECM execute correctly across all nodes, and for alerts to have consistent timestamps.
  - The time zone can also be set just for ECM instead of the whole server, by changing the *rightitnow/bin/catalina.sh* file and adding the following to *CATALINA_OPTS*: **-Duser.timezone=Timezone** where the time zone matches the one on the Master ECM node. For the whole list see https://www.iana.org/time-zones e.g. **-Duser.timezone=Europe/London**
  - To handle events happening in different time zones, the actual event timestamp can be set by using the *time* token in the event (if sending via SOAP, REST or

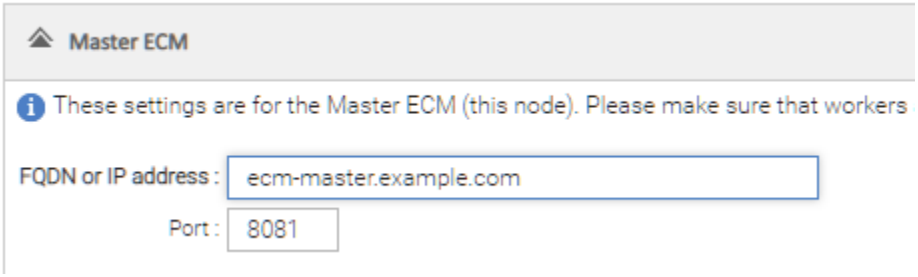
ActiveMQ), while the user preferences let users pick the time zone they want to see alerts in.

- Although optional, externalizing the JMS queues that ECM uses for event and alert processing can be beneficial if running additional worker nodes, since all nodes can share the queues and event/alert processing becomes automatically load balanced. If this is not done, then some sort of load balancing needs to be implemented by the sender when sending events to ECM via SOAP or REST, to make use of all nodes rather than just one. For further information please access the "Setup external queues" documentation from the Getting Started displet on the Dashboard.

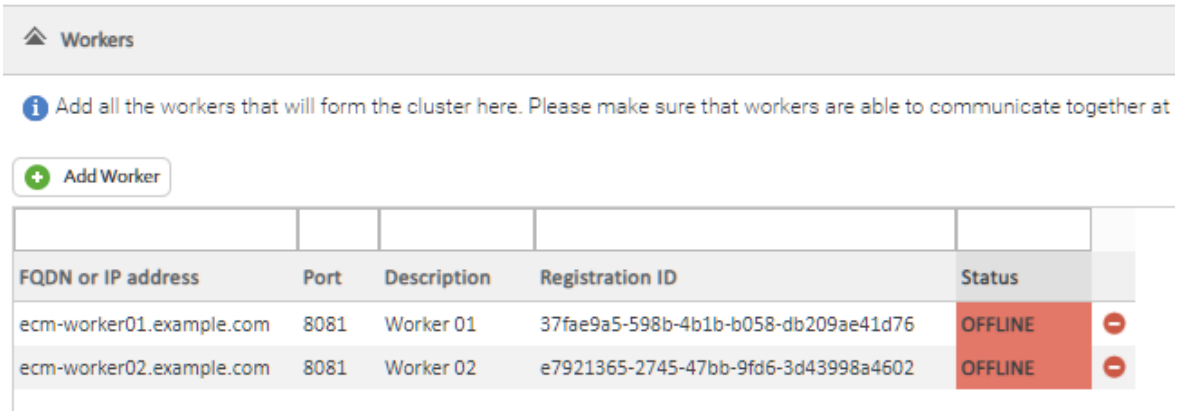
Configuring Workers

If all prerequisites mentioned above are satisfied, log on to the Master ECM as an Administrator and proceed with the following steps:

1. Open the Configuration → Distributed Workers tab.
2. In the Master ECM section enter the FQDN or IP address of the Master ECM and change the port if desired.

