Simple OpenStack Monitoring Tool

User Documentation

Version 1.1

Team Name: Oceans11

Team Members:

- Tarun Aluguri.
- Nikhil Reddy Araga.
- J N S Sri Harsha Vardhan Kamisetty.
- Prathisrihas Reddy Konduru.
- Sai Anoop Nadella.
- Rohit Pothuraju.
- Dilip Renkila.
- Venkat Sivendra Tipirisetty.
- Sai Bhargava Ravi Teja Vedantam.
- S. Sai Srinivas Jayapala Vemula.
- Rahul Vudutha

I) PREFACE:

The main concern of the project is to develop a simple and intuitive web based drill down GUI that provides an overview of an OpenStack environment. The tool monitors the OpenStack services Nova, Neutron, Cinder, Swift, Keystone, Glance and Heat existing on the nodes. It displays the status of monitored nodes and allows the user to restart the services on demand. This guide gives the information about usage of the monitoring tool. This is second version of the document (version 1.1).

Release version 1.1

- Made changes regarding all features and final outputs

Release version 1.0

- Initial release

II) GLOSSARY AND ABBREVIATIONS:

1. **GUI: Graphical User Interface:**
   A **GUI** is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces.

2. **PHP: Hypertext Pre-processor**:

   PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language.

3. **PERL: Practical Extraction and Report Language**
   Perl is a general-purpose programming language developed for text manipulation and also used for tasks including system administration, web development, network programming, GUI development, and etc.

4. **HTTP: Hyper Text Transfer Protocol**

   The Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems.

5. **SSH: Secure Shell:**

   Secure shell is a cryptographic (encrypted) network_protocol for initiating text-based shell sessions on remote machines in a secure way.

III) PREREQUISITES

- Knowledge about Linux and running commands in terminal for installing prerequisite packages for the running of the tool
- Knowledge about using a web browser
- Knowledge about HTTP and SSH
- Basic knowledge about MySQL-server

IV) FEAUTURES OF THE TOOL:

The tool consists of a Frontend, Database and a Backend. The Frontend consists of web pages written in PHP and HTML used to create an intuitive, drill down GUI that provides a user-friendly interface.

The Database is used by the Frontend and Backend. The tool uses MySQL database management tool to store and manage data. A database is created with a name specified in db.conf file provided with the tool. By default, the database name is Oceans11 and can be changed manually. The database consists of tables required for storing user credentials as well as status and uptime data retrieved by the backend

The Backend consists of a series of perl files that need to be running in the background in order to ensure the proper functioning of the tool. Firstly, a SSH connection must be established between the tool and the OpenStack node. For this, the tool is provided with a keys folder containing private key and public needed to establish the connection. The passphrase is provided in the db.conf file. After this, other perl files are used to retrieve status and uptime of each service continuously while the backend is connected to the OpenStack node. This data is inserted into the MySQL database into their respective tables which is later fetched by the Frontend to be displayed on the web page

1. **User Registration:**

   A default user is created using credentials mentioned in db.conf under the variables $user and $pass. Using these credentials, he/she can enter the tool dashboard through the login page. After logging in, the user can enter Accounts tab in the navigation panel on the left hand side of the web page. After clicking on this tab, two options arise:

   i. Register new user
   ii. Change credentials

   When first option is selected, username and password are requested. After these necessary credentials are entered and submitted the user table in the database is updated and a new user is created

   When the second option is selected, an existing account holder can change his/her password

2. **Status and Uptime of each service:**

   After logging in, the user is brought to the home page of the tool Dashboard. On the home page, information pertaining to the working of the tool is displayed along with information related with each OpenStack service that the tool monitors.

   On the right hand side of the web page, the service status panel contains list of services along with their current status. This data is fetched from the tables in the tool database. The backend continuously monitors the services and inserts their respective status and uptimes in the MySQL tables.

In the Statistics tab, Uptime of each service is displayed in D-HH:MM:SS format. The Uptime is updated automatically whenever a service status is altered.

