

## Quick Guide: Setting Up the SSHResourceManager

This quick guide shows you how to setup the SSHResourceManager (RSM). Please see also the admin guide for further information which can be found in the installation directory:

`<CAESESInstallDir>/tools/sshResourceManager/doc/adminguide.pdf`

SSH Configuration Manager

[Operating Systems](#) | [Applications](#) | [Floating Licenses](#) | [Hosts](#) | [Users](#) | **[State](#)** | [Administration](#) | [Logout](#)

State of SshResourceManager at <http://192.168.25.14:8080>

[clean up](#)

Statistics

Uptime	32 minutes, 50 seconds
Started jobs	0
Finished jobs	0
Failed jobs	0
Rescheduled jobs	0
Hosts	1
Applications	1
Users Total	0
Super Users	0
Users	0
Locked Users	0

Scheduler State

Global

Jobs: 0
Limit:

Running Jobs: 0

Job Id	Task	Submitter	Owner	Start time	Application	Execution host	State
0							

Waiting Jobs: 0

Job Id	Submitter	Owner	Submit time	Application	Needs auth	State
0						

Finished Jobs: 0

Job Id	Task	Submitter	Owner	Finish time	Application	Execution host	State
0							

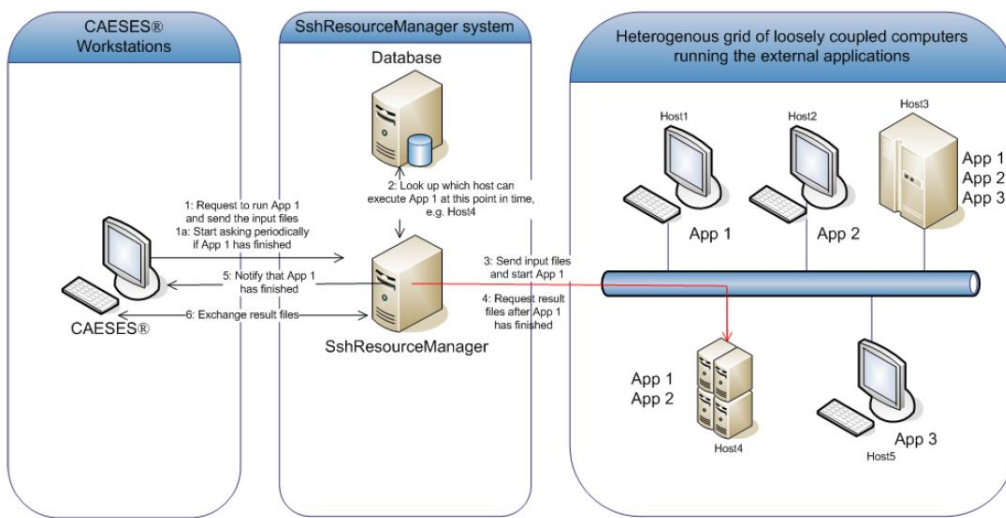
Hosts: 1

Name	CPUs				Active	Known host	Has auth	# Apps
	total	free	used	locked				
localhost	6	6	0	0	true	true	true	0

1

### Location of the RSM

The first decision you have to take is where you want to install the RSM. In general this should be a computer which is always running and is accessible by the local workstations, where CAESES is installed. It needs to have a connection to the host where the actual computation should be triggered. In most cases these computers are Linux servers or Linux clusters. However, it could also be a Windows server.



An additional **important** prerequisite is that the RSM can connect to a **database**.

## 2

## Creating a Database

First, we need to set up a database, which is used to store all necessary information. In most cases this will be MySQL:

- ▶ Install the database server on a computer that is reachable from the computer and that is supposed to run the RSM (can be the same computer). You can download MySQL here: <https://dev.mysql.com/downloads/mysql/>
- ▶ Create a database, a user and assign the user to the database, as shown in the next steps:
- ▶ Log into MySQL: -> `mysql -u root -p`
- ▶ Create a database: -> `create database rsm;`
- ▶ Create a user: -> `create user rsmuser;`
- ▶ Assign user to database: -> `grant all on rsm.* to 'rsmuser'@'localhost' identified by 'pw';`

```
MariaDB [(none)]> create database rsm;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]> create user rsmuser;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on rsm.* to 'rsmuser'@'localhost' identified by 'pw';
Query OK, 0 rows affected (0.00 sec)
```

✓ The OS of our test computer is Debian (Linux). On this system MariaDB is installed by default, which is a fork of MySQL.

