

How to Integrate Pandora FMS with PagerDuty step by step

Introduction

Pandora FMS allows monitoring in a visual way the status and performance of several parameters from different operating systems, servers, applications and hardware systems such as firewalls or routers.

The result of this monitoring can be used basically in reports and alerts. With Pandora FMS' alert system you can configure an infinity number of ways to alert the users about strange behaviors. An alert can send an email or SMS, trigger a sound alert, write a log... or do anything you are able to do executing your own scripts.

With PagerDuty, you can externalize all the notification tasks, avoiding configure and maintain them on your systems.

This integration is done translating Pandora FMS' alerts into PagerDuty's incidents.

To perform this translation Pandora FMS' alert will execute one simple perl script to create a PagerDuty's incident through an API call. Then, the user will configure his PagerDuty's service to manage the incidents (including validation and notifications) via email, sms, phone calls or mobile push notifications.

Understanding the integration

PagerDuty incidents

In PagerDuty the incidents have three possible status: Triggered, Acknowledged and Resolved.

Triggered is the initial status, and then, the user can set it as Acknowledged or Resolved (closed) using the web interface, the mobile apps or replying the notification SMS.

The incident status can be changed by the API too. This feature is used in the integration to change the status automatically when a Pandora FMS alert has been recovered.

Pandora FMS alerts

In Pandora FMS, the alerts are fired when one specific condition occurs. When this condition stops happening, the alert is recovered. It's possible to define a different action when an alert is fired and when it's recovered. I.E.: To send an email with different messages.

With this integration is possible close the PagerDuty incident (changing the status to Resolved) when an alert is recovered.

From Pandora FMS alert to PagerDuty incident

Each Pandora FMS' alert match up with a PagerDuty's incident, but not necessarily the same incident all the time.

A new incident will be created by a Pandora FMS' alert when:

- An PagerDuty's incident associated to the alert doesn't exist (The alert was never fired before)
- When the PagerDuty's incident associated with the alert is closed (status: Resolved). This status can be changed by:
 - The user: The incident is considered closed by the user but still happening, so a new incident will be created and associated to the alert.
 - Pandora FMS: It means that a Pandora FMS alert is configured to close the incident when it has been recovered. So when it's fired again a new incident will be created and associated to the alert.

The integration script

A perl based script has been created to make a call to the PagerDuty API.

The script is called *pandorafmsalert2pagerduty.pl* and have the following execution syntax:

```
./pandorafmsalert2pagerduty.pl [service_api_key] [id_alert] [event_type] [description] [fired_timestamp*]  
[severity*] [agent_name*] [module_name*] [module_data*]
```

- service_api_key: The API key of PagerDuty's service where the incident will be created.
- id_alert: Identification number of the alert in Pandora FMS
- event_type: Incident type on PagerDuty. 'trigger' for create an incident, 'resolve' to close it.
- description: Description stored in PagerDuty's incident.
- fired_timestamp: Timestamp with format 'yy-mm-dd hh:mm:ss' when the alert was fired on Pandora FMS.
- severity: alert severity (Maintenance, Informational, Normal, Minor, Warning, Major, Critical)
- agent_name: Name of the Pandora FMS agent that fired the alert.
- module_name: Name of the Pandora FMS monitor that fired the alert.
- module_data: Monitor data that caused the alert to be fired on Pandora FMS.

** Pandora FMS alert info passed to the PagerDuty's incident as extra details.*

Setting up the PagerDuty / Pandora FMS integration

Integration script configuration

The integration script doesn't need to be configured, because the necessary data is fully parametrized.

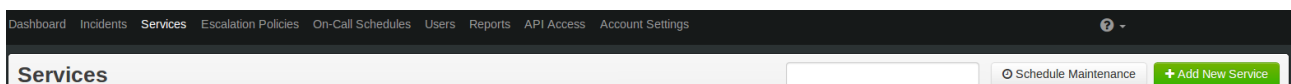
However, we can define an advanced configuration value. The script need to maintain a list with the relationship between Pandora FMS and PagerDuty Ids. It's done creating a hidden file for each Pandora FMS alert that contain the Pandora FMS ID as the name and the PagerDuty ID in the content. It means that one file will be created for each Pandora FMS alert. These files are stored by default in the same path where the script is located, but you can change it modifying a variable at the starting of the file:
\$matchup_dir

```
1  #! /usr/bin/perl
2
3  use LWP::UserAgent;
4  use JSON qw(decode_json);
5  use File::Basename;
6
7  ### Configuration parameters ###
8
9  # Directory where is stored necessary files with match up ids between Pandora FMS and PagerDuty
10 # One hidden tiny file per alert will be created
11 my $matchup_dir = dirname(__FILE__);
12
13 #####
14
```

