

Saagie[©]



Platform Quick Guide

— v0.10 Luna-9 —

Table of contents ^{1/2}

Luna-9 : New release, new features	4
Manager	5
Your Platform Manager	5
Available capsules	6
Job status	7
Create a job	8
Create a job - Scheduling	9
Access options within a job overview	9
Search for a job	10
Delete a job	10
Run a job	10
Display logs	11
Edit settings	12
Upgrade a job	13
Manage versions of a job	14
Duplicate a job	15
Create a Sqoop job	16
Create a Talend job	17
Create a Java/Scala job	18
Create a R job	19
Create a Spark job	20
Create a Datascience Notebook	21
Access a Datascience Notebook	21
Create a smart app using Docker	22
Open a smart app	22
Explore your datalake with HUE	23
Show connection details	23

Table of contents 2/2

Promote jobs between platforms	24
Switch between platforms	25
ACL with Sentry	26
Environment variables	27
Access the environment variables display	27
Create a variable	28
Delete a variable	28
Search for a variable	29
Edit a variable	29
Platform Manager API	30
Network Architecture	31
Report a bug	32

Luna-9 : new release, new features

History:

Luna 9, internal designation Ye-6 No.13, was an unmanned space mission of the Soviet Union's Luna programme. On the 3 February 1966 the Luna 9 spacecraft became the first spacecraft to achieve a soft landing on the Moon, or any planetary body other than Earth, and to transmit photographic data to Earth from the surface of another planetary body.

New features:

- New “Revolution” UI
- Create a job from another
- New panel to display capsules connection details
- Set resources (CPU, RAM, DISK) for all kind of jobs

Capsules:

- Datalake: Impala - upgrade to version 2.5
- Datalake: Drill 1.7
- Datamart: MySQL 5.7

Your Platform Manager

<https://manager.prod.saagie.io/>

The screenshot displays the Saagie Platform Manager interface. At the top left, it says "Saagie Manager" and "PLATFORM My Platform". A user profile "victor" is visible in the top right. The main content is divided into three columns: "Extraction jobs", "Processing jobs", and "Smart apps".

- Extraction jobs:** Lists categories like Docker (0 jobs), SQOOP (29 jobs), Talend (1 job), Java/Scala (4 jobs), R (0 jobs), and Spark (3 jobs).
- Processing jobs:** Features a "Datascience notebook" section with 4 jobs (R test, spark notebook, R notebook, Notebook Python) and a "Talend" section with 7 jobs.
- Smart apps:** Lists Docker (4 jobs) with specific applications like Saagie Present Up2Date, caravel, shiny-roxy1, and Wetty.

At the bottom, there are two service sections: "Datalake services" (Impala, Hive, HDFS, Explore datalake, Drill) and "Datamart services" (Mongo).

All rights reserved Saagie - v0.10.0 (build 376)

Available capsules

Extract



Talend

Datalake



Processing



Datascience
with  jupyter

Datamart



Dataviz







Job status

▶  Hello World ▼ 

▶  Hello World ▼ 

▶  Hello World ▼ 

  Hello World ▼ 

  Hello World ▼ 

▶  Hello World ▼ 

1. Job never launched

2. Job successful

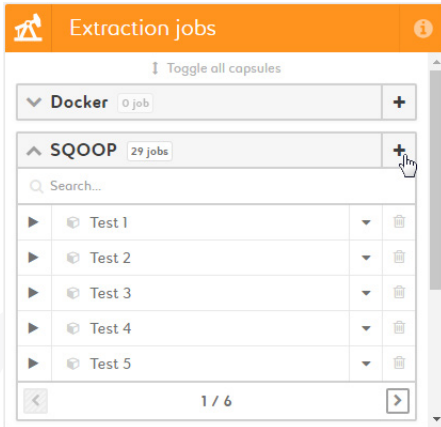
3. Job failed

4. Job processing

5. Job pending

6. Job killed

Create a job



1. Click on the “+” symbol next to the capsule name

Job name Name required

2. Name your job (required)

Email

An email will be sent to this email if the job fails.

3. Enter an email address to be alerted if and when the job fails

Run type

Manual Scheduled

4. Run your job manually or set up a schedule

Create a job - Scheduling

Your job starts the...

Play the job...