## 3

**RSM and JDBC**

In this step, we extract the RSM and add the JDBC Driver:

- ▶ Create a folder on your system, where you want to run the RSM as a user  
-> `mkdir ~/rsm`
- ▶ Copy the RSM zip archive to this folder and extract it  
-> `cp CAESESInstallDir/tools/sshResourceanager/sshresourceanager.zip ~/rsm`  
-> `unzip ~/rsm/sshresourceanager.zip`

***Adding the JDBC Driver***

Now we need a driver which is used for the communication of the RSM with the database. This driver is called JDBC driver.

- ▶ Download a JDBC driver for the database system:  
<https://dev.mysql.com/downloads/connector/j/>
- ▶ Extract the downloaded package and search for the .jar file, e.g.  
`~/Downloads/usr/share/java/mysql-connector-java-8.0.11.jar`  
if you extracted the JDBC Connector inside the downloads folder.
- ▶ Copy the .jar file to the “dbdriver” subdirectory of the folder “rsm”.

## 4

### Edit the Database Settings

Now we have to set the properties of the RSM to make sure that the RSM, the database and the JDBC driver can communicate with each other.

- ▶ Open the RSM configuration file “SshResourceManager.properties” located inside the RSM installation directory.
- ▶ Find the chapter “Database Settings”.
- ▶ Set the JDBC Driver: **databaseJDBCdriver=com.mysql.jdbc.Driver**  
(it may change to “**com.mysql.cj.jdbc.Driver**” or if you use mariaDB “**org.mariadb.jdbc.Driver**”)
- ▶ Set the dialect which works in most cases:  
**databaseDialect=org.hibernate.dialect.MYSQLDialect**
- ▶ Set the URL: **databaseConnectionURL=jdbc:mysql://localhost/rsm**  
(in case you use mariaDB it is **jdbc:mariadb://localhost/rsm**)
- ▶ Set the database user: **databaseUser=rsmuser**
- ▶ Set the user’s password: **databasePassword=pw**
- ▶ Start the RSM and test the connection on port 8080: `java -jar sshresourcemanager.jar 8080`

Here is an output message of a successful start of the RSM:

```

2018-05-03 10:48:41,265 DEBUG [localhost-startStop-1] FScheduler: Setting scheduler priority to "priority"
2018-05-03 10:48:41,269 INFO [localhost-startStop-1] FSSHrmStateInitializer: X forwarding is defaulted to: localhost:6000
2018-05-03 10:48:41,269 INFO [localhost-startStop-1] FSSHrmStateInitializer: Setting file transfer timeout to : 60000
2018-05-03 10:48:41,269 INFO [localhost-startStop-1] FSSHrmStateInitializer: Output buffer is set to: 10MB
2018-05-03 10:48:41,270 INFO [localhost-startStop-1] FSSHrmStateInitializer: Result file server started on port 39943
2018-05-03 10:48:41,270 INFO [localhost-startStop-1] FSSHrmStateInitializer: Input file server started on port 42797
2018-05-03 10:48:41,270 INFO [localhost-startStop-1] FSSHrmStateInitializer: Users will NOT be able to execute multiple
|| or '!'
2018-05-03 10:48:41,271 INFO [localhost-startStop-1] FSSHrmStateInitializer: Using command line "echo HELLO" for configu
2018-05-03 10:48:41,274 INFO [localhost-startStop-1] FSSHrmStateInitializer: No LDAP authentication configured
2018-05-03 10:48:41,278 INFO [localhost-startStop-1] FSSHrmState: SshResourceManager initialization complet
2018-05-03 10:48:41,316 INFO [localhost-startStop-1] FSSHrmState: Database was resetted
2018-05-03 10:48:41,344 INFO [localhost-startStop-1] FSSHrmState: Startup FSSHrmState version 4.1
May 03, 2018 10:48:41 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
May 03, 2018 10:48:41 AM com.friendshipsystems.FSSHrmState start
INFO: Server started on port 8080
May 03, 2018 10:48:41 AM com.friendshipsystems.FSSHrmState$3 run
INFO: Started discovery service on port 49152

```

✓ You can find a list of supported dialects here:

<http://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/dialect/package-summary.html>

✓ Errors can occur, when the database settings do not correspond to the created database. Also there are incompatibilities of certain database server with certain JDBC driver. When you use MariaDB database then make sure to use the latest version, to avoid conflicts with the JDBC driver.

