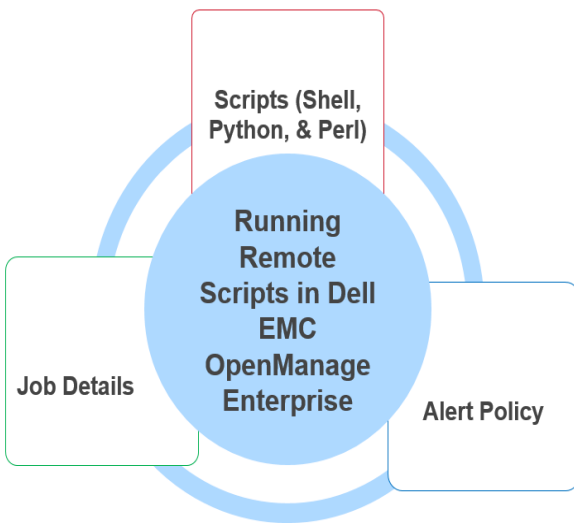


# Remote script execution with Dell EMC OpenManage Enterprise



## Abstract

This technical white paper describes the Remote Script Execution feature of OpenManage Enterprise. It gives an overview of the feature along with a use case to enable and use the feature as a System Administrator.

July 2019

## Revisions

Date	Description
July 2019	Initial release

## Acknowledgements

This paper was produced by the following members of the Dell EMC Enterprise Systems Management Programs:

Author: Subhakant (Test Engineer), Soumya Aggarwal (Test Engineer), and Nagaraj K (Test Engineer)

The information in this publication is provided "as is." Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any software described in this publication requires an applicable software license.

Copyright © July 2019 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

# Table of contents

Revisions.....	2
Acknowledgements.....	2
Table of contents .....	3
Executive summary.....	4
1 Some use cases for the Remote Execution feature.....	5
2 Basic details of Remote Script Execution .....	6
3 Configure, Edit, and Delete scripts for remote execution.....	7
3.1 Configure Remote Script Execution with token substitution.....	7
3.2 Configure Remote Script Execution with password for remote execution. ....	9
3.3 Configure Remote Script Execution for script with SSH key .....	10
3.3.1 Configure SSH key .....	10
3.3.2 Procedure to configure Remote Script Execution for script with SSH key.....	10
3.4 Configure Remote Script Execution for RACADM / IPMI command.....	11
3.5 Edit scripts or commands used in Remote Script Execution.....	12
3.6 Delete scripts or commands used in Remote Script Execution .....	13
4 Create an alert policy and link to the remote execution scripts and commands.....	14
5 Validate remote script execution .....	19
6 Remote script execution using RESTful APIs .....	21
6.1 Create IPMI Commands from Remote Script Execution page using REST APIs.....	21
6.2 Create RACADM Commands from Remote Script Execution page using REST APIs.....	22
6.3 Create scripts with SSH key authentication using REST APIs .....	23
6.4 Create Alert policy and link with action “Remote Script” .....	24
A Technical support and resources .....	27

## Executive summary

Addressing device alerts manually, especially in large setups, is a time-consuming and laborious exercise for the system administrators. OpenManage Enterprise with the Remote Script Execution feature is a solution to this concern.

The Remote Script Execution feature enables the system administrators to run up to 100 scripts, RACADM, or IPMI commands remotely in response to alerts received such as change in device health, power status, and connectivity status.

# 1 Some use cases for the Remote Script Execution feature

Listed below are some of the use cases where the Remote Script Execution feature can be used:

## **USE CASE 1:**

Remote Script Execution can be used to prevent over heating of servers. When temperature reaches a critical state in a server, alert is generated. In such instances, the user can create an alert policy to trigger a remote IPMI command to shut down the device.

## **USE CASE 2:**

Remote scripts can be used to get the system event logs using RACADM. For instance, when power redundancy is lost for iDRAC servers, the user is notified with the alerts. An alert policy could be created on such alerts to trigger remote RACADM command to get SEL logs for that device.

## **USE CASE 3:**

Customer gets an SNMP trap on an appliance in a private network, based on which he/she wants to set up a policy to run a script on the remote server to send a mail to the group, with the device and alert details. For more information, refer the topic [Configure Remote Script Execution with token substitution](#)

## 2 Basic details of Remote Script Execution

Here are a few facts about the Remote Script Execution feature in OpenManage Enterprise that you need to be aware of before getting started:

- Only the OpenManage Enterprise users with Administrator privileges can use this feature

Features	User levels for accessing Dell EMC OpenManage Enterprise		
<b>Manage traps with Alert Polices</b>	Admin	Device Manager	Viewer
	Yes	No	No

- This feature supports up to 4 custom remote commands
- Script execution is supported only on Linux-based systems
- Token substitution is only supported for the “script” execution
- The nodes where the scripts are executed may or may not be managed by the console
- If any change is made to the “command name”, user would need to link the alert action again to the alert policy
- Batched commands with up to 100 scripts with token substitution or to a maximum of 3072 characters can be created and passed
- Batched command with up to 100 RACADM or IPMI commands containing a maximum of 3072 characters can be created and passed

## 3 Configure, Edit, and Delete scripts for remote execution

Scripts for remote execution can be customized to the different types of devices in the data center. This section provides a step-by-step explanation of how to create, edit and delete the scripts or commands for remote execution.

