การวิจัยเพื่อศึกษาการนำระบบโอเพ่นสแตกมาใช้ในองค์กร

RESEARCH PROJECT FOR OPENSTACK IMPLEMENTATION

การะเกด ศาสดา
ณัฐสิมา เงินไทย

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาศาตร์คอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESEARCH PROJECT FOR OPENSTACK IMPLEMENTATION

KARAGADE SASSADA
NATSIMA NGOENTHAI

A COOPERATIVE EDUCATION SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การวิจัยเพื่อศึกษาการนำระบบโอเพ่นสแตกมาใช้ในองค์กร Research project for Openstack Implementation
ชื่อนักศึกษา	นางสาวการะเกด ศาสดา รหัสนักศึกษา 57050181 นางสาวณัฐสิมา เงินไทย รหัสนักศึกษา 57050229
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	ผศ.กฤษฎา บุศรา

คณะวิทยาศาตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2559

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.กฤษฎา บุศรา ประธานกรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา การวิจัยเพื่อศึกษาการนำระบบโอเพ่นสแตกมาใช้ในองค์กร
ชื่อนักศึกษา นางสาวการะเกด ศาสดา รหัสนักศึกษา 57050181
นางสาวณัฐสิมา เงินไทย รหัสนักศึกษา 57050229
ปริญญา วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา วิทยาการคอมพิวเตอร์
คณะ วิทยาศาสตร์
ปีการศึกษา 2559
อาจารย์ที่ปรึกษา ผศ.กฤษฎา บุศรา

บทคัดย่อ

สหกิจนี้นำเสนอระบบบริหารจัดการ IT Infrastructure เป็นรูปแบบการให้บริการของระบบประมวลผลกลุ่มเมฆ ในรูปแบบระบบประมวลผลกลุ่มเมฆแบบส่วนตัว ซึ่งเป็นแพลตฟอร์มที่นำมาพัฒนาระบบคอมพิวเตอร์ ระบบเครือข่าย และการจัดสรรทรัพยากรคอมพิวเตอร์ขององค์กร ที่ทำให้ผู้ใช้สามารถเข้าถึงระบบ สร้าง จัดการและใช้งานเครื่องเสมือน (virtual machine) ได้ ผ่านหน้าเว็บ โดยใช้ซอฟต์แวร์ โอเพ่นซอร์ส ได้แก่ โอเพ่นสแตก อูบุนตู และ Ceph ทำหน้าที่ในการเป็นซอฟท์แวร์ที่มีประสิทธิภาพในการจัดเก็บข้อมูล มีความยืดหยุ่นง่ายต่อการจัดการ ซึ่งโอเพ่นสแตก ทำหน้าที่ในการบริหารเครื่องเสมือน มีหน้าเว็บที่สามารถ ช่วยให้ผู้ใช้ควบคุม จัดการเครื่องเสมือน ได้ง่ายขึ้น และยังช่วยลดค่าใช้จ่ายในด้านของฮาร์ดแวร์ มีการติดตั้งและทดสอบ ประสิทธิภาพของ openstack โดยไม่ใช้เครื่องคอมพิวเตอร์ ผลการทดสอบ High Availability Infrastructure และ Application โดยการปิดเครื่อง controller และ ceph ทีละเครื่อง ระบบสามารถทำงานได้ปกติ เมื่อปิดเครื่อง compute instance ที่ทำงานอยู่บน compute ตัวที่ปิด ไม่สามารถใช้งานได้ โดยแนวทางแก้ปัญหาคือต้องทำ Load balance เฉพาะ compute server ซึ่งจะทำให้ในการปิดเครื่อง compute server ระบบยังสามารถใช้งานได้ปกติ

คำสำคัญ : เครื่องเสมือน โอเพ่นซอรส โอเพ่นสแตก ระบบประมวลผลกลุ่มเมฆแบบส่วนตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Research project for Openstack Implementation
Student Name	Miss Karagade Sassada Student ID 57050181
	Miss Natsima Ngoenthai Student ID 57050229
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Year	2016
Advisor	Asst.Prof.Kridsada Budsara

Abstract

This cooperative education project present about Cloud computing format private cloud of Infrastructure as a Service (IaaS), This platform for the development of Infrastructure Networking and to allocate computer resources to the organization. Users can access, create, manage and run virtual machines by Dashboard. Using Open source software Openstack Ubuntu and Ceph. Ceph is a flexible and easy to use storage platform. Openstack is software manage virtual machine. dashboard of Openstack user can to control Virtual machine management simplifies and reduces the cost of hardware. Installed and performance manual testing of the Openstack. To bring the conclusion that Openstack. Result of high availability infrastructure and application testing by shutting down both the controller and the ceph machine. When shutting down the compute instance running on the compute, the shutdown does not work. The solution is to load balance compute server only, which will cause the computer to shut down the compute server is still usable.

Keywords: Opensource ,Openstack ,Private Cloud, Virtual Machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

สหกิจศึกษา หัวข้อการวิจัยเพื่อศึกษาการนำระบบโอเพ่นสแตกมาใช้ในองค์กร สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากคณะผู้จัดทำได้รับความร่วมมือจากบุคคลผู้ มีพระคุณหลายท่าน ดังนี้

ขอขอบคุณพระคุณ บิดา มารดา และสมาชิกครอบครัวทุกท่าน ผู้ซึ่งอยู่เบื้อง คอยอบรมสั่งสอน ให้กาลังใจมาตลอดในการจัดทาปัญหาพิเศษ และความอบอุ่นตลอดมา

ขอขอบพระคุณ ผศ.กฤษฎา บุศรา อาจารย์ภาควิชา วิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อาจารย์ที่ปรึกษาในการทำสหกิจศึกษาที่ได้ให้คำปรึกษาและคำแนะนาต่างๆอย่างใกล้ชิดและเสนอแนะแนวทางแก้ปัญหา รวมถึงการตรวจสอบและแก้ไขรายงานสหกิจศึกษาฉบับนี้ให้มีความสมบูรณ์มากขึ้น

ขอขอบพระคุณ ผศ.ดร.วรางคณา กิ้มปาน อาจารย์ภาควิชา วิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้คำแนะนาและชี้จุดบกพร่องที่ควรแก้ไข ช่วย ตรวจสอบเพิ่มความสมบูรณ์ให้กับปัญหาพิเศษนี้

ขอขอบคุณ พี่ศุภกฤษฎ์ บุญสาตร์ พี่สนธิ ณ พัทลุง พี่ก้องศักดิ์ ลิมปิติเกียรติ จาก บริษัท GEC (GENERAL ELECTRONIC COMMERCE SERVICES CO.,LTD.) ที่ได้ให้ความรู้ทางวิชาการและความรู้อันเป็น พื้นฐาน ความช่วยเหลือในการทำงาน จนกระทั่งคณะผู้จัดทำสามารถที่จะดำเนิน สหกิจศึกษาสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ต้องขอขอบคุณเพื่อน ๆ ทุกคนที่ให้คำปรึกษาและวางใจเสมอมา

การะเกด ศาสดา
ณัฐสิมา เงินไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ช
สารบัญรูป ..... ซ
บทที่ 1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของงานวิจัย/ปัญหา 1
1.2 วัตถุประสงค์ของงานวิจัย 1
1.3 ขอบเขตของงานวิจัย 2
1.4 ประโยชน์ที่คาดว่าจะได้รับ 2
1.5 ขั้นตอนการดำเนินงาน 2
1.6 อุปกรณ์ที่ใช้ 2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3
2.1 ระบบประมวลผลกลุ่มเมฆ หรือ Cloud Computing 3
2.1.1 รูปแบบการให้บริการของ ระบบประมวลผลกลุ่มเมฆ 4
2.1.2 ประเภทของ ระบบประมวลผลกลุ่มเมฆ 5
2.1.3 คุณสมบัติของบริการ ระบบประมวลผลกลุ่มเมฆ 5
2.1.4 องค์ประกอบของ ระบบประมวลผลกลุ่มเมฆ 6
2.1.5 ข้อดีของ ระบบประมวลผลกลุ่มเมฆ 6
2.1.6 ข้อเสียของ ระบบประมวลผลกลุ่มเมฆ 6
2.1.7 จุดเด่นของ ระบบประมวลผลกลุ่มเมฆ 6
2.2 openstack 7
2.2.1 Module การทำงานของ openstack 8
2.2.2 Enterprise Cloud by OpenStack 9
2.2.3 เหตุผลที่เหล่าองค์กรและผู้ดูแลระบบ ควรเริ่มต้นใช้ openStack สร้าง ระบบประมวลผลกลุ่มเมฆแบบส่วนตัว ภายในองค์กร 10
2.3 Ceph 12
2.3.1 Ceph Storage is a Distributed Storage 12
2.3.2 Ceph Storage is Software-Defined Storage 13
2.3.3 Ceph Storage is Unified Storage System 13
2.3.4 Ceph Storage is a Greater Peace of Mind 14
2.3.5 Ceph Storage is Designed for Scale 14
2.3.6 แล้ว Ceph Storage เหมาะกับอะไรบ้าง 14
2.4 Virtualization Technology 15
2.4.1 เหตุผลที่ใช้ Virtualization 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4.2 องค์กรแบบใดที่เหมาะสำหรับการนำเทคโนโลยี Server Virtualization ... 17
2.5 โครงสร้างพื้นฐานทางไอที (IT Infrastructure) ... 17
2.5.1 ปัจจัยที่ผลักดันให้เกิดวิวัฒนาการของโครงสร้างพื้นฐานทางด้านไอที ... 18
2.5.2 การจัดเก็บข้อมูลดิจิตอลเป็นจำนวนมาก ... 18
2.5.3 ปัจจัยที่ผลักดันให้เกิดวิวัฒนาการของโครงสร้างพื้นฐานทางด้านไอที ... 18
2.5.4 ส่วนประกอบของโครงสร้างพื้นฐานทางด้านไอที ... 19
2.6 Ubuntu ... 19
2.6.1 ประวัติและลำดับการพัฒนา ... 19
2.6.2 หลักการสำคัญ ... 20
2.7 Ansible ... 20
2.7.1 inventory file ... 20
2.7.2 playbook.yml ... 20
2.8 วิจัยที่เกี่ยวข้อง ... 21
บทที่ 3 วิธีการดำเนินงานวิจัย ... 22
3.1 ขั้นตอนการศึกษาและรวบรวมข้อมูล ... 22
3.2 วิเคราะห์ปัญหาขององค์กร ... 22
3.3 สถาปัตยกรรมของ VM ware ... 23
3.4 สถาปัตยกรรมของ openstack บนระบบประมวลผลแบบกลุ่มเมฆส่วนตัว ... 23
3.5 ความสามารถของ openstack ... 24
3.5.1 Admin ... 25
3.5.2 User ... 25
3.6 ขั้นตอนการทำงานของ openstack ... 25
3.7 การทดสอบประสิทธิภาพของ openstack ... 26
3.7.1 สภาพแวดล้อมโดยรวมของระบบในการทดสอบ ... 27
3.8 ขั้นตอนการทดสอบระบบ ... 28
3.9 ส่วนติดต่อกับผู้ใช้งานของ openstack ... 28
3.9.1 หน้าจอแรกของ openstack ... 28
3.9.2 หน้าจอแสดงภาพรวมของระบบ ... 29
3.9.3 หน้าจอแสดงรายละเอียดของเครื่องเสมอน(Instances) ... 30
3.9.4 หน้าจอแสดงรายละเอียดของอิมเมจไฟล์(Image) ... 30
3.9.5 หน้าจอแสดงการเข้าถึงและความปลอดภัยของเครื่องเสมือน (Access & Security) ... 30
บทที่ 4 ผลการวิจัยและการอภิปรายผล ... 32
4.1 ผลการทดสอบระบบ ... 32
4.2 ภาพรวมส่วนติดต่อผู้ใช้ผ่านหน้าแดชบอร์ด (dashboard) ... 33
4.3 การจัดการไฟล์อิมเมจ (Image) ... 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.4 การสร้างเครื่องเสมือน (Instances) 37
4.5 การจัดการบัญชีผู้ใช้งาน 39
4.6 การทำงานของ Access & Security 40
4.7 ความคุ้มค่าของการนำ openstack มาใช้ในองค์กร 42
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 44
5.1 สรุปผลการดำเนินการ 44
5.2 สรุปผลการทดลอง 44
5.3 แนวทางในการศึกษาต่อและข้อเสนอแนะ 44
5.4 ปัญหาที่พบ 45
เอกสารอ้างอิง 46
ภาคผนวก 49
ภาคผนวก ก คู่มือการใช้งาน 50
ภาคผนวก ข รายละเอียดการพัฒนาระบบประมวลผลกลุ่มเมฆแบบส่วนตัวด้วย openstack และ ceph 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

3.1 คุณสมบัติของแต่ละเครื่องเสมือน 28
4.1 ขั้นตอนการทดสอบ High Availability Infrastructure 32
4.2 ขั้นตอนการทดสอบ High Availability Application 32
4.3 แสดงค่าใช้จ่ายเครื่องเสมือน 43
ก.1 ฟังก์ชันของ Identity 54
ก.2 ฟังก์ชัน Setting 55
ก.3 แสดงรายละเอียดการสร้างอิมเมจ 56
ก.4 รายละเอียดการสร้างเครือข่าย 58
ก.5 Details tab 63
ข.1 แสดงการตั้งค่าเครือข่าย (Network) 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	Cloud computing หรือการประมวลผลกลุ่มเมฆ	4
2.2	openstack diagram	7
2.3	Module การทำงานของ openstack	8
2.4	Ceph	12
2.5	Ceph Storage Cluster	12
2.6	Ceph Storage Interface	13
2.7	การทำงานของ Virtualization	16
2.8	เปรียบเทียบการทำงานระหว่างการนำ Virtualization มาใช้กับไม่ได้นำมาใช้	16
2.9	การเชื่อมโยงระหว่างองค์กรกับโครงสร้างพื้นฐานทางไอที และความสามารถทางธุรกิจ	18
2.10	โลโก้ Ubuntu	19
3.1	สถาปัตยกรรมของ VMware	23
3.2	โครงสร้างการทำงานของ openstack	23
3.3	Use case diagram	24
3.4	Flowchart การทำงานร่วมกันของ opensatck และ ceph	26
3.5	โครงสร้างระบบเพื่อทดสอบการใช้งาน(POC)	27
3.6	หน้าจอแรกของเว็บไซต์ openstack	29
3.7	หน้าจอ Overview ของระบบ	29
3.8	หน้าจอ Instance ของระบบ	30
3.9	หน้าจอ Images ของระบบ	30
3.10	หน้าจอ Security Group ของระบบ	31
4.1	หน้าจอเข้าสู่ openstack	34
4.2	หน้าจอระบบ openstack	34
4.3	หน้าจอแสดงภาพรวมของระบบ openstack	35
4.4	หน้าจอแสดงบัญชีผู้ใช้งาน	35
4.5	หน้าจอแสดงรายการไฟล์อิมเมจ	35
4.6	หน้าจอการสร้างไฟล์อิมเมจ	36
4.7	การนำไฟล์อิมเมจเข้าสู่ระบบ	36
4.8	หน้าจอการลบไฟล์อิมเมจ	37
4.9	หน้าจอแสดงเครื่องเสมือนในระบบ	37
4.10	หน้าจอแสดงรายละเอียดของเครื่องเสมือน (1)	38
4.11	หน้าจอแสดงรายละเอียดของเครื่องเสมือน (2)	38
4.12	หน้าจอแสดงรายละเอียดของเครื่องเสมือน (3)	39
4.13	หน้าจอแสดงบัญชีผู้ใช้	39
4.14	หน้าจอแสดงรายละเอียดของการสร้างบัญชีผู้ใช้	40
4.15	หน้าจอควบคุมเครื่องเสมือน	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

4.16 หน้าจอแสดง Access & Security 41
4.17 หน้าจอ Create Security Group 41
4.18 หน้าจอ Add Rule 41
4.19 การใช้ Heat สร้าง Instance ในระบบ 42
4.20 Template การสร้าง Heat 42
ก.1 หน้าจอเข้าสู่ระบบ 50
ก.2 หน้าจอภาพรวมของ project tab 51
ก.3 หน้าจอผู้ดูแลระบบ 52
ก.4 ฟังก์ชัน Identity 53
ก.5 ฟังก์ชัน Setting Tab 53
ก.6 หน้าจอ Create image (1) 54
ก.7 หน้าจอ Create image (2) 54
ก.8 หน้าจอ Create Network 55
ก.9 หน้าจอ Create Subnet 56
ก.10 หน้าจอ Subnet Details 56
ก.11 แผนผังเครือข่าย Network Topology 57
ก.12 หน้าจอ Create Router 58
ก.13 หน้าจอ Add Interface 58
ก.14 หน้าจอ Create Security Group 59
ก.15 หน้าจอ Add Rule 59
ก.16 หน้าจอ Import Key Pair 60
ก.17 หน้าจอ Allocate A floating IP 60
ก.18 หน้าจอการสร้างเครื่องเสมือน 61
ก.19 หน้าจอ Source ของเครื่องเสมือน 61
ก.20 หน้าจอ flavor ของเครื่องเสมือน 62
ก.21 หน้าจอ Network ของเครื่องเสมือน 62
ก.22 หน้าจอ Security Groups tab 63
ก.23 หน้าจอ Key Pair ของเครื่องเสมือน 63
ก.24 หน้าจอ Configuration ของเครื่องเสมือน 64
ก.25 หน้าจอ Metadata ของเครื่องเสมือน 64
ก.26 หน้าจอ Volume ของเครื่องเสมือน 65
ข.1 สถาปัตยกรรมของระบบ openstack 66
ข.2 สภาพแวดล้อมและทรัพยากรโดยรวมของระบบ 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กระบวนการทำงานของระบบสารสนเทศในองค์กรปัจจุบันใช้เทคโนโลยีการจำลองอุปกรณ์เสมือน ในการบริหารจัดการ IT Infrastructure โดยในกระบวนการทำงานมีซอฟต์แวร์ที่ทำหน้าที่จัดสรรทรัพยากร รวมถึงระบบเครือข่าย และอุปกรณ์จัดเก็บข้อมูลให้กับเครื่องเสมือนที่ทำงานอยู่บนเครื่องเซิร์ฟเวอร์เดียวกัน เรียกซอฟต์แวร์ที่ทำหน้าที่นี้ว่า ไฮเปอร์ไวเซอร์ ซึ่งองค์กรใช้ VMware หลักการทำงานของ VMware ในการเพิ่มทรัพยากร คือเพิ่มฮาร์ดแวร์ที่เป็นอุปกรณ์จัดเก็บข้อมูลหรือถ้าต้องการเครื่องเสมือนเพิ่ม ทีมที่คอยจัดการจะดำเนินการจัดหาทรัพยากรตามที่ผู้ใช้งานต้องการ ทำให้การเพิ่มทรัพยากรต้องเสียค่าใช้จ่ายและเวลาในการจัดหา

จากปัญหาที่เกิดขึ้นองค์กรจึงพิจารณาแพลตฟอร์มระบบประมวลผลกลุ่มเมฆแบบส่วนตัวซึ่งเป็นรูปแบบการใช้งานของระบบประมวลผลกลุ่มเมฆ โดยมีคุณสมบัติ On-demand self-service ที่ทำให้ผู้ใช้สามารถสร้างหรือเพิ่มทรัพยากรด้วยตัวเองโดยที่ไม่ต้องเสียเวลารอผู้ดูแลระบบ รวมถึงคุณสมบัติอื่นๆที่น่าสนใจ และพิจารณานำ openstack ที่เป็นซอฟต์แวร์โอเพ่นซอร์ส และรองรับ IT Infrastructure ได้หลากหลาย รวมถึงมีเทคโนโลยีที่น่าสนใจให้พร้อมใช้ในตัว

โดยเป้าหมายขององค์กรคือ ถ้าหากองค์กรสามารถทำในลักษณะระบบประมวลผลกลุ่มเมฆแบบส่วนตัวได้ ทำให้อุปกรณ์หลายๆตัวจะถูกปรับเปลี่ยนไปเป็นในลักษณะของเครื่องเสมือนซึ่งจะทำให้มีความยืดหยุ่นในการทำงานและลดปริมาณของจำนวนฮาร์ดแวร์ลงได้เป็นอย่างมาก ส่งผลให้ในการลดต้นทุน ค่าติดตั้ง และค่าบำรุงรักษา รวมไปถึงมีความยืดหยุ่นในลักษณะขยายได้ในอนาคต

1.2 วัตถุประสงค์

1) เพื่อศึกษาระบบโครงสร้างพื้นฐานของการประมวลผลกลุ่มเมฆ
2) เพื่อศึกษาความเป็นไปได้ในแนวทางของการใช้งาน openstack ในระดับองค์กร
3) เพื่อลดเวลาและความซับซ้อนในการติดตั้งและใช้งาน ฮาร์ดแวร์และ ระบบเครือข่ายภายในองค์กร

1.3 ขอบเขตของงานวิจัย

1) งานวิจัยนี้เป็นกรณีศึกษาระบบโครงสร้างพื้นฐานของระบบประมวลผลกลุ่มเมฆ เพื่อนำไปพัฒนา IT Infrastructure ขององค์กร โดยใช้แพลตฟอร์มระบบประมวลผลกลุ่มเมฆแบบส่วนตัวและเป็นประโยชน์ในการลดต้นทุนทางด้านฮาร์ดแวร์ ค่าติดตั้ง ค่าบำรุงรักษา รวมถึงค่าลิขสิทธ์ของซอฟต์แวร์
2) ศึกษาโครงสร้างพื้นฐานของ openstack ในการใช้งานระดับองค์กร และทำการทดสอบการใช้งาน (POC) เพื่อวัดประสิทธิภาพในการทำงานว่าสามารถทำงานได้เหมือนกับ VMware
3) ขอบเขตเครื่องมือที่ใช้ในการพัฒนา
   3.1) เครื่องเซิรฟเวอร์ สำหรับงานวิจัยนี้ใช้ ฮาร์ดแวร์ ที่มีการใช้อยู่ภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2) ด้านซอฟต์แวร์

