Simple OpenStack monitoring tool

Project Specifications

Version 1.4

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.

1) 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 the revised version of the document (version 1.4)

A. *Release version 1.4 on 2015-06-01 :*
   - Risks list is provided only in tabular format. (Section XI).

B. *Release version 1.3 on 2015-05-20:*
   - Changes made in tabular column under Risk management (Section XI).

C. *Release version 1.2 on 2015-05-14:*
   - Changes made in section 8 regarding version management using git (Section VIII).
   - Changes made in section quality management (Section X).
   - Changes made in risk management (Section XI).
   - Changes made in system release plan (Section XII).

D. *Release version 1.1 on 2015-05-06:*
   - Implementation milestone has been included and Time Plane flow charts have been changed in the Time Plan section (section VI).
   - Changes made project organization (section VII).
   - Made changes in configuration management (section VIII).
   - Changes made in progress tracking (section IX).
   - Made changes in Quality control (section X).
   - Made changes in Risk management (section XI).
   - Made date changes in Testing plan and packaging plan (section XII).
   - Added developer documentation section (section XIII).

E. *Release version 1.0 on 2015-04-27:*
   - Initial release

In the remainder of the document, Section II describes briefly about the basic abbreviations used in this document. Section III describes the background, specifying requirement of the customer and process implemented by the developers. Section IV describes the functionality of the proposed tool. Section V specifies the limitations of the project. Section VI dictates the time plan of the project, and schedule for product development. Section VII shows the project organization among the members of development team, work is broken down and each division assigned to different members of the team. Section VIII describes the Configuration Management which involves managing versions, system building and release management. Section IX shows Progress Tracking by assigning checkpoints. Section X gives Quality control measures and ensures effective execution of tool. Section XI defines Risk management. Section XII gives an estimate of System Release Plan, which involves testing, packaging and documentation plans. Lastly, Reference section cites references used for documentation and tool development.

2) 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

3) 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 of the services at two months' time scale.

4) 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.

5) 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.

6) TIME PLAN:

Milestones:

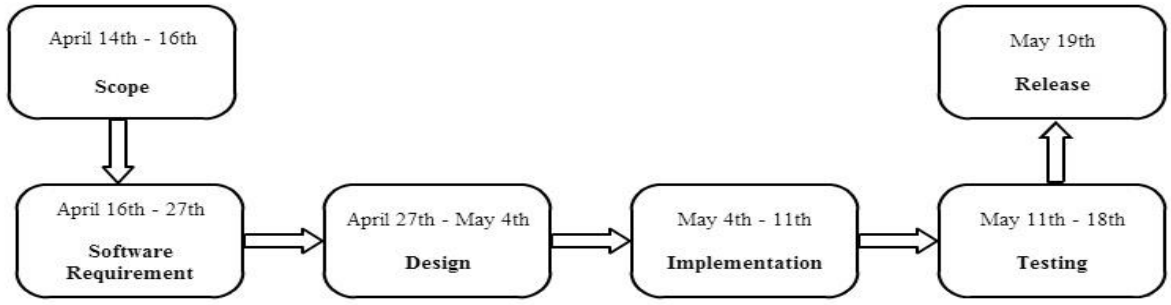

Fig. 1: Time Plan flow chart

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 Rquirement**	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