Delay between jobs:

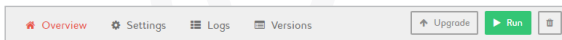
Year(s)	Month(s)	Day(s)	Hour(s)	Minute(s)	Second(s)
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

A retry on failure is done during:

Year(s)	Month(s)	Day(s)	Hour(s)	Minute(s)	Second(s)
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

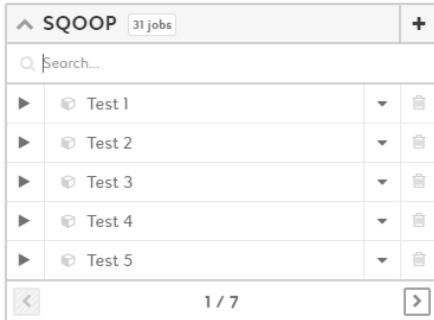
1. Select the date at which the job will start
2. Select how many times the job will play
3. Select a delay between each jobs (required)
4. Set up timing for a retry in case the job fails

Access options within a job overview



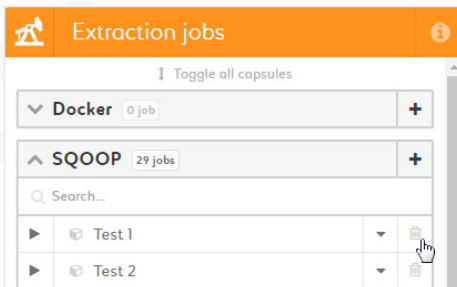
1. Select the action you want to perform
2. If in mobile view, click on the "Actions" button located in the top bar of a job overview

Search for a job



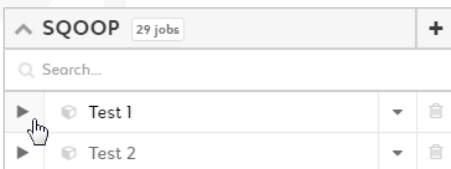
1. Click on the search box
2. Type the name of the job you are searching for

Delete a job



1. Click on the "bin" icon next to a job
2. Confirm your action by clicking on the "Delete" button

Run a job



1. Click on the "play" icon
2. To stop your job, click on the "stop" icon

Display logs 1/2

SQOOP		29 jobs		
Search...				
▶	🔄	Test 1	▼	🗑️
▶	🔄	Test 2	▼	🗑️
▶	🔄	Test 3	▼	🗑️
▶	🔄	Test 4	▼	🗑️
▶	🔄	Test 5	▼	🗑️

1. Click on a job to display the logs screen

Extraction / SQOOP
Hello world Actions

Job logs

Refresh

- SUCCESS
11/08/2016 15:50:31 →
- FAILED
11/08/2016 15:50:09 →
- SUCCESS
11/08/2016 15:49:48 →
- SUCCESS
11/08/2016 15:49:16 →

2. Select which version of the job you want to display logs for

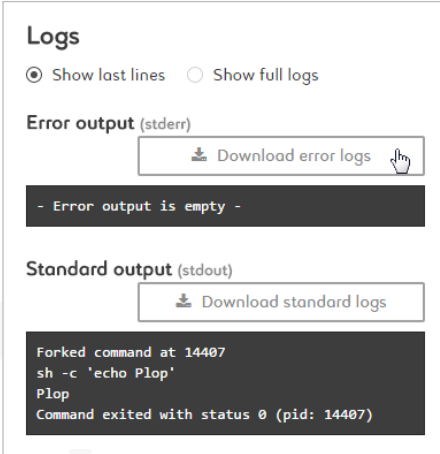
Job execution SUCCESS

Start date time: 11/08/2016 15:49:48 - End date time: 11/08/2016 15:49:48 - Duration: Less than 1 sec.