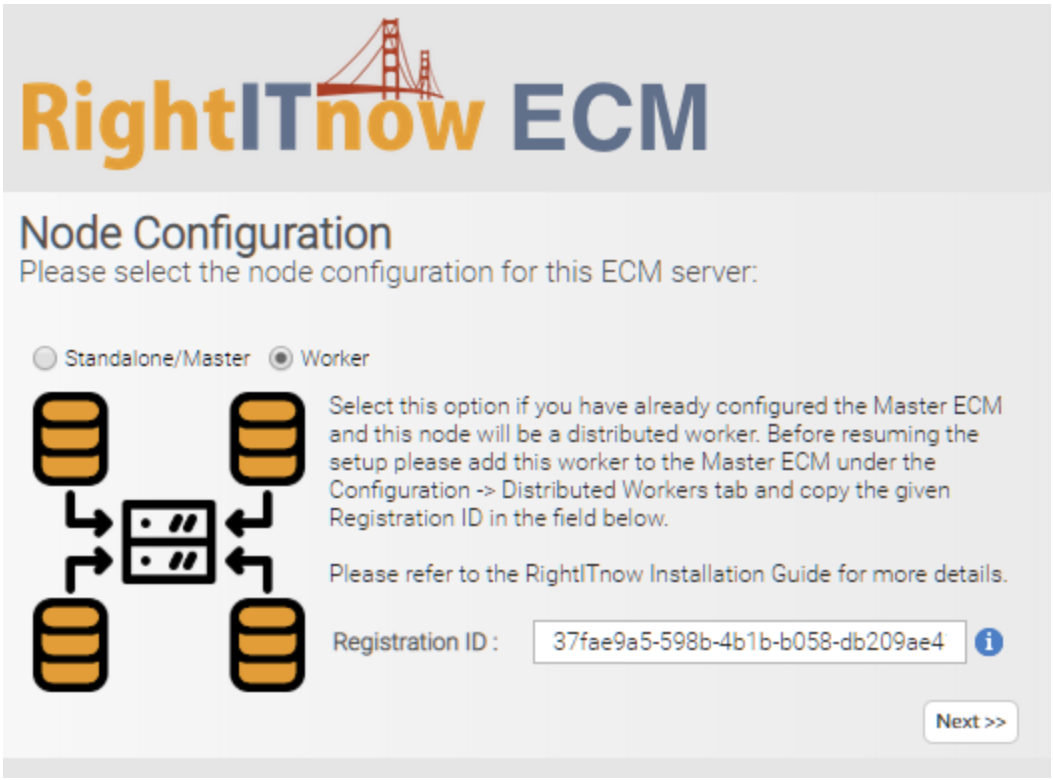


3. In the Workers section add the required workers using the Add Worker button and click Save. Each worker should have a registration ID created for it automatically. Note that the Status will say OFFLINE since the worker ECMs have not been started yet.



4. For each worker created, setup the worker ECM node on its server by following the Installing and Configuring RightITnow ECM section. Note that Step 3 (database creation) is not required since the worker will use the same database as the Master ECM. In Step 6 (Node Configuration) select Worker and enter the registration ID for the

corresponding node that was generated on the Master ECM, as shown in the example below.



5. Make sure to enter the same database details of the Master ECM node in Step 7, otherwise the setup will not continue since the registration ID of the worker will not be found.
6. If the setup is successful, the worker ECM will prompt for login. Use the same credentials as you would on the Master ECM, however, note that the worker ECM UI is restricted to read-only access and only certain tabs are available.
7. If the cluster seems to be working correctly, it is recommended to do a clean startup to verify the setup, by shutting down all the workers and the Master, then starting up the Master ECM followed by the workers.
8. If an existing worker's address or port needs to be changed, do the following:
    a. Shut down the worker ECM and remove (or backup) the installation files.
    b. Apply any changes, such as adding the new port to the firewall or changing the hostname, to the server.
    c. Remove the existing worker from the Master ECM and re-add it with the new settings.
    d. Setup the worker again following the steps above. Since the worker registration ID has changed, the worker ECM needs to be installed from scratch.

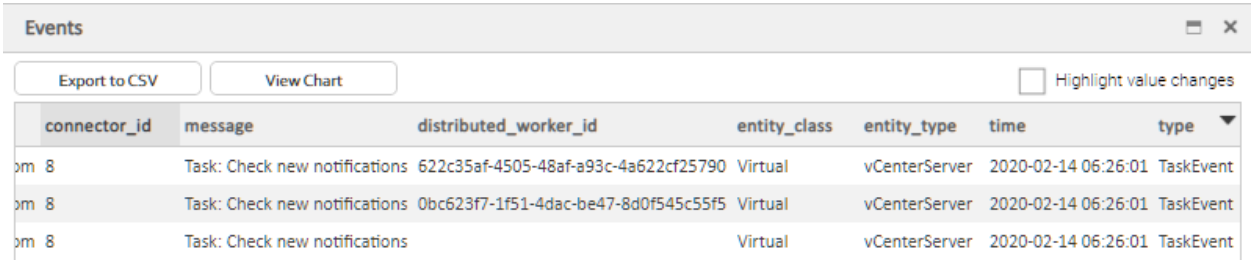
Troubleshooting and Limitations

In the Distributed Workers tab, the worker and cluster status should be “ONLINE”, and no errors should be present in the Hazelcast log, otherwise it can indicate that the cluster is not functioning correctly. All of the online members of the cluster will be assigned a unique ID as shown below (this ID is different from the registration ID and can change if the worker is restarted):

```
Members {size:3, ver:3} [
    Member [ecm-master.example.com]:8081 - 06bd5fca-9bdc-40d3-892b-c59942376f94 this
    Member [ecm-worker01.example.com]:8081 - e69ea390-cfb0-4040-938e-95ec29b55bd3
    Member [ecm-worker02.example.com]:8081 - 196030b5-9692-4f3d-99be-0e885ac94152
]
```