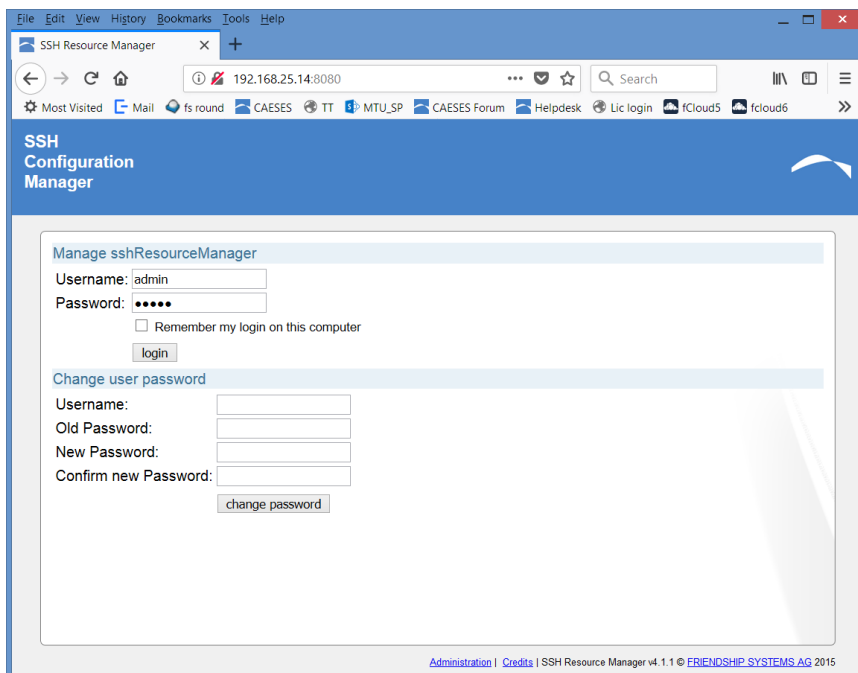
# 5

## RSM Configuration

If the previous steps have been successful, you should now be able to log in to the RSM browser interface, to set up your host and applications.

✓ Remember that you can log into the RSM from any computer in the network

- ▶ Open a web browser and navigate to `http://<IPAddress or name of the computer running SshRM>:8080/`
- ▶ Now log in to the RSM itself. In the menu "Manage sshResourceManager" enter "admin" (without quotes) as *Username* and an arbitrary password. There is no need to remember this password, as long as no users are configured.



In the following steps, the network settings and the applications are configured.

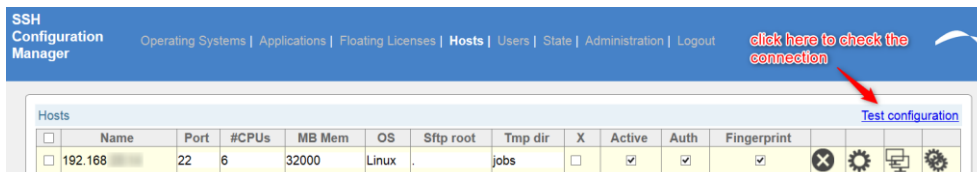
- ▶ Go to the "Applications" page. This is what will be seen by the CAESES when it connects to the RSM. Let's enter a new application for testing purposes only

**Name:** date





Press the "+" sign

- ▶ Go to the "Hosts" page. These are the computers the RSM will connect to via SSH. If you installed the RSM on the head node, add the following host:
  - Name:** localhost (otherwise enter the IP address)
  - Port:** 22 should be ok, except when you have set up the SSH server to listen on a different port.
  - CPUs:** Normally the RSM uses this information to determine which host has free resources. So, for every instance of an application it starts, it will subtract the number of required CPUs of the application (see Application page) from the number of available ones on a certain host and will only select the host for execution if the number is larger than 0.
  - MB Mem:** Currently unused, just set anything.
  - OS:** Linux/Windows
  - Sftp Root:** "." (no quotes) This is used for temporary files that are created on the host. It should be the initial directory of a user when connecting via SFTP. For Linux "." means that the user's home directory is used.
  - Tmp Dir:** This is the actual directory where the temporary data is stored relative to the Sftp Root. It will be created if it doesn't exist. e.g. "jobs" (no quotes)
  - Active:** true