### 3.1 Configure Remote Script Execution with token substitution

- Select Create button a wizard for “Add Remote Command “will open.
- Provide a Command name.
- Select “script” radio button.
- In Authentication method select “Password”.
- Provide Admin username and password of remote box.
- Place the script file with name demo.sh in the remote system.
  - Example command in Script file to send a mail: `$ mail -s $1+$2+$3+$4 user1@domain.com`
- Provide Command to initiate the script .Ex-`./demo.sh $SEVERITY $IP $SERVICETAG $MSG`
- Passing Tokens to a Script: If you are using a batch file or a script, use \$1, \$2, \$3 , and so on to receive the values passed from OpenManage Enterprise. The values are passed in the order they are entered from left to right in the Arguments field.
  - For example, if you use \$IP \$SERVICETAG as arguments, a batch file with the following Echo `$1 $2` displays the following result: 172.168.10.10 ABC1234

The following table lists the supported tokens for Remote Script Execution Commands

Description	Tokens
Device IP Address	\$IP
Message	\$MSG
Date	\$DATE
Time	\$TIME
Severity	\$SEVERITY
Servicetag	\$SERVICETAG
Recommended Resolution	\$RESOLUTION
Alert Category Name	\$CATEGORY
Asset tag	\$ASSETTAG
Model Name	\$MODEL
Os hostname	\$HOSTNAME



Figure: Screen shot from linux terminal.

A screenshot of a web-based configuration dialog titled "Add Remote Command". The dialog has a close button (X) in the top right corner. It is divided into several sections:

- General Information**:
  - Command Name: remote command for script
  - Remote Command Type:  Script  RACADM  IPMI Tool
  - IP Address: 100.97.135.71
- Credentials**:
  - Authentication method:  Password  SSH Key
  - User Name: root
  - Password: .....
- Command**:
  - Command field: /demo|sh SIP SRESOLUTION

At the bottom right, there are "Finish" and "Cancel" buttons.

Figure: Screen shot from the Remote Script Execution page.



## 3.2 Configure Remote Script Execution with password for remote execution.

Here are the steps to configure the Remote Script Execution feature in OpenManage Enterprise using SSH with password.

- Activate the **Add Remote Command** wizard in OpenManage Enterprise (Application Setting > Script Execution > Create)
- Provide a Command Name
- For the Remote Command Type, select **Script**
- For the Authentication method select **Password**
- Provide **Admin Username** and **Password** of the remote box
- Place the script file in the remote system
- Provide Command to initiate the script. For example, `./demo.sh $IP $DATE`

The screenshot shows the 'Add Remote Command' wizard interface. It is divided into several sections:

- General Information:**
  - Command Name: remote command for script
  - Remote Command Type:  Script,  RACADM,  IPMI Tool
  - IP Address: 100.97.135.71
- Credentials:**
  - Authentication method:  Password,  SSH Key
  - User Name: root
  - Password: .....
- Command:**
  - ./demo.sh \$IP \$RESOLUTION

At the bottom right, there are 'Finish' and 'Cancel' buttons.

Figure 1: Configuring Remote Script Execution for script with password.

## 3.3 Configure Remote Script Execution for script with SSH key

### 3.3.1 Configure SSH key

- In the Linux host, run the following command to generate SSH key: **ssh-keygen -b 4096**.
- Enter path in which you want to save SSH key (/root/.ssh/id\_rsa): **/root/testKey/redhat\_id\_rsa**
- Your public key and private key has been saved in location **/root/testKey/**
- We get keys private key and public key respectively: **redhat\_id\_rsa redhat\_id\_rsa.pub**
- Use SSH with -i option to add public, for example - **ssh -i redhat\_id\_rsa.pub root@100.97.135.71**
- With ssh-copy-id command, make keys available to authorize logins on a remote machine. For example **ssh-copy-id -i /root/testKey/redhat\_id\_rsa.pub root@100.97.135.71**
- Now login to Linux SSH using the private key. **ssh -i redhat\_id\_rsa root@100.97.135.71**
- Copy contains of private key “redhat\_id\_rsa” in of SSH key box.

### 3.3.2 Procedure to configure Remote Script Execution for script with SSH key

- Activate the **Add Remote Command** wizard in OpenManage Enterprise (Application Setting > Script Execution > Create)
- Provide a Command Name
- For the Remote Command Type, select **Script**
- For the Authentication Method, select **SSH Key**
- Provide Username and “SSH KEY”.  
**Note:** SSH Private key is entered in this box. (id\_rsa)
- Provide Command to initiate the script. For example, **./demo.sh \$IP \$DATE**

The screenshot shows the 'Add Remote Command' wizard with the following configuration:

- General Information:**
  - Command Name: Linux script command
  - Remote Command Type:  Script  RACADM  IPMI Tool
  - IP Address: 100.97.135.71
- Credentials:**
  - Authentication method:  Password  SSH Key
  - User Name: root
  - SSH Key: eL8NjN8yhTFxipE7k6NTTdIUPIH  
Jc/HSXfwd  
-----END RSA PRIVATE KEY-----
- Command:**
  - Command: ./demo.sh \$IP \$RESOLUTION

Buttons: Finish, Cancel

**Figure 2:** Configuring Remote Script Execution for script with SSH Key.

### 3.4 Configure Remote Script Execution for RACADM / IPMI command

- Activate the **Add Remote Command** wizard in OpenManage Enterprise (Application Setting > Script Execution > Create)
- Provide a Command Name
- For the Remote Command Type, select either RACADM or IPMI
- Depending on your selection provide either RACADM or IPMI command in the Command Box.