The following can be done to solve certain issues, especially if a network outage occurred recently which in some cases can cause the cluster to split into two:

- Shut down each worker and then restart the Master ECM node. When the node is fully online, start the workers in turn.
- If only the Master ECM fails, the workers can continue to process events and alerts, and the Master ECM can be started later, and it will re-join the cluster. However, if all the nodes fail, the Master ECM should always be the first one to be started, followed by the workers.
- Access to the worker ECM UI is limited to the following tabs, all of which are read-only:
    - Dashboard and Alerts Console
    - Manage Connectors, Distributed Workers and ECM Server Monitor
    - Audit Log and SLA Breach Log
    - The REST API can be used for read-only purposes eg. fetching alerts from a worker
- Events and alerts processed by a worker have a *distributed_worker* token added to the event with the registration ID of the worker, as shown below. This can help in troubleshooting issues when processing events and alerts on a specific node and can also be used in rules to target alerts coming from different workers.

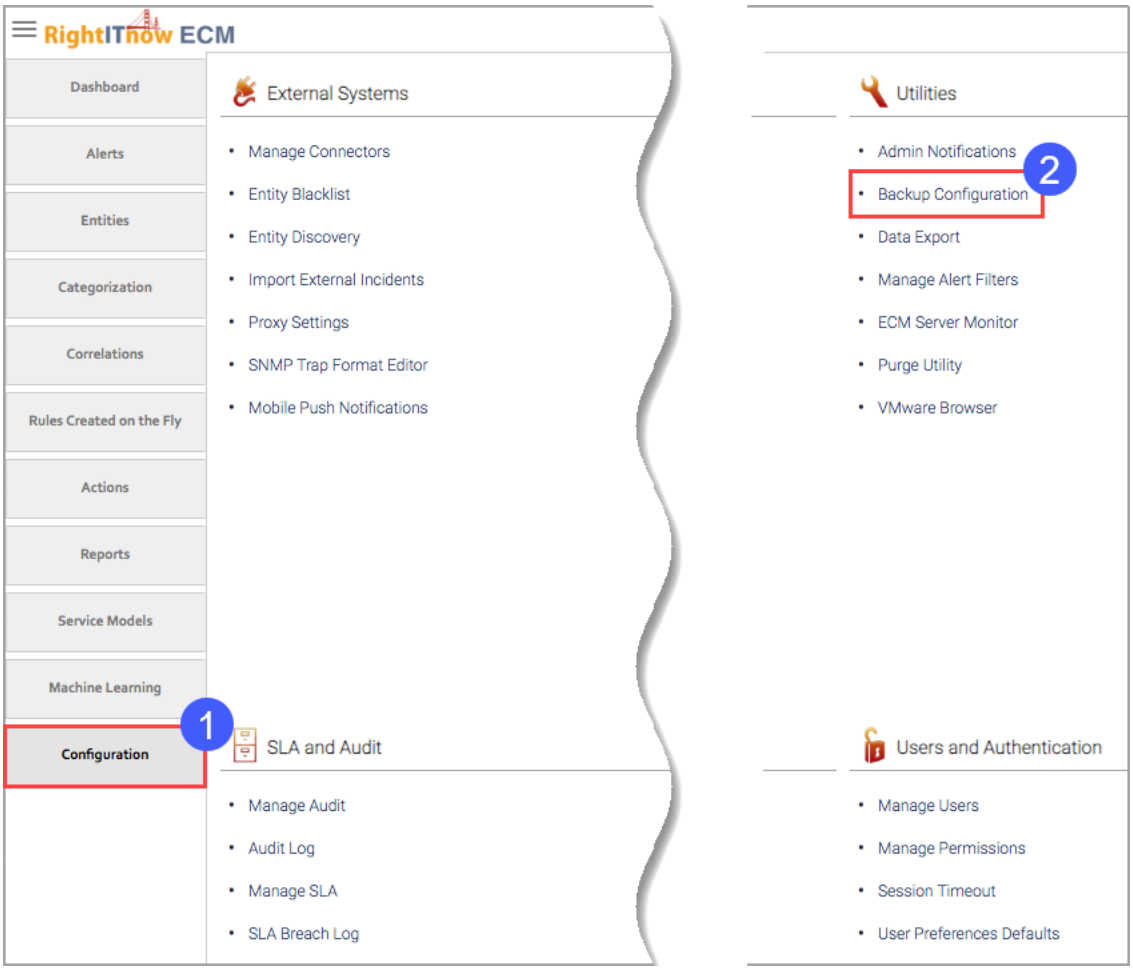


Chapter 3. Backup, Restoring and Troubleshooting

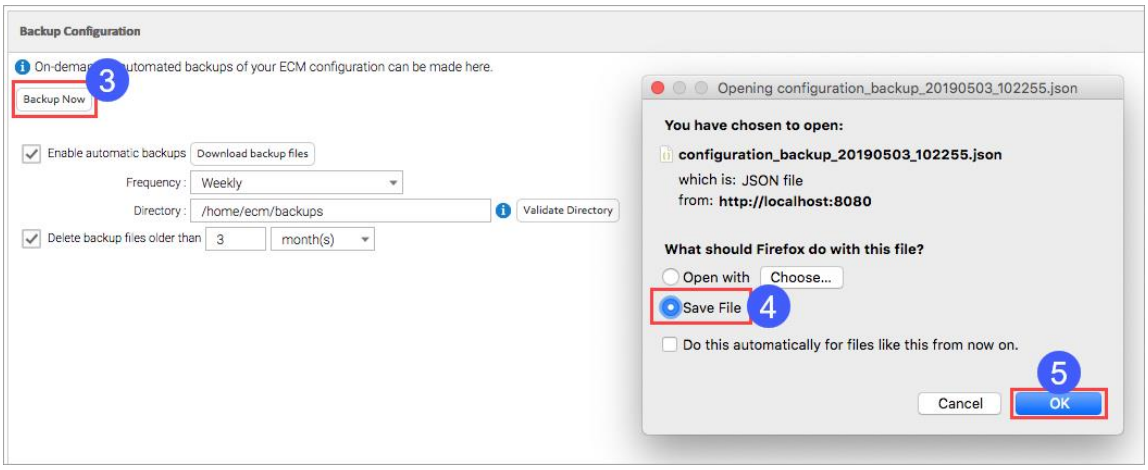
Backing up the configuration

To backup the RightITnow ECM configuration:

9. Click the **Configuration** tab.



10. Click **Backup Configuration**.



11. Click **Backup Now**.

    A dialog box appears prompting you to save the backup configuration file. You can also set up automated backups.