1) VWware EXSi ใช้สร้างเครื่องเสมือน
2) openstack ใช้สร้างระบบ ระบบประมวลผลกลุ่มเมฆแบบส่วนตัว
3) ceph ในการจัดการ Storage ซึ่งใช้ทำงานร่วมกับ openstack

3.3) ระบบเครือข่าย Network เป็นระบบเครือข่ายภายในองค์กร

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) เพื่อเป็นแนวทางในการนำ openstack มาใช้งานในองค์กรแทนซอฟต์แวร์ตัวเดิม และลดค่าใช้จ่ายในการดำเนินการขององค์กรได้
2) องค์กรทราบถึงปัญหาของการนำ openstack มาใช้จริงเพื่อหาแนวทางแก้ไขในอนาคตและเพื่อให้ ทราบถึงการใช้งานในระดับ การผลิต (Production)
3) เป็นแนวทางในการจัดฝึกอบรมให้แก่บุคลากรในองค์กรเพื่อเข้าใจ openstack มากขึ้น

1.5 ขั้นตอนการดำเนินงาน

1) ศึกษา IT Infrastructure ขององค์กร
   1.1) ศึกษาขั้นตอนการทำงานและหลักการในการควบคุม
   1.2) ศึกษาประสิทธิภาพการทำงานของ VMware ในเทคโนโลยีการจำลองอุปกรณ์เสมือน
2) ศึกษาและรวบรวมข้อมูลการทำงานของ openstack
   2.1) ศึกษาหน้าที่และการทำงาน แต่ละ โมดูล (Module) ใน openstack
   2.2) ศึกษาภาพรวมการทำงานของ openstack ใน ระบบประมวลผลกลุ่มเมฆแบบส่วนตัว
3) วิเคราะห์ปัญหา IT Infrastructure แบบเดิมขององค์กร และการทำงานของ openstack
4) ทำการติดตั้ง openstack และ ceph เพื่อทำการทดสอบการใช้งาน (POC)
5) ทดสอบเพื่อปรับปรุงข้อผิดพลาดของระบบ
6) สรุปผลของการทดสอบการใช้งาน (POC)
7) จัดทำคู่มือการใช้งานและคู่มือการติดตั้ง
8) จัดทำรูปเล่มสหกิจศึกษา

1.6 อุปกรณ์ที่ใช้

1) ฮาร์ดแวร์เสมือน จำนวน 9 เครื่อง
2) ซอฟต์แวร์
   2.1) openstack เป็นซอฟต์แวร์ที่ใช้สร้างระบบการประมวลผลกลุ่มเมฆ
   2.2) ceph เป็นซอฟต์แวร์ที่ใช้จัดการอุปกรณ์เก็บข้อมูล (storage)
   2.3) VMWare ESXi เป็นซอฟต์แวร์ใช้ในการสร้างเครื่องเสมือน
3) ระบบปฏิบัติการ

   3.1) ระบบปฏิบัติการ Ubuntu เป็นระบบปฏิบัติการคอมพิวเตอร์ เป็นระบบที่วางตัวอยู่บนพื้นฐานของระบบ UNIX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

งานวิจัยเรื่องการนำระบบ openstack มาใช้ในองค์กร ผู้วิจัยได้ศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง โดยมีรายละเอียดดังนี้

2.1 ระบบประมวลผลกลุ่มเมฆ (Cloud Computing)

Wikipedia อธิบาย ระบบประมวลผลแบบกลุ่มเมฆ (อังกฤษ: cloud computing)[1] เป็นลักษณะของการทำงานของผู้ใช้งานคอมพิวเตอร์ผ่านอินเทอร์เน็ต ที่ให้บริการใดบริการหนึ่งกับผู้ใช้ โดยผู้ให้บริการจะแบ่งปันทรัพยากรให้กับผู้ต้องการใช้งานนั้น การประมวลผลแบบกลุ่มเมฆ เป็นลักษณะที่พัฒนาขึ้นต่อมาจากความคิดและบริการของเวอร์ชัวไลเซชันและเว็บเซอร์วิซ โดยผู้ใช้งานนั้นไม่จำเป็นต้องมีความรู้ในเชิงเทคนิคสำหรับตัวพื้นฐานการทำงานนั้น

บริษัท Gartner กล่าวเกี่ยวกับความหมายของ ระบบประมวลผลกลุ่มเมฆว่า[2] ระบบการประมวลผลแบบกลุ่มเมฆคือ แนวทางการประมวลผลที่พลังของโครงสร้างทางไอทีขนาดใหญ่ที่ขยายตัวได้ถูกนำเสนอยังลูกค้าภายนอกจำนวนมหาศาลในรูปแบบของบริการ

ฟอเรสเตอร์กรุ๊ป กล่าวกล่าวเกี่ยวกับความหมายของ ระบบประมวลผลกลุ่มเมฆ ว่า[3] การประมวลผลแบบกลุ่มเมฆคือ กลุ่มของโครงสร้างพื้นฐานที่ถูกบริหารจัดการและขยายตัวได้อย่างมาก ซึ่งมีขีดความสามารถในการรองรับโปรแกรมประยุกต์ต่างๆของผู้ใช้และเก็บค่า บริการตามการใช้งาน โดยการอิงหลักการของ Grid Computing, Utility Computing และ SaaS

“การประมวลผลที่อิงกับความต้องการของผู้ใช้ โดยผู้ใช้สามารถระบุความต้องการไปยังซอฟต์แวร์ของระบบประมวลผลกลุ่มเมฆ จากนั้นซอฟต์แวร์จะร้องขอให้ ระบบ จัดสรรทรัพยากรและบริการให้ตรงกับความต้องการของผู้ใช้ โดยระบบสามารถเพิ่มหรือลดจำนวนทรัพยากรให้พอเหมาะกับความต้องการของผู้ใช้ โดยที่ผู้ใช้ไม่ต้องทราบการทำงานเบื้องหลังว่าเป็นอย่างไร ” โดย JavaBoom Collection [4]

สรุปได้ว่า ระบบประมวลผลกลุ่มเมฆ คือการประมวลผลที่นำความต้องการของผู้ใช้งานที่ระบุไปประมวลผลหรือจัดสรรทรัพยากรตามที่ผู้ใช้งานต้องการ โดยที่ผู้ใช้งาน ไม่จำเป็นต้องทราบถึงวิธีการประมวลผลทั้งหมด และเมื่อเกิดปัญหา ระบบประมวลผลกลุ่มเมฆจะแก้ปัญหาเองโดยที่ผู้ใช้งานไม่ต้องมีบทบาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 Cloud computing [5]

2.1.1) รูปแบบการให้บริการของระบบประมวลผลกลุ่มเมฆ [6]

1) Software as a Service: SaaS เป็นการให้บริการด้านซอฟต์แวร์และแอปพลิเคชันผ่านทางอินเทอร์เน็ต คล้ายกับการเช่าใช้การคิดค่าบริการตามลักษณะการใช้งาน (Pay as you go) เช่น ตามจำนวนผู้ใช้งาน ระยะเวลาการใช้งาน ทำให้ผู้ใช้บริการไม่จำเป็นต้องลงทุนเพิ่มเติมในส่วนของฮาร์ดแวร์และค่าลิขสิทธิ์ซอฟต์แวร์ รวมไปถึงค่าใช้จ่ายในการดูแลรักษาระบบเพราะผู้ให้บริการจะเป็นผู้ดูแลระบบทั้งหมดให้ ผู้ใช้งานสามารถเข้าใช้งานซอฟต์แวร์และแอพพลิเคชั่นผ่านการเชื่อมต่ออินเทอร์เน็ตจากที่ไหนก็ได้ โดยไม่สนใจว่าซอฟต์แวร์และแอปพลิเคชั่นเหล่านี้ติดตั้งอยู่ที่ไหน ประมวลผลอยู่บนเซิร์ฟเวอร์อะไร

2) Platform as a Service: PaaS เป็นการให้บริการด้านแพลตฟอร์ม สาหรับการพัฒนาซอฟต์แวร์และแอปพลิเคชันโดยผู้ให้บริการจะจัดเตรียมสิ่งที่จำเป็นต้องใช้ในการพัฒนาซอฟต์แวร์และแอปพลิเคชันเช่น เว็บ แอพพลิเคชั่น เซิร์ฟเวอร์ที่จัดเก็บฐานข้อมูล ระบบประมวลผลกลางสำหรับองค์กรขนาดใหญ่ และ Middleware อื่นๆ โดยบริการทั้งหมดทางานภายใต้ระบบรักษาความปลอดภัยเครือข่าย และสามารถเรียกใช้งานได้ผ่านเว็บแอพพลิเคชั่น

3) Infrastructure as a Service: IaaS เป็นการให้บริการเฉพาะโครงสร้างพื้นฐาน เช่น เซิร์ฟเวอร์ ส่วนต่อประสานกับผู้ใช้ และระบบจัดเก็บข้อมูล ที่ทางานอยู่บนระบบเสมือน (Virtualization) เพื่อรองรับการใช้งาน ซอฟต์แวร์ และแอพพลิเคชั่น

4) Storage as a Service: SaaS เป็นการให้บริการระบบการจัดเก็บข้อมูลที่มีขนาดใหญ่ไม่จำกัด รองรับการสืบค้นและการจัดการข้อมูลขั้นสูง

5) Composite Service: CaaS เป็นการให้บริการส่วนที่ทำหน้าที่รวมโปรแกรมประยุกต์หลายๆ โปรแกรม หรือ จัดลำดับการเชื่อมโยงแบบกระแสงานข้ามเครือข่าย รวมถึงการจัดการด้านความปลอดภัย

เอกสารนี้มีลิขสิทธิ์ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2) ประเภทของระบบประมวลผลกลุ่มเมฆ [7]

1) Private Cloud ระบบที่ทางานอยู่บน ระบบประมวลผลกลุ่มเมฆ และบริหารจัดการโดยบริษัทหรือองค์กร เพื่อการใช้งานภายในองค์กรเท่านั้น โดยที่ผู้ให้บริการและผู้ใช้สามารถควบคุมและปรับปรุงระบบความปลอดภัยได้ด้วยตนเอง บริหารจัดการทรัพยากรของตนเอง

2) Public Cloud ระบบที่ทำงานอยู่บน ระบบประมวลผลกลุ่มเมฆ แบบสาธารณะที่ดูแลจัดการโดยผู้ให้บริการภายนอกผ่านอินเทอร์เน็ต ผู้ใช้จะมีสิทธิในการควบคุมที่จำกัดขึ้นอยู่กับการมอบสิทธิของผู้ให้บริการ Public Cloud มีทั้งบริการที่เสียค่าใช้จ่าย เช่น Windows Azure, SQL Azure และบริการฟรี เช่น Windows Live

3) Community Cloud ระบบที่สร้างขึ้นมาระหว่างองค์กรเป็นเครือข่ายสังคมเฉพาะกลุ่มที่มีความสนใจหรือต้องทำงานร่วมกัน สามารถเข้าใช้งานได้ เพื่อใช้เป็นมาตรฐาน หรือใช้ในการแก้ไขปัญหา

4) Hybrid Cloud ระบบที่ผสมผสานระหว่าง Private Cloud และ Public Cloud ทำให้สามารถทำงานเชื่อมต่อกันได้ ผู้ใช้สามารถขยายศูนย์ข้อมูลไปยัง Public Cloud เพื่อการใช้งานเฉพาะอย่าง และยังกลับมาใช้ Private Cloud ได้เมื่อต้องการเช่นกัน

2.1.3) คุณสมบัติของบริการระบบประมวลผลกลุ่มเมฆ [8]

1) บริการด้วยตัวเองเมื่อต้องการ (On-demand self-service) ผู้ใช้สามารถระบุความต้องการ และขอใช้ทรัพยากรคอมพิวเตอร์ เช่น เครื่องเซิร์ฟเวอร์ และระบบบันทึกข้อมูลที่เป็นเครือข่ายด้วยตนเองได้ทุกเวลา โดยไม่ต้องอาศัยความช่วยเหลือจากผู้ให้บริการ

2) เข้าถึงทรัพยากรคอมพิวเตอร์ได้ในวงกว้างผ่านเครือข่าย (Broad network access) ทรัพยากรคอมพิวเตอร์ถูกจัดสรรให้ผู้ใช้ใช้งานผ่านเครือข่าย ด้วยกลไกที่เป็นมาตรฐาน ผู้ใช้ทำงาน ได้ด้วยอุปกรณ์หลากหลายชนิด เช่น เครื่องโทรศัพท์พกพา เครื่องแล็บท็อปคอมพิวเตอร์ และเครื่อง พีดีเอ (Personal Digital Assistant, PDA)

3) ทรัพยากรถูกรวบรวมจากที่ต่าง ๆ (Resource pooling) ทรัพยากรคอมพิวเตอร์ที่จัดสรรให้กลุ่มผู้ใช้นั้น อาจมาจากศูนย์คอมพิวเตอร์หลายๆแห่ง ผู้ใช้แต่ละรายใช้ทรัพยากรที่ถูกจัดสรรให้โดยไม่รบกวนกันและกัน ทรัพยากรชุดเดียวกันแต่บริการหลายๆคนได้โดยไม่รบกวนกัน เรียกว่า Multi-tenant model ซึ่งต่างกับ Multiinstance ในกรณีหลัง เป็นการบริการด้วยทรัพยากรหลายชุด ผู้ใช้แต่ละรายจะมีหนึ่งชุดเป็นของตนเองทรัพยากรคอมพิวเตอร์ ทั้งที่เป็นกายภาพ (Physical) และที่เป็นเสมือน การจัดสรรอุปกรณ์จากกองกลาง ที่ผู้ใช้รู้สึกเสมือนหนึ่งว่าเป็นอุปกรณ์ที่ใช้แต่ผู้เดียว ทางเทคนิคเรียกว่า Virtualization ที่นำมาจัดสรรให้ผู้ใช้นั้น ไม่จำเป็นต้องอยู่ที่เดียวกัน อาจมาจากศูนย์คอมพิวเตอร์หลาย ๆ แห่ง หรือจากหลาย ๆ ประเทศได้ ผู้ใช้ไม่จำเป็นต้องรู้ว่าตนกำลังใช้อุปกรณ์คอมพิวเตอร์จากสถานที่ใดในโลก ผู้ใช้สนใจเพียงว่าได้ใช้ทรัพยากรคอมพิวเตอร์ ตามขนาด และประสิทธิภาพที่ต้องการ แต่ผู้ใช้อาจขอสิทธิ์ที่จะเลือกตำแหน่งศูนย์คอมพิวเตอร์ที่ตนต้องการใช้ได้

4) มีความยืดหยุ่นและปรับตัวได้รวดเร็ว (Rapid elasticity) การบริการคลาวต้องสามารถเพิ่มและลดขนาดอุปกรณ์ที่ให้บริการลูกค้าตามความต้องการและสามารถจัดสรรโดยอัตโนมัติได้ในสายตาของผู้ใช้ทรัพยากรคอมพิวเตอร์ที่นำมาให้บริการมีขนาดไม่จำกัดผู้ใช้สามารถสั่งให้เพิ่มหรือลดได้ตลอดเวลาตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) มีความยืดหยุ่นและปรับตัวได้รวดเร็ว (Rapid elasticity) การบริการคลาวต้องสามารถเพิ่มและลดขนาดอุปกรณ์ที่ให้บริการลูกค้าตามความต้องการและสามารถจัดสรรโดยอัตโนมัติได้ ในสายตาของผู้ใช้ ทรัพยากรคอมพิวเตอร์ที่นำมาให้บริการ มีขนาดไม่จำกัด ผู้ใช้สามารถสั่งให้เพิ่มหรือลดได้ตลอดเวลาตามความต้องการ

2.1.4) องค์ประกอบของระบบประมวลผลกลุ่มเมฆ [9]

ระบบประมวลผลกลุ่มเมฆ จำเป็นต้องอาศัยองค์ประกอบที่สำคัญคือ

1) อินเตอร์เน็ตที่มีช่องสัญญาณสูงจนเกือบจะไม่มีจำกัด (Nearly unlimited bandwidth)

2) เทคโนโลยีระบบเสมือนจริง (Increasingly sophisticated virtualization technologies)

2.1.5 ข้อดีของระบบประมวลผลกลุ่มเมฆ [10]

1) ลดต้นทุนค่าดูแลบำรุงรักษาเนื่องจากค่าบริการได้รวมค่าใช้จ่ายตามที่ใช้งาน จริง เช่น ค่าจ้างพนักงาน ค่าซ่อมแซม ค่าลิขสิทธิ์ ค่าไฟฟ้า ค่าน้ำ ค่าน้ำมันเชื้อเพลิง ค่าอัพเกรด และค่าเช่าคู่สาย เป็นต้น

2) ลดความเสี่ยงการเริ่มต้น หรือการทดลองโครงการ

3) สามารถลดหรือขยายได้ตามความต้องการ

4) ได้เครื่องแม่ข่ายที่มีประสิทธิภาพ มีระบบสำรองข้อมูลที่ดี มีเครือข่ายความเร็วสูง

5) อยู่ภายใต้การดูแลของผู้เชี่ยวชาญ

2.1.6) ข้อเสียของระบบประมวลผลกลุ่มเมฆ [11]

1) จากการที่มีทรัพยากรที่มาจากหลายแห่ง จึงอาจเกิดปัญหาด้านความต่อเนื่องและความรวดเร็ว

2) ยังไม่มีการรับประกันในการทำงานอย่างต่อเนื่องของระบบและความปลอดภัยของข้อมูล

3) แพลทฟอร์มยังไม่ได้มาตรฐาน ทำให้ลูกค้ามีข้อจำกัดสำหรับตัวเลือกในการพัฒนาหรือติดตั้งระบบ site

4) เนื่อง จากเป็นการใช้ทรัพยากรที่มาจากหลายที่หลายแห่งทำให้อาจมีปัญหาใน

2.1.7) จุดเด่นของระบบประมวลผลกลุ่มเมฆ [12]

1) ความสามารถในการขยายสูง (Scalability) ระบบประมวลผลกลุ่มเมฆสามารถรองรับการขยายได้อย่างรวดเร็วเพียง ปลายนิ้ว ไม่ว่าจะเป็นการขยายเพิ่มความจุ CPU หน่วยความจำ หรือ Storage ในการจัดเก็บข้อมูล หากต้องการเพิ่มประสิทธิภาพก็สามารถเข้าไปปรับแต่งค่าโดยใช้เวลาเพียงไม่กี่นาทีเท่านั้น ซึ่ง แตกต่างจากระบบ On-Premise ที่จะต้องจัดหาอุปกรณ์มาจัดเตรียม ไว้ล่วงหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

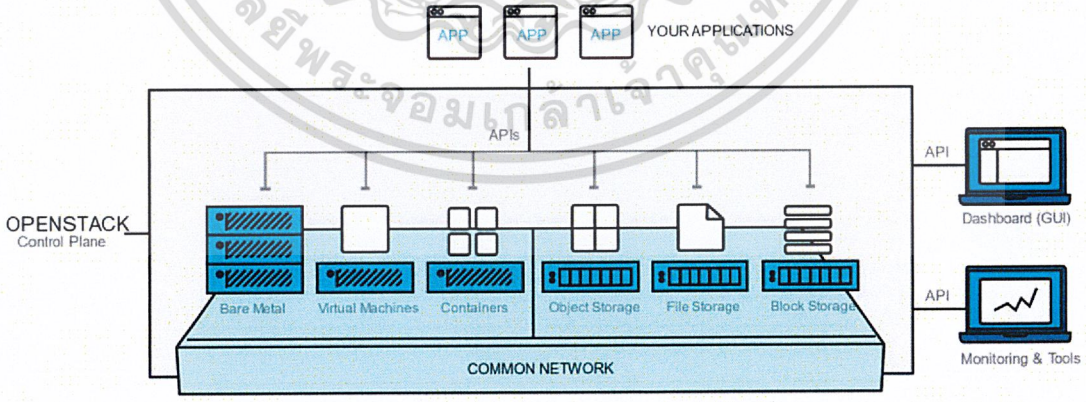
2) ความรวดเร็วในการปรับตัวเพื่อตอบสนองธุรกิจ (Speed & Agility) ทุกวันนี้ธุรกิจมีการแข่งขันกันสูง ระบบ Cloud สามารถปรับเปลี่ยนให้ สามารถรองรับการ

เปลี่ยนแปลงและความต้องการทางธุรกิจได้ อย่างทันท่วงที เช่น สมมติว่าบริษัทมีแผนจะออกแคมเปญใหม่ใน เดือนหน้าซึ่งจะต้องใช้เครื่องเซิร์ฟเวอร์และ Storage สำหรับการ พัฒนาระบบใหม่ ถ้าเป็นระบบ Cloud เราก็สามารถเข้าไปคลิก เลือกเพื่อขอใช้บริการได้ทันที แต่หากเป็นวิธีการแบบเดิมๆ ก็คงไม่ สามารถทำได้ แม้เพียงแค่การเตรียมการณ์ในการจัดหาอุปกรณ์ก็ ไม่ทันการเสียแล้ว

3) ค่าใช้จ่ายต่ำกว่า (Lower Cost) ระบบ Cloud จะคิดค่าใช้จ่ายตามที่ใช้งาน หรือที่เรียกว่า Pay-As-You-Go ซึ่งนำเอาแนวคิดของการให้บริการทางด้านระบบ สาธารณูปโภคมาใช้ นั่นคือใช้เท่าไหร่ ก็จ่ายเท่านั้น ทำให้การใช้ งาน Cloud นั้นมีค่าใช้จ่ายต่ำกว่าการลงทุนซื้อระบบมาติดตั้งเอง แม้ในบางกรณีค่าใช้จ่ายของระบบ Cloud อาจจะไม่ได้ต่ำกว่าการ ลงทุนแบบ On-Premise อย่างมีนัยนัก แต่องค์กรก็ยังได้ประโยชน์ ในแง่ของการบริหารต้นทุน เนื่องจากการใช้จ่ายทางด้านระบบ Cloud จะถูกมองเป็นค่าใช้จ่าย (OpEx) แทนที่จะเป็นการลงทุน (CapEx)