3. **Restart of service:**

   After logging in, the navigation panel consists of a Monitor tab that must be selected in order to perform a RESTART operation.

   On clicking the Monitor tab, a table containing list of services are displayed along with a column containing number of restarts.

   Checkboxes are provided to select the services to be restarted. The user can simply select a single or multiple services and click restart to perform a restart operation

   When a restart of service is done, a change in status can be seen if a change occurs. The uptime of the service is automatically updated and can be seen in the Statistics tab

4. **Graphs on Uptime of each service**

   In the navigation panel, the Statistics tab shows the current Uptime of each service in D-HH:MM:SS format. A particular service must be selected amongst the list of services displayed. Upon clicking a particular service, the uptime of that service is displayed along with the uptime graph.

   The sub-services are displayed along the X-axis

   Uptime is displayed along the Y-axis in hours

   The graphs are generated by perl script based on data inserted in the MySQL tables by the backend.

5. **RESTful API:**

   The tool also provides a RESTful interface in order to export service data to a 3rd party. The Frontend contains a PHP file that retrieves data from the tool’s database and encodes the data in JSON format. This data is displayed on the web page upon GET request by the user.

Firstly, the Oceans11_rest_api php file must be entered in the address bar. Then, at the end of the address bar GET request must be entered like as follows

Oceans11_rest_api.php/?service=<service_name>

After this, status and uptime of each sub-service are fetched from the database and displayed on the web browser in JSON format.

V) USAGE GUIDE:

The tool requires a SSH connection to the OpenStack server to be run in the background continuously.

- This connection facilitates the retrieval of status and Uptime data from OpenStack server with the help of backend perl scripts.
- The user will also be given db.conf file containing database parameters which needs to be executed before starting.
- The user can change the name of the database manually. The file also contains ssh hostname and passphrase needed for establishment of the ssh connection
- The user must enter the login page of the tool. Username and Password must be entered to gain access to the Dashboard

Username: Oceans11 Password: Oceans11

- These are the default user credentials used to initially login the tool login page
- The main sections of the tool Dashboard are:

1. Home
2. Monitor
3. Statistics
4. Accounts

- After entering, new users can be entered in the Accounts section
- Upon entering the dashboard, the user will be shown a complete list of services monitored by the tool in the home page.
- On the right side of the web page, the status of each service is shown, in the service status panel
- Whether the service is started/running or ERROR is displayed depending on the data entered in the MySQL database
- Whenever the service has stopped unexpectedly, on the web interface, the status will be shown in red text as ERROR

- In the Monitor tab, list of services are displayed along with a column for number of restarts. Single/Multiple services can be selected and restarted simultaneously by the user.
- Once the service is restarted, a change in status can be observed in status panel if a change occurs.
- The Uptime changes automatically when the restart operation is performed

- In the Statistics tab, the Uptime of each service is displayed in D-HH:MM:SS format in a table

- All the services being monitored are displayed on a horizontal bar

- A service can be selected to show an Uptime graph pertaining to that service.

VI) FINAL OUTPUTS:

The user is shown the current status of every OpenStack service. The status of a stopped service is shown in red colour text as ERROR.

The uptime of each service is also shown and updated automatically by the backend

When the user clicks on RESTART button in the web page. The service is restarted and Uptime is updated immediately.

Also, graph is plotted for Uptime of every service. This graph is changes every time the service is restarted.
REST API is provided in order to export service data to a 3rd party. The user can perform GET request in the web browser and view the status and uptime in JSON format

In conclusion, the Tool's outputs contains mainly six parts

- It shows about the status of services on open stack node .
- It shows about the service uptime in D-HH-MM-SS .
- It shows the graphs between uptime to the service .
- It shows the number of restarts done for a service when failure occurs on a service.
- It also provides an option for the user to add another user and also to change his own credentials.
- It provides a RESTful interface in order to export OpenStack service data to a 3rd party