Job version: 2 [Go to this version](#)

3. Click on “Go to this version” to rollback to a specific version of a job

Display logs 2/2



Logs

Show last lines Show full logs

Error output (stderr)

[Download error logs](#)

- Error output is empty -

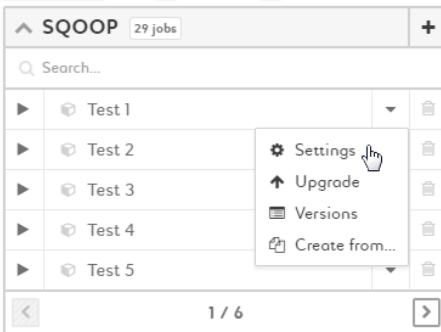
Standard output (stdout)

[Download standard logs](#)

```
Forked command at 14407
sh -c 'echo Plop'
Plop
Command exited with status 0 (pid: 14407)
```

4. Select between only showing the last lines of the logs or full logs
5. Click on “Download error/standard logs” to download your logs or read them through the online viewer

Edit settings



^ SQOOP 29 jobs +

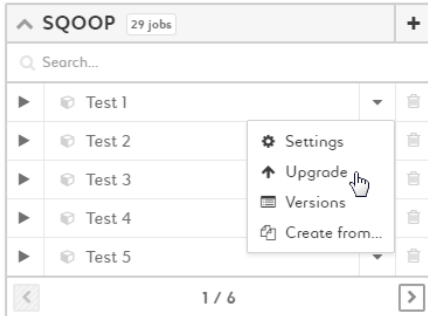
Search...

▶	Test 1	⌵	🗑️
▶	Test 2	⚙️	🗑️
▶	Test 3	⬆️	🗑️
▶	Test 4	📄	🗑️
▶	Test 5	🏠	🗑️

1 / 6

1. Click on the arrow to display the options dropdown menu
2. Click on “Settings” to edit settings for a specific job
3. Edit the email address used for alerts and the run type (manual or scheduled)

Upgrade a job



1. Click on the arrow to display the options dropdown menu
2. Click on “Upgrade” to access the job upgrade display

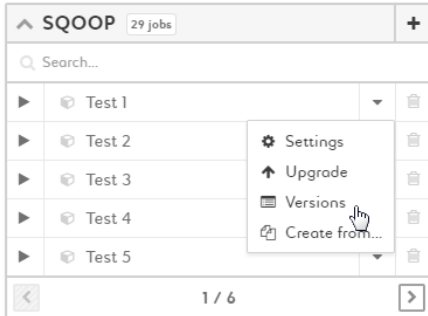
Create a new version and upgrade the job to this version.

Command to launch job
Linux shell command line to launch the job. You can use [environment variables](#) here.

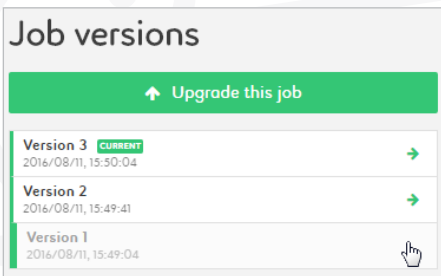
```
echo "plop"
```

3. Edit your command
4. Click on “Upgrade to new version” to set this version as the current version of your job

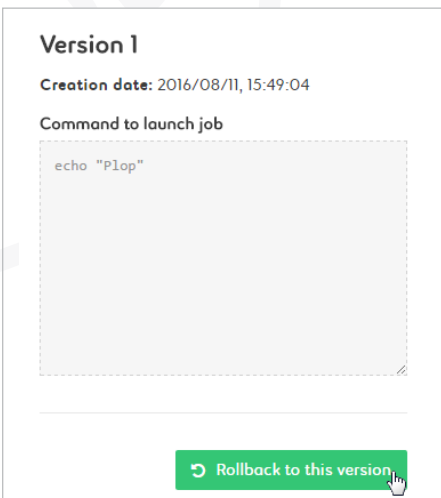
Manage versions of a job



1. Click on the arrow to display the options dropdown menu
2. Click on “Versions” to access the job versions display

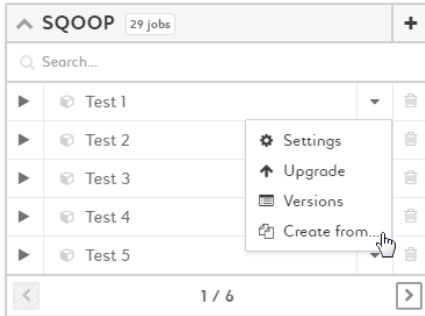


3. Select which version of the job you want to display



4. Check the details of the version and click on "Rollback to this version" if you want to set this version as the current version of the job
5. You can download the package of the job if one has been uploaded