4) การพัฒนาและปรับปรุงอย่างต่อเนื่อง (Development And Continuous Delivery)คงปฏิเสธไม่ได้ว่าเทคโนโลยีทางด้าน IT มีการเปลี่ยนแปลงไปค่อนข้างมากโดยเฉพาะในช่วง 2-3 ปีมานี้ แนวทางในการพัฒนา ซอฟต์แวร์ก็มีการเปลี่ยนแปลงเช่นเดียวกัน ระบบ Cloud จะเอื้อให้ปรับเปลี่ยนรูปแบบการพัฒนาซอฟต์แวร์จากเดิมที่เป็นการพัฒนา แบบ Waterfall ซึ่งกว่าจะเห็นโปรแกรมต้นแบบ (Prototype) ก็ ต้องใช้เวลาหลายเดือน เปลี่ยนมาเป็นการพัฒนาแบบใหม่ที่ใช้แนวคิดเรื่อง Agile และ Continuous Delivery ที่แบ่งซอฟต์แวร์ เป็นโมดูลย่อย และมีการออกเวอร์ชั่นใหม่อยู่เรื่อย ซึ่งวิธีการนี้มี ข้อดี คือ ผู้ใช้จะเห็นความสามารถใหม่ๆ ตลอดเวลา หากพบว่ามี Feature/Function ที่ไม่เป็นไปตามความคาดหวังก็สามารถปรับ เปลี่ยนได้ทันที ไม่จำเป็นต้องรอไปหลายเดือนซึ่งกว่าจะรู้ตัวก็ไม่ สามารถทำอะไรได้มากนัก เพราะการปรับเปลี่ยนอาจส่งผลกระทบ ต่อโครงสร้างโดยรวมของทั้งระบบ

2.2 openstack

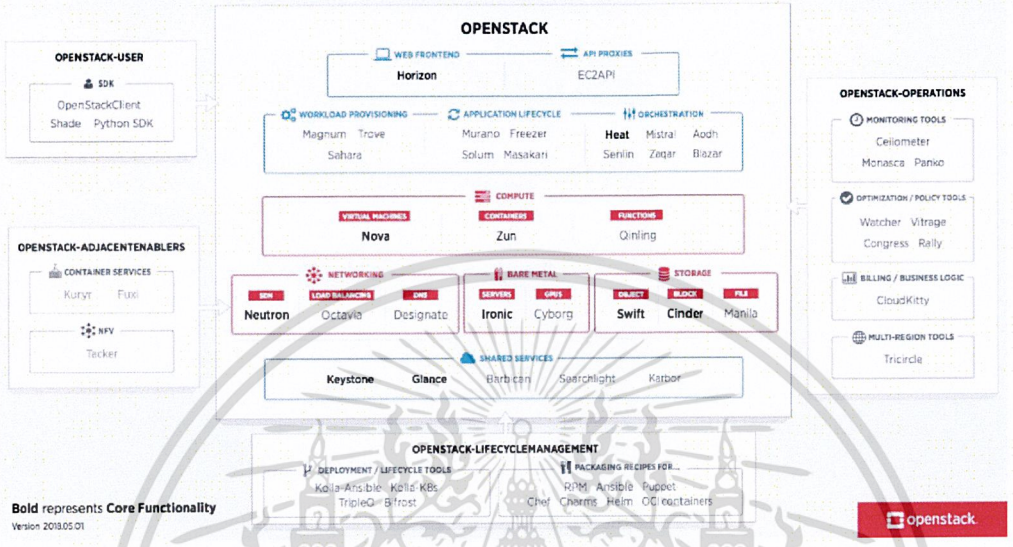


รูปที่ 2.2 openstack diagram[13]

Openstack [14] คือ Cloud Operating System แบบ Open Source ทำหน้าที่บริหารจัดการทรัพยากรด้าน การประมวนผล(compute), อุปกรณ์จัดเก็บข้อมูล(storage), Networking ของ Data Center ให้มีประสิทธิภาพในการทำงานสูงที่สุด แบ่งการทำงานออกเป็น Module ซึ่งมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่การทำงานที่แตกต่างกัน เพื่อความง่ายในการบริหารจัดการทรัพยากรในแต่ละด้าน และคงไว้ซึ่งประสิทธิภาพสูงสุดในการทำงานของระบบ โปรแกรมนี้มีส่วนประกอบ Module ดังต่อไปนี้



รูปที่ 2.3 Module การทำงานของ openstack[14]

2.2.1) Module การทำงานของ openstack

1) Nova Module เป็นพื้นฐานของการทำงานในส่วน Compute ของ Software OpenStack แบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนการทำงาน API และ Compute Worker ซึ่งทั้ง 2 ส่วนนี้ ทำหน้าที่ควบคุมการประมวลผล

2) Neutron Module เป็นพื้นฐานของการทำงานในส่วน Network ของ Software openstack ผสานการทำงานของ Physical Network เพิ่มประสิทธิภาพการทำงานด้วยคุณสมบัติ Network Management ด้วยเทคนิคการทำงานของ Virtual Interface ในรูปแบบต่าง ๆ ไม่ว่าจะเป็น Router, Switch, Network Interface Card หรือ Firewall

3) Swift Module เป็นพื้นฐานของการทำงานในส่วน Storage ในรูปแบบ ObjectStorage ของ openstack ที่มาพร้อมคุณสมบัติ Scalability, Redundancy และ Durability โดยพัฒนาโครงสร้างการจัดเก็บข้อมูลต่าง ๆ ตามประเภทของข้อมูล เช่น Images, Photo Storage,Email Storage, Backups และ Archives พร้อมทั้งการบันทึกข้อมูลต่าง ๆ แบบ Multiple Disks

4) Cinder Module เป็นพื้นฐานของการทำงานในส่วน Storage ในรูปแบบ Block Storage ของ OpenStack ซึ่งเป็นการทำงานร่วมกับ Nova Module โดยลักษณะการทำงานของ Cinder Module สามารถแบ่งออกได้ 2 รูปแบบ คือ

4.1) แบบ Ephemeral เป็นลักษณะการเก็บข้อมูลชั่วคราวซึ่งข้อมูลเหล่านั้นจะหายไปทันทีเมื่อมีการหยุดการทำงานของ Instance

4.2) แบบ Persistant เป็นลักษณะการเก็บข้อมูลแบบถาวร ซึ่งข้อมูลเหล่านั้นจะไม่สูญหายไป แม้ว่า Instance จะกำลังทำงานอยู่ หรือยุติการทำงานไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงลิขสิทธิ์ของมหาวิทยาลัยเชียงใหม่ทุกครั้งที่มีการนำไปใช้

5) Keystone Module เป็นระบบพื้นฐานที่ใช้ในการจัดการสิทธิ์การใช้งานแต่ละ Module ของ OpenStack โดยจะดำเนินการตรวจสอบสิทธิ์การใช้งาน Module ของผู้ใช้ เพื่อระบุตัวตน และความสามารถในเรียกใช้งานแต่ละ Module ของผู้ใช้ ช่วยเพิ่มประสิทธิภาพด้านเสถียรภาพและความปลอดภัยของ OpenStack

6) Glance Module จัดเป็นระบบ Stored Images ของ OpenStack ทำหน้าที่จัดเก็บ Server Image ต่าง ๆ ที่จัดเตรียมไว้เพื่อรองรับความต้องการใช้งาน Server Image ของระบบ ไม่ว่าจะเป็นการค้นหา บันทึก หรือ เรียกใช้ อีกทั้งความสามารถในการ Copy หรือ Snapshot Disk และ Server Image รวมทั้งการจัดเก็บข้อมูลแบบ Object Storage อีกด้วย

7) Horizon Module จัดเป็นระบบ Dashboard ที่ OpenStack พัฒนาขึ้นเพื่อให้ผู้ดูแลระบบ และผู้ใช้ สามารถบริหารจัดการทรัพยากรด้าน Compute ,Storage, Networking ได้อย่างมีประสิทธิภาพ

8) Heat Module จัดเป็น Template-Driven Engine ที่สามารถอนุญาตให้ผู้พัฒนาโปรแกรมสามารถ ตั้งค่า/ปรับแต่ง Template โครงสร้างของทรัพยากรแต่ละด้าน (Compute, Storage และ Networking) เพื่อรองรับการใช้งาน Cloud Application ของผู้ใช้ ซึ่งรวมไปถึงคุณสมบัติ Autoscaling เมื่อนำ Heat Module และ Ceilometer Module มาทำงานร่วมกัน อีกทั้งยังสามารถตั้งค่า Scaling Group ของทรัพยากรของแต่ละ Template

2.2.2) Enterprise Cloud by OpenStack [15]

แนวความคิดและเทคโนโลยีใหม่ที่ได้รับความสนใจในปัจจุบัน และมีแนวโน้มที่สูงขึ้นอีกในอนาคต ด้วยรูปแบบการพัฒนาและความสามารถในการทำงานของเทคโนโลยีใหม่นี้ ทำให้มีจำนวนผู้สนใจและจำนวนผู้ใช้งานเพิ่มมากขึ้นในทุก ๆ วงการ แนวความคิดและเทคโนโลยีที่ว่านั้น คือ ระบบประมวลผลกลุ่มเมฆ ซึ่งเป็นเทคโนโลยีที่เน้นการใช้งานง่าย และสามารถจัดการกับทรัพยากรต่าง ๆ ได้ด้วยตนเอง ไม่ว่าจะเป็น Compute , Network หรือ Storage นำมาซึ่งความสามารถในการพัฒนาปรับปรุง Infrastructure ให้สามารถ ปรับเพิ่มขึ้น หรือ ลดลง ได้อย่างง่ายดาย ปัจจุบันมีการพัฒนา Software ต่าง ๆ มากมาย เพื่อให้สอดคล้องกับแนวความคิดของ ระบบประมวลผลกลุ่มเมฆทั้งแบบ Open Source และแบบ License ด้วยเหตุนี้จึงทำให้องค์กรมากมายให้ความสนใจและนำแนวความคิดของ ระบบประมวลผลกลุ่มเมฆมาพัฒนาเป็น Platform ต่าง ๆ เพื่อใช้ในจัดสรรในส่วนของ Infrastructure , Platform และ Software ของระบบ IT แต่ละองค์กร ในรูปแบบระบบเสมือนและสามารถเรียกใช้งานได้ทันที เมื่อมีความต้องการใช้งาน อีกทั้งยังมี “ความยืดหยุ่นในการปรับเพิ่มหรือลดขนาดโครงสร้างพื้นฐาน” ให้เป็นไปตามความต้องการใช้งานที่แท้จริงของผู้ใช้งาน*โฉมใหม่* ของระบบ IT ที่สามารถปรับเปลี่ยนได้ตามความต้องการใช้งาน ทำให้องค์กร หรือบริษัทที่มีการลงทุนด้าน Cloud Platform สามารถที่จะวางแผนการบริหารงาน ในการพัฒนาดำเนินธุรกิจได้อย่างต่อเนื่องและรวดเร็ว ปรับเปลี่ยนภาพลักษณ์ของระบบ IT แบบเดิม ๆ ให้มีประสิทธิภาพ มีความพร้อม ที่จะรองรับการปรับเปลี่ยนแผนการดำเนินธุรกิจขององค์กร หรือบริษัท โดยมุ่งเน้นการตัดสินใจ และความรวดเร็วในดำเนินงานตามเป้าหมายที่มีการปรับเปลี่ยนอยู่ตลอดเวลา เพื่อให้ทันต่อเหตุการณ์ มีสอดคล้องกับการแข่งขันในตลาดการค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3) เหตุผลที่เหล่าองค์กรและผู้ดูแลระบบ ควรเริ่มต้นใช้ openStack สร้าง ระบบประมวลผลกลุ่มเมฆแบบส่วนตัว ภายในองค์กร [16]

1) เรียนรู้และลองผิดลองถูกกับการสร้าง Private Cloud ขั้นตอนหนึ่งที่กินเวลามาก ๆ ของการเริ่มต้นนำระบบ Cloud ในระดับ Infrastructure as a Service (IaaS) หรือ Platform as a Service (PaaS) มาใช้นั้นก็คือการลองผิดลองถูก ไม่ว่าจะเป็นการคัดเลือกเทคโนโลยี, การปรับแต่งการตั้งค่า, การปรับแต่ง Workflow หรือ Process ให้เหมาะสมกับธุรกิจ, การหา Plugin เชื่อมต่อที่ตอบโจทย์ความต้องการขององค์กร และการแก้ปัญหาเฉพาะหน้าต่าง ๆการเริ่มต้นลองใช้ OpenStack ให้เร็วขึ้นนั้นจะหมายถึงการที่องค์กรและผู้ดูแลระบบจะได้ลองผิดลองถูกก่อนกับเทคโนโลยีที่ใหม่และมีสโคปใหญ่อย่าง Private Cloud เพื่อค้นหาว่าความต้องการในเชิงธุรกิจที่ OpenStack จะสามารถมาช่วยตอบโจทย์ให้กับองค์กรนั้นได้คืออะไร ในขณะที่ผู้ดูแลระบบเองก็จะได้สัมผัสและเรียนรู้กับเทคโนโลยีใหม่ๆ เพื่อปรับตัวไปในเวลาเดียวกัน

การเรียนรู้ตรงนี้ถือเป็นส่วนที่สำคัญมาก เพราะถ้าหากองค์กรเริ่มมีความต้องการระบบ Private Cloud หรือ IT Infrastructure ที่มีความยืดหยุ่นภายในองค์กรสำหรับใช้งานแล้ว ความรู้ในส่วนนี้จะเป็นประโยชน์อย่างมากในการออกแบบและกำหนดสโคปของงานทั้งในเชิงธุรกิจและเชิงเทคนิค และทำให้การปรับปรุง IT Infrastructure ครั้งใหญ่ขององค์กรลุล่วงไปได้ด้วยดี แน่นอนว่าด้วยข้อดีของ OpenStack ที่เป็น Open Source นี้ก็ช่วยให้องค์กรสามารถประหยัดค่าใช้จ่ายไปได้มากทีเดียวในระหว่างลองผิดลองถูกครั้งนี้

2) สร้าง Environment สำหรับให้องค์กรได้ลองผิดลองถูกในการสร้างเทคโนโลยีใหม่ๆ ทั้งในแง่ของเทคโนโลยีและกระบวนการไม่เพียงแต่การลองผิดลองถูกในเชิง Infrastructure เท่านั้น หนึ่งในสาเหตุหลักที่เหล่าองค์กรเริ่มนำ OpenStack ไปใช้งานนั้นก็เพื่อสร้าง IT Infrastructure สำหรับทดลองเทคโนโลยีใหม่ๆ หรือพัฒนาเทคโนโลยีของตัวเองขึ้นมาทดสอบได้อย่างง่ายดาย ไม่ว่าจะเป็นการทดลองใช้งานระบบ Big Data Analytics อย่าง Apache Hadoop, Cloudera หรืออื่น ๆ, การทดสอบพัฒนา Mobile Application หรือ Web Application ใหม่ขึ้นมาเพื่อให้บริการลูกค้า, การทดสอบเทคโนโลยี Database แนวคิดใหม่ๆ หรือแม้แต่การทดสอบเทคโนโลยี Open Source หรือ Commercial ใหม่ๆ ให้ได้อย่างคล่องตัวนั่นเอง

การลองผิดลองถูกถือเป็นขั้นตอนที่สำคัญมากสำหรับทุก ๆ การพัฒนาเทคโนโลยีใหม่ๆ และการพัฒนาเทคโนโลยีใหม่ๆ เองก็เป็นส่วนสำคัญของการก้าวไปสู่การทำ Digital Transformation ดังนั้นการสร้างสนามสำหรับลองผิดลองถูกโดยเฉพาะนี้จึงเป็นสิ่งสำคัญอันดับต้นๆ ของหลายๆ องค์กรไปแล้วในเวลานี้ และแน่นอนว่านอกจากการลองผิดลองถูกในเชิงเทคโนโลยีแล้ว การลองผิดลองถูกในการวางกระบวนการ DevOps เพื่อให้เหล่าผู้ดูแลระบบและเหล่านักพัฒนาสามารถสร้างสรรค์เทคโนโลยีใหม่ๆ ขึ้นมาได้ภายใต้กระบวนการที่เป็นระบบก็ถือเป็นอีกปัจจัยที่สำคัญในระยะยาว เพราะการออกแบบกระบวนการการพัฒนา Software ไปจนถึงการ Deploy ระบบที่เป็นมาตรฐานนั้น จะเป็นตัวเร่งให้การพัฒนาเทคโนโลยีต่างๆ มีความรวดเร็วสูงยิ่งขึ้นนั่นเอง ซึ่ง OpenStack เองนั้นก็รองรับการทำงานร่วมกับเครื่องมือต่างๆ ในการทำ DevOps ได้เป็นอย่างดี

3) เริ่มสร้างวัฒนธรรมองค์กรแบบ Self service เปลี่ยนแนวคิดในการทำงานของผู้ดูแลระบบ วัฒนธรรมองค์กรแบบ Self-Service นี้ถือเป็นอีกจุดเปลี่ยนของระบบ IT ภายในองค์กรเลยก็ว่าได้ และเป็นอีกวัฒนธรรมใหม่ที่องค์กรเองก็ต้องเรียนรู้จากการลองผิดลองถูกด้วยเช่นกัน เพราะต่อไปนั้นแนวโน้มของระบบ IT ต่างๆ ทั้งในระดับของ Data, Application และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Infrastructure นั้นก็มีแนวโน้มที่จะกลายเป็น Self Service มากขึ้นเรื่อยๆ และผู้ใช้งานทั่วๆ ไปที่ไม่ใช่คนสาย IT เองก็ต้องเข้ามาใช้บริการเหล่านี้ด้วยเช่นกัน การเริ่มต้นลองผิดลองถูกด้วยระบบ Self-Service สำหรับให้บริการ IT Infrastructure แก่คนสาย IT ด้วยกันเองก่อนโดยใช้ OpenStack จึงถือเป็นทางเลือกที่น่าสนใจ

สิ่งที่องค์กรต้องเรียนรู้ในการสร้างวัฒนธรรม Self-Service ที่ประสบความสำเร็จนี้ ก็คือกระบวนการในการสร้าง Workflow สำหรับการออกแบบบริการ Self-Service แต่ละบริการให้ประสบความสำเร็จในเชิงการใช้งาน ไม่ว่าจะเป็นการเก็บความต้องการ, การออกแบบระบบและบริการ, การดูแลรักษาและติดตาม Resource ที่มีการใช้งาน, การวางแผนว่าควรจะปรับบริการ Self-Service เดิมที่มีอยู่อย่างไร และการ Maintenance หรือ Upgrade แต่ละบริการที่จะต้องมีการสื่อสารและวางแผนกันล่วงหน้าทั้งหมด Self-Service นั้นถือเป็นเทคโนโลยีที่จะมาเพิ่มความเร็วในการสร้างเทคโนโลยีใหม่ๆ ให้มีความง่ายดายมากยิ่งขึ้น และจำเป็นเป็นอย่างยิ่งสำหรับองค์กรที่ต้องการขับเคลื่อนด้วยเทคโนโลยีและข้อมูลในอนาคต แต่ในขณะเดียวกันการสร้างบริการ Self-Service ที่ดีให้ได้นั้นก็ไม่ใช่เรื่องง่ายเท่าไหร่ และเป็นสิ่งที่องค์กรควรต้องเริ่มเรียนรู้ด้วยการลองผิดลองถูกกับคน IT กันเองก่อนจะต้องเปิดบริการ Self-Service สำหรับผู้ใช้งานทั่วไปในอนาคต

4) เปิดทางเลือกใหม่ๆ สำหรับ IT Infrastructrure สำหรับองค์กรในอนาคตหนึ่งในสาเหตุที่ OpenStack นั้นกลายเป็นที่นิยมก็เป็นเพราะความที่ OpenStack นั้นเป็น Open Source Software ที่ไม่มีค่าใช้จ่ายในการจัดซื้อแต่อย่างใด และรองรับ IT Infrastructure ได้หลากหลาย รวมถึงมีเทคโนโลยีที่น่าสนใจให้พร้อมใช้ในตัว การเริ่มต้นเรียนรู้ Open Stack นั้นเรียกได้ว่าเป็นการเปิดโอกาสและทางเลือกใหม่ๆ สำหรับระบบ IT Infrastructure ที่จะมาช่วยประหยัดค่าใช้จ่ายให้กับองค์กรได้ในระยาวเลยก็ไม่ผิดนัก

