

Simple OpenStack monitoring tool

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. This is revised version of the tool (version 1.2).

A. Release Version 1.3 on 2015-05-5:

- Implementation milestone has been included and Time Plane flow charts have been changed in the Time Plan section (See Sections VI i.e., Time plan).

B. Release Version 1.2 on 2015-04-27:

- OpenStack services being monitored are included in the preface. Made changes regarding the statistics that will be collected and scope of the project. (See Sections III and VI i.e., Background and Time plan respectively).

C. Release Version 1.1 on 2015-04-20:

- Made changes regarding the performance monitoring parameters. (See Sections III, IV and V i.e., Background, Proposed Solution and Limitations respectively).

D. Release Version 1.0 on 2015-04-13:

- Initial Release

In the remainder of the document, section II describes briefly about the basic abbreviations used in this document. Section III emphasizes on the problem issues of the OpenStack services faced by the customer and how the tool addresses those problems. Section IV describes the functionality of the proposed tool. Section V specifies the limitations of the project. The last section describes the time plan for the project.

II. GLOSSARY AND ABBREVIATIONS:

1) API: Application Programming Interface:

An **API** is a set of routines, protocols, and tools for building software applications.

2) 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.

3) CPU: Central Processing Unit

III. BACKGROUND:

The stack deployment company (the Customer) contains certain nodes that allow implementation of OpenStack services. These services need to be monitored periodically as they may encounter certain issues over time. We address this problem by taking help of certain restoration methods to rectify them and using historical data to analyse periodicity of failures. We develop a tool to monitor the status (Started/Stopped/Failed) of the services on the nodes and to graphically represent the statistics (Uptime, Downtime, Number of Failures, Number of Restarts) collected about the services at two months' time scale.

IV. PROPOSED SOLUTION:

The customer will be provided with a tool that monitors the status of the OpenStack services. The tool we develop includes a web GUI that facilitates the user to monitor OpenStack services and to collect the historical data on Uptime, Accessibility of services and detects periodicity of failures through a RESTful interface.

V. LIMITATIONS:

The tool is limited to the monitoring of OpenStack services. The customer is limited to restart the service via GUI soon after the service is down irrespective of what caused it. If an issue persists even after restarting the service, it can't be handled from the web interface and in-depth troubleshooting is required.

VI. TIME PLAN:

Milestones:

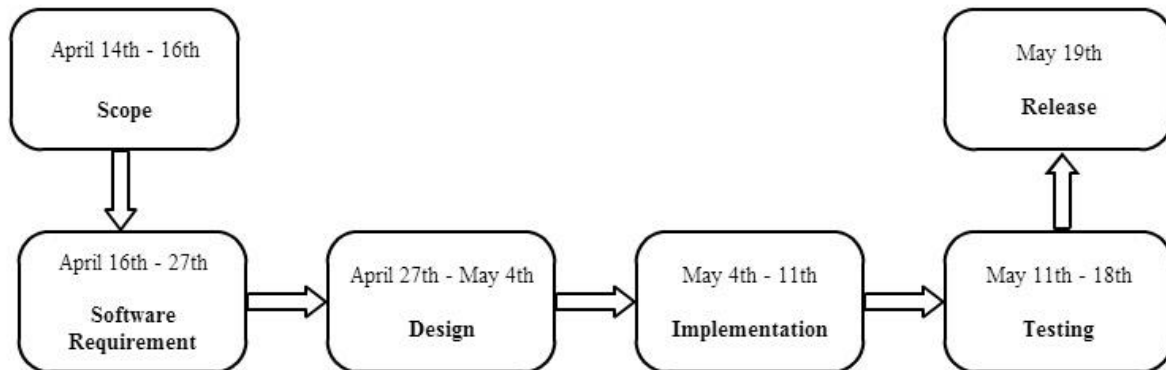


Fig. 1 Time Plan flow chart

Toll Gates:

1. 13-04-2015: Project proposal
2. 27-04-2015: Project Specification
3. 27-04-2015: Software Requirements Specifications
4. 04-05-2015: Design document
5. 11-05-2015: Acceptance test plan
6. 16-05-2015: Project Demo
7. 18-05-2015: Project Submission and Documentation