In PagerDuty

Create a "Generic API system" service:

1. In your account, under the **Services** tab, click "Add New Service".




2. Enter a name for the service and select an escalation policy. Then, select "Generic API system" for the Service Type.

Name

Description

Escalation Policy
The policy specifies **who will be alerted** when faults are reported by your monitoring tool.

Type 

- ☐ Generic email system
Any system that can be configured to send email.
- ☒ Generic API system
Any system that can be configured to make an HTTP API call.
- ☐ HP SiteScope
The [HP SiteScope](#) application monitoring software.
- ☐ Keynote
The [Keynote](#) web and mobile performance monitoring system.
- ☐ Nagios
The [Nagios](#) network and infrastructure monitoring system.
- ☐ Pingdom
The [Pingdom](#) uptime monitoring system.
- ☐ Server Density
The [Server Density](#) server monitoring system.
- ☐ SQL Monitor
The [Red Gate SQL Monitor](#) database monitoring system.
- ☐ Solarwinds NPM
The [Solarwinds Network Performance Monitor](#) system.
- ☐ Amazon CloudWatch
The [Amazon CloudWatch](#) monitoring system.




3. Click the “Add Service” button.

4. Once the service is created, you’ll be taken to the service page. On this page, you’ll see the “Service key”, which will be needed to configure Pandora FMS to send alerts to PagerDuty.




PagerDuty example service

Your open incidents	
0 triggered	0 acknowledged



General Settings

Service Name 	PagerDuty example service
Enabled 	Yes
Description	Example service to integrate alerts from Pandora FMS
Escalation Policy 	Default

Integration Settings

Service Type 	 Generic API View PagerDuty API documentation
Service API Key 	c3845b5136b748559a5090a2015aa834

Incident Settings

Incident Ack Timeout 	Enabled: Acknowledged incidents time out after 30 minutes
Incident Auto-Resolution 	Enabled: Open incidents will auto-resolve in 4 hours

5.You can configure PagerDuty users and notifications for this service.

6.At this moment the dashboard view is empty.

Dashboard

Your open incidents

0 triggered 0 acknowledged

All open incidents

0 triggered 0 acknowledged

Incidents

Open 0Triggered 0Acknowledged 0All 0

Assigned to meAll

Selected Incidents: ResolveAcknowledgeReassign

Go to incident #:

	#	Created On	Details	Service	Assigned To	Status
No incidents.						

Selected Incidents: ResolveAcknowledgeReassign

Go to incident #:

Show 10 incidents per page

In Pandora FMS

A little bit of theory to understand the Pandora FMS alerts:

An alert of PandoraFMS is composed by three kind of levels:

- Template (upper level)
- Action (middle level)
- Command (lowest level)

The configuration of these levels includes 10 possible custom fields to be used as parameters in the execution of the alert.

The fields filled in command override fields filled in action. In the same way the fields filled in action override fields filled in the template. Template recovery is an exception, in this case template overrides action fields this adds more flexibility.

Create a Pandora FMS alert integrated with PagerDuty step by step:

Requirements:

- **Pandora FMS 5.0 or greater system installed.** To know requirements and how to install/configure check the online documentation at: <http://wiki.pandorafms.com>
- **Perl installed with following dependences:**
 - **LWP::UserAgent** (To perform the Post call to the API)
 - **JSON** (To build the data structure on the call)
 - **File::Basename** (To find the script path and store the temporary files)
 - **IO::Socket::SSL** (To perform the API call using HTTPS)
- **Internet connection.** The integration will be performed calling to PagerDuty web API. The Pandora FMS server will need to have access to this resource.

Since Pandora FMS 5.0 SP3, the integration script and an Alert command called 'PagerDuty incident' are

included out-of-the-box.