12. Click **Save** to save the configuration file.

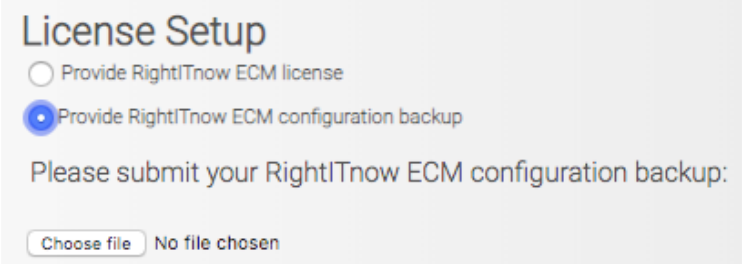
13. Click **OK**.

    The system backs up the following configuration components:

    - Categorisation Rules
    - Correlation Rules
    - Entity Groups Definitions
    - Actions and Action Groups
    - Application Settings (Workflow, Severities Labels, Correlation By Example, Alert Context Menu, Authentication Method, Grid and displet preferences)
    - Users
    - Alert Console's Filters
    - Connectors
    - Users
    - License information

Restoring the configuration

You can only restore the configuration to a new empty database. The option to restore the ECM configuration will be presented after the user has created and pointed to the database:



Once you have selected the configuration backup file, the system restores the configuration components automatically and you can log back in to RightITnow ECM.

NOTE: SolarWinds connectors will be undeployed. You need to validate the connection to get the certificate before deploying them again.

NOTE: If restoring a configuration that was a Master node (i.e. it had a FQDN or IP address of itself and the workers configured in the Distributed Workers configuration), make sure to either shut down the original node, or to isolate the restored node on the network, otherwise it will attempt to join the network of nodes and might result in a split network with two Master nodes.

Troubleshooting

The following table suggests some steps you can take to troubleshoot your RightITnow ECM installation.

