System Requirements Specification

Justin Anderson
John Nagel
Cesar Ramirez
John Rensberger

Table of Contents

1. Project Overview

1.1 Vision

The purpose of this project is to provide the Computer Science department at North Dakota State University with a open source cloud using OpenStack. The cloud will provide the department with many new possibilities by offering faculty and students virtualized resources through the Infrastructure as a Service model. This will allow better utilization of resources as specific courses will be able to create purpose-built images that are easy to modify and scale while also offering many other solutions to faculty and staff for research needs. Throughout the duration of the semester, it is our goal to deliver a working cloud to the department that is not only fully operational, but is actively being utilized and is hopefully incorporated into future educational use cases.

1.2 Overall Description

OpenStack creates an Infrastructure as a Service (IaaS) that provisions computing resources to provide various cloud-based services. IBM approached North Dakota State University last year to create a private cloud running on four nodes as a proof-of-concept that such an application could be achieved using open source products. Being able to implement OpenStack also presents many possibilities to the university, the primary being cost savings on software. The capstone group that worked on this project last year delivered a working prototype and developed a few use cases to demonstrate its abilities.

Based on last year's success, IBM would like to continue to pursue the possibilities that OpenStack can offer the CS department at NDSU. Throughout the duration of the semester, the group is expected to deliver a fully-functional production instance of the cloud to the department. In addition to delivering this instance, the group will need to reach out to the faculty and staff in the department to develop several use cases for the cloud. It is then their responsibility to deliver as many of these use cases as possible. The true success of this project will be measured by how heavily utilized the cloud is within the department.

To complete these goals, the group will need to spend significant time researching OpenStack to streamline processes in order to allow further development. All work will need to be thoroughly documented and packages/scripts should be provided to allow future builds of the cloud, modules, and configurations.

2. System Requirements

2.1 High-Level Requirements

The **ID** column provides a high-level ID for each requirement.
The **added** column lists when the requirement was added to the project, R = revision.
The **description** column gives a description of the high-level requirement.
The **status** column indicates whether these high level requirements are:

- C = committed, will be completed by the team
- T = targeted, will be completed if the team has time after committed requirements are completed
- NC = not committed, will not be completed by the team

ID	Added	Description	Status
1100	Onset	Provide visual interface for System Administrator and other users of OpenStack	C
1200	Onset	LDAP and AD integration with OpenStack	C
1300	Onset	Quantum (network) component working	C
1400	Onset	Cinder (block storage) component working	C
1500	Onset	Nova (compute) component working	C
1600	Onset	Documentation	C
1700	R1	Glance (image services) working	C
2100	Onset	Integration from OpenStack with Hadoop	T
2200	Onset	Integration GRASS-GIS and OpenStack over a distributed file system	T
2300	Onset	Post deployment script for VMs	T
2400	Onset	Swift (object storage) component working	T
3100	Onset	Integration with Jenkins/CI systems for large scale testing	NC
3200	R1	Chef or Puppet integration for orchestration	NC

2.2 Low-Level Requirements

ID	Description	Verification
1101	Main controller node and 2+ compute nodes	Fault tolerance by adding and removing nodes in real time
1102	Reporting and monitoring of VMs that allows a system administrator to confirm the correct functioning of the entire system	Running gkrellm or equivalent monitoring tools
1103	System should be able to deploy a new VM in less than 10 minutes	Record instance deployment duration and average run time
1104	System should be capable of deploying a new cluster of VMs, quantity based on hardware	Stress test use cases to determine the limits of our hardware
1105	Updates should be able follow standard OpenStack procedures	Ensure that updates run without any additional configurations outside of the preconfigured steps
1201	Integration with the CS department's Active Directory for authentication	CS department credentials can be used with OpenStack services
1301	VLAN for network management	Will not have to blacklist MAC addresses, IP ranges will be outside of NDSU's network
1302	All internal networks should have a connection speed of at least 1Gb/s	All hardware supporting infrastructure will need to be verified for consistent performance through load tests
1401	Redundancy of data storage	Storage system can survive a simulated device failure
1601	Documentation for administrators	Resources are made readily available from common sources like the CS wiki and CS webpage
1602	Documentation for VM deployment and management for end-users	Inexperienced user can understand and complete tasks without aid

1701	Provide images with Hadoop installed and configured	Interested faculty can easily utilize Hadoop on VMs
1702	Provide Windows and Linux images	Deployment of new VM instances using images
2301	Scripts should exist to allow the quick deployment of the system	Script will create a standard output to indicate completion status
2401	Object storage is only accessible by the tenant owner and the system administrator	Verify permissions by performing test cases with restricted users
2402	Both high-speed (SSD) and high-capacity drives should be available for the users	Users can choose storage types based on usage requirements

3. Dependencies and Constraints

ID	Dependencies and Constraints
D1	System Administrator physical hardware/availability
D2	Folsom release having all features implemented for the project
C1	Conflicting schedules between all group members
C2	Relaying requirements and development progress between
C3	Make sure mentor has access to servers, Dashboard interface and other tools

4. Definitions and Acronyms

Term	Definition
Backend	A term that refers to a computer that stores data
Cinder	The OpenStack component that provides block storage for guest VMs
Flavor	An available hardware configuration for a server
Glance	The OpenStack component that provides catalog and repository for virtual disk images
Horizon	The OpenStack component that provides a web UI for easy administration
Keystone	The OpenStack component that provides authorization and authentication for the services
NFS	An acronym that stands for Network File System, a system that allows access of remote files and directories over a network
Nova	The OpenStack component that provides creation of virtual servers on demand
Quantum	The OpenStack component that provides networking as a service
SAN	An acronym that stands for Storage Area Network, a network of systems that are dedicated to file access and storage
Swift	The OpenStack component that provides redundant object storage
VM	An acronym that stands for Virtual Machine, software that allows multiple instances of operating systems to run on the same system

5. Requirements Review

Project Manager

Name:	John Rensberger
Date:	
Signature:	

Mentor

Name:	Michael Fork, Lance Bragstad, Mathew Odden, Adam Reznechek
Date:	
Signature:	

6. Revision History

Date	Author/Updater	Status	Item(s) Changed	Short Description of Change
02/04/2013	Team	Draft	None	Created initial document
02/05/2013	Michael F, Adam R	In Review	None	
02/07/2013	John R	Approved	1.1 Vision, 2.1 Requirements	Updated requested items
02/13/2013	Cesar R, Jack N, John R	Revision 1	2.1, 2.2, additional language updates	Updates from meeting with mentor