Duplicate a job



1. Click on the arrow to display the options dropdown menu
2. Click on “Create from...” to duplicate the selected job
3. Set up your job as explained in the "Create a job" section of this guide

Create a Sqoop job

Job name **Nome required**

Command to launch job
Linux shell command line to launch the job. You can use [environment variables](#) here.

```
#!/bin/bash
# https://sqoop.apache.org/docs/1.4.0-incubating/SqoopUserGuide.html
# This script fetch the selected tables
driver=xxx #sqlserver
ip="x.x.x.x"
username="xxx"
password="xxxx"
database="xxxx"
hdfsdest=/user/creativedata
```

CPU

Memory (MB)

Disk (MB)

Email

An email will be send to this email if the job fails.

Run type

Manual Scheduled

1. Name your job
2. Type your command using the ready-to-use template provided. Add environment variables as \$VARIABLE or access your variable library by clicking the link
3. Set your CPU, memory and disk settings
4. Enter an email address to be alerted if and when the job fails
5. Run your job manually or set up a schedule

Create a Talend job

Job name

Package File or url required

Upload URL

Example : jar, tar.gz or .zip talend job.

Command to launch job

Linux shell command line to launch the job. You can use [environment variables](#) here.

```
sh {file} arg1 arg2
```

CPU

Memory (MB)

Disk (MB)

Email

An email will be send to this email if the job fails.

Run type

Manual Scheduled

1. Name your job
2. Add a package by uploading a zip of by entering a URL
3. Type your command. {file} is a parameter (don't change it). Customize arg1, arg2, etc. Add environment variables as \$VARIABLE or access your variable library by clicking the link
4. Set your CPU, memory and disk settings
5. Enter an email address to be alerted if and when the job fails
6. Run your job manually or set up a schedule

Create a Java/Scala job

Job name Name required

Package File or url required

Upload URL

Example : jar, tar.gz or zip talend job.

Command to launch job

Linux shell command line to launch the job. You can use [environment variables](#) here.

```
java -jar {file} arg1 arg2
```

Choose the language version:

JAVA 8 (recommended) ▾

CPU

Memory (MB)
Disk (MB)
Email

An email will be sent to this email if the job fails.

Run type

Manual Scheduled

1. Name your job
2. Add a package by uploading a jar or by entering a URL
3. Type your command. {file} is a parameter (don't change it). Customize arg1, arg2, etc. Add environment variables as \$VARIABLE or access your variable library by clicking the link
4. Choose the language version: JAVA 8 (recommended) or JAVA 7
5. Set your CPU, memory and disk settings
6. Enter an email address to be alerted if and when the job fails
7. Run your job manually or set up a schedule

Create a R job

Job name **Name required**

Package **File or url required**

Upload URL

Example : jar, targz or .zip talend job.

Command to launch job

Linux shell command line to launch the job. You can use [environment variables](#) here.

```
Rscript {file} arg1 arg2
```

CPU

Memory (MB)

Disk (MB)

Email

An email will be send to this email if the job fails.

Run type

Manual Scheduled

1. Name your job
2. Add a package by uploading a R file or by entering a URL
3. Type your command. {file} is a parameter (don't change it). Customize arg1, arg2, etc. Add environment variables as \$VARIABLE or access your variable library by clicking the link
4. Set your CPU, memory and disk settings
5. Enter an email address to be alerted if and when the job fails
6. Run your job manually or set up a schedule

Create a Spark job

Job name **Name required**

Package **File or url required**

Upload URL

Example : jar, tar.gz or .zip talend job.

Command to launch job

Linux shell command line to launch the job. You can use [environment variables](#) here.

```
spark-submit {driver_options} --class=MyClass {file}
arg1 arg2
```

Choose the language version:

SPARK 1.6.1 (recommended) ▾

CPU

Memory (MB)

Disk (MB)

Email

An email will be send to this email if the job fails.

Streaming

Enable long job
Choose this option for streaming process.