จากเดิมที่ธุรกิจองค์กรมักจะคุ้นชินกับการซื้อ Server มาลง Hypervisor สร้างเป็นระบบ Virtualization และซื้อ Storage มาเชื่อมต่อเข้าไปในระบบ Virtualization กันเป็นหลักนั้น ก็จะกลายเป็นแค่ทางเลือกหนึ่งในอนาคตหลังการมาของ OpenStack เพราะ OpenStack นั้นมีครบถ้วนทั้งระบบ Compute, Storage, Network และ Management ในตัว ทำให้องค์กรสามารถเพิ่มทรัพยากรใด ๆ ก็ได้จากการลงทุนเพียงแค่ Server ซึ่งถือเป็น Component ที่มีราคาถูกที่สุดใน Data Center แล้วในเวลานี้ ซึ่งข้อดีตรงนี้ก็จะช่วยทำให้องค์กรวางแผนการลงทุนทรัพยากรต่าง ๆ ได้อย่างง่ายดายไปด้วยอีกทางในทางกลับกัน OpenStack นั้นสามารถใช้ในการควบคุมและบริการจัดการ Hypervisor ต่าง ๆ ได้หลากหลายค่าย รวมถึงสร้างระบบ Self Service ให้แก่ IT Infrastructure นั้น ๆ ได้ด้วย การนำ OpenStack มาใช้จึงไม่ต้องละทิ้งเทคโนโลยีเดิมที่มีการใช้งานอยู่แต่อย่างใด ในขณะที่ยังเปิดทางเลือกใหม่ในการเลือกใช้เทคโนโลยีได้หลากหลายยิ่งขึ้นกว่าเดิมในอนาคตไปพร้อม ๆ กัน ปัจจุบันผู้ผลิตรายใหญ่แทบทุกรายต่างก็ให้การสนับสนุน OpenStack กันทั้งนั้น ดังนั้นจึงไม่ต้องกังวลกันเลยในเรื่องของความเข้ากันได้กับระบบเดิม หรืออนาคตของ openStack เองก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

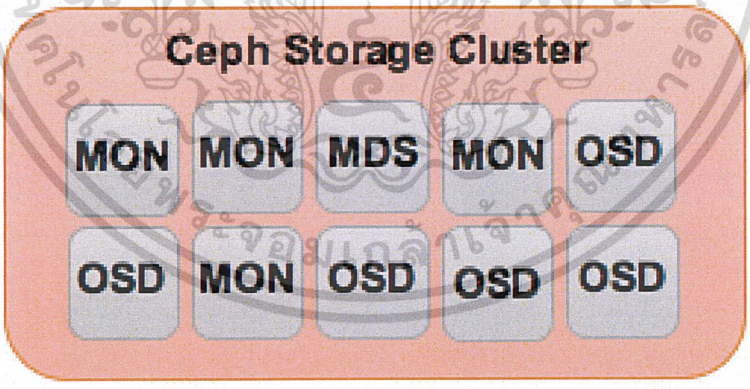
2.3 Ceph

รูปที่ 2.4 Ceph[17]

2.3.1) Ceph Storage is a Distributed Storage [18]

Ceph Storage สามารถเปลี่ยนโครงสร้างพื้นฐานด้านไอทีขององค์กรและความสามารถในการจัดการข้อมูลจำนวนมหาศาล หากองค์กรทำงานที่มีความต้องการ ใช้งานในการจัดเก็บข้อมูลที่แตกต่างกัน Ceph Storage สามารถตอบโจทย์ได้ Ceph Storage มีพื้นฐานคือ Reliable Autonomic Distributed Object Store (Rados) ซึ่งมีการใช้งานด้วย object, block, และ file system ในการจัดเก็บข้อมูลแบบคลัสเตอร์ (cluster) ทำให้ Ceph Storage มีความยืดหยุ่น ความน่าเชื่อถือสูงและง่ายต่อการจัดการ

Ceph Storage เป็นระบบ distributed storage แบบหนึ่ง หรือพูดง่ายๆ คือ เป็นหน่วยเก็บข้อมูล(storage) ที่ทำงานบน คลัสเตอร์(cluster) ของ computer node โดยในระบบ Ceph Storage ประกอบไปด้วย node 3 ประเภท แสดงดังรูป 2.5



รูปที่ 2.5 Ceph Storage Cluster

1) Monitor ทำหน้าที่ดูแลสถานะของคลัสเตอร์ (cluster)

2) OSD (Object Storage Device) ทำหน้าที่อ่าน/เขียนข้อมูลตามคำสั่งของผู้ใช้

3) MDS (Metadata Server) ทำหน้าที่ดูแลสถานะของ file hierarchy

มี node ประเภทนี้เฉพาะใช้งาน Ceph Storage กับ CephFS เท่านั้น เพราะว่า Ceph Storage ทำงานบนคลัสเตอร์(cluster) ที่แต่ละ node ทำหน้าที่แตกต่างกันอย่างชัดเจน ทำให้การ

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้เพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

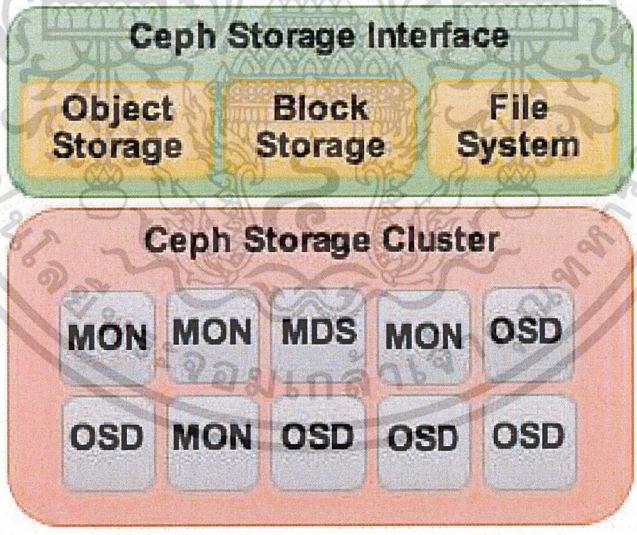
บริหารจัดการทำได้สะดวก workload ถูกกระจายออกไปอย่างเหมาะสม ลดความเสี่ยงจาก system failure และทำให้ scale ได้ง่ายอีกด้วย

2.3.2) Ceph Storage is Software-Defined Storage

Ceph Storage เป็น Software-defined storage (SDS) ที่ถูกออกแบบมาให้ติดตั้งได้ตั้งแต่บน All-flash server-class hardware ไปจนถึง Commodity off-theshelf hardware (computer ที่ใช้ทั่วไปตามบ้าน) หรือแม้กระทั่ง Single-board computer เช่น Raspberry Pi นอกจากนี้ Ceph Storage ไม่ได้ต้องการฮาร์ดแวร์แบบพิเศษ เหมือนระบบหน่วยเก็บข้อมูล(storage) แบบอื่น ๆ เพื่อที่จะใช้งานลักษณะต่าง ๆ เช่น RAID (Redundant Array of Independent Disks), replication, erasure codingหรือ storage virtualization ทำให้ผู้ใช้มีความอิสระในการเลือกฮาร์ดแวร์สำหรับหน่วยเก็บข้อมูล(storage) มากขึ้น และสามารถเลือกให้ตรงตามงบประมาณและ performance requirements ได้มากยิ่งขึ้นกว่าเดิม

2.3.3) Ceph Storage is Unified Storage System

Ceph Storage มาพร้อมกับ 3 interface ในการติดต่อเข้าใช้งานหน่วยเก็บข้อมูล(storage) โดยผู้ใช้สามารถเลือกได้ทั้ง Object storage interface, Block storage interface หรือ File system interface แสดงดังรูป 2.6



รูปที่ 2.6 Ceph Storage Interface

Ceph Object Storage รองรับทั้ง Amazon S3 และ OpenStack Swift protocol สามารถทำงานร่วมกับ application ที่รองรับ protocol เหล่านั้นได้ทันที

Ceph Block Storage รองรับการทำงานในลักษณะที่คล้ายกับ SAN (Storage Area Network) โดยสามารถสร้าง resizable logical block devices และ mount มาใช้งานที่เครื่อง client ปลายทางได้ และยังสามารถทำงานร่วมกับ KVM (Kernel-based Virtual Machine) เช่น QEMU (Quick Emulator) ได้อีกด้วย เหมาะเป็นอย่างยิ่งเมื่อนำไปใช้จัดการ virtual machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(VM) image เพราะสามารถทำ snapshot layering ได้ ทำให้สามารถ clone VM image ออกมาใช้งานได้อย่างรวดเร็วมาก

Ceph File System Storage รองรับการทำงานในลักษณะคล้าย NAS (Network-Attached Storage) คือ ผู้ใช้สามารถสร้าง CephFS (Ceph File System) ขึ้นมาใน Ceph Storage แล้ว mount ไปใช้งานยังเครื่องปลายทางได้เช่นเดียวกับ NAS ทั่วไปโดยใช้งานผ่าน CephFS Gateway เมื่อเปรียบเทียบกับ หน่วยเก็บข้อมูล(storage) อื่น ๆ ในท้องตลาดแล้ว นับว่า Ceph Storage นั้นเป็นหน่วยเก็บข้อมูล(storage) ที่มีความเพียบพร้อมมากเพราะมาพร้อมกับทั้ง 3 interface ในขณะที่เจ้าอื่น ๆส่วนใหญ่จะให้มามากที่สุดเพียง 2 interface เท่านั้น

2.3.4) Ceph Storage is a Greater Peace of Mind

Ceph Storage มีคุณสมบัติและความสามารถต่าง ๆ มากมายที่ช่วยรักษาข้อมูลให้คงทนถาวรมากขึ้น เช่น data replication, erasure coding, across-cluster mirroring ซึ่งส่วนใหญ่สามารถตั้งให้ทำงานได้โดยอัตโนมัติโดยที่ผู้ใช้ไม่ต้องทำอะไรเพิ่มพิเศษ เพียงแค่บันทึกข้อมูลลงไปในระบบข้อมูลนั้นจะถูก replicate หรือ กระจายไปเก็บหลายๆ ที่เองโดยอัตโนมัติความสามารถหนึ่งที่สำคัญและทำให้การเก็บข้อมูลมีความปลอดภัยมากขึ้นใน Ceph Storage คือ configurable data placement rule (CRUSH ruleset) กล่าวคือ ผู้ใช้สามารถกำหนดเองได้ว่าจะให้ข้อมูลกลุ่มไหน ถูก replicate หรือ กระจายไปเก็บไว้ที่ใดบ้าง เช่น ผู้ใช้สามารถกำหนดให้เก็บข้อมูลการจ่ายเงินไว้ 3 replica โดยต้องเก็บไว้คนละ host ที่อยู่คนละ room กันด้วย ในลักษณะนี้ก็สามารถกำหนดลงไปใน Ceph Storage ได้อย่างง่ายดาย ทำให้ผู้ใช้สามารถออกแบบการจัดวางข้อมูล (data placement) ได้อย่างมีประสิทธิภาพและตรงตาม data disaster และ recovery plan ของแต่ละองค์กร ความสามารถนี้เป็นไปได้เพราะความสามารถของ CRUSH

2.3.5) Ceph Storage is Designed for Scale

CRUSH เป็น data placement algorithm ที่ใช้คำนวณตำแหน่งการจัดวางข้อมูลใน Ceph Storage โดย algorithm นี้สามารถคำนวณได้ทันทีไม่ว่าจะเป็นบน OSD node หรือ client computer ของผู้ใช้ ทำให้ไม่จำเป็นต้องมี centralized service มาคอยเก็บว่าข้อมูลนี้เก็ไว้ที่ใด หาก service นี้เกิดปัญหาสามารถส่งผลให้ทั้งระบบหน่วยเก็บข้อมูล(storage) ใช้การไม่ได้ไปด้วย และการมี service แบบนี้ยังทำให้การ scale out ทำได้ลำบาก แต่ Ceph Storage นั้นไม่มีปัญหานี้ นอกจากนี้ CRUSH ยังช่วย distribute load ของข้อมูลออกไปในแต่ละ OSD node อย่างเท่า ๆกัน และยังสามารถกำหนด weight ที่แตกต่างกันบนแต่ละ OSD node ได้ ด้วยความสามารถนี้ ทำให้ไม่เกิดการคอขวดขึ้นในระบบ และช่วยลดความเสี่ยงในเรื่อง data loss ได้อีกด้วย เพราะหากข้อมูลมากระจุกตัวกันที่ node เพียงไม่กี่ node แล้ว node เหล่านั้นเกิดปัญหาขึ้น อาจทำให้สูญเสียข้อมูลจำนวนมากไปในคราวเดียวกันได้ เปรียบเหมือนการเอาไข่มาใส่ตะกร้าใบเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

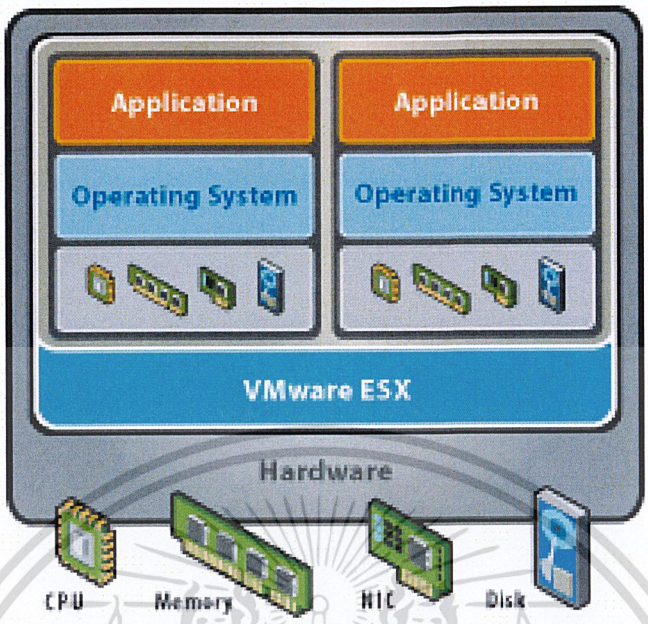
2.3.6) แล้ว Ceph Storage เหมาะกับอะไรบ้าง

ตัวอย่างของความสามารถของ Ceph Storage ข้างบนก็น่าจะพอเพียงที่จะทำให้หลายคนต้องหันมาสนใจ ติดตาม และทดลองใช้หน่วยเก็บข้อมูล(storage) ตัวนี้กัน โดยถ้าหากเปรียบเทียบกับหน่วยเก็บข้อมูล(storage) ในท้องตลาดแล้วจะพบว่า Ceph Storage มีคุณสมบัติและความสามารถที่มากกว่า หรือเทียบเท่าหน่วยเก็บข้อมูล(storage) ส่วนใหญ่ทั้งหมด และที่สำคัญที่สุด คือ Ceph Storage license นั้น ฟรี! ไม่มีค่าใช้จ่ายเนื่องจากเป็น Open Source แต่ก็ยังมีบริษัทมากมายที่ นำ Ceph มา package ร่วมเป็นส่วนหนึ่งของผลิตภัณฑ์ขายเช่นกัน ซึ่งก็จะเป็นทางออกให้องค์กรที่ต้องการใช้ Ceph แต่ยังต้องการcommercial support แต่สิ่งที่น่าสนใจที่สุดสำหรับ Ceph ก็คือเราสามารถปรับค่าของระบบให้สามารถทำงานได้ทั้งงานแบบที่ต้องการความเร็วของการเขียนข้อมูลสูงๆ (high write transaction rate หรือ IOPS สูง) อย่างในงานพวก Database หรือ Log เก็บข้อมูลทั้งหลาย ไปจนถึงงานที่ต้องการความเร็วในการอ่านพร้อมกับความปลอดภัยสูงๆ ซึ่งก็น่าจะเหมาะกับงานเก็บข้อมูลที่นาน ๆครั้งจะทำการเปลี่ยนแปลง ไม่ว่าจะเป็น ระบบเก็บข้อมูลของ e-commerce catalog หรืองานเก็บข้อมูลย้อนหลังทางบัญชีทั้งหลาย โดยมีการจัดเก็บข้อมูลหลายชุดพร้อม ๆกัน ซึ่งใช้ความสามารถในการทำ replication แบบอัตโนมัติของ Ceph เรียกได้ว่า Ceph ระบบเดียวสามารถนำมาใช้ตอบโจทย์ได้หลายอย่างทีเดียว

2.4 Virtualization Technology

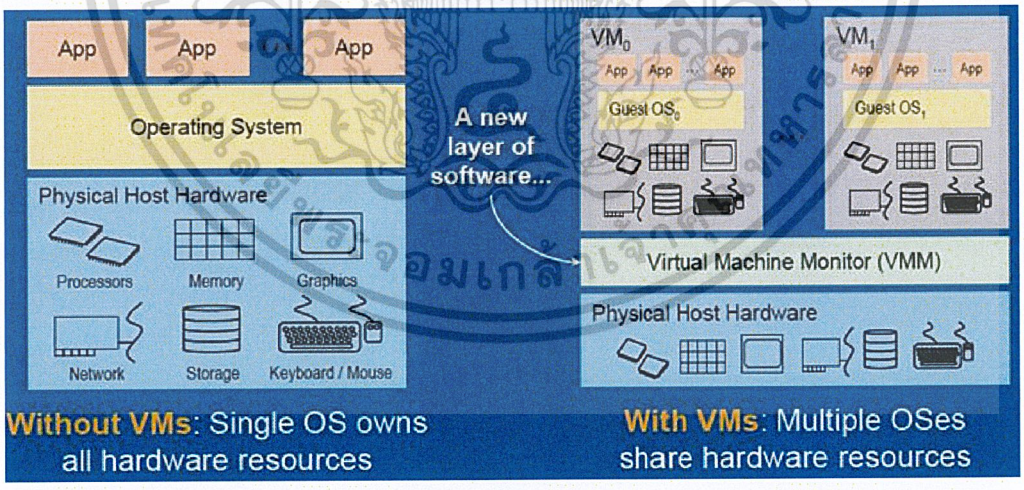
เมื่อกล่าวถึงเทคโนโลยี Virtualization Technology[19] ซึ่งปัจจุบันเริ่มเป็นที่รู้จักกันอย่างแพร่หลายนั้น ไม่ได้เป็นเทคโนโลยีใหม่เลย โดยในอดีตนั้น มีการใช้งานบนระบบขนาดใหญ่อยู่แล้ว แต่เมื่อมีผู้นำมาใช้งานบนระบบที่มีขนาดเล็กลง ก็เริ่มมีการใช้งานอย่างแพร่หลายมากขึ้น หลายบริษัทต่างพัฒนาผลิตภัณฑ์ของตัวเองออกมาแข่งขันกันในตลาด ที่มีชื่อเสียงมากก็คือ VMWare อีกตัวที่พอจะรู้จักก็คือ Virtual PC ซึ่งไมโครซอฟต์ได้นำไปพัฒนาต่อจนกลายเป็น Hyper V และอีกค่ายได้แก่ Citrix ซึ่งการทำงานหลักโดยรวมของแต่ผลิตภัณฑ์ก็คือ สามารถจำลองการทำงานของคอมพิวเตอร์ขึ้นมาหลายๆ เครื่องโดยไม่ได้จำลองเฉพาะซีพียูเท่านั้น ทำการนรัระบบปฏิบัติการหลายระบบได้พร้อมกัน บนเครื่องจริง ๆ เพียงเครื่องเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การทำงานของ Virtualization

สรุปได้ว่า Virtualization Technology คือ เทคโนโลยีที่ช่วยให้สามารถใช้ Resource เช่นซีพียู, หน่วยความจำ, ฮาร์ดดิสก์ เป็นต้น ของคอมพิวเตอร์ 1 เครื่องหรือมากกว่านั้น ให้สามารถรันซอฟต์แวร์ หรือ application ในปริมาณมาก ๆ หรือแม้แต่รันระบบปฏิบัติการหลายๆ ตัวให้สามารถทำงานพร้อมกันหลายๆ อย่างได้ ถึงแม้ว่าจะต่าง platform ก็ตาม



รูปที่ 2.8 เปรียบเทียบการทำงานระหว่างการนำ Virtualization มาใช้กับไม่ได้นำมาใช้

2.4.1) เหตุผลที่ใช้ Virtualization

1) ลดต้นทุนในการซื้อ Server เพียงตัวเดียวแต่สามารถลงได้หลายระบบปฏิบัติการ (OS) เช่น ลง Windows Server, Unix, Linux, Ubuntu, FreeDBS หรือ Solaris

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) รองรับการเชื่อมต่อเข้าจากเครื่อง Thin Client ได้จากทุกที่ทุกเวลา ง่ายต่อการจัดการดูแลระบบ

3) ง่ายต่อการโอนย้ายระบบ กรณีเปลี่ยน Server การดูแลรักษาเซิร์ฟเวอร์ได้ง่าย

4) เพิ่มประสิทธิภาพการทำงานของระบบโดยรวม

5) ตอบสนองทางธุรกิจได้อย่างเต็มประสิทธิภาพ

6) เมื่อลดจำนวนเครื่องเซิร์ฟเวอร์ลงได้ ก็ประหยัดพลังงานและลดค่าใช้จ่ายบางองค์กรมีเซิร์ฟเวอร์ และเครื่อง Client อยู่เป็นจำนวนมาก ซึ่งลงระบบปฏิบัติการได้เพียงตัวเดียว ก็ลดให้เหลือน้อยที่สุด เพื่อลดต้นทุนค่าไฟฟ้า และค่าใช้จ่ายในด้านบุคลากร เพราะเป็นระบบที่ง่ายต่อการดูแลระบบ ไม่ต้องใช้คนมากนัก

7) แบ่งทรัพยากรทางด้านเครือข่ายเพื่อให้สามารถใช้ทรัพยากรร่วมกันได้อย่างมีประสิทธิภาพ

2.4.2) องค์กรแบบใดที่เหมาะสำหรับการนำเทคโนโลยี Server Virtualization

1) องค์กร หรือหน่วยงานที่มีจำนวนเซิร์ฟเวอร์และเครื่อง client อยู่จำนวนมาก การใช้เทคโนโลยี Virtual Machine เป็นการลดจำนวนเซิร์ฟเวอร์ในองค์กรลงได้มาก ประหยัดงบประมาณด้านบำรุงรักษา ด้านบุคลากร สามารถจัดการได้ง่ายขึ้นและเป็นการรวมเซิร์ฟเวอร์ (Server Consolidation) ในองค์กรให้เหลือน้อยลงเป็นหนทางที่ดีที่สุด

2) หน่วยงานที่มีผู้ใช้งานซอฟแวร์หลายๆ ลักษณะงาน ต้องใช้ระบบปฏิบัติการต่างระบบกันมาก ๆ