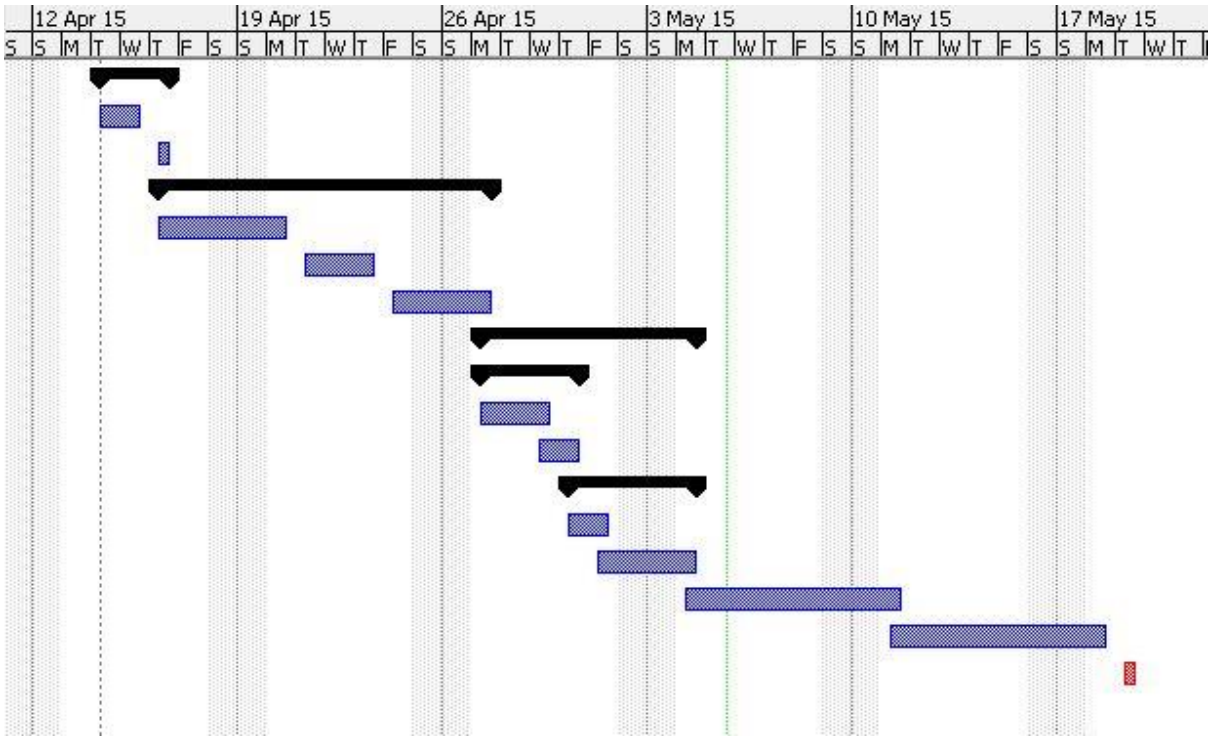

Fig. 3: Time Plan flow chart

(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:

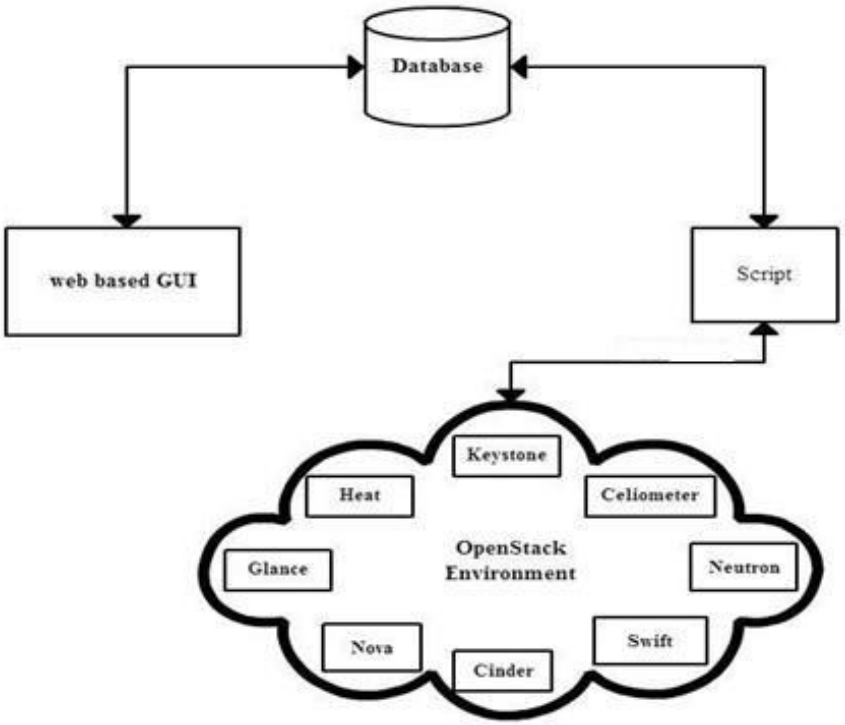

Fig.4: Structured Layout

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.

7) Project organization

WORK	TEAM MEMBER
1. Programming :	
a. Frontend :	J N S SRI HARSHA VARDHAN KAMISETTY, ROHIT POTHURAJU, RAHUL VUDUTHA, SAI BHARGAVA RAVI TEJA VEDANTAM.
b. Database :	VENKAT SIVENDRA TIPIRISETTY, TARUN ALUGURI, NIKHIL REDDY ARAGA.
c. Backend :	PRATHISRIHAS REDDY KONDURU, S. SAI SRINIVAS JAYAPALA VEMULA, SAI ANOOP NADELLA, DILIP RENKILA.
2. Testing :	SAI BHARAGAVA RAVI TEJA VEDANTAM, DILIP RENKILA.
3. Management :	NIKHIL REDDY ARAGA.
4. Documentation :	RAHUL VUDUTHA.