The screenshot shows a dialog box titled "Add Remote Command" with a help icon and a close button in the top right corner. The dialog is divided into three sections: "General Information", "Command", and "Remote Command Type". In the "General Information" section, the "Command Name" field contains the text "ipmi command". The "Remote Command Type" section has three radio buttons: "Script", "RACADM", and "IPMI Tool", with "IPMI Tool" being the selected option. The "Command" section contains a text area with the text "-l lanplus power status". At the bottom right of the dialog, there are two buttons: "Finish" and "Cancel".

Figure 3: Configuring Remote Script Execution for IPMI commands

The screenshot shows a dialog box titled "Add Remote Command" with a help icon and a close button in the top right corner. The dialog is divided into three sections: "General Information", "Command", and "Remote Command Type". In the "General Information" section, the "Command Name" field contains the text "racadm command". The "Remote Command Type" section has three radio buttons: "Script", "RACADM", and "IPMI Tool", with "RACADM" being the selected option. The "Command" section contains a text area with the text "getniccfg". At the bottom right of the dialog, there are two buttons: "Finish" and "Cancel".

Figure 4: Configuring Remote Script Execution for RACADM commands.

### 3.5 Edit scripts or commands used in Remote Script Execution

Listed below are the steps to make changes to the existing scripts or commands meant for remote execution.

- Select the existing remote command from the Script Execution page (Application Setting > Script Execution)
- Click **Edit** to activate the Edit Remote Command wizard. You can edit the Command Name, Remote Command Type and the commands.

---

Note: If any change is made to the “command name”, user would need to link the alert action again to the alert policy

---

## 3.6 Delete scripts or commands used in Remote Script Execution

Listed below are the steps to delete the existing scripts or commands meant for remote execution.

- Select the existing remote command from the Script Execution page (Application Setting > Script Execution)
- Click **Delete**.

## 4 Create an alert policy and link to the remote execution scripts and commands

To activate the scripts and commands created for remote execution, you need to create alert policies and link those policies to the remote execution scripts or commands. A step-by-step explanation using screenshots is provided on creating alert policies and linking them to the remote-execution scripts or commands.

- Click **Create** on the Alerts > Alert Policies page to activate the Create Alert Policy wizard.
- Provide a name and description of the alert policy in the **Name and Description** page of the Create Alert Policy wizard.

The screenshot shows the 'Create Alert Policy' wizard interface. On the left, a vertical list of steps is shown, each with a green checkmark: 'Name and Description', 'Category', 'Target', 'Date and Time', 'Severity', 'Actions', and 'Summary'. The 'Name and Description' step is highlighted in blue. The main area contains the following fields:

- Name:** A text input field containing 'remote script with Linux ip'.
- Description:** A larger text input field that is currently empty.
- Enable Policy:** A checkbox that is checked.

At the bottom right, there are 'Next' and 'Cancel' buttons. The text 'Step 1 of 7' is visible at the bottom left of the wizard area.

Figure : Create Alert Policy Wizard - Name and Description for Alert policy

- From the **Category** page of the wizard, select category(ies) of the devices on which the alert policy would apply.

Create Alert Policy

Name and Description ✓

**Category** ✓

Target ✓

Date and Time ✓

Severity ✓

Actions ✓

Summary ✓

Step 2 of 7

All  
 >  Application  
 >  Dell Storage  
 >  iDRAC  
 >  IF-MIB  
 >  MM  
 >  Networking  
 >  OMSA  
 >  OpenManage Enterprise  
 >  OpenManage Essentials  
 >  Power Manager  
 >  RFC1215  
 >  SNMPv2-MIB  
 >  VMWare

Previous Next Cancel

- Select the target devices on the **Target** page. Only the devices belonging to the earlier-selected category(ies) would be available for selection.

Job Target

Select the target from devices or groups.

Name and Description ✓

Category ✓

**Target** ✓

Date and Time ✓

Severity ✓

Actions ✓

Summary ✓

Step 3 of 7

Select Devices  
 Select Groups **32 selected**  
 Specific Undiscovered Devices  
 Any Undiscovered Devices

ⓘ The selection of target devices is not applicable to all the events generated by the appliance. For example, the audit logs including the appliance settings, user login attempts, and others do not require the selection of target devices.

Previous Next Cancel

- In the **Date and Time** page of the Create Alert Policy wizard, specify the Date range, Time Frame or the Days.