3) บริษัทที่รับพัฒนาซอฟแวร์ เพราะจะช่วยในเรื่องของการทดสอบระบบจะทำได้ง่ายขึ้น ยิ่งต้องทดสอบบนระบบปฏิบัติการ (OS) ที่แตกต่างกันยิ่งใช้ประโยชน์ของการทำ Server Virtualization ได้มากขึ้น

4) หน่วยงานการศึกษา เพราะมีนักศึกษาและบุคลากรอยู่เป็นจำนวนมาก

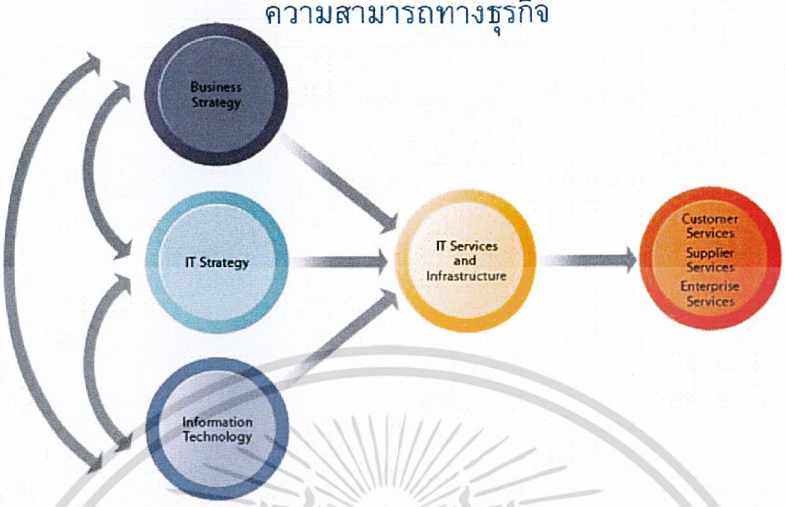
5) หน่วยงานทั่วไปที่ต้องการประหยัดค่าใช้จ่ายในด้านซอฟแวร์และฮาร์ดแวร์ ค่าไฟฟ้า, ประหยัดพื้นที่

2.5 โครงสร้างพื้นฐานทางไอที (IT Infrastructure)

ความหมายของโครงสร้างพื้นฐานทางไอที (IT Infrastructure) [20]คือ สิ่งอำนวยความสะดวกทางกายภาพต่าง ๆ รวมถึงการให้บริการและการจัดการที่นำมาสนับสนุนการปฏิบัติงานได้ทั่วองค์กร รวมถึงการลงทุนโดยการวาง รากฐาน งานบริการให้แก่ลูกค้า การทำงานร่วมกับผู้ขายปัจจัยการผลิต และการนำมาใช้เพื่อจัดการกระบวนการทางธุรกิจภายในองค์กร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมโยงระหว่างองค์กรกับโครงสร้างพื้นฐานทางไอที และ
ความสามารถทางธุรกิจ



รูปที่ 2.9 การเชื่อมโยงระหว่างองค์กรกับโครงสร้างพื้นฐานทางไอที และความสามารถทางธุรกิจ

2.5.1) ปัจจัยที่ผลักดันให้เกิดวิวัฒนาการของโครงสร้างพื้นฐานทางด้านไอที (IT Technology drivers of infrastructure evolution)

1) การพัฒนาความสามารถในการประมวลผลของไมโครโพรเซสเซอร์

2) กฎของ Moore (พลังของไมโครโพรเซสเซอร์จะเพิ่มขึ้นสองเท่าในทุก ๆ 18เดือน, ประสิทธิภาพในการประมวลผลจะเพิ่มเป็นสองเท่าในทุก ๆ 18 เดือน, ราคาของการประมวลผลจะลดลงครึ่งหนึ่งในทุกๆ 18 เดือน)

3) เทคโนโลยีนาโน (Nanotechnology) ทำให้สามารถผลิตชิพคอมพิวเตอร์และอุปกรณ์อิเล็กทรกนิกส์ต่าง ๆ ที่มีขนาดเล็กลงกว่าเดิมมาก

2.5.2) การจัดเก็บข้อมูลดิจิตอลเป็นจำนวนมาก

ปริมาณข้อมูลดิจิตอลมีอัตราการเพิ่มขึ้นอย่างมาก ทำให้มีการพัฒนาขีดความสามารถในการจัดเก็บข้อมูลของอุปกรณ์

2.5.3) ปัจจัยที่ผลักดันให้เกิดวิวัฒนาการของโครงสร้างพื้นฐานทางด้านไอที

1) ค่าใช้จ่ายในการสื่อสารข้อมูลลดลง การเติบโตอย่างรวดเร็วของระบบอินเทอร์เน็ตผู้ใช้ที่สามารถใช้งานระบบอินเตอร์เน็ตเพิ่มมากขึ้น จำนวนเว็บไซต์ต่างๆก็เพิ่มมากขึ้น เนื่องจากค่าใช้จ่ายในการสื่อสารข้อมูลลดลง

2) การกำหนดมาตรฐานของเทคโนโลยีข้อ ตกลงระหว่างบริษัทผู้ผลิต และผู้บริโภคทั่วไป ในการยอมรับมาตรฐานของเทคโนโลยีต่าง ๆ ส่งผลให้ค่าใช้จ่ายในการผลิตสินค้าลดลง เนื่องจากบริษัทสามารถผลิตสินค้าที่มีมาตรฐานเดียวกัน สามารถทำงานร่วมกัน และสื่อสารกันผ่านระบบเครือข่ายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4) ส่วนประกอบของโครงสร้างพื้นฐานทางด้านไอที (IT Infrastructure)

1) ฮาร์ดแวร์คอมพิวเตอร์ (Computer Hardware Platform)

2) ซอฟต์แวร์ระบบ (Operating System Platforms)

3) ซอฟต์แวร์ประยุกต์ระดับองค์กร (Enterprise Software Application)

4) การจัดการข้อมูลและการเก็บข้อมูล (Data management and storage)

5) การสื่อสารโทรคมนาคมและเครือข่าย (Networking / Telecommunications Platforms)

6) อินเตอร์เน็ต (Internet Platforms)

7) ที่ปรึกษาทางด้านระบบ (Consulting and system Integration services)

2.6 อูบันตู (Ubuntu)

รูปที่ 2.10 โลโก้ Ubuntu

Ubuntu [21] เป็นระบบปฏิบัติการคอมพิวเตอร์ที่เป็นระบบปฏิบัติการแบบเปิดซึ่งมีพื้นฐานบนลินุกซ์ดิสทริบิวชันที่พัฒนาต่อมาจากเดเบียน การพัฒนาสนับสนุนโดยบริษัท Canonical Ltd ซึ่งเป็นบริษัทของนายมาร์ก ชัทเทิลเวิร์ธ ชื่อของดิสทริบิวชันนั้นมาจากคำในภาษาซูลู และภาษาโคซา (ภาษาในแอฟริกาใต้) ว่า Ubuntu ซึ่งมีความหมายในภาษาอังกฤษคือ "humanity towards others อูบุนตูต่างจากเดเบียนตรงที่ออกรุ่นใหม่ทุก 6 เดือน และแต่ละรุ่นจะมีระยะเวลาในการสนับสนุนเป็นเวลา 18 เดือน รุ่นปัจจุบันของ Ubuntu คือ 15.10ซอฟต์แวร์ต่างๆ ที่รวมมาใน อูบุนตูนั้นเป็นซอฟต์แวร์เสรีเกือบทั้งหมด(มีบางส่วนที่เป็นลิขสิทธิ์ เช่น ไดรเวอร์) โดยจุดมุ่งหมายหลักของ อูบุนตูคือเป็นระบบปฏิบัติการสำหรับคนทั่วไป ที่มีโปรแกรมทันสมัย และมีเสถียรภาพในระดับที่ยอมรับได้

2.6.1) ประวัติและลำดับการพัฒนา

Ubuntu เปิดตัวเป็นครั้งแรกเมื่อวันที่ 20 ตุลาคม ค.ศ. 2004 โดยเริ่มจากการแยกตัวชั่วคราวออกมาทำจากโครงการ Debian GNU/Linux เมื่อเสร็จสิ้นคราวนั้นแล้วก็ได้มีการออกตัวใหม่ๆทุก 6 เดือน และมีการอับเดตระบบอยู่เรื่อยๆ Ubuntu เวอร์ชันใหม่ๆที่ออกมาก็ได้ใส่ GNOME เวอร์ชันล่าสุดเข้าไปด้วย โดยแผนการเปิดตัวทุกครั้งจะออกหลังจาก GNOME ออกหนึ่งเดือน ซึ่งตรงข้ามกับทางฝั่งที่แยกออกมาจาก Debian อื่น ๆ เช่นพวก MEPIS, Xandros, Linspire, Progeny และ Libranet ทั้งหมดล้วนมีกรรมสิทธิ์ และไม่เปิดเผยCode ซึ่งเป็นส่วนที่อยู่ในรูปแบบธุรกิจ Ubuntu เป็นตัวปิดฉากหลักการของ Debian และมีการใช้งานฟรีมากที่สุดในเวลานี้ โลโก้ของ Ubuntu ยังคงใช้รูปแบบเดิมตั้งแต่เปิดตัวครั้งแรก ซึ่งสร้างโดย แอนดี้ ฟิสสิมอน ฟอนต์ได้รับการแจกมาจาก Lesser General Public License แล้วก็ได้มาเป็นโลโก้ Ubuntu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบต่างๆของUbuntu ส่วนใหญ่มีพื้นฐานมาจากความไม่แน่นอนของ Debian โดยทั้งสองใช้ Debian's deb package format และ APT/Synaptic เป็นตัวจัดการการ ติดตั้งส่วนประกอบต่างๆ

2.6.2) หลักการสำคัญ

1) นักพัฒนา Ubuntu จำนวนมากมาจากชุมชนเดเบียนและ GNOME โดยการออก Ubuntu รุ่นใหม่จะตรงกับรุ่นใหม่ของ GNOME อยู่เสมอ มีนักพัฒนาอีกหลายกลุ่มพยายามที่จะใช้ KDE กับ Ubuntu และทำให้เกิดโครงการ Kubuntu ขึ้น นอกจากนี้ยังมี

- โครงการ Xubuntu สำหรับ XFCE และตัว Shuttleworth เองยังประกาศ
- โครงการ Gnubuntu ซึ่งใช้ซอฟต์แวร์เสรีทั้งหมด ตามอุดมคติของริชาร์ด สตอลแมน และโครงการ Edubuntu ซึ่งเป็นลีนุกซ์ที่ใช้ภายในโรงเรียนอีกด้วย

2) Ubuntu นั้นเน้นในเรื่องความง่ายในการใช้งานเป็นหลัก ใช้เครื่องมือ sudo สำหรับงานบริหารระบบ เช่นเดียวกับ Mac OS X3) รองรับการทำงานกับทั้ง CPU ชนิด 32bit , 64bit และ CPU แบบ ARM

4) รูปแบบการติดตั้งแบบ Live CD ที่รันระบบปฏิบัติการจากแผ่นซีดี ให้ทดลองใช้ก่อนการติดตั้งจริง

5) รูปแบบการติดตั้งแบบ Live USB ที่รันระบบปฏิบัติการจากแฟลชเมมโมรี่ ให้ทดลองใช้ก่อนการติดตั้งจริง

6) ทุกโครงการของ Ubuntu นั้นไม่เสียค่าใช้จ่ายในการใช้งาน ผู้ใช้ทุกคนจากทุกประเทศสามารถขอรับซีดี Ubuntu ได้ฟรี (ทาง Ubuntu จะเป็นฝ่ายเสียค่าจัดส่งให้ทางไปรษณีย์) ใต้ชื่อโครงการ Ubuntu Shipit โครงการนี้ยังแบ่งย่อยเป็น Kubuntu Shipit, และ Edubuntu Shipit ด้วย แตทว่า Edubuntu ShipIt ได้ปิดตัวลงไปตั้งแต่ออกเวอร์ชัน 8.10 มา

7) ส่วนติดต่อผู้ใช้หลังจากติดตั้งเสร็จจะเป็นสีน้ำตาลและส้ม ใช้ชื่อชุดตกแต่งนี้ว่า Human ซึ่งสามารถปรับเปลี่ยนได้ แตในเวอร์ชัน 10.04 ได้เปลี่ยนโทนสีทั้งหมดเป็นสีดำม่วงและส้ม

8) ใช้ระบบ APT และ Synaptic ในการจัดการโปรแกรมของระบบ

9) ลินุกซ์ทะเล (Linux TLE) ซึ่งเป็นลินุกซ์ที่พัฒนาโดยคนไทย ก็ได้ใช้ Ubuntu เป็นฐานในการพัฒนา ตั้งแต่ลินุกซ์ทะเลเวอร์ชัน 8.0 เป็นต้นมา

2.7 Ansible

Ansible [22] คือ เครื่องมือที่ใช้สำหรับการจัดการ configuration โดยจะทำงานผ่านทาง ssh พัฒนาด้วย ภาษา python การทำงานจะงานในรูปแบบที่ต้องมี management node ที่ทำหน้าเป็น server หลังที่จะทำหน้าเป็น ผู้ที่จะต้องไปติดตั้ง package บน เครื่องserverในระบบ การทำงานจะประกอบด้วย file ที่เกี่ยวข้อง 2 file คือ inventory file และ playbook

2.7.1) inventory file ทำหน้าเก็บรายชื่อของ server หรือ ip ของ server ที่จะใช้สำหรับเป้าหมายสำหรับการติดตั้ง โดยdefault หาไม่มีการกำหนด ใน option -i ก็จะใช้ inventory คือ /etc/ansible/hosts เราสามารถที่จะแก้ไขค่านี้ได้โดยทำการแก้ไข file/etc/ansible/ansible.cfg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2) playbook.yml ไฟล์นี้จะเป็นตัวระบุว่า จะให้ ansible ทำอะไรบ้างในแต่ละ server การเขียนไฟล์จะใช้เป็น format ของ yaml format โดยสามารถเขียนแบบที่เป็น static และ

dynamic ก็ได้โดยถ้าหาเขียนแบบ dynamic โดยใช้ jinja 2 templateในการทดสอบการใช้งาน จะทดสอบการใช้งานผ่านทาง Vagrantfile โดยจะเป็นการสร้าง infrastructure ด้วย Vagrantfile

2.8 งานวิจัยที่เกี่ยวข้อง

การพัฒนา IT Infrastructure ในงานวิจัยนี้ได้ทำการศึกษางานวิจัยที่เกี่ยวข้องและบทความด้าน หลักการทำงานของ Cloud computing ในระดับองค์กร การนำ Software Openstack ซึ่งเป็นโอเพ่นซอสมาใช้ในองค์กร

พันตรี อนันต์ สมไร่ขิง(2556) [23]เรื่อง “การเปรียบเทียบการทำงานของเซิร์ฟเวอร์การจัดเก็บข้อมูลแบบคลาวด์ระหว่างฮาร์ดแวร์ทางกายภาพกับฮาร์ดแวร์เสมือน” โดยงานวิจัยนี้ได้จัดทำเครื่องเซิร์เวอร์ การจัดเก็บข้อมูลแบบคลาวด์ขึ้นด้วยฮาร์ดแวร์ 2 แบบ คือ ฮาร์ดแวร์ทางกายภาพและฮาร์ดแวร์เสมือนเพื่อทดลองเพื่อทดลองและเทียบผลการทำงานของฮาร์ดแวร์ทั้ง 2 แบบ โดยมีการทดลอง 2 วิธี คือวิธีแรกทำการจำลองสถานการณ์ให้เครื่องเซิร์ฟเวอร์รองรับการทำงานถ่ายโอนข้อมูลจากเครื่องลูกข่ายพร้อมๆกันด้วยจำนวนเครื่องลูกข่ายปริมาณต่างๆ และวิธีที่ 2 ทำการจำลองสร้างการเชื่อมต่อในปริมาณต่างๆ เพื่อดูผลการทำงานของเครื่องเซิร์ฟเวอร์ที่เกิดขึ้นตามจำนวนการเชื่อมต่อในจำนวนต่างๆ ทั้งนี้ได้ดำเนินการทดลองภายใต้โครงสร้างพื้นฐานภายในกรมเทคโนโลยีสารสนเทศและอวกาศกลาโหม

สุชาติ คุ้มมะณี(2556) [24] เรื่อง “ความมั่นคงของเทคโนโลยีการประมวลผลกลุ่มเมฆ” บทความนี้กล่าวถึง ปัญหา และแนวทางการแก้ไขปัญหาด้านความมั่นคงของเทคโนโลยีการประมวลผลกลุ่มเมฆ ซึ่งเป็นระบบคอมพวเตอร์ที่มีสมรรถนะสูง และกำลังได้รับความสนใจเป็นอย่างมากในปัจจุบัน วัตถุประสงค์ของเทคโนโลยีดังกล่าวเพื่อตอบสนองต่อการประมวลผลข้อมูลที่มีปริมาณมาก ซับซ้อน ยืดหยุ่นต่อการใช้งาน และลดต้นทุนของการประมวลผลลง ปัจจุบันเทคโนโลยีดังกล่าวยังอยู่ในช่วงเริ่มต้นการใช้งานจริง ด้วยเหตุนี้ปัญหาด้านความมั่นคงยังคงเป็นประเด็นปัญหาที่สำคัญและน่าสนใจเป็นอย่างยิ่งในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงานวิจัย

3.1 ขั้นตอนการศึกษาและรวบรวมข้อมูล

1) ศึกษาปัญหาและรวบรวมความต้องการขององค์กร

ศึกษาปัญหาและรวบรวมข้อมูลของการพัฒนา IT Infrastructure โดยใช้แพลต์ฟอร์มระบบประมวลผลกลุ่มเมฆแบบส่วนตัวและศึกษาเครื่องมือต่างๆที่ใช้ในการพัฒนารวมถึงศึกษาปัญหาขององค์กรเพื่อนำมาวิเคราะห์และออกแบบระบบ สถาปัตยกรรมในขั้นตอนต่อไป

2) ศึกษาขั้นตอนการพัฒนาระบบ

ศึกษาขั้นตอนการพัฒนา IT Infrastructure เริ่มจากการศึกษาและรวบรวมข้อมูลสถาปัตยกรรมต่างๆ ที่ใช้ในการพัฒนา IT Infrastructure และทำการทดสอบ

3) ศึกษาข้อมูลและเครื่องมือที่ใช้ในการพัฒนา

ศึกษาการพัฒนา IT Infrastructure โดยใช้ openstack สร้างระบบประมวลผลกลุ่มเมฆแบบส่วนตัว และจัดสรรทรัพยากรต่างๆ ส่วนซอฟต์แวร์การจัดเก็บข้อมูลใช้ ceph ซึ่งเป็นซอฟต์แวร์ที่มีประสิทธิภาพในการจัดเก็บข้อมูล มีความยืดหยุ่น ความน่าเชื่อถือสูงและง่ายต่อการจัดการ

3.2 วิเคราะห์ปัญหาขององค์กร

IT Infrastructure แบบเดิมขององค์กรใช้ซอฟต์แวร์เพื่อสร้างเครื่องเสมือน โดยมีระบบควบคุมการใช้งานและดูแลเครื่องเสมือน แต่ละตัวเรียกว่า Hypervisor ซึ่งปัจจุบันทางองค์กรประสบปัญหาการใช้งานระบบดังกล่าวดังนี้

1) ซอฟต์แวร์ที่ทำหน้าที่เป็น Hypervisor มีค่าลิขสิทธิ์ค่อนข้างสูง และจำเป็นต้องจ่ายค่าลิขสิทธิ์เพิ่มเติมทุกครั้งที่มีการขยายทรัพยากร เช่น ในการซื้อเครื่องใหม่มาเพิ่มเติม3.2.2 ในการพัฒนาซอฟต์แวร์ของทางองค์กร มีการทดสอบซอฟต์แวร์ในหลายๆระบบ เช่น Dev, SIT, UAT, Pre-Production etc. ดังนั้นการจัดการระบบต่างๆเหล่านี้ ส่งผลให้ IT Infrastructure มีความซับซ้อนสูง รวมถึงมีความยากในการจัดการอย่างมาก

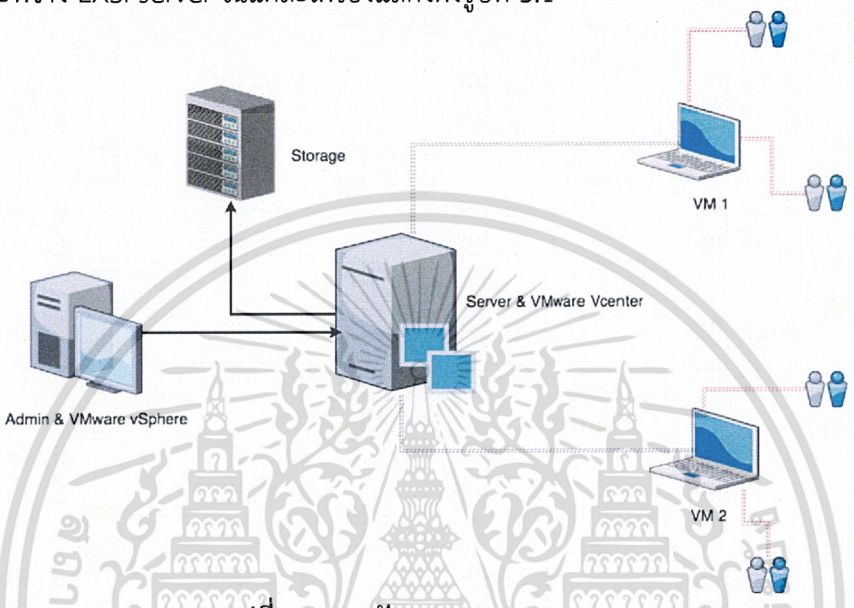
2) เนื่องจาก IT Infrastructure มีความซับซ้อนมาก จึงมีค่าใช้จ่ายในการดูแลรักษาสูง และใช้เวลามากในการแก้ไขเปลี่ยนแปลงแต่ละครั้ง ส่งผลให้การพัฒนาซอฟต์แวร์ใช้เวลามากกว่าที่ควรจะเป็น