Please, check if you got it on your system to skip steps 1 and 2:

- The integration script's path is:
`[SERVER_INSTALLATION_PATH]/pandora_server/util/pagerduty/pandorafmsalert2pagerduty.pl`
- The Alert command list of your system is available in the section *Administration->Manage alerts->Commands* of your Pandora FMS console.

Configuration steps:

1. Download the integration script to your system

Download the script called *pandorafmsalert2pagerduty.pl* from the Pandora FMS modules library:

http://pandorafms.com/index.php?sec=Library&sec2=repository&lng=es&action=view_PUI&id_PUI=599

Place it in a path reachable by the server. The recommended path is:

`[SERVER_INSTALLATION_PATH]/pandora_server/util/pagerduty/`

2. Create an "Alert Command" with the generic execution of the script

We can create the necessary alert command in two ways. Automatically with a predefined SQL script or manually in the Pandora FMS interface.

1. Automatically

Execute on your Pandora FMS (5 or greater) database the SQL script *PagerDutyPandoraFMSCommand.sql*. You can find this script in the Pandora FMS modules library:

http://pandorafms.com/index.php?sec=Library&sec2=repository&lng=es&action=view_PUI&id_PUI=599

2. Manually

Go to menu *Administration->Manage alerts->Commands* and create an alert command with the call to the script that will be executed when alert being fired:

The screenshot shows the Pandora FMS web interface. The left sidebar contains navigation menus for 'Operation' and 'Administration'. The main content area is titled 'Alerts > Configure alert command'. The form has the following sections:

- Name:** PagerDuty incident
- Command:** `/usr/share/pandora_server/util/pagerduty/pandorafmsalert2pagerduty.pl "_field8_" "_id_alert_" "_field9_" "_field10_" "_timestamp_" "_alert_text_severity_" "_agent_" "_module_" "_data_"`
- Description:** Call a script that create an incident in the online service PagerDuty calling its web API
field8 : Service API key assigned a service created on PagerDuty
field9 : Type of the incident call 'trigger' to new incident and 'resolve' to close it
field10 : Description of the incident
- Field table:** A table with 10 rows and 4 columns: Field description, Field 1 values, Field 2 values, and Field 3 values. The last three columns are currently empty.

At the bottom right of the form is an 'Update' button.

The command is the call to the integration script using the absolute path:

```
/usr/share/pandora_server/util/pagerduty/pandorafmsalert2pagerduty.pl "_field8_" "_id_alert_"  
"_field9_" "_field10_" "_timestamp_" "_alert_text_severity_" "_agent_" "_module_" "_data_"
```

The custom fields will be:

field8: Service API key retrieved from PagerDuty service

field9: Type of the incident. Possible values:

trigger: For standard call for create or update incidents.

resolve: To close incident.

field10: Description of the Incident.

NOTE: Highest fields are used to make it easily compatible with typical templates and actions which use lowest fields.

3. Create an "Alert Action" for each PagerDuty's service

An alert in Pandora FMS can have several actions. And the actions use generic commands adding desired parameters.

In this case, we create an action for each PagerDuty's service. To do this, we go to menu *Administration->Manage alerts->Actions* and create a new alert action.

The screenshot shows the 'Configure alert action' page in Pandora FMS. The left sidebar has a menu with 'Operation' and 'Administration' sections. The 'Administration' section is expanded, showing 'Manage monitoring', 'Massive operations', 'Manage modules', 'Manage alerts', 'Templates', 'Actions', 'Commands', and 'List of special days'. The 'Manage alerts' option is selected. The main area shows the 'Configure alert action' form. The 'Name' field is 'New PagerDuty incident'. The 'Group' is 'All'. The 'Command' field contains a script that calls the PagerDuty API using custom fields _field8_, _field9_, and _field10_. The 'Threshold' is set to 0 seconds. The 'Service API key' field contains a long alphanumeric string. The 'Type' is set to 'Trigger'. The 'Description' field contains the text 'Alert has been fired in Pandora FMS'. The 'Command preview' field shows the execution of the integration script.

In the alert configuration we will select the PagerDuty command.

Three configuration fields will appear:

- **Service API key:** This is the key that we get from our PagerDuty service.
- **Type:** Can be 'Trigger' to create a new incident or 'Resolve' to close it. Resolve will typically used on recovery alert parameters.
- **Description:** Description that will be sent to the PagerDuty incident. A generic example can be "Alert has been fired in Pandora FMS".