The screenshot shows the 'Create Alert Policy' wizard at Step 4 of 7. The left sidebar lists the steps: Name and Description, Category, Target, Date and Time (highlighted), Severity, Actions, and Summary. The main area is titled 'Date and Time' and contains the following fields:

- Date Range:** From: 2019-06-25, To: No End Date.
- Time Frame:** 12 : 34 PM.
- Days:** A list of days with checkboxes: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. All are currently unchecked.

At the bottom right, there are three buttons: Previous, Next, and Cancel. The status 'Step 4 of 7' is shown at the bottom left.

- On the **Severity** page, specify the severity of the alerts based on which the alert policy would apply.

The screenshot shows the 'Create Alert Policy' wizard at Step 5 of 7. The left sidebar lists the steps: Name and Description, Category, Target, Date and Time, Severity (highlighted), Actions, and Summary. The main area is titled 'Severity' and contains a list of severity levels with checkboxes:

- All
- Unknown
- Info
- Normal
- Warning
- Critical

At the bottom right, there are three buttons: Previous, Next, and Cancel. The status 'Step 5 of 7' is shown at the bottom left.



- On the **Actions** page, select Remote Script Execution.

Create Alert Policy ? X

Name and Description	✓	<input type="checkbox"/> Email
Category	✓	<input type="checkbox"/> SNMP Trap Forwarding ( Enable )
Target	✓	<input type="checkbox"/> Syslog ( Enable )
Date and Time	✓	<input type="checkbox"/> Ignore
Severity	✓	<input type="checkbox"/> SMS
<b>Actions</b>	✓	<input checked="" type="checkbox"/> Remote Script Execution
Summary	✓	<input type="checkbox"/> Mobile

Linux script commands : 100.97.135.;

Step 6 of 7

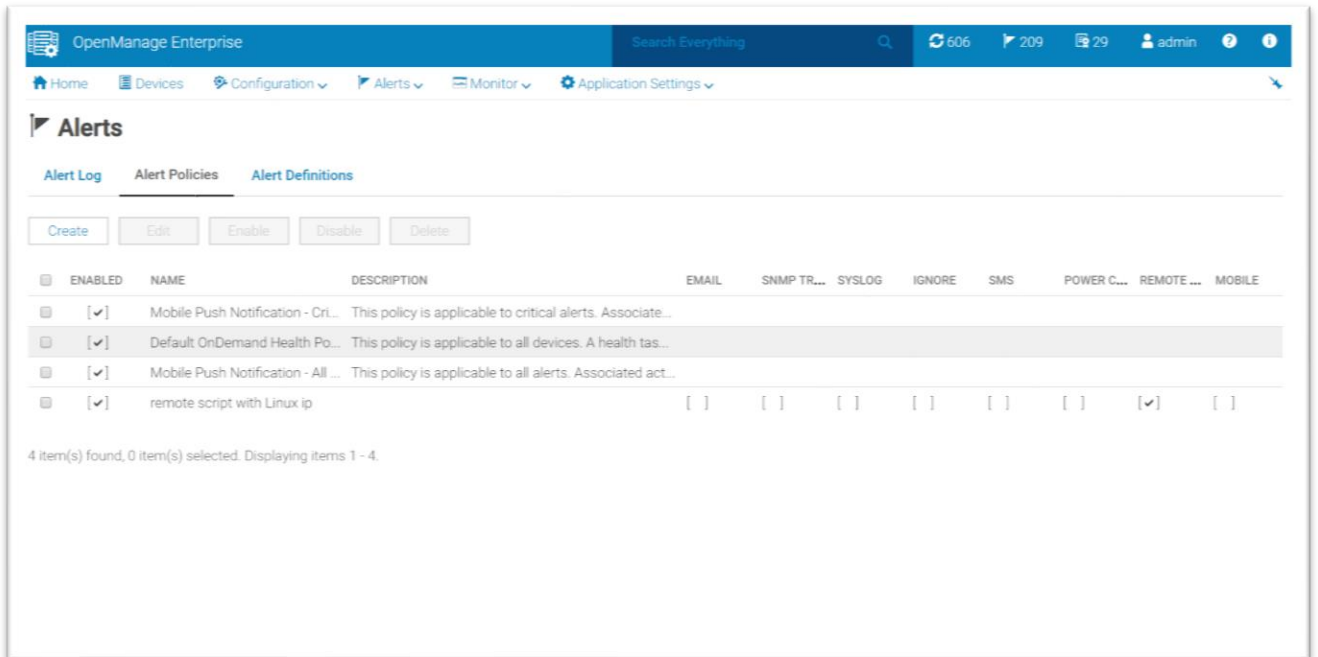
- On the **Summary** page, click Finish after reviewing the newly-created alert policy