Press the "+" sign
- ▶ A new host is created. Press the "Fingerprint" checkbox and confirm the RSA fingerprint (if it matches the expected value). This is a security feature to avoid man-in-the-middle attacks and is similar to the mechanism used by "regular" SSH clients.
- ▶ There are two ways to specify the log in data for SSH for this host: 1. inside CAESES (see later) or 2. on this page. If you set them on this page, you will not have to set them inside CAESES. To do so, press the "**Auth**" checkbox. Enter the username and password and check the "Check login" checkbox. If the login with the given credentials succeeds, it will say so, otherwise, you can try again. The second option should be preferred.
- ▶ Now check the connection to the new host:



The screenshot shows the 'SSH Configuration Manager' interface. At the top, there is a navigation bar with links for 'Operating Systems', 'Applications', 'Floating Licenses', 'Hosts', 'Users', 'State', 'Administration', and 'Logout'. A red arrow points to a link that says 'click here to check the connection'. Below this is a table with the following columns: Name, Port, #CPUs, MB Mem, OS, Sftp root, Tmp dir, X, Active, Auth, Fingerprint, and a set of icons. The table contains one row with the following data: Name: 192.168..., Port: 22, #CPUs: 6, MB Mem: 32000, OS: Linux, Sftp root: ., Tmp dir: jobs, X: , Active: , Auth: , Fingerprint: . The icons include a close button (X), a refresh button, a print button, and a settings button.

	Name	Port	#CPUs	MB Mem	OS	Sftp root	Tmp dir	X	Active	Auth	Fingerprint	
<input type="checkbox"/>	192.168...	22	6	32000	Linux	.	jobs	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>    

SSH username

SSH password

Close  click here to run the test

Host	Start time	Finish time	Result
192.168	2018-05-03 15:54:54,231	2018-05-03 15:54:55,398	SUCCESS

- ▶ Press the single gear at the end of the line to add an application to the host.

SSH Configuration Manager Operating Systems | Applications | Floating Licenses | **Hosts** | Users | State | Administration | Logout click here to add an application to the host

Hosts Test configuration

<input type="checkbox"/>	Name	Port	#CPUs	MB Mem	OS	Sftp root	Tmp dir	X	Active	Auth	Fingerprint	
<input type="checkbox"/>	192.168	22	6	32000	Linux	.	jobs	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

- ▶ Select the application (e.g. "date") from the drop-down menu. Then set the execution path, which is simply "date" on Linux. Press the "+" sign.

SSH Configuration Manager Operating Systems | Applications | Floating Licenses | **Hosts** | Users | State | Administration | Logout

Applications of host localhost(CPU: 6, memory: 32000, OS: Linux, X-support: false) back Test configuration

Name	Description	Req lics	Req Cpus	Needs X	Priority	Hard lics	Exe path
Add new application to localhost							
date		0	0	false	0	0	date <input style="float: right;" type="button" value="+"/>

- ▶ Create a new user and set the role as "user" and set a password.

SSH Configuration Manager Operating Systems | Applications | Floating Licenses | Hosts | **Users** | State | Administration | Logout

Users

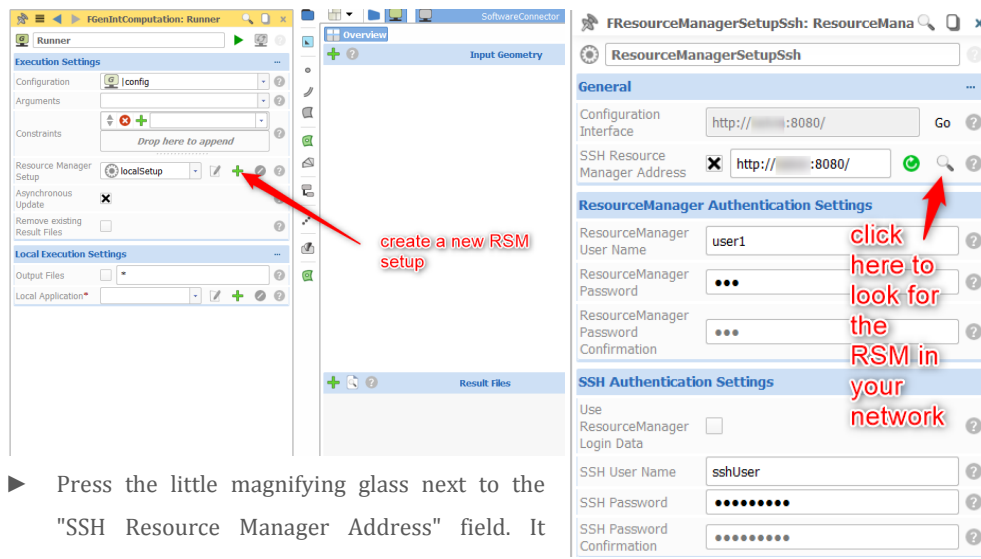
Username	Role	Email	Password
user1	user	[none]	<input type="password"/> <input checked="" type="checkbox"/>
Add new user			
[edit]	lockedUser	[none]	<input type="password"/> <input style="float: right;" type="button" value="+"/>