จากการศึกษากระบวนการทำงาน IT Infrastructure แบบเดิมและวิเคราะห์ความเป็นไปได้ในการพัฒนาระบบ IT Infrastructure เพื่อให้กระบวนการทำงานดำเนินไปอย่างมีประสิทธิภาพมากขึ้น และสามารถตอบสนองความต้องการขององค์กรในการบริหารทรัพยากร รวมถึงแก้ปัญหาขององค์กร และยังสามารถลดต้นทุนในการดำเนินการ เช่น ค่าลิขสิทธิ์ซอฟต์แวร์ จึงได้พัฒนา IT Infrastructure โดยใช้ openstack ในการดูแลและจัดสรรทรัพยากร นอกจากนี้ได้นำ ceph เข้ามาทำงานร่วมกับ openstack เพื่อทำให้การจัดเก็บข้อมูลมีประสิทธิภาพเพิ่มมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

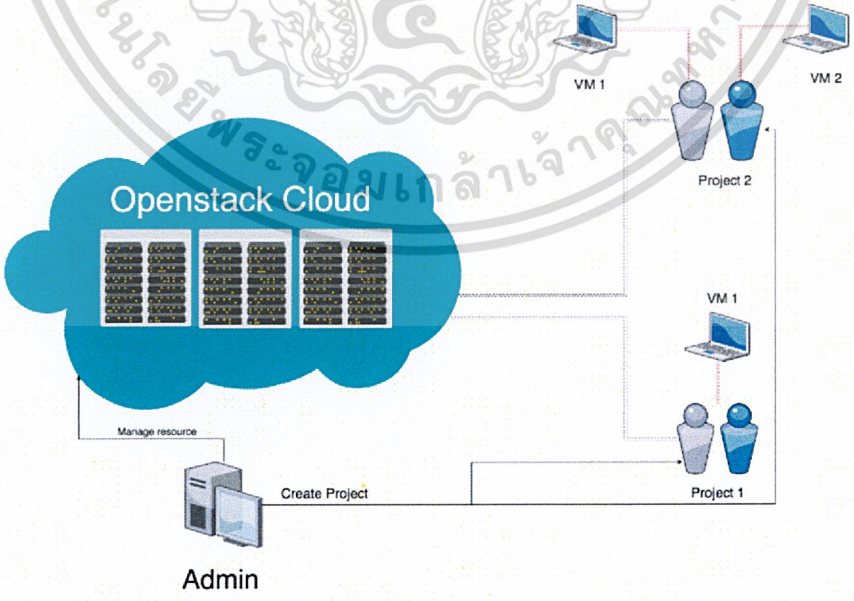
3.3 สถาปัตยกรรมของ VMware

สถาปัตยกรรมระบบขององค์กรใช้ VMware ผู้ดูแลระบบ (Admin) จะใช้ VMware vSphere รีโมทเข้าไปในเครื่องเซิร์ฟเวอร์ เพื่อทำการสร้างเครื่องเสมือน และบริหารทรัพยากรของเซิร์ฟเวอร์ สร้างเครือข่ายให้กับผู้ใช้งานในองค์กรผ่าน VMware Vcenter มีหน้าที่ควบคุมการเชื่อมต่อระหว่าง EXSi server ในแต่ละเครื่องแสดงดังรูปที่ 3.1



รูปที่ 3.1 สถาปัตยกรรมของ VMware

3.4 สถาปัตยกรรมของ openstack บน ระบบประมวลผลกลุ่มเมฆแบบส่วนตัว

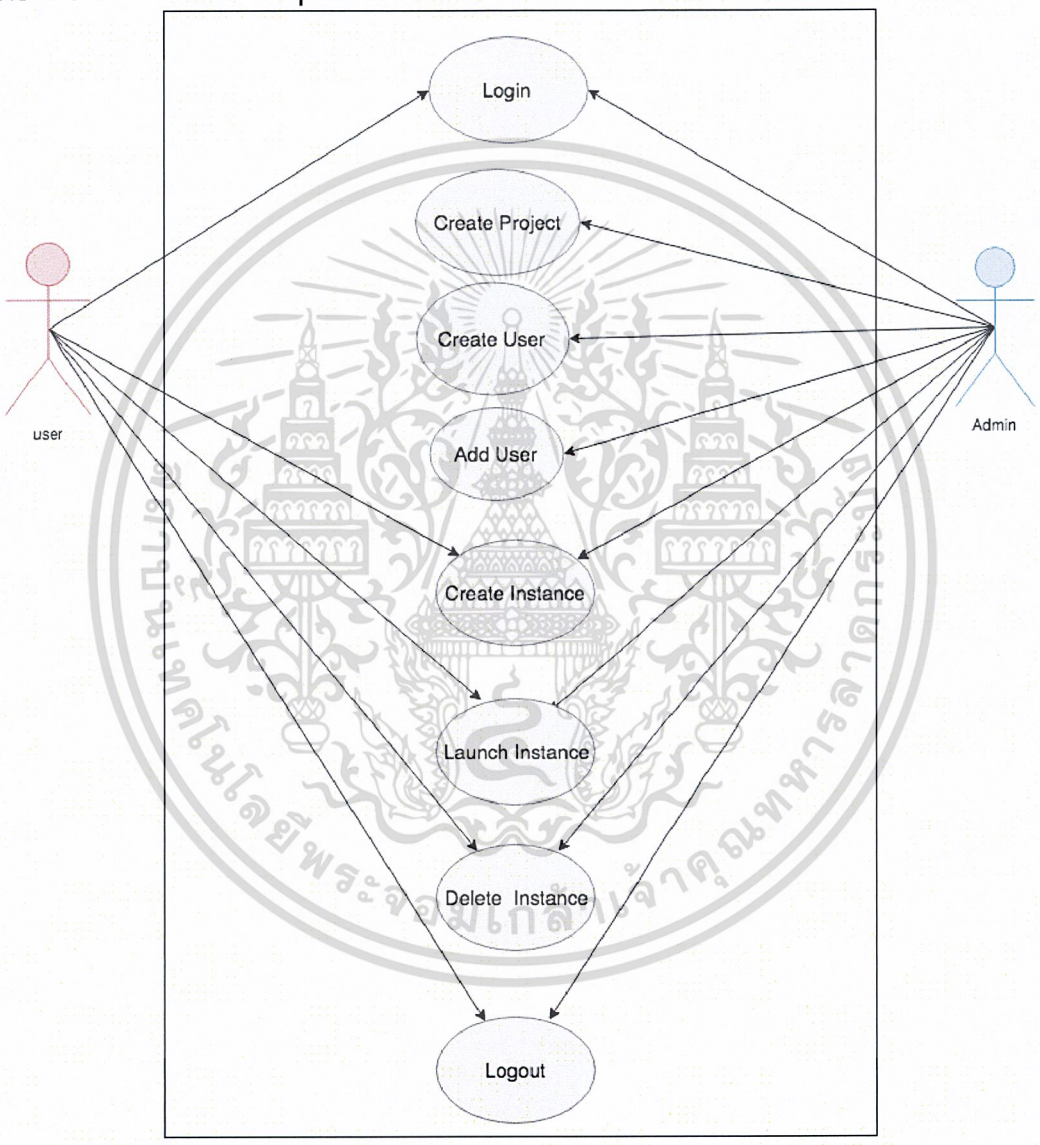


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 โครงสร้างการทำงานของ openstack ในองค์กร

การนำ openstack มาใช้ในองค์เป็นการเปลี่ยนการจัดการทรัพยากรให้อยู่ในรูปแบบบริการของระบบประมวลผลกลุ่มเมฆ ทำให้สะดวกและประหยัดเวลาในการจัดการและใช้งาน แสดงดังรูปที่ 3.2 คือโครงสร้างของการทำงานของ openstack โดยผู้ดูแลระบบ (Admin) จะเป็นออกโควต้าทรัพยากรแต่ละโปรเจคเพื่อให้ผู้ใช้เข้ามาใช้งานระบบประมวลผลกลุ่มเมฆ ได้อย่างอิสระ

3.5 ความสามารถของ opensatck



รูปที่ 3.3 Use case diagram

3.5.1) Admin

มีหน้าที่การทำงานรวมถึงสิทธิ์ในการเข้าถึงดังต่อไปนี้

1) Login การเข้าสู่ระบบเพื่อเข้าใช้และจัดการทรัพยากรในระบบ openstack โดยข้อมูลที่จำเป็นสำหรับ การทำงานนี้คือ User/Password ใช้ยืนยันตัวตนการเข้าใช้งาน

เอกสารนี้มีไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Create Project การสร้างโปรเจคใน openstack คือการสร้างกลุ่มงานขึ้นมาสำหรับงานหนึ่งชิ้น โดยในหนึ่งโปรเจคสามารถกำหนดสิทธิ์ User ที่จะเข้าถึงโปรเจคได้

3) Create User การสร้าง User คือการสร้างผู้เข้าใช้ขึ้นมาในระบบ

4) Add User ผู้ใช้จะถูกจัดเข้าอยู่ตามโปรเจคที่ผู้ให้ร้องขอหรือ Admin จัดการเข้าตามกลุ่มโปรเจค

5) Create Instance การสร้างเครื่องเสมือนสำหรับโปรเจคนั้น สามารถจัดการเลือกอิมเมจสร้างเน็ตเวิร์กภายใน

6) Launch Instance เมื่อสร้างเครื่องเสมือนขึ้นมา สามารถเปิดใช้งานได้ผ่านการ Launch Instance

7) Delete Instance การลบเครื่องเสมือนออกจากโปรเจค

8) Logout การออกจากระบบ

3.5.2) User

1) Login การเข้าสู่ระบบเพื่อเข้าใช้และจัดการทรัพยากรในระบบ openstack โดยข้อมูลที่จำเป็นสำหรับ การทำงานนี้คือ User/Password ใช้ยืนยันตัวตนการเข้าใช้งาน

2) Create Instance การสร้างเครื่องเสมือนสำหรับโปรเจคนั้น สามารถจัดการเลือกอิมเมจสร้างเน็ตเวิร์กภายใน

3) Launch Instance เมื่อสร้างเครื่องเสมือนขึ้นมา สามารถเปิดใช้งานได้ผ่านการ Launch Instance

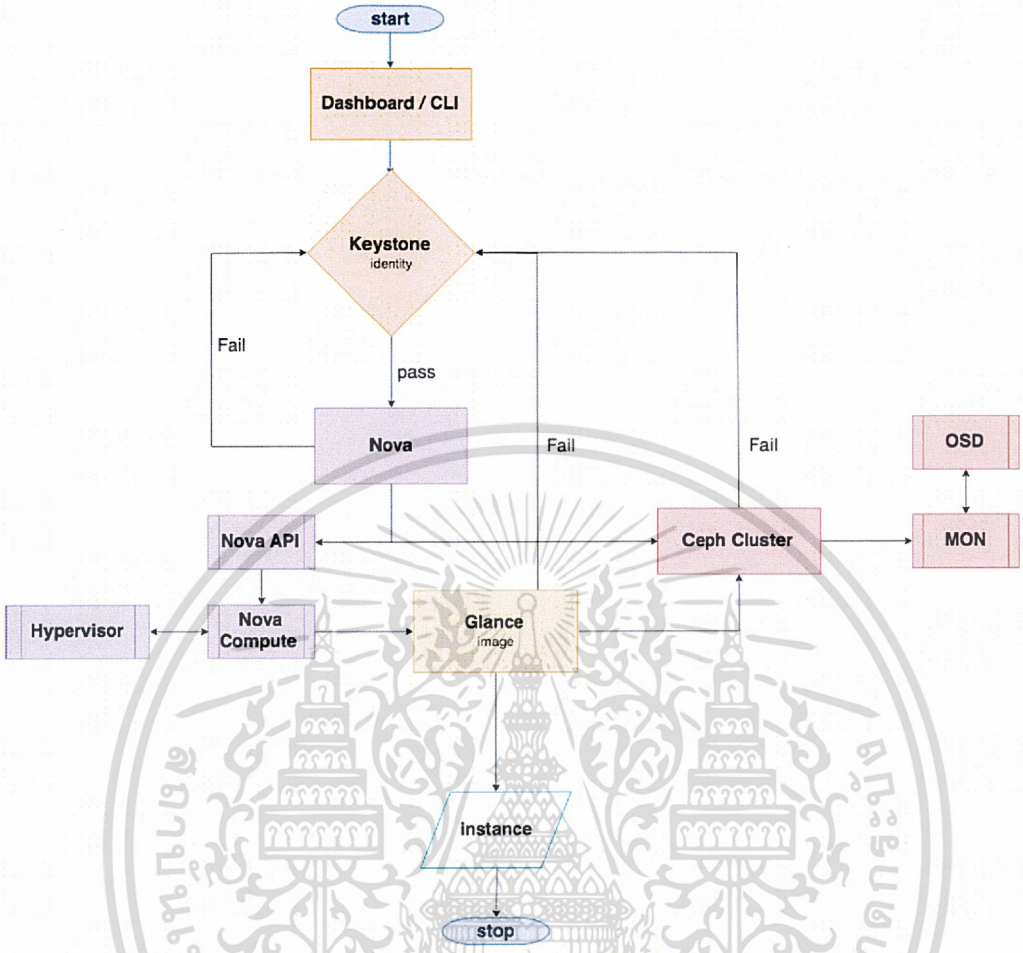
4) Delete Instance การสร้างและลบเครื่องเสมือนสำหรับโปรเจคนั้น

5) Logout การออกจากระบบ

3.6 ขั้นตอนการทำงานของ openstack

ขั้นตอนระหว่างการสร้าง Instance ของระบบ openstack ที่ทำงานร่วมกับ Ceph แสดงดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

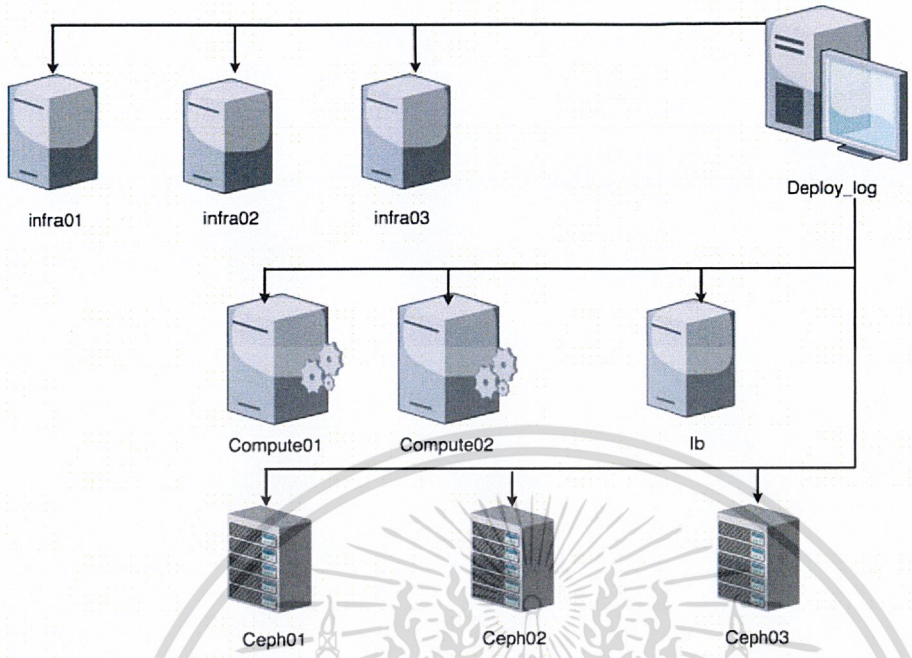


รูปที่ 3.4 Flowchart การทำงานของ opensatck และ ceph

3.7 การทดสอบประสิทธิภาพของ openstack

เป็นการทดสอบเพื่อแสดงให้เห็นว่าองค์สามารถนำ openstack มาพัฒนา IT Infrastructure ขององค์กรได้ โดยไม่ทำให้ประสิทธิภาพของงานลดลง สถาปัตยกรรมระบบเพื่อทดสอบการใช้งาน(POC) ทุกเครื่องจะลงระบบปฏิบัติการ Ubuntu และมีหน้าที่การทำงานแสดงดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 โครงสร้างระบบเพื่อทดสอบการใช้งาน(POC)

1) infra01-infra03 เป็นเครื่องที่ใช้จัดการระบบของระบบประมวลผลกลุ่มเมฆแบบส่วนตัวทั้งหมด ในเครื่องนี้จะมีทุกโมดูลของ openstack ที่ใช้ในระบบ เพื่อควบคุมการทำงานของเครื่องอื่นๆ ที่เหลือ ในระบบนี้จะมี infra จำนวน 3 เครื่องสำหรับการทดสอบการใช้งาน (POC) โดยเครื่อง infra เป็นการลงโมดูล Nova

2) Deploy_log เป็นเครื่องที่ประกอบด้วย เครื่อง deploy และ log ไว้เครื่องเดียวกัน deploy เครื่องที่จะเข้าไปติดตั้งและตั้งค่าแต่ละเครื่อง ส่วน log เป็นการเก็บ log การทำงานของแต่ละเครื่องไว้

3) lb เครื่องนี้จะเป็นเครื่อง loadbalancer ที่จะลง ha proxy เอาไว้

4) Compute เป็นเครื่องที่มีการทำงานร่วมกันของ openstack โดยในเครื่องนี้มีโมดูล Nova และจำนวนเครื่องของ Compute มี 1 เครื่อง และสามารถเพิ่มขึ้นและลดลงได้ตาม

5) ceph 01-03 เป็นเครื่องที่ใช้จัดการงานเกี่ยวกับการเก็บข้อมูลหรืออุปกรณ์จัดเก็บข้อมูล (storage) ซึ่งจะเป็นการนำ ceph มาใช้งานในระบบ เพื่อให้การจัดเก็บข้อมูลมี ประสิทธิภาพเพิ่มขึ้น โดยในเครื่อง cephประกอบไปด้วย node 2 ประเภท คือ osd และ monitor โดยรายละเอียดการติดตั้งทั้งหมดจะแสดงในภาคผนวก

3.7.1) สภาพแวดล้อมโดยรวมของระบบในการทดสอบ

มีการกำหนดคุณสมบัติและการเชื่อมต่อกันของแต่ละเครื่องผ่านเครือข่ายอินเตอร์เน็ต (Network) ของแต่ละเครื่องดังตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 คุณสมบัติของแต่ละเครื่องเสมือน

No.	Name	CPU (vCore)	RAM (GB)	Disk (GB)
1	Deploy	4	4	120
2	Lb	4	2	120
3	Infra01	4	10	120
4	Infra02	4	10	120
5	Infra03	4	10	120
6	Compute01	16	4	120
7	Compute02	16	4	120
8	Ceph01	4	2	300
9	Ceph02	4	2	300
10	Ceph03	4	2	300

3.8 ขั้นตอนการทดสอบระบบ

จากขั้นตอนศึกษาและรวบรวมข้อมูล วิเคราะห์ความต้องการขององค์กร ออกแบบโครงสร้าง ทำให้เราทราบถึงปัญหาในการทำงานและขั้นตอนการทำงานต่างๆของกระบวนการทำงานแบบเดิม ซึ่งการนำ openstack มาใช้งานแทน VMware นั้น เราต้องทดสอบการทำงานของ openstack ว่ามีประสิทธิภาพที่จะจัดการและควบคุมการทำงาน IT Infrastructure ได้ ในขั้นตอนการทดสอบจะทดสอบในส่วนของ High Availability Infrastructure และ High Availability Application เพื่อดูประสิทธิภาพด้าน High Availability ของ openstack รวมถึงทดสอบในส่วนของการจัดการการเก็บข้อมูลที่ ceph เป็นตัวจัดการข้อมูลทั้งหมดขององค์กร

3.9 ส่วนติดต่อกับผู้ใช้งานของ openstack

3.9.1) หน้าจอแรกของ openstack

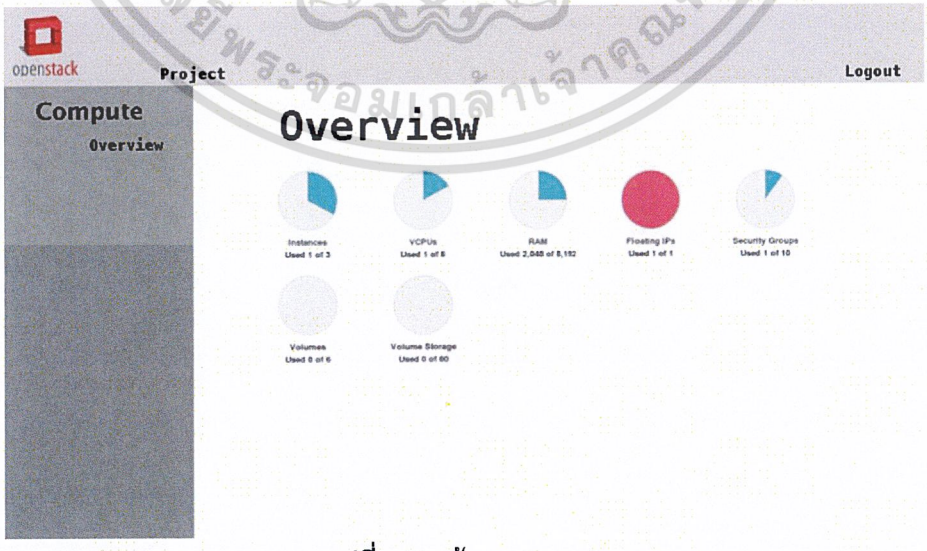
หน้าจอแรกให้สำหรับ login เข้าสู่ระบบเพื่อใช้จัดการทรัพยากรของระบบคลาวด์ เข้าผ่านบราวเซอร์โดยใช้ IP เป็น URL การจะเข้าระบบได้จำเป็นต้องมี User และ Password แสดงดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.6 หน้าจอแรกของเว็บไซต์ openstack