While we change the configuration fields, we observe that the command preview is updated showing us the execution of the integration script.

4. Configure a Pandora FMS alert integrated with PagerDuty

Once created the necessary elements we're going to create an example alert on a network monitor.

We create an example Agent in *Administration->Manage monitoring->Manage agents* clicking on the button 'Create agent'.

Agent manager

Agent name: Agent example

IP Address: 8.8.8.8

Parent: Cascade protection

Group: Network

Interval: 5 minutes

OS: Network

Server: burbuja

Description:

Advanced options

Custom fields

Create

Inside this agent, we create a simple network monitor.

Agent example - Modules

Search: Filter

Create a new network server module

Create

No available data to show

Get more modules in Pandora FMS Library

This monitor will be a 'Host alive' that make a ping to a defined IP every 5 minutes.

Agent example - Modules

Using module component: Network Management

Name: Host Alive

Type: Remote ICMP network agent, book

Warning status: Min: 0, Max: 0, Inverse interval

FF threshold: 0

Target IP: 8.8.8.8

Disabled: ☐

Module group: Networking

Critical status: Min: 0, Max: 0, Inverse interval

Historical data: ☒

Port: 0

Advanced options

Module macros

Create

Associated to this module, we will create an alert with the template 'Critical condition' with the action 'New PagerDuty incident'.

Agent example - Alert

Alert control filter

Total items: 0

No alerts defined

Module: Host Alive

Template: Critical condition

Actions: New PagerDuty incident

Threshold: 0 seconds

Create Template

Create Action

Add alert

Here we get our alert ready

Full list of monitors

Form filter

Total items: 1

F.	P.	Type	Module name	Description	Status	Warn	Data	Graph	Last contact
Networking									
			Host Alive			N/A - N/A	1		58 seconds

Total items: 1

Full list of alerts

Total items: 1

P.	S.	F.	Module	Template	Action	Last fired	Status	Validate
			Host Alive	Critical condition	New PagerDuty incident	Unknown		

Validate

Now, when this monitor goes to critical status (the ping fails), the alert will be fired.

Full list of monitors

Form filter

Total items: 1

F.	P.	Type	Module name	Description	Status	Warn	Data	Graph	Last contact
Networking									
			Host Alive			N/A - N/A	0	100	16 seconds

Total items: 1

Full list of alerts

Total items: 1

P.	S.	F.	Module	Template	Action	Last fired	Status	Validate
			Host Alive	Critical condition	► New PagerDuty incident	16 seconds		

Validate

When the alert is fired, an incident is created on PagerDuty via API

Your open incidents

1 triggered0 acknowledged

All open incidents

1 triggered0 acknowledged

Incidents

Open 1

Triggered 1

Acknowledged 0

All 1

Assigned to meAll

Selected Incidents:

ResolveAcknowledgeReassign

Go to incident #:

<input type="checkbox"/>	#	Created On	Details	Service	Assigned To	Status											
<input type="checkbox"/>	21	Jan 9, 2014 at 10:13 AM	Failure: Alert has been fired in Pandora FMS	PagerDuty example service	Sancho Lerena	Triggered	Details										
Service Key																	
c3845b5136b748559a5090a2015aa834																	
Description																	
Alert has been fired in Pandora FMS																	
Incident Key																	
bfe3f87d13a64022a1434ca52a353971																	
Details																	
<table><tr><td>agent</td><td>Agent example</td></tr><tr><td>fired_timestamp</td><td>2014-01-09 09:59:20</td></tr><tr><td>module</td><td>Host Alive</td></tr><tr><td>module_data</td><td>0.00</td></tr><tr><td>severity</td><td>Critical</td></tr></table>								agent	Agent example	fired_timestamp	2014-01-09 09:59:20	module	Host Alive	module_data	0.00	severity	Critical
agent	Agent example																
fired_timestamp	2014-01-09 09:59:20																
module	Host Alive																
module_data	0.00																
severity	Critical																

We can see the description of the fired alert and the extra details sent by Pandora FMS alert.

After incident triggering, if configured, PagerDuty sends the alerts to the desired contacts by phone call, SMS, emails or mobile apps push notifications.