Run type

Manual Scheduled

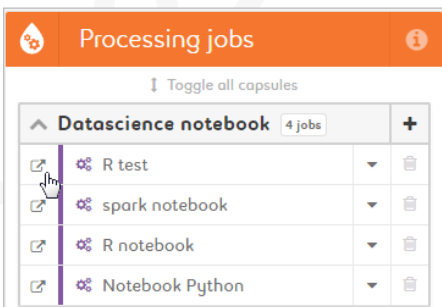
1. Name your job
2. Add a package by uploading a jar of by entering a URL
3. Type your command. {file} and {driver_options} are parameters (don't change it). Customize arg1, arg2, etc. Add environment variables as \$VARIABLE or access your variable library by clicking the link
4. Choose the language version: Spark 1.6.1 (recommended) or Spark 1.5.2
5. Set your CPU, memory and disk settings
6. Enter an email address to be alerted if and when the job fails
7. Activate the streaming option
8. Run your job manually or set up a schedule

Create a Datascience Notebook

The screenshot shows a 'Create job' dialog box. At the top, there is a 'Job name' field with a red 'Name required' error message. Below it is a 'Choose the notebook:' dropdown menu. The dropdown is open, showing a list of notebook options: 'Python 2 et 3' (selected), 'R', 'Scala 2.11.7 / Spark 1.6.1', 'Scala 2.11.7 / Spark 1.5.2', 'Ruby', 'Haskell', and 'Julia'. Below the dropdown is a 'Disk (MB)' field with the value '1024'. At the bottom, there are two buttons: 'Cancel' and '+ Create job'.

1. Name your job
2. Choose the notebook: Python 2 & 3, R, Scala 2.11.7 / Spark 1.6.1, Scala 2.11.7 / Spark 1.5.2, Ruby, Haskell, Julia
3. Set your CPU, memory and disk settings

Access a Datascience Notebook



1. Click on the "Open in new window" icon next to a notebook

Create a smart app using Docker

Job name **Name required**

Docker image URL **Uri required**

Public image name or repository_url/image

Container port

CPU

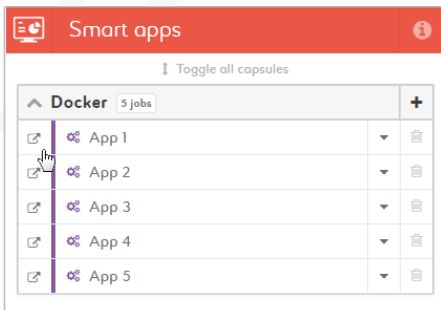
Memory (MB)

Disk (MB)

Cancel + Create job

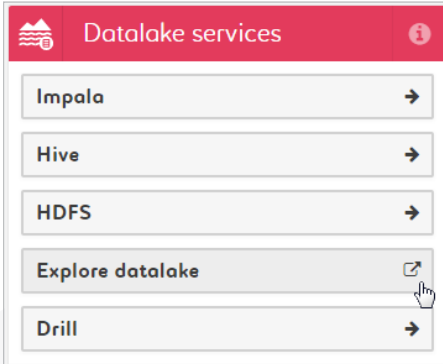
1. Name your job
2. Type the Docker image URL (only public Docker images are available)
3. Enter a container port (HTTP(s) only, TCP/UDP not supported)
4. Set your CPU, memory and disk settings depending on your app/Docker image

Open a smart app



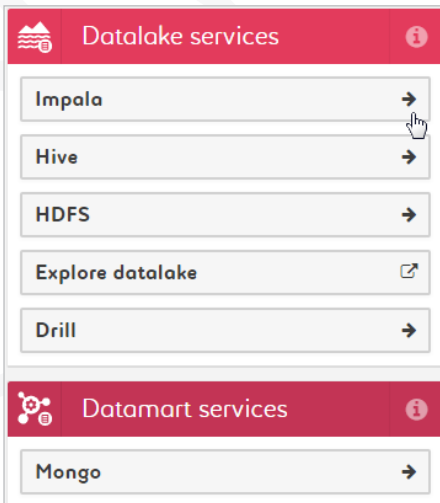
1. Click on the "Open in new window" icon next to a smart app

Explore your datalake with HUE



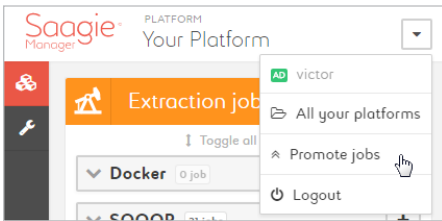
1. Click on the "Explore datalake" button in the "Datalake services" module
2. Explore your datalake in the now opened HUE interface