3.9.2) **หน้าจอแสดงภาพรวมของระบบ**

Overview แสดงการรวบรวมข้อมูลทรัพยากรของระบบนำเสนอเป็นแผนภูมิวงกลม เช่นข้อมูล Instance, RAM, VCPUs, Volumes แสดงดังรูปที่ 3.7

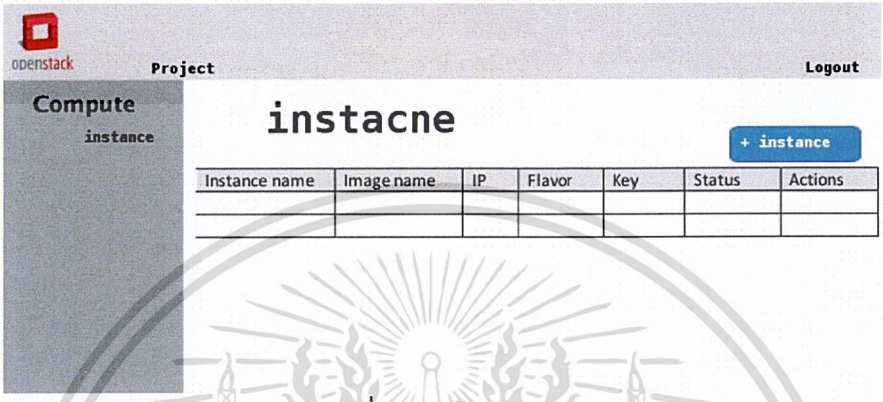


รูปที่ 3.7 หน้าจอ Overview ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9.3) หน้าจอแสดงรายละเอียดของเครื่องเสมือน (Instances)

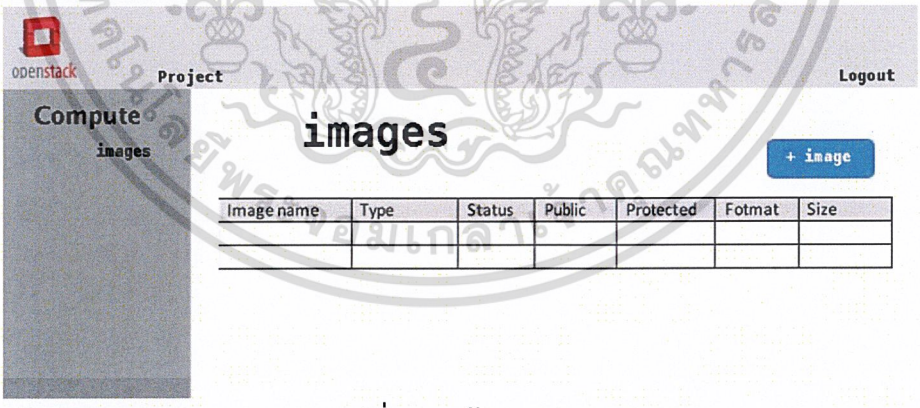
หน้าจอนี้จะมีรายชื่อของ Instance ที่มีทั้งหมดภายในระบบ โดยจะแสดงข้อมูลที่เกี่ยวกับ Instance เช่น Name, Image name, IP, Flavor ใช้จัดการ การทำงานของ Instance สามารถทำการเพิ่มเครื่องได้ตามความต้องการ แสดงดังรูปที่ 3.8



รูปที่ 3.8 หน้าจอ Instance ของระบบ

3.9.4) หน้าจอแสดงรายละเอียดของอิมเมจไฟล์ (Image)

หน้าจอที่ใช้จัดการไฟล์ images สำหรับใช้สร้าง Instance โดยจะแสดง image name, Type, Format ของ Images สามารถเพิ่ม images ได้ตามความต้องการ แสดงดังรูปที่ 3.9

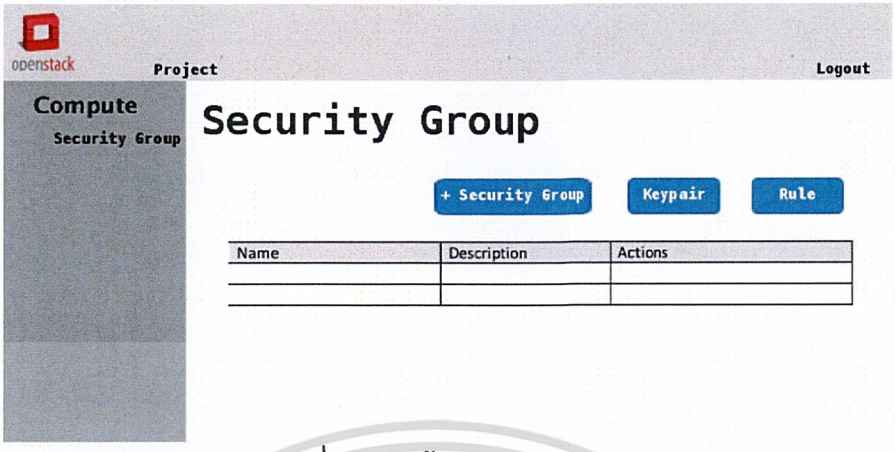


รูปที่ 3.9 หน้าจอ Images ของระบบ

3.9.5) หน้าจอแสดงการเข้าถึงและความปลอดภัยของเครื่องเสมือน (Security Group)

หน้าจอที่แสดงการตั้งค่า Security ที่ใช้สำหรับ Instance แต่ละเครื่อง จะสามารถกำหนดการใช้งานที่แตกต่างกันได้ โดยใช้เมนูนี้เป็นตัวจัดการยกตัวอย่างเช่น การกำหนดให้ใช้โปรโตคอลใดในการเข้าและออกของข้อมูล แสดงดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 หน้าจอ Security Group ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัยและการอภิปรายผล

4.1 ผลการทดสอบระบบ

ในการทดสอบจะวัดเรื่อง High Availability Infrastructure และ Application แสดงดังตาราง 4.1 และ 4.2

ตารางที่ 4.1 ขั้นตอนการทดสอบ High Availability Infrastructure

ขั้นตอนที่	การทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ (ผ่าน/ไม่ผ่าน)	หมายเหตุ
1	ทำการปิดเครื่อง Instances ที่เป็น controller server จำนวน 1 เครื่อง	ระบบสามารถทำงานได้ปกติ	ผ่าน	
2	ทำการปิดเครื่อง Instances ที่เป็น compute server จำนวน 1 เครื่อง	ระบบสามารถทำงานได้ปกติ	ไม่ผ่าน	ต้องทำการสร้าง เครื่อง load balanc เฉพาะ compute
3	ทำการปิดเครื่อง Instances ที่เป็น ceph จำนวน 1 เครื่อง	ระบบสามารถทำงานได้ปกติ	ผ่าน	

ตารางที่ 4.2 ขั้นตอนการทดสอบ High Availability Application

ขั้นตอนที่	การทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ (ผ่าน/ไม่ผ่าน)	หมายเหตุ
1	ทำการปิดเครื่อง Instances ที่เป็น controller server และติดตั้ง Application จำนวน 1 เครื่อง	ระบบสามารถทำงานได้ปกติ	ผ่าน	
2	ทำการปิดเครื่อง	ระบบสามารถ	ไม่ผ่าน	ต้องทำการ

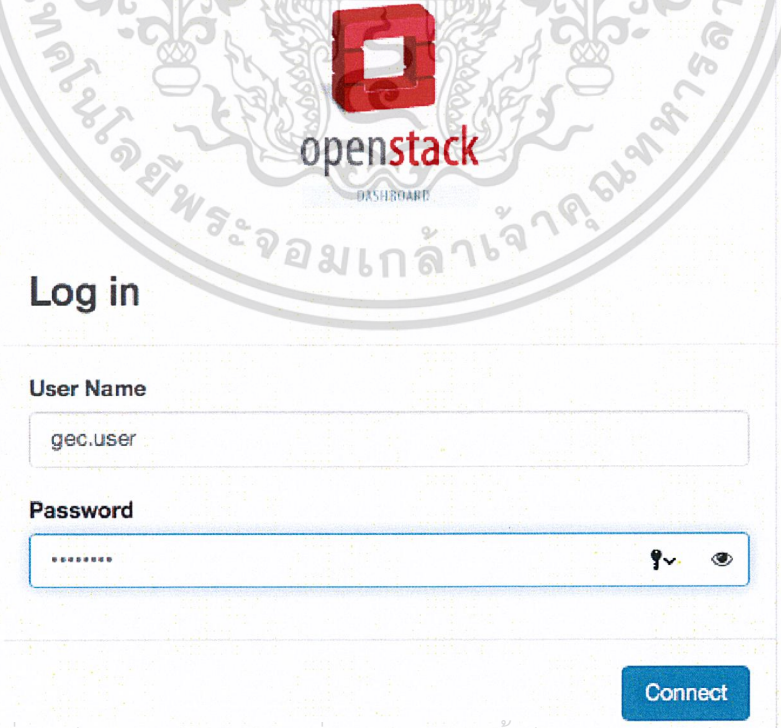
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Instances ที่เป็น compute server และติดตั้ง Application จำนวน 1 เครื่อง	ทำงานได้ปกติ		สร้าง เครื่องที่ทำ load balanc เฉพาะ compute server
3	ทำการปิดเครื่อง Instances ที่เป็น ceph และติดตั้ง Application จำนวน 1 เครื่อง	ระบบสามารถทำงานได้ปกติ	ผ่าน	

4.2 ภาพรวมส่วนติดต่อผู้ใช้ผ่านหน้าแดชบอร์ด (dashboard)

ในส่วนของหน้าแดชบอร์ด (dashboard) การเข้าใช้งาน openstack ต้องทำการเข้าสู่ระบบด้วยบัญชีผู้ใช้ และรหัสผ่านก่อนเข้าใช้งาน แสดงดังรูปที่ 4.1

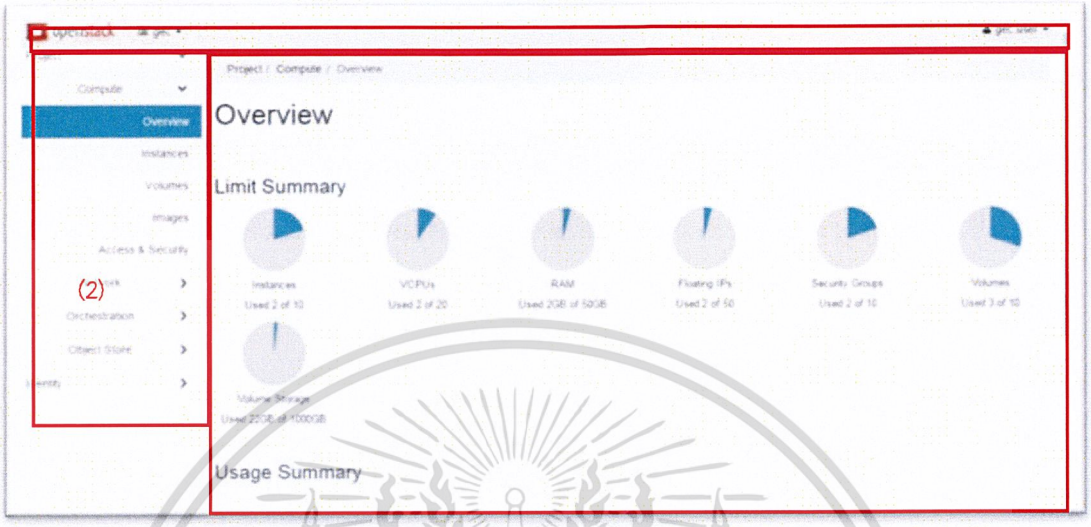
บัญชีผู้ใช้งานระบบจะแบ่งออกเป็นสองกลุ่มคือ กลุ่มของผู้ดูแลระบบและกลุ่มผู้ใช้งานทั่วไปภายในองค์กร โดยบัญชีผู้ใช้ที่ดูและระบบจะสามารถตรวจสอบและแก้ไขภายในระบบโดยรวมได้ รวมถึงจัดการบัญชีผู้ใช้อื่นๆได้ ส่วนบัญชีของผู้ใช้ทั่วไปจะไม่สามารถตรวจ และแก้ไขข้อมูลต่างๆ บัญชีผู้ใช้คนอื่นของระบบได้ จะสามารถใช้งานเฉพาะเครื่องเสมือนของผู้ใช้เองเท่านั้น



รูปที่ 4.1 หน้าจอเข้าสู่ openstack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

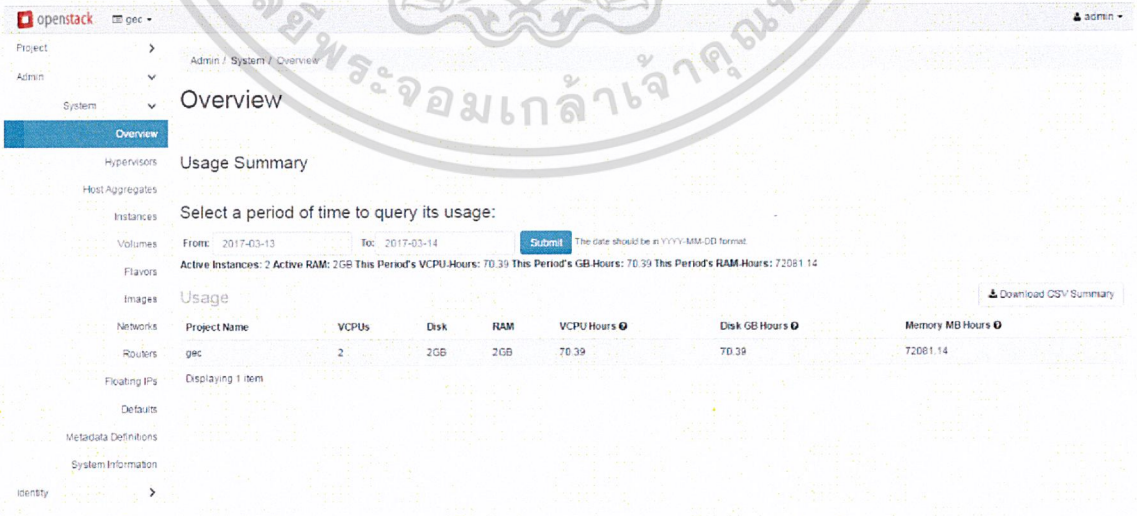
หน้าจอแสดงของระบบ openstack แบ่งออกเป็น 3 ส่วนหลัก โดยมีส่วนของเมนู ส่วนของการทำงานหรือฟังก์ชันของระบบ และส่วนการแสดงข้อมูลของระบบ



รูปที่ 4.2 หน้าจอระบบ openstack

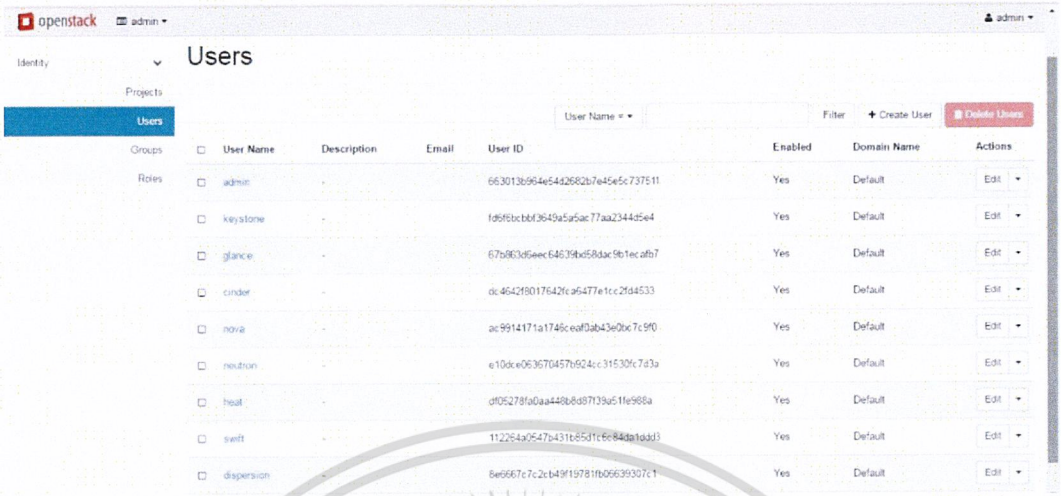
จากรูปที่ 4.2 คือหน้าจอแดชบอร์ด (dashboard) ที่ใช้ติดต่อของ openstack โดยมีส่วน (1) เมนูที่ใช้ในการเลือกตั้งค่าบัญชีผู้ใช้ และการลงชื่อออก ส่วน (2) ฟังก์ชันของระบบ (3) ส่วนแสดงข้อมูลของระบบ

เมื่อเข้าสู่ระบบ openstack แล้วหน้า แดชบอร์ด (dashboard) จะแสดงการใช้งานโดยรวมของระบบ ได้แก่ จำนวนเครื่องเสมือนที่กำลังทำงานอยู่ เวลาที่เครื่องเสมือนทำงาน และสามารถดาวน์โหลดไฟล์สถิตินี้ได้ โดยกดปุ่ม Download CSV Summary แสดงดังรูปที่ 4.3



รูปที่ 4.3 หน้าจอแสดงภาพรวมของระบบ openstack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

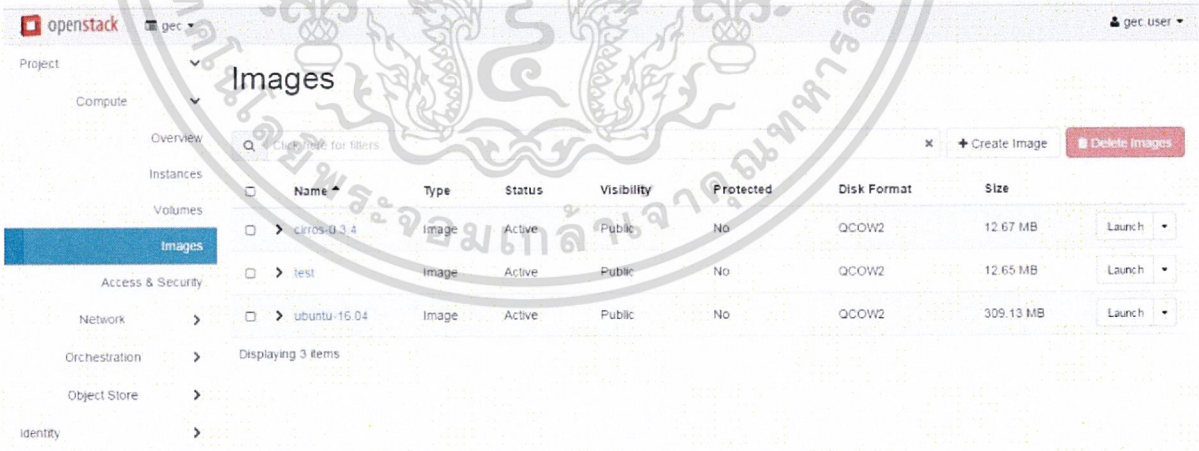


รูปที่ 4.4 หน้าจอแสดงบัญชีผู้ใช้งาน

ในบัญชีผู้ใช้ทั่วไปจะไม่สามารถจัดการกับระบบโดยรวมได้นอกจัดการกับบัญชี และเครื่องเสมือนของตัวเองเท่านั้น

4.3 การจัดการไฟล์อิมเมจ (Image)

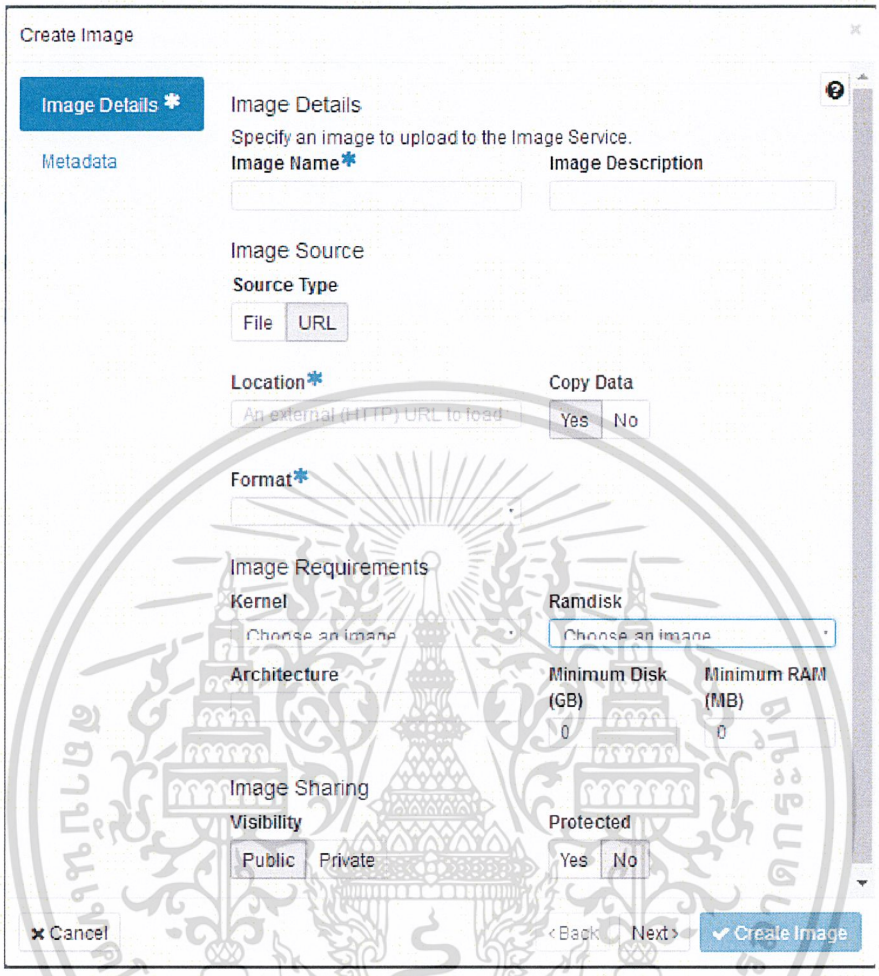
รายการไฟล์อิมเมจทั้งหมด จะใช้ในการบูตระบบปฏิบัติการ สำหรับเครื่องเสมือนเพื่อใช้งาน แสดงดังรูปที่ 4.5