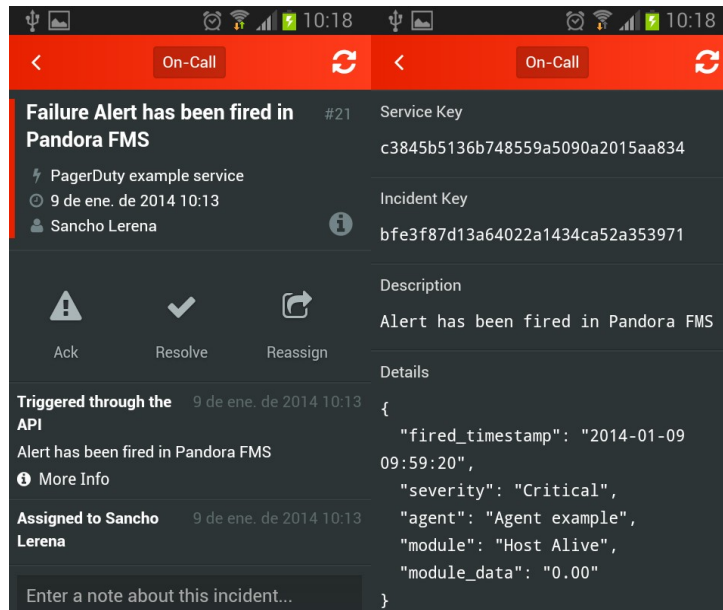
For example:

Email

Hello Sancho Lerena,
You are assigned 1 triggered incident in PagerDuty:
Please visit the following URL to manage this incident.
<https://artica-st.pagerduty.com/dashboard>

1) Incident #21
Opened on: Jan 9 at 10:13am CET
Service: PagerDuty example service
Description: Alert has been fired in Pandora FMS
Link: <https://artica-st.pagerduty.com/i/21>

Mobile app push notification

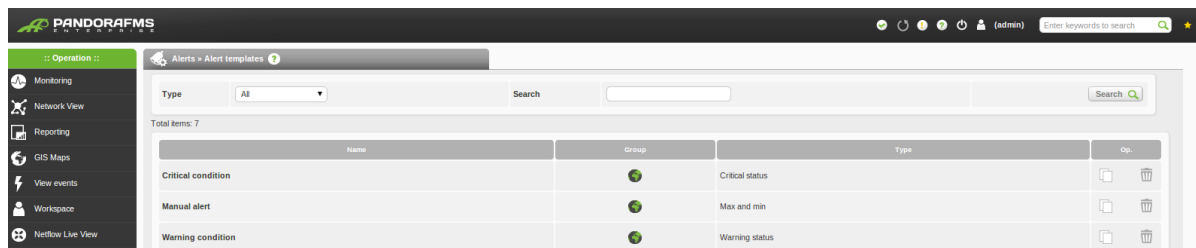


5. Configure an alert template to close PagerDuty incidents when alert been recovered

This step is only required if you want that when an alert on Pandora FMS has been recovered, the incident associated to it in PagerDuty will be closed automatically.

We go to *Administration->Manage alerts->Templates* and edit the desired template.

We are going to configure one of the templates that Pandora FMS have configured by default: "Critical condition". But this configuration is compatible with any alert template.



In the template editor we go to the Step 3 > Recovery.

Alert recovery Enabled

Field 2: [PANDORA] Alert RECOVERED for CRITICAL status on _agent_/_module_

Field 3: Hello, this is an automated email coming from Pandora FMS
This alert has been RECOVERED from a CRITICAL condition in one of your monitored items:

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

Field 9: resolve

Field 10: Alert has been recovered in Pandora FMS

As we remember the fields configured on the Command used to PagerDuty integration are 8,9 and 10.

- **Field 8: API key of PagerDuty's service.** This key is configured in the action so we left it blank.
- **Field 9: Type of the incident.** We selected Trigger in the action to create the incident on PagerDuty when Pandora FMS' alert has been fired. In recovery cases we override the value with "resolve". This type will close the PagerDuty incident.
- **Field 10: Description of the incident.** We change the description to register a different description in the incident when it is closed.

Other option is left this field in blank and use the action field. In this case will be added to the field the prefix [RECOVER] in recovering cases.

When the alert has been recovered (The ping response works again), the incident on PagerDuty will be closed (Resolved).