Create Alert Policy ? X

Name and Description	✓	<h3>Review your inputs and click Finish to continue</h3> <table border="1"> <thead> <tr> <th>ATTRIBUTE</th> <th>VALUE</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>remote script with Linux ip</td> </tr> <tr> <td>Description</td> <td></td> </tr> <tr> <td>Enabled</td> <td>true</td> </tr> <tr> <td>Actions</td> <td>Remote Script Execution</td> </tr> <tr> <td>Targets</td> <td>32 Groups</td> </tr> <tr> <td>Start Date</td> <td>Jun 25, 2019 12:34:57 PM</td> </tr> <tr> <td>End Date</td> <td></td> </tr> <tr> <td>Days</td> <td>All</td> </tr> </tbody> </table>	ATTRIBUTE	VALUE	Name	remote script with Linux ip	Description		Enabled	true	Actions	Remote Script Execution	Targets	32 Groups	Start Date	Jun 25, 2019 12:34:57 PM	End Date		Days	All
ATTRIBUTE	VALUE																			
Name	remote script with Linux ip																			
Description																				
Enabled	true																			
Actions	Remote Script Execution																			
Targets	32 Groups																			
Start Date	Jun 25, 2019 12:34:57 PM																			
End Date																				
Days	All																			
Category	✓																			
Target	✓																			
Date and Time	✓																			
Severity	✓																			
Actions	✓																			
<b>Summary</b>	✓																			

Step 7 of 7

- Once an alert policy is successfully created, you can view the policy on the Alerts > Alert Policies page.



## 5 Validate remote script execution

Remote scripts/commands are executed when alerts are received specific to the linked alert policy. A job is created whenever a remote script is executed.

The status of all jobs can be viewed on the Jobs page (**Monitor > Jobs**). Following screenshots are of the completed remote-execution jobs.

The screenshot displays the OpenManage Enterprise interface. The top navigation bar includes 'Home', 'Devices', 'Configuration', 'Alerts', 'Monitor', and 'Application Settings'. The main content area is titled 'Return to Jobs' and 'Job Details'. The job details show:

- Job Name: Script Execution on 100.97.135.71
- Job Type: Device Action
- Description: Script Execution on external event
- Status: Completed (indicated by a green checkmark)

Below the details is a 'Restart Job' button and a 'Execution History' section with an 'Export' button. The execution history table shows one completed job:

STATUS	START TIME	END TIME	ELAPSED	PERCENTAGE COMPLETE
Completed	Jun 25, 2019 5:30:14 PM	Jun 25, 2019 5:30:23 PM	00:00:08	100%

Below the table, it says 'Item(s) found. Displaying items 1 - 1.' and there is an 'Execution Details' section. The execution details table shows:

STATUS	TARGET SYSTEM	START TIME	END TIME	ELAPSED TIME
Completed	bdc-s470-host03.austinlab.vrm	Jun 25, 2019 5:30:14 PM	Jun 25, 2019 5:30:23 PM	00:00:08

Below the table, it says '1 item(s) found. Displaying items 1 - 1.' and there is a 'Results' section showing the command output:

```
Result:
Target System: bdc-s470-host03.austinlab.vrm
Messages:
Running
Executing Script on device - 100.97.135.71
***** Command Started *****
Command: /!sh 100.97.131.203 12:00:13

bash: /!sh: No such file or directory
***** Command Completed *****

Completed
```

Figure: Completed remote-execution script.