Table 1: work management

8) Configuration Management:

8.1 Version Management:

Identifiers are assigned to the different codes or documents while they are submitted to a version management system. These identifiers follow a specific format consisting of attribute stating the document or code type, followed by team name and Version number. The initial version of the document starts with version 1.0 and the number increments depending on the type of the changes made. For example, in the case of a major change the version changes to 2.0 and for minor changes the version changes to 1.1. A central Git Repository is maintained to monitor the documents and source code being developed by team members, which is accessible to the CEO. Different development branches are made in the repository and simultaneous development is done. Later the changes are submitted to main branch which can be monitored by CEO for changes and conflicts

8.2 System Building:

System building has three phases. Developing of the tool, version management and build server followed by execution of target system. The tool development is done in a private workspace and checked out to an interactive version management system for easy management by the team members. The build system includes code lines and baselines according to the changes done and the concluding build is checked out to build server to build definitive editions of the software. The final software is compiled to change into executable.

8.3 Release Management:

The early build version of software is released to customer. The customer uses the product a reports the problems. Depending on the types of problems and bug fixes the release can be classified to major release and minor release. Each release is documented to ensure that it can be re-created exactly in the future. The source code copies must be maintained along with the versions of the libraries, compilers and tools used for development for future release purposes.

9) Progress Tracking:

The entire project is divided among all the team members and the progress is tracked through colloquial meetings. The project manager acts as a medium to coordinate between the different members of the group and help them to keep informed about the progress. A detailed excel spread sheet regarding our progress can be found on git: https://herkules.comproj.bth.se/puts-2015/Oceans11.

10) Quality Control:

In order to maintain the expected quality of the customer, the testing wing of the group plays a vital role. They perform thorough testing of the build environment against a target platform and report the bugs to the development wing for rectification and debugging. The build environment can be finally validated by testing and taking feedback from the costumer and by making the relevant changes. This involves:

- Unit testing: individual program units or object classes are tested.
- Component testing: focus on testing component interfaces.
- System testing: focus on testing component interactions.

11) Risk Management:

Risk management deals with expectancy of risks that might affect the project.

Risk	**Strategy**	**Level of impact**
Recruitment problems	Alert about the possible delays.	minimal
Staff illness	Maintain communication between the remaining team members, to share the pending work.	minor
Component malfunctions	Test for defects an do the necessary replacements.	major
Change in requirements	Asses the requirements change and their impacts	major
Underestimated development time	Investigate the buy in components	critical

Table 2: Risk management.

12) System Release Plan:

12.1 Testing Plan:

After developing the complete product, tests will be performed. The testing members will run tests to detect bugs and ensure efficiency of the product. Tests are performed to check easy accessibility of the tool to the customer. Tests are mainly performed to check whether the tool runs in the work environment and to ensure that the tool meets customer requirements.

The testing process is carried out in three stages:

1. Unit testing where individual program units or object class are tested, focusing on the functionality of the objects.
2. Component testing involves integration of several individual components focusing on testing of component interfaces.
3. System testing where the whole system is taken and focused on component interactions. The tests to be conducted under this section include verifying the functionality of GUI for displaying of errors, warning and monitoring of open stack services.

   The testing tram is responsible for conducting all the relevant tests at different stages according to time plan, analyse the test results and display the statistics.

12.2 Packaging plan:

A compressed tar.gz archive is provided to the user consisting of the alpha code, library files, tools and related documentation. The details of the release plan are as follows:

Release candidate: 2015/05/18

Final release: 2015/05/28.

12.3 Documentation Plan:

12.3.1 Installation Documentation:

The Installation document is released in a PDF format. It covers the installation procedures of different software components and also the configuration settings for different components.

Time Schedule:

The time schedule for preparing installation document is divided into three phases with regard to the time plan.

Documentation before Testing: 2015/5/10

Documentation during the Testing phase: 2015/5/15

Documentation after Testing: 2015/5/18

.

12.3.2 User Documentation:

The user document is released in a PDF format. It covers the scope of the tool, functionality of the tool, linking different modules in the tool, procedure for supplying inputs to the tool, generation of output and various scripts.

Documentation before Testing: 2015/5/11

Documentation during the Testing phase: 2015/5/16

Documentation after Testing: 2015/5/19

12.3.3 Developer Documentation:

This document gives us about the future scope of the project. This document covers the sources code and all the related information about the API's used in this project.

Documentation before Testing: 2015/5/12

Documentation during the Testing phase: 2015/5/17

Documentation after Testing: 2015/5/20

Time Schedule:

User Documentation is prepared after the installation documentation is done considering the fact that the user has no prior knowledge regarding the usage of the tool.

Documentation before Testing: 2015/5/10

Documentation during the Testing phase: 2015/5/15

Documentation after Testing: 2015/5/19

13) References:

[1]. http://docs.openstack.org/: OpenStack Documentation

[2]. Ian Sommerville. *Software Engineering*. 9th Ed.