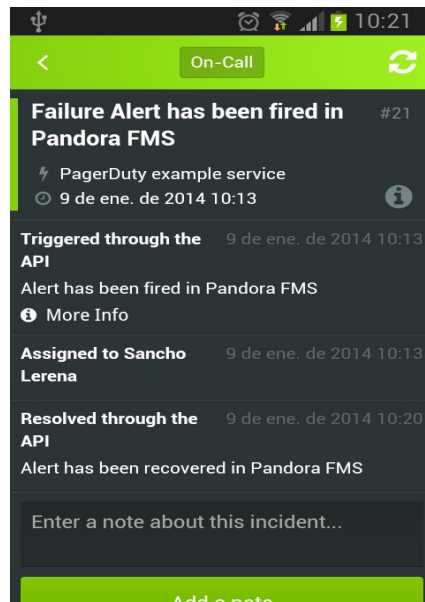
Incident #21

Service	PagerDuty example service
Current status	Resolved automatically by timeout
Incident times:	Opened from Jan 9 at 10:13am to Jan 9 at 10:20am (Opened for 7 min)

Incident Log

Time	Activity
Jan 9, 2014 at 10:20 AM	Resolved through the API. Description: Alert has been recovered in Pandora FMS
Jan 9, 2014 at 10:13 AM	Notified Sancho Lerena by push notification to GT-I9100.
Jan 9, 2014 at 10:13 AM	Notified Sancho Lerena by email at zarzuelo@artica.es.
Jan 9, 2014 at 10:13 AM	Assigned to Sancho Lerena .
Jan 9, 2014 at 10:13 AM	Triggered through the API. Description: Alert has been fired in Pandora FMS (View Message)

In the mobile app the incident will pass to the resolved incidents too.



Real example

To have a more real vision of the usage of the integration, we are going to do a summary of the steps of a real example.

We **create a service on PagerDuty for each kind of alert.**

In this example:

- Communications alert
- System alert

We can **configure a different group of users** that will receive the alerts for each service.

We **create one alert action for each service in our Pandora FMS console** using the API keys given by PagerDuty.

We **configure the desired template with recovery conditions.** For example, the Critical conditions template. In this case, we will left the Field 10 (description of the alert) in blank to use a different message for each action, because the actions will be used for different purposes.

We **create alerts with this template in our monitors**, associating the communication or system alert actions depending on the type of module.

For example for CPU, memory or processes monitors we will assign the system alert action. In the other hand, for the router interfaces checks or host pings, we will assign the communication alert.

In this way, **the alerts of each monitor will be notified to the appropriate people.**

FAQ

What happen when user make changes on PagerDuty incidents?

In PagerDuty is possible to add comments to the incidents, set its as acknowledged or resolved (closed).

Comment the incident or acknowledged its doesn't affect to the integration. The new triggered alerts from Pandora FMS will append the information to the incident history.

However if you mark an incident as resolved in PagerDuty, this incident will be closed and when the same alert is fired again in Pandora FMS, a new incident will be created.

The Pandora FMS alert is fired but it doesn't create anything on PagerDuty

Possible causes:

- **Bad alert configuration:** Be sure that your Pandora FMS command, action and template is properly configured using this step by step guide.
- **SSL dependences:** The integration script makes a call to the PagerDuty API using HTTPS. To perform it is necessary the Perl dependence **IO::Socket::SSL** installed on the system. Maybe you can edit the integration script and change the call protocol to HTTP, but it's not recommended.

Incidents triggered on PagerDuty but not resolved when recovering

Possible causes:

- **Bad alert configuration:** You need to configure the alert in Pandora FMS with a template where the recovery is enabled and the field9 configured with 'resolve'. More information in the step by step guide.
- **Error creating auxiliary files:** The integration script maintains a list with the match up between the Pandora FMS and PagerDuty's IDs. This hidden files are created by default at the same path of the script.

This path may not be writable if you have this problem.

You need to change it editing the integration script and setting a writable path in the variable \$matchup_dir at the starting of the file.

I need to debug the script execution

The alert execution debugging on Pandora FMS server is in the level 8 of verbosity. You need to set at least this level in the verbosity token of the pandora_server.conf on your system (located at /etc/pandora/) and restart the server (/etc/pandora_server restart). Then you will be able to read the executing command on the pandora_server.log file (located at /var/log).

Obtaining the executed command, you can do tests manually on command line from your server shell and get more feedback of the execution results.