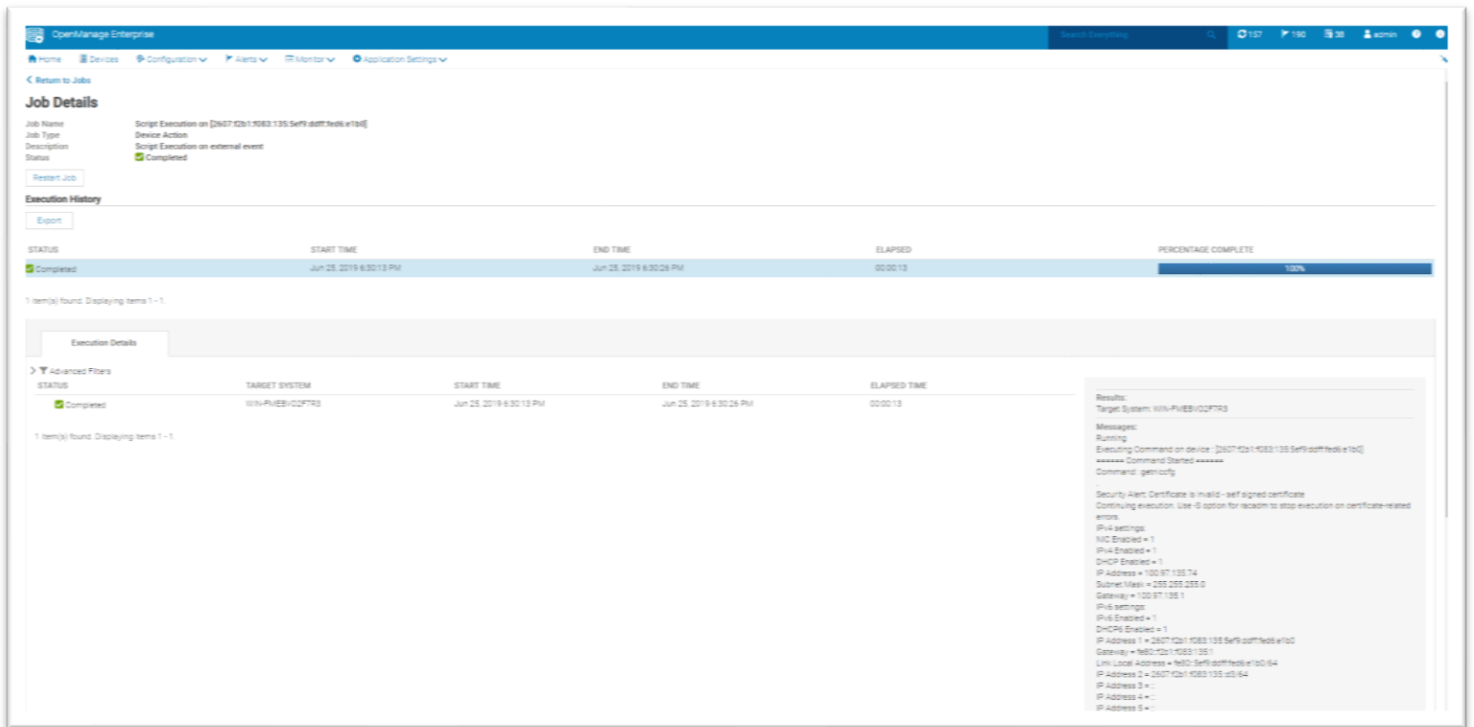


Figure: A completed RACADM job which was remotely executed.

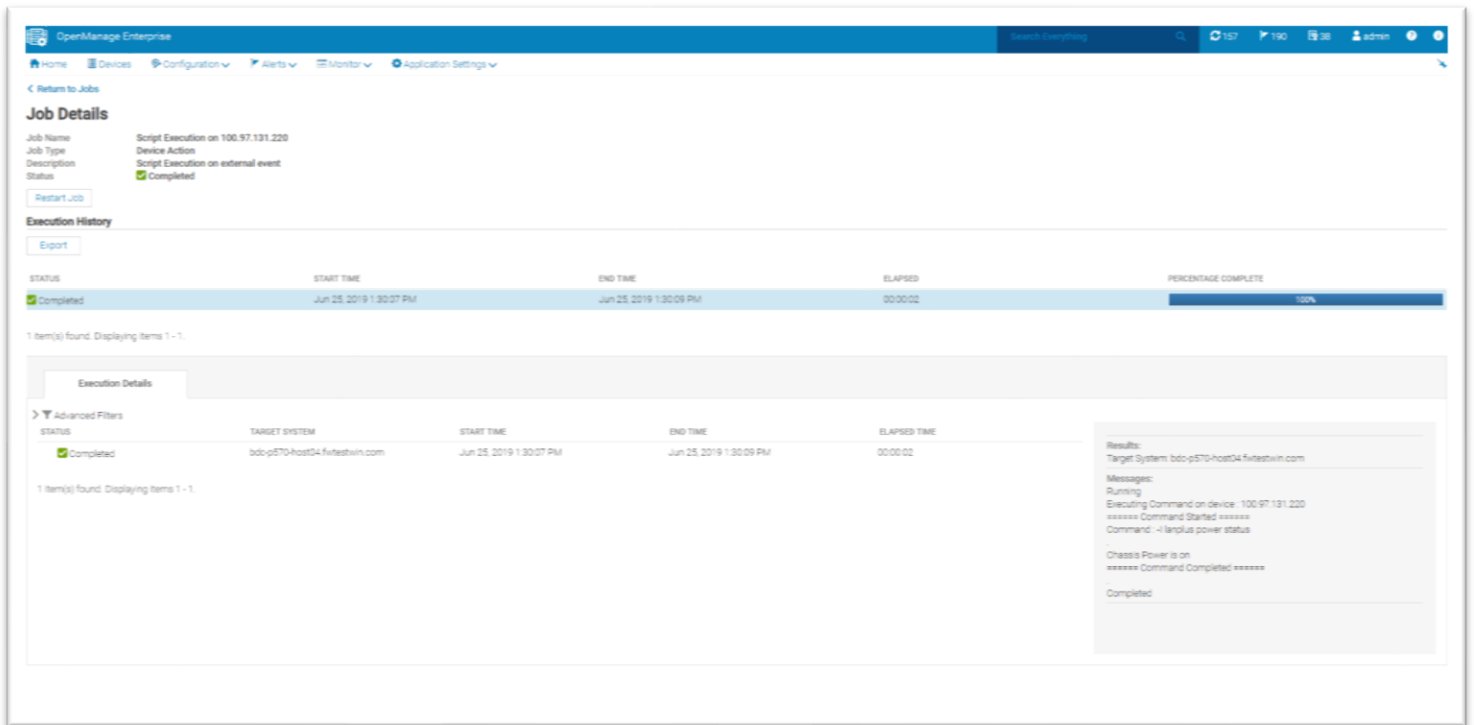


Figure: Successful IPMI command remote script execution.