Issue	Suggested Actions
LDAP	
Having difficulty logging in your LDAP store?	Use the **ldapsearch** utility to identify the origin of the issue For example, the following command validates a connection for jack mango: `ldapsearch -h 192.168.12.12 -p 389 -b "cn=Users,dc=rivertest,dc=co,dc=uk" -D "admin" -w "4dm1nPasswd" -x "(&(objectCategory=person)(cn=Jack mango))"`

Logging

ECM writes logs to the rightitnow/logs directory. The **rightitnow.log** should always be checked when troubleshooting issues and can also be viewed directly in the application by going to Configuration → ECM Server Monitor → System Log. The System Log Configuration button can be used to configure logging settings.

Troubleshooting Kerberos

Windows authentication using Kerberos requires precise configuration and often proves difficult to get working correctly. The following section attempts to describe some common issues and how to fix them.

Supported Encryption Types

Different versions of Windows support different encryption types and you must ensure that settings described in this document conform to the encryption types supported by your environment. A detailed look at how to change supported encryption types in Windows is given here.

This document assumes that **rc4-hmac** is the encryption type to be used as it is the most compatible with many versions of Windows. The following configuration items need attention if you change the supported encryption types in your environment:

- The **keytab** file needs to include the relevant keys for every encryption type supported, this is controlled with the */crypto* parameter.
- The encryption settings in **krb5.conf** should be changed to reflect the desired types.
- If AES256 is used, then the **Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files** should be installed on the ECM server https://www.oracle.com/java/technologies/javase-jce8-downloads.html

Domain Name Resolution

It is important that DNS is working properly in your environment. The following should be checked:

- The domain name used in the SPN should be a DNS A record pointing to the server and not a CNAME record.
- A domain using a CNAME record can be used to access ECM if it points to the domain used in the SPN. This is useful to setup vanity domains.
- Reverse lookup needs to work correctly for all domains in question.
- Avoid using IPs when accessing ECM and always use the domain name.

Logging

You can enable extra logging by adding the following line to the JAVA_OPTS environment variable in catalina.sh or catalina.bat:

```
-Dsun.security.krb5.debug=true -Dsun.security.jgss.debug=true
```

Chapter 4.
Event Processing and Performance

Internal Event Processing

ECM makes use of internal ActiveMQ queues when processing events: one queue for events and one queue for alerts. An event coming into the system will typically go through both queues, with the categorization process retrieving events from the events queue, processing them and placing them on the alerts queue, and then the correlation modules pick up the alerts from the alerts queue, process them and save them in the database.

The purpose of these queues, apart from conforming to standard JMS development practices, is to be able to handle event floods within ECM, so that no events are dropped or lost. During normal operation, these queues should be empty since ECM should be processing events immediately without any delays. If the queues start to grow, then there will be a delay between the time an event enters the system and the time it is fully processed and visible on the alerts console. The queues will typically hold a maximum of between 10,000 to 20,000 items, depending on the event/alert size. If this limit is reached, then ECM cannot queue any more events and will have to drop them.

It is possible for users of ECM to create and configure the event and alert queues on their own infrastructure and then configure ECM to use these queues instead of the internal ones. This has several benefits such as increased reliability, improved HA/failover and better flexibility, since it allows users to allocate more disk space and memory for the queues and back up the queues to disk. For further information please access the “Setup external queues” documentation from the Getting Started displet on the Dashboard.

Event Rates

The event rate is the rate at which ECM is able to process events into alerts **without** any delay i.e. the event is processed and shows up immediately on the alerts console. This rate varies mostly on the configuration of ECM itself (number of correlation rules and the actions that they perform, alert workflow settings, purging schedule etc), the size of the database, and also on the capability of the underlying hardware. A typical server with standard specifications will achieve a rate of at least 30 events/sec (equivalent to 1,800/minute = 108,000/hour = 2.6 million/day).

Recommended Settings

The following sections discuss recommended settings for RighITnow ECM supporting components.

Tomcat/Apache

The ECM application is bundled within a Tomcat container, and as such many of the environment settings are standard Tomcat/Apache settings. The following are some settings that typically require tuning or configuration:

System memory: The amount of system memory available to the ECM application is dictated by the standard JVM memory settings. ECM ships with a default of 2GB (restricted up to 8GB) allocated for heap memory. For large installations, the heap setting should be bumped up to use 16GB of memory (or more if required). This can be changed under rightitnow/bin/catalina.sh or catalina.bat. The default setting is as follows:

```
CATALINA_OPTS="$CATALINA_OPTS -Xms2g -Xmx8g -
XX:+HeapDumpOnOutOfMemoryError -XX:+UseG1GC -
Dorg.apache.activemq.SERIALIZABLE_PACKAGES=*"
```