		Name	Duration	Start	Finish	Notes
1		☐ Scope	3 days	4/14/15 8:00 AM	4/16/15 5:00 PM	
2		OpenStack basics	2 days	4/14/15 8:00 AM	4/15/15 5:00 PM	
3		Services to monitor	1 day	4/16/15 8:00 AM	4/16/15 5:00 PM	
4		☐ Software Requirement	8 days	4/16/15 8:00 AM	4/27/15 5:00 PM	
5		Openstack Environment	3 days	4/16/15 8:00 AM	4/20/15 5:00 PM	
6		Programming Languages	3 days	4/21/15 8:00 AM	4/23/15 5:00 PM	
7		Database management tools	2 days	4/24/15 8:00 AM	4/27/15 5:00 PM	
8		☐ Design	6 days?	4/27/15 8:00 AM	5/4/15 5:00 PM	
9		☐ Monitoring Tool(Backend) scripts	4 days	4/27/15 8:00 AM	4/30/15 5:00 PM	prko15, dire15, veti15, taal15
10		Script	3 days	4/27/15 8:00 AM	4/29/15 5:00 PM	prko15, dire15, save15
11		Database-MySQL	2 days	4/29/15 8:00 AM	4/30/15 5:00 PM	veti15, taal15
12		☐ Web GUI (Frontend)	3 days?	4/30/15 8:00 AM	5/4/15 5:00 PM	ropo15, jnka15, niar15, sanc15
13		Script	2 days	4/30/15 8:00 AM	5/1/15 5:00 PM	ropo15, jnka15, seve15
14		Database-RRDTool	2 days?	5/1/15 8:00 AM	5/4/15 5:00 PM	niar15, sanc15
15		Implementation	6 days?	5/4/15 8:00 AM	5/11/15 5:00 PM	save15, seve15
16		Testing	6 days	5/11/15 8:00 AM	5/18/15 5:00 PM	rava15
17		Release	1 day	5/19/15 8:00 AM	5/19/15 5:00 PM	