6

### Using the RSM

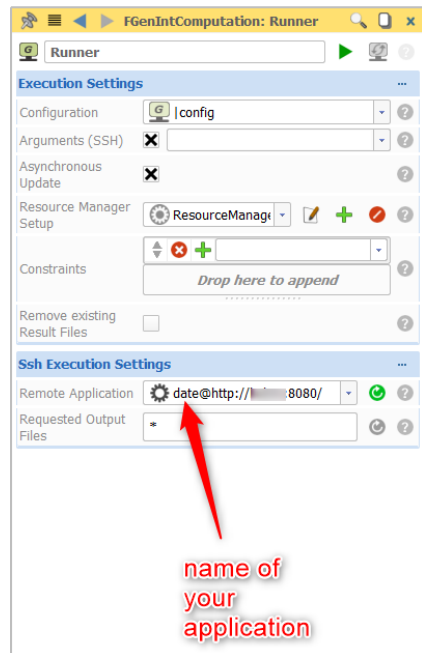
Now that we have the RSM up and running as well as an existing application, we can access the RSM from CAESES on your local machine.

- ▶ Start CAESES on your local machine
- ▶ Create a new *Software Connector* and select the *Runner*
- ▶ In the "Object Editor" see the "Execution Settings" and press the green "+" button next to the "Resource Manager Setup" combo box.

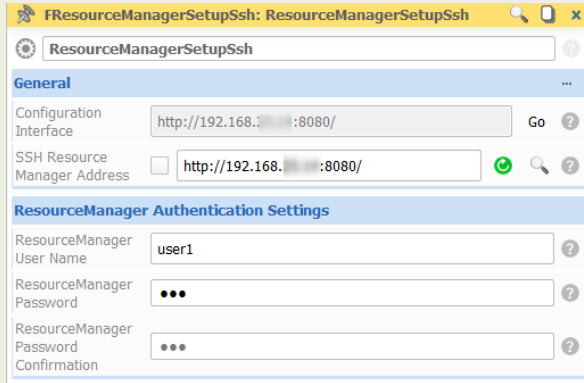


- ▶ Press the little magnifying glass next to the "SSH Resource Manager Address" field. It should show your running RSM. Select it and press "Ok". If it doesn't show the RSM, enter the address manually.
- ▶ Enter your RSM user data.
- ▶ Enter your SSH user account data.
- ▶ Click on the green refresh button to load your applications.

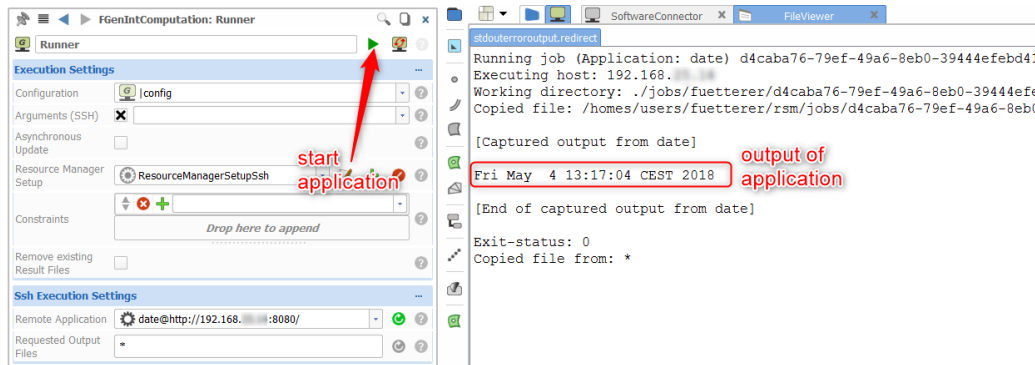
Now you should be able to select your application.



✓ If you are having trouble to connect to the RSM and retrieve the application list, then disable “DNS resolving” and enter the IP address of the RSM.



► Start your application by pressing the run button, just to see if it works:



Now you are ready to set up your application (e.g. CFD tools). Go back to the RSM web interface and add a new application. Add it to the Host as you have done with the date testing application (which, of course, can be removed now). If you now refresh the list of applications inside CAESES, you should see the new application and you should be able to run it.