## 6 Remote script execution using RESTful APIs

Remote script execution can also be implemented using RESTful APIs. Using REST APIs you can create remote RACADM commands, IPMI commands and SSH script and link them to alert policies. Click on the following links for more information:

[Create IPMI Commands from Remote Script Execution page using REST APIs](#)

[Create RACADM Commands from Remote Script Execution page using REST APIs](#)

[Create scripts with SSH key authentication using REST APIs](#)

[Create Alert policy and link with action "Remote Script"](#)

### 6.1 Create IPMI Commands from Remote Script Execution page using REST APIs

- Below payload creates IPMI remote command with name test\_ipmi and

Command used: -l lanplus sel time get

URI: https://100.97.140.61/api/ApplicationService/Settings

Method: POST

Payload:

```
{
  "ConsoleSetting": [
    {
      "Name": "REMOTE_COMMAND_NAME2",
      "DefaultValue": "",
      "Value": "test_ipmi",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING2"
    },
    {
      "Name": "REMOTE_COMMAND_IP2",
      "DefaultValue": "",
      "Value": "",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING2"
    },
    {
      "Name": "REMOTE_COMMAND_USER2",
      "DefaultValue": "",
      "Value": "",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING2"
    },
    {
      "Name": "REMOTE_COMMAND_TYPE2",
      "DefaultValue": "",
```

```

        "Value": "Ipmi",
        "DataType": "java.lang.String",
        "GroupName": "REMOTE_COMMAND_ACTION_SETTING2"
    },
    {
        "Name": "REMOTE_COMMAND_CMD2",
        "DefaultValue": "",
        "Value": "-Ilanplus sel time get",
        "DataType": "java.lang.String",
        "GroupName": "REMOTE_COMMAND_ACTION_SETTING2"
    }
]
}

```

## 6.2 Create RACADM Commands from Remote Script Execution page using REST APIs

- Below payload creates remote racadm command with name test\_racadm and

Command Used: getniccfg

URI: https://100.97.140.61/api/ApplicationService/Settings

METHOD: POST

Payload:

```

{
  "ConsoleSetting": [
    {
      "Name": "REMOTE_COMMAND_NAME3",
      "DefaultValue": "",
      "Value": "test_racadm",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
      "Name": "REMOTE_COMMAND_IP3",
      "DefaultValue": "",
      "Value": "",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
      "Name": "REMOTE_COMMAND_USER2",
      "DefaultValue": "",
      "Value": "",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
      "Name": "REMOTE_COMMAND_TYPE3",
      "DefaultValue": "",
      "Value": "Racadm",

```

```

        "DataType": "java.lang.String",
        "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
        "Name": "REMOTE_COMMAND_CMD3",
        "DefaultValue": "",
        "Value": "getniccfg",
        "DataType": "java.lang.String",
        "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    }
]
}

```

## 6.3 Create scripts with SSH key authentication using REST APIs

- Below payload creates remote racadm command with name test\_racadm and

Command Used: getniccfg

URI: https://100.97.140.61/api/ApplicationService/Settings

METHOD: POST

Payload:

```

{
  "ConsoleSetting": [
    {
      "Name": "REMOTE_COMMAND_NAME3",
      "DefaultValue": "",
      "Value": "test_racadm",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
      "Name": "REMOTE_COMMAND_IP3",
      "DefaultValue": "",
      "Value": "",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
      "Name": "REMOTE_COMMAND_USER2",
      "DefaultValue": "",
      "Value": "",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
      "Name": "REMOTE_COMMAND_TYPE3",
      "DefaultValue": "",
      "Value": "Racadm",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    }
  ]
}

```

```

    },
    {
      "Name": "REMOTE_COMMAND_CMD3",
      "DefaultValue": "",
      "Value": "getniccfg",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    }
  ]
}

```

## 6.4 Create Alert policy and link with action “Remote Script”

•Below payload creates remote command with remote command type as Script and authentication method as SSH key and command used as: ./demo.sh \$IP \$HOSTNAME