Fig. 2 Time Plan flow chart

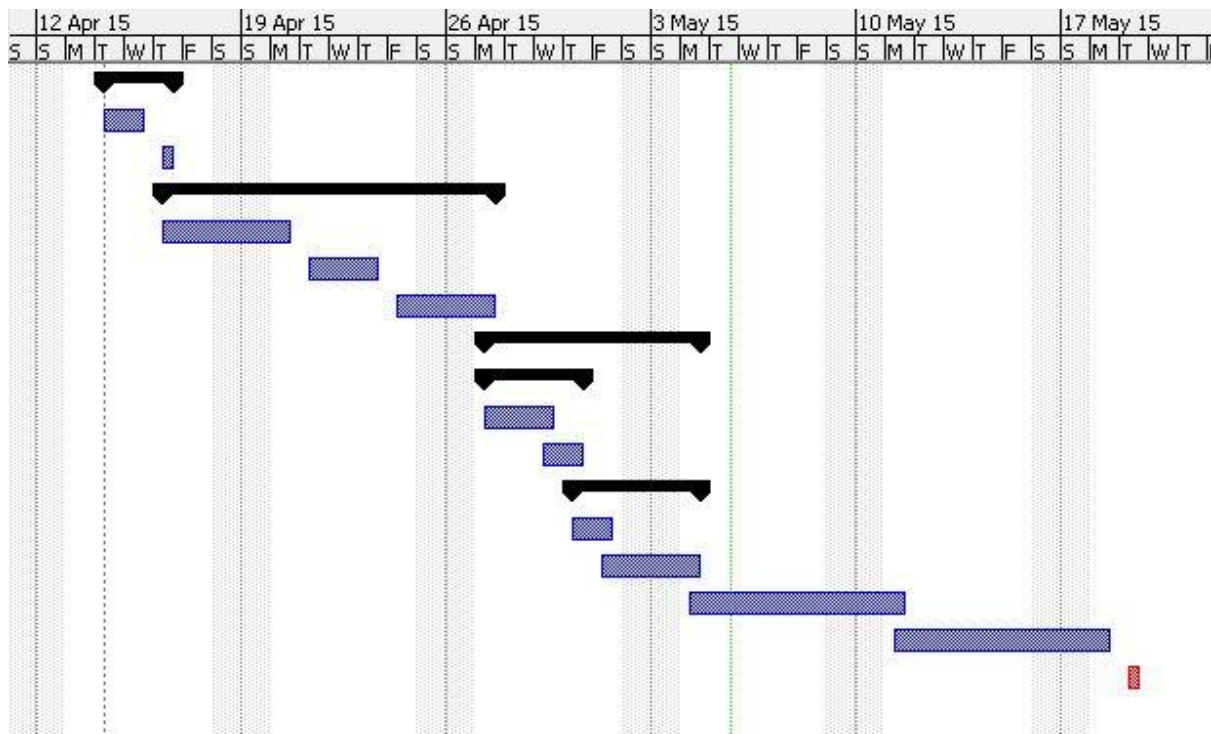


Fig. 3 Time Plan flow chart

(a) Scope:

The monitoring tool checks the status of various OpenStack services like Keystone, Glance, Nova, Neutron, Cinder, Swift and Ceilometer running on the nodes.

The tool can only monitor the OpenStack services but cannot control the functionality/operations of the services such as networking functionality of neutron or provisioning of new virtual machines.

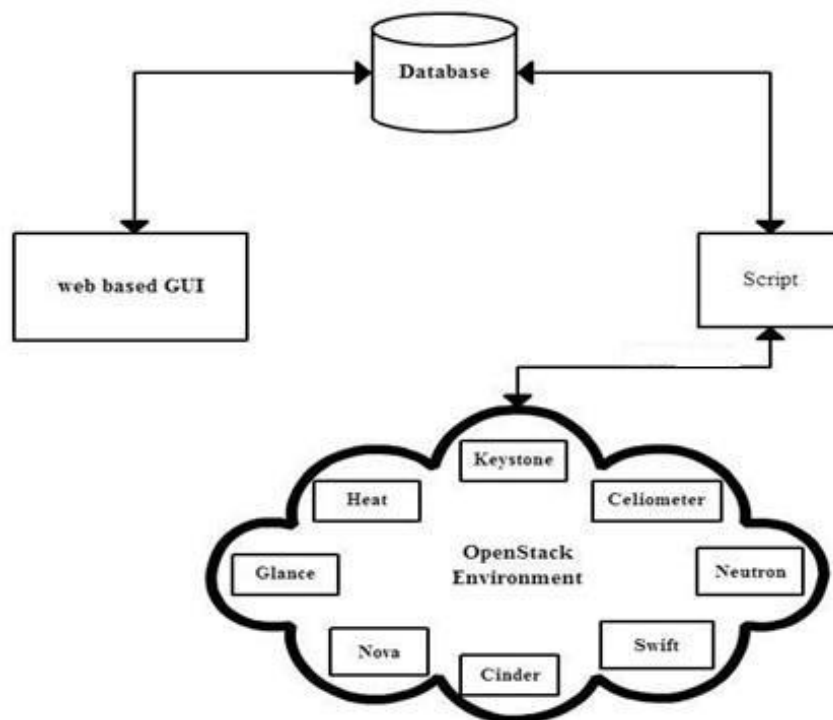
(b) Design:

Fig.4: Structured Layout

The project layout is designed to have clear understanding of the approach to achieve the required goal. It defines the interaction between the monitoring tool and OpenStack environment via RESTful API.

- **Web based GUI / Dashboard:** Provides a simple and intuitive user interface through which the user can view the status of the services (working or failed) and restart them if needed. It also includes a provision to represent the status graphically.

- **Database:** Stores the status information like Uptime, Downtime, etc. retrieved from the OpenStack environment through the script.
- **Script:** The script extracts the service status information from the OpenStack environment and stores it in the database.
- **OpenStack Environment:** Contains the nodes running OpenStack services Nova, Neutron, Cinder, etc.

(c) Testing:

Testing of the tool includes verification and validation. Tests are done to confirm that the tool has met all the specifications specified by the customer and the tool is ready for operational use.

(d) Release:

The monitoring tool which we developed is made available for the customer after all the specifications mentioned by the customer are met.