Show connection details



1. Click on any datalake or datamart capsule to display its connection details

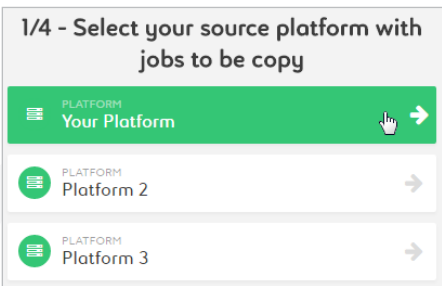
Promote jobs between platforms



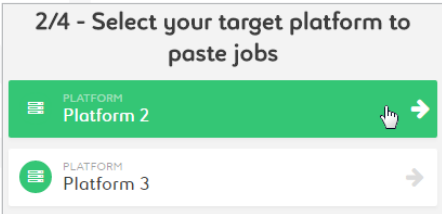
1. Click on the arrow at the right of the manager's top bar

2. Click on "Promote jobs"

3. Select the platform you want to copy jobs from



4. Select the platform you want to copy jobs to

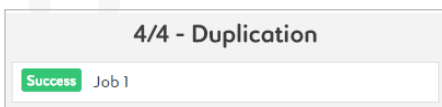


5. Select the jobs you want to copy

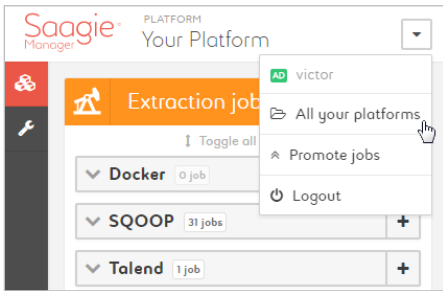
6. Confirm copy at the bottom of the page



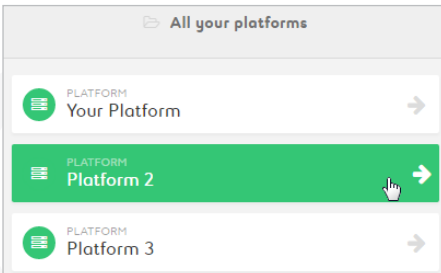
7. Check the status of your copy



Switch between platforms



1. Click on the arrow at the right of the manager's top bar
2. Click on "All your platforms"



3. Select the platform you want to switch to

ACL with Sentry

Option on demand

```
//create role
```

```
CREATE ROLE loblanalyst;
```

```
// assign role to group
```

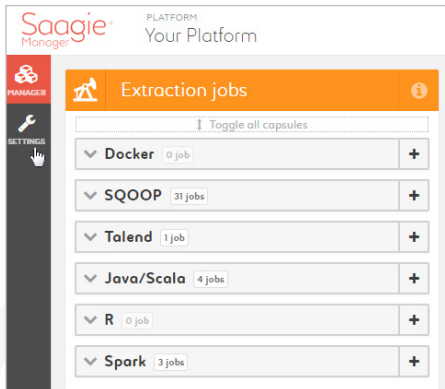
```
GRANT ROLE lobadministrator TO GROUP lobladm WITH  
GRANT OPTION;
```

```
// manage access
```

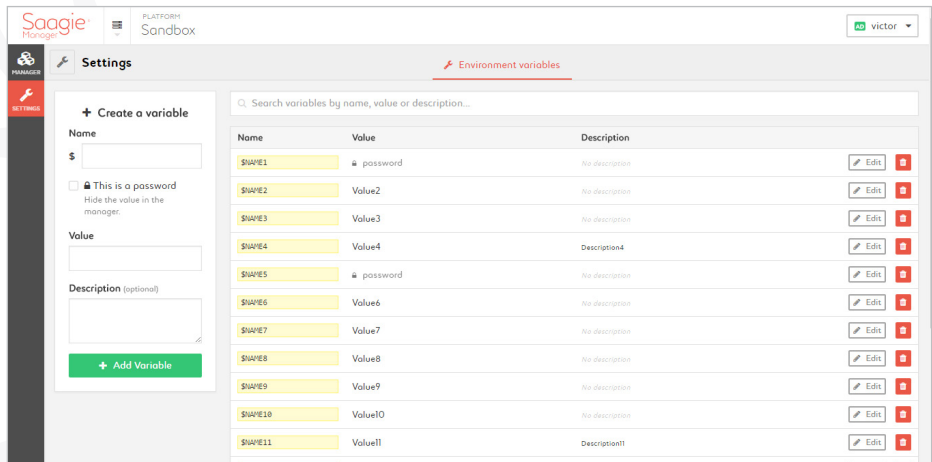
```
GRANT ALL ON DATABASE lob1 TO role lobadministrator;
```

Environment variables 1/3

Access the Environment variables display



1. From the Platform Manager, click on the "Settings" button in the upper-left corner



Environment variables 2/3

Create a variable

+ Create a variable

Name

\$

🔒 This is a password
Hide the value in the manager.