URI: https://100.97.140.61/api/ApplicationService/Settings

METHOD: POST

PAYLOAD

```

{
  "ConsoleSetting": [
    {
      "Name": "REMOTE_COMMAND_PWD3",
      "DefaultValue": "",
      "Value": "",
      "DataType": "java.lang.String",
      "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
    },
    {
      "Name": "REMOTE_COMMAND_SSHKEY3",
      "DefaultValue": "",
      "Value": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEoQIBAAKCAQEAI2iOQ/yOzjIqrzeBrLYFhWhOqWqNKZMBM0LktkfrRyEVEK3d\nUEmszrvfNfS/
/ICN47xKNHSubFwsK/Fo5ZPcOrGGIM/SPpN1mwzYXoYtrhu3o7jJ\nW6hNwhEyuErl142VHJPNI8o1AB
ECgPD9L1U+uj9wJYobnxysl5QUshZcDJSzBLRJ\nnrtK+hDhMwvSaZm0ppIlmOa4nLXUNcKWfSYnMKPcF
IwGO4li6188qUcN1N2B5VZG4\nbqNGvklIydcDLw+CDOnvDB6UYmnn5gsYS8Mjmhns0T0vGtRrrqse+
lW81cMl3z0\nQWjc0bPAL5sW/09LVkVLJrxZftZoVy5Gn5qgRQIBJQKCAQAlrY4742TdyFeJTUWk\nnTN
7lySoERk+qXxUUxfif98MMTiFedC36BhMhcQv1EbdZACZZOc7dvqSnqjVzqQeX\n/nLfbj/tMT+94qop
58vGhSEaTK4sP8e36Exd9s9wFD4elPfeNci9kJiz6Ovrf2cF\nn37AWqVyUhjD6kiDPBWlgh/0z1L617+
EeTzZaLCn1cvpkew3sArjJHx9PIbuD6/AY\nnlcCiM9V7mJXYWlAHUqABzhXAwnbaeodAyv00iHTTaeLz
b3EhvRxZ9ZOJa0M+tdID\nnlvblcc2neNu6NKiT3Eeuh+sVWc0ctDrXwdBvealEgRJYmzOWT4SF6zVNFZ
STGxL7\nn9x8tAoGBAP8THSP9bJ+AnpHUOKkQH19qqrFkpaigSbwQHuh5dXyqWz/KSXi/0LI\nnXrrcug
CZJHA6JRokKMINz+/2drWcaLqYhXC9MYC4f7nzH+eEm2ZR0QmtTWWkMjNP\nnEaZrVohJKcJIMlbQhaJC
lmagVHcvBza42PE+dpXRob6Lyg0ZXV/VAoGBAIVqBf5a\nn2qK191LikoYVubBQoW+Ge73ekKpLfCtMrR
PeJ2inVwte2BzBJR9uW727bISMp000\nnWGVhTUIri7kN8NgRKMibrT6T3U6+F0FTeSvEI9CAuGIadkg8
k5aKzyjzyQZ+r524\nn/IQ12zQRz3WicDwkeVX4xeTcyzdEs0AVKuaxAoGADcmujFLjRuRbms4Q5os/91
Hf\nto/RmmoPSQYRUsp7IFJOWOsYqBobor65Ay6GmD+hGtK15btcJipC11luM1SQChYV\nnDQNOyLAG53
Ts98H6j+i/MPScZlVnCbE/OwWzuS13ytpjlf1vAdoV97Wjs2pTar3i\nnNo3BOIffOrtytJmWWDUCgYB5
Ab/+k8QCWC77k4WkZdEHMPpSndLNuZHfY+EC2ofZ\nn1OPQIGYXq/ktocABLwM8TxG+vtXZj92/dr9b0q

```



```
Ja25/ddp/QF+/HayBEGgZGVgfu\ngB75hBUWk203j1YhVXT1Jd5Kzm5e5S1rXwLFP9YEyQcfQiO/+b/o
h5r/XgLpO5q5\nrQKBgQCyCt7kPOTfmTpTDmws0wHYRoWrmdib6qXLeIR4W8k5hxnBMtfcuFwalIMb\n
GLUOcW1RVg4X+axQCKc2MpzE6I1ZsuXK5fQTol8Y+8aldRLCTTN6Jb3cJ+jRPwPT\nnpRk75dAmSFNoUY
BU1yJblf7lHMklr5KHgnNZIoUHkgtoLsFSVg==\n-----END RSA PRIVATE KEY-----",
```

```
    "DataType": "java.lang.String",
    "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
  },
  {
    "Name": "REMOTE_COMMAND_NAME3",
    "DefaultValue": "",
    "Value": "test_script",
    "DataType": "java.lang.String",
    "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
  },
  {
    "Name": "REMOTE_COMMAND_IP3",
    "DefaultValue": "",
    "Value": "10.255.3.183",
    "DataType": "java.lang.String",
    "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
  },
  {
    "Name": "REMOTE_COMMAND_USER3",
    "DefaultValue": "",
    "Value": "root",
    "DataType": "java.lang.String",
    "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
  },
  {
    "Name": "REMOTE_COMMAND_TYPE3",
    "DefaultValue": "",
    "Value": "script",
    "DataType": "java.lang.String",
    "GroupName": "REMOTE_COMMAND_ACTION_SETTING1"
  },
  {
    "Name": "REMOTE_COMMAND_CMD3",
    "DefaultValue": "",
    "Value": "./demo.sh $IP $HOSTNAME ",
    "DataType": "java.lang.String",
    "GroupName": "REMOTE_COMMAND_ACTION_SETTING3"
  }
]
}
```

d) Create Alert policy and link with action "Remote Script"

```
URI: https://100.97.140.61/api/AlertService/AlertPolicies
METHOD: POST
PAYLOAD:
```

```

{
  "Name": "Remote Script",
  "Enabled": true,
  "DefaultPolicy": false,
  "PolicyData": {
    "Severities": [],
    "Devices": [],
    "DeviceTypes": [],
    "Groups": [500, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008,
1009, 1010, 1011, 1023, 1024, 1025, 1012, 1013, 1014, 1015, 1016, 1017, 1018,
1019],
    "UndiscoveredTargets": [],
    "Schedule": {
      "StartTime": "2019-01-22 20:27:49.453",
      "EndTime": "",
      "CronString": "* * * ? * * *"
    },
    "Actions": [{
      "Name": "RemoteCommand",
      "TemplateId": 111,
      "ParameterDetails": [{
        "Name": "remotecommandaction",
        "Id": 1,
        "Value": "test_ipmi"
      }]
    }]
  },
  "State": true
}

```

## A Technical support and resources

- [Dell.com/support](https://www.dell.com/support) is focused on meeting customer needs with proven services and support.
- For OpenManage Enterprise documentation on the Dell support site, see [OpenManage Enterprise documents](#)