Value

Description (optional)

+ Add Variable

1. Name your variable
2. Select "This is a password" to hide the value of your variable in the manager
3. Enter your value
4. Describe your value for documentation purposes
5. Click the "Add variable" button when finished

Delete a variable

description...

Description	Actions
No description	<input type="checkbox"/> <input type="checkbox"/>
No description	<input type="checkbox"/> <input type="checkbox"/>
Description 3	<input type="checkbox"/> Edit <input type="checkbox"/>
No description	<input type="checkbox"/> Edit <input type="checkbox"/>
No description	<input type="checkbox"/> Edit <input type="checkbox"/>
No description	<input type="checkbox"/> Edit <input type="checkbox"/>
No description	<input type="checkbox"/> Edit <input type="checkbox"/>
No description	<input type="checkbox"/> Edit <input type="checkbox"/>
No description	<input type="checkbox"/> Edit <input type="checkbox"/>
No description	<input type="checkbox"/> Edit <input type="checkbox"/>

Are you sure?
You are about to remove the variable "PRESENT_METASTORE_THRIFT_URL".

1. Click on the "bin" icon next to a variable
2. Confirm your action by clicking on the "Remove" button

Environment variables 3/3

Search for a variable

Name	Value	Desc
\$NAME1	password	No de
\$NAME2	Value2	No de
\$NAME3	Value3	No de
\$NAME4	Value4	Descri
\$NAME5	password	No de

1. Click on the search box located on top of the list of variables
2. Search variables by name, value or description

Edit a variable

Description	Edit	Delete
No description	Edit	Delete
No description	Edit	Delete
Description 3	Edit	Delete
No description	Edit	Delete
No description	Edit	Delete
No description	Edit	Delete
No description	Edit	Delete

1. Click on the "edit" button next to a variable
2. Edit the name, value and description fields
3. Save your edits by clicking on the "Save" button

Platform Manager API

API

Everything you can do through the UI is available through an API:

- You can automatize deployment with Jenkins
- You can gather job status in your centralized monitoring system

Documentation is available here:

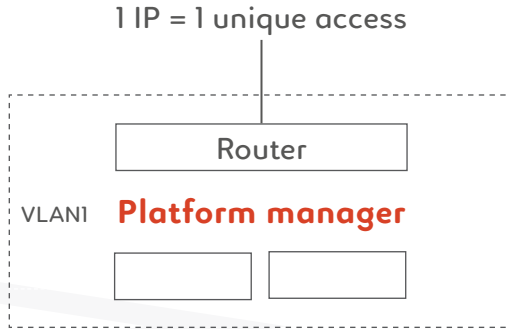
<https://manager.prod.saagie.io/api/doc>

<https://manager.prod.saagie.io/api/doc/admin>

(or with your private Cloud URL)

Network architecture

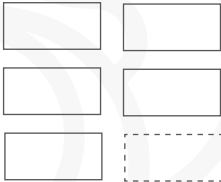
Managed version



VLAN1 can access to VLAN2, 3 & 4 but not in the other direction. VLAN2, 3 & 4 are silos

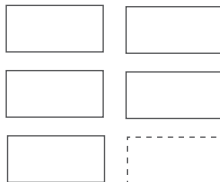
VLAN2

DEV



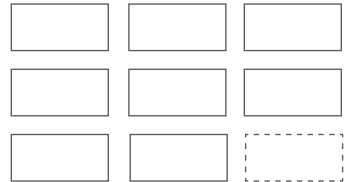
VLAN3

TEST



VLAN4

PROD



Report a bug

bug-platform@saagie.com

1. Send an email at that address explaining the bug you encountered
2. Please attach relevant files and screenshots