รูปที่ 4.5 หน้าจอแสดงรายการไฟล์อิมเมจ

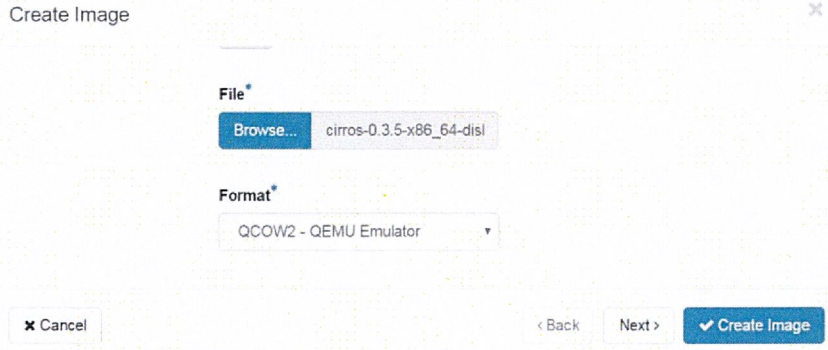
เมื่อกดปุ่ม Create Image จะแสดงหน้าจอดังรูปที่ 4.6 ให้กรอกรายละเอียดต่างๆในการสร้างไฟล์อิมเมจได้แก่ ชื่อไฟล์อิมเมจ ที่อยู่ไฟลอิมเมจ ชนิดของไฟล์ จำนวนทรัพยากรขั้นต่ำที่ระบบต้องการ จากนั้นกดปุ่ม Create Image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 หน้าจอการสร้างไฟล์อิมเมจ

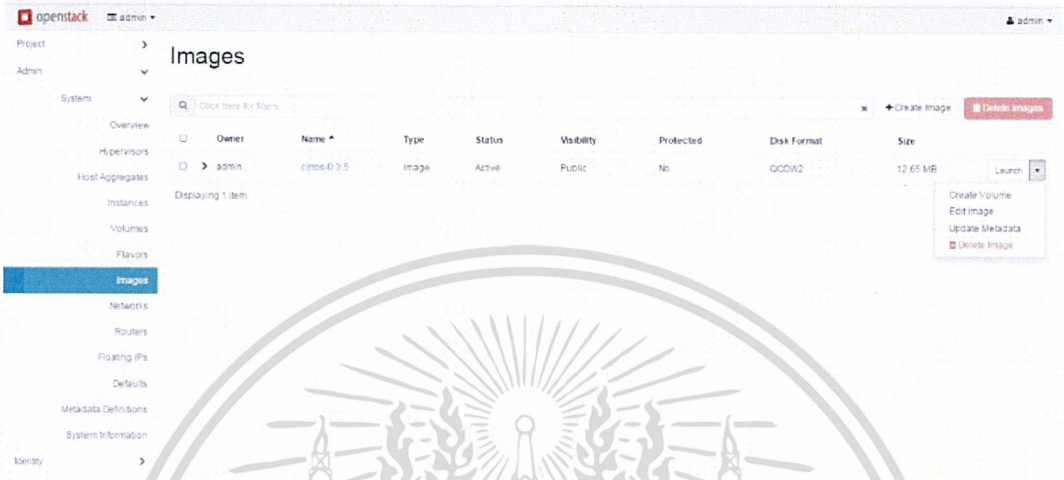
การโหลดไฟล์อิมเมจเข้าสู่ระบบสามารถนำไฟล์จากเครื่องคอมพิวเตอร์ โดยการ Browse และเลือกไฟล์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 การนำไฟล์อิมเมจเข้าสู่ระบบ

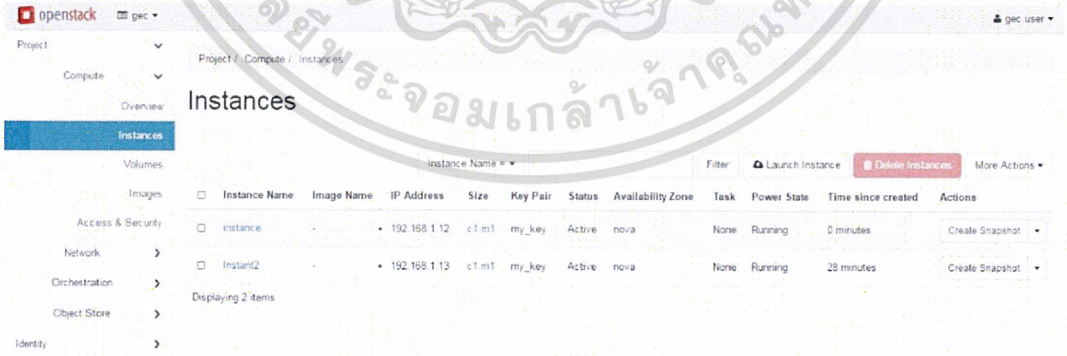
เมื่อต้องการลบไฟล์อิมเมจให้กดปุ่ม Launch ตรงลูกศรเพื่อ edit ไฟล์อิมเมจ Deleteไฟล์อิมเมจแสดงดังรูป 4.8



รูปที่ 4.8 หน้าจอการลบไฟล์อิมเมจ

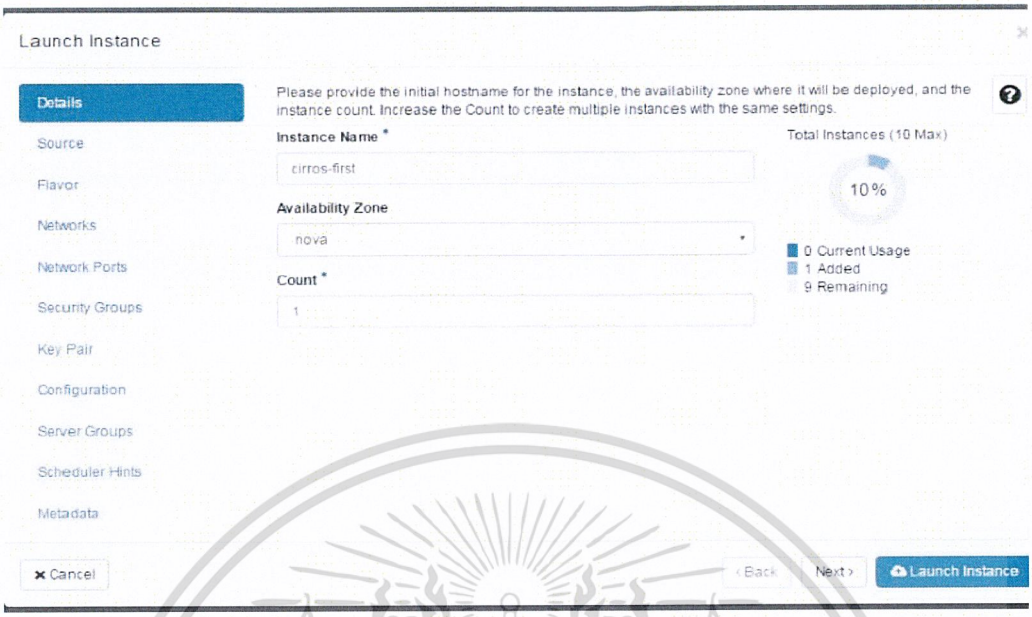
4.4 การสร้างเครื่องเสมือน (Instances)

การสร้างเครื่องเสมือนให้กดปุ่ม Launch Instance จะแสดงหน้าจอดังรูปที่ 4.9 จากนั้นกรอกรายละเอียดต่างๆ ได้แก่ ชื่อของเครื่องเสมือน ขนาด หน่วยความจำของเครื่องเสมือน จำนวนที่ต้องสร้างและระบบปฏิบัติการที่ต้องการบูตระบบ รวมถึงระบบเครือข่ายเน็ตเวิร์คของเครื่องเสมือนแสดงดังรูป 4.10 – 4.12

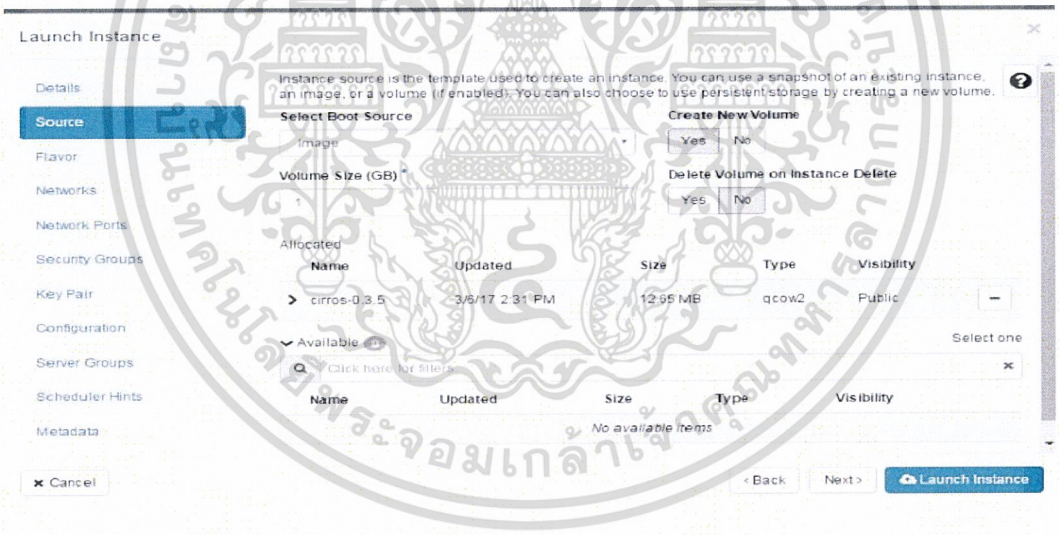


รูปที่ 4.9 หน้าจอแสดงเครื่องเสมือนในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

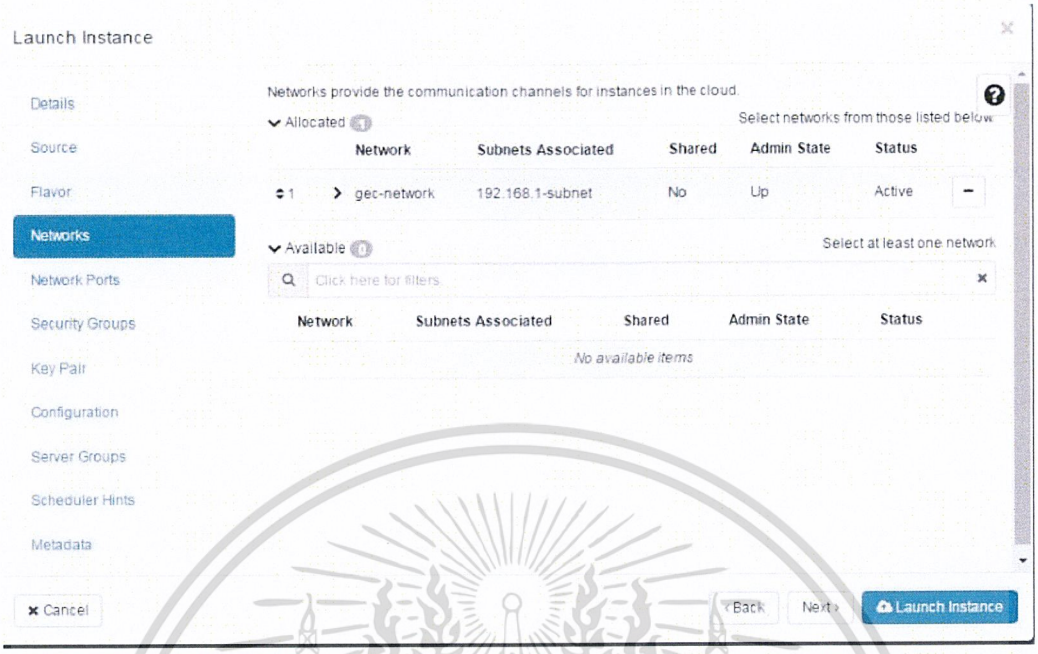


รูปที่ 4.10 หน้าจอแสดงรายละเอียดของเครื่องเสมือน (1)



รูปที่ 4.11 หน้าจอแสดงรายละเอียดของเครื่องเสมือน (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



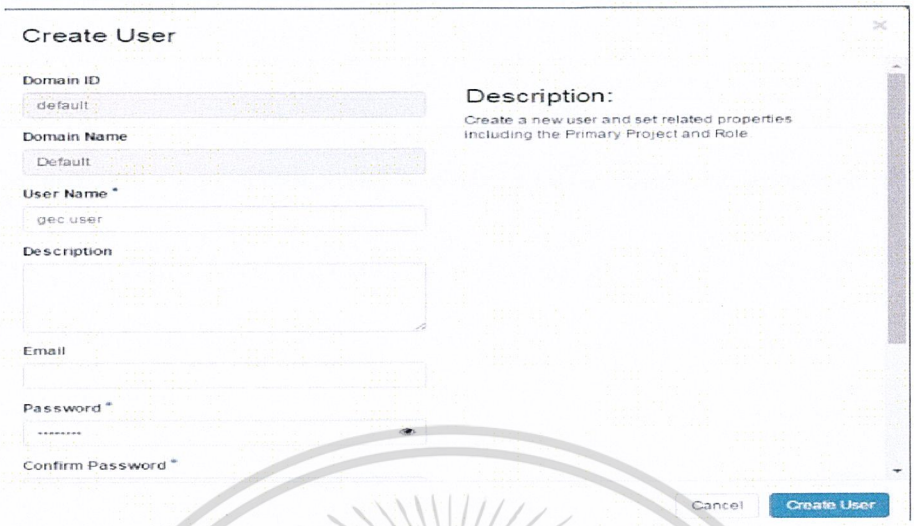
รูปที่ 4.12 หน้าจอแสดงรายละเอียดของเครื่องเสมือน (3)

4.5 การจัดการบัญชีผู้ใช้งาน

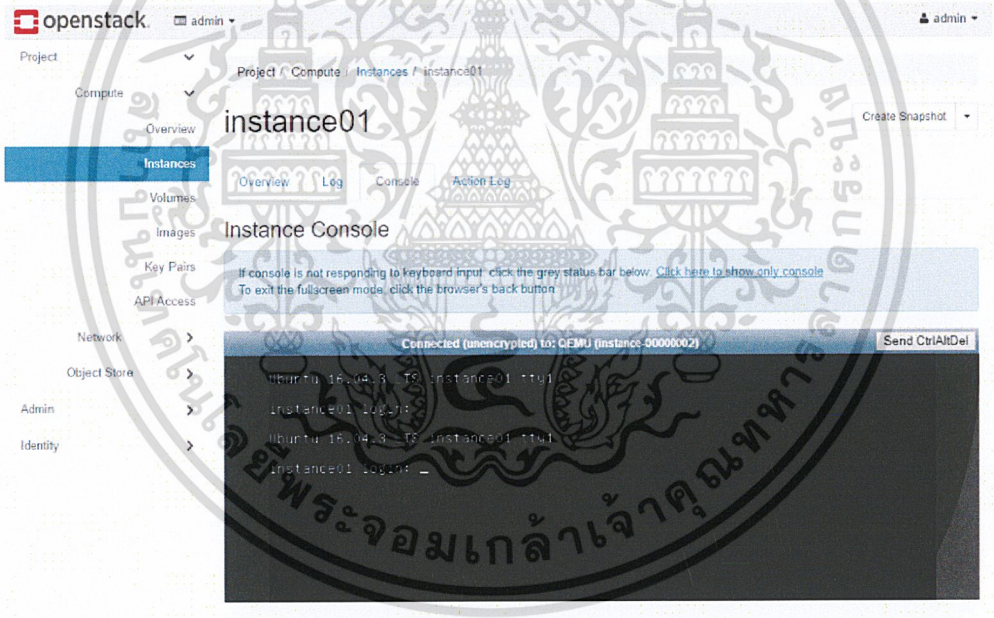
การสร้างบัญชีผู้ใช้งานให้กดปุ่ม Create User แสดงดังรูปที่ 4.14 จากนั้นจะปรากฏหน้าจอดังรูปที่ 4.15 ให้กรอกรายละเอียดของผู้ใช้ ได้แก่ ชื่อบัญชีผู้ใช้ อีเมล รหัสผ่าน จากนั้นเลือกงานหรือ project ที่ผู้ใช้มีสิทธ์

รูปที่ 4.13 หน้าจอแสดงบัญชีผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 หน้าจอแสดงรายละเอียดของการสร้างบัญชีผู้ใช้

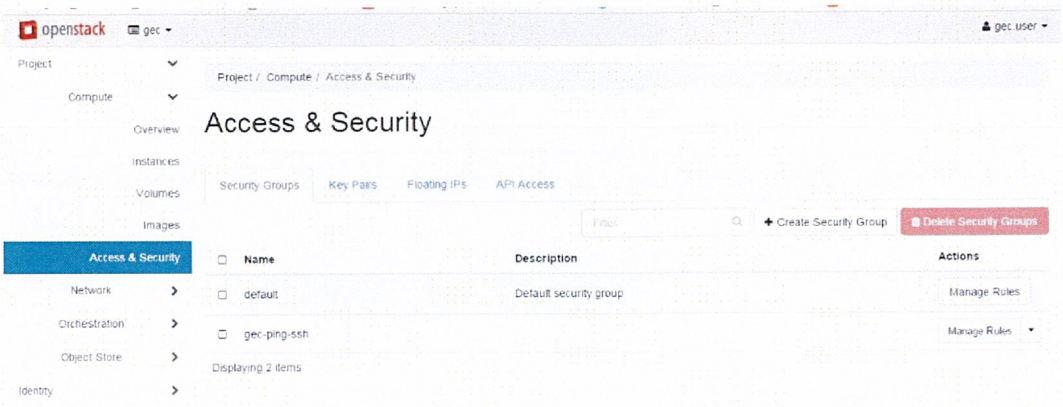


รูปที่ 4.15 หน้าจอควบคุมเครื่องเสมือน

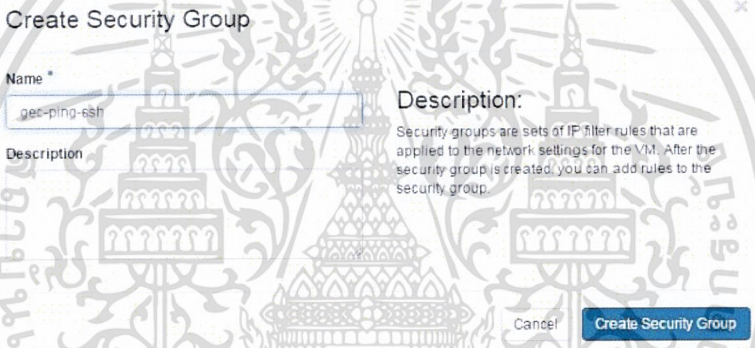
4.6 การทำงานของ Access & Security

ก่อนที่จะสร้าง เครื่องเสมือน หรือ Instance ควรเพิ่มกฎของ Security groups แสดงดังรูปที่ 4.17 เพื่อให้ผู้ใช้สามารถ ping และใช้ SSH เพื่อเชื่อมต่อกับ Instance ได้ Security groups คือ ชุดของการตรวจสอบ IP ที่กำหนดการเข้าถึงเครือข่ายและใช้กับ Instance ทั้งหมดภายใน Project ซึ่งในการทำคุณสามารถเพิ่มกฎ (Add Rule) แสดงดังรูปที่ 4.18 ลงใน Security groups

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 หน้าจอแสดง Access & Security



รูปที่ 4.17 หน้าจอ Create Security Group

รูปที่ 4.18 หน้าจอ Add Rule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

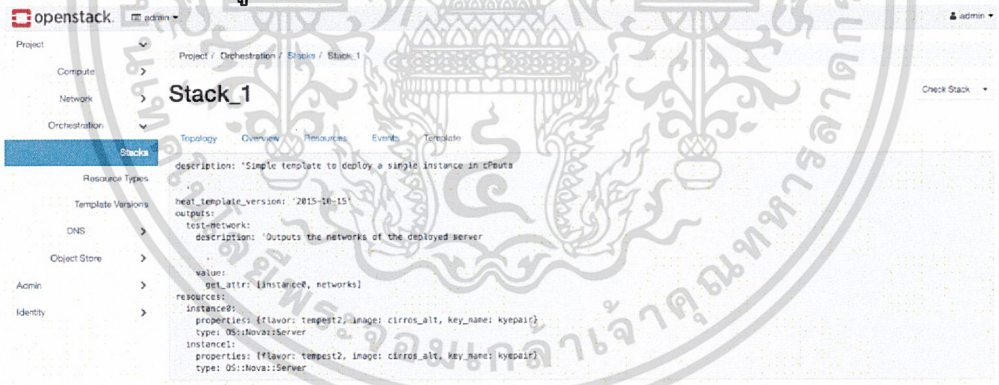
4.7 ความคุ้มค่าของการนำ openstack มาใช้ในองค์กร

เนื่องจากค่าใช้จ่ายในการลงทุนและประโยชน์จากการนำ openstack มาใช้งานในระดับองค์กร ตรงจุดนี้เป็นหนึ่งปัจจัยสำคัญที่ทำให้องค์กรเลือกใช้งาน openstack มาพัฒนา IT Infrastructure

4.7.1 ประโยชน์ที่ได้รับ

ประโยชน์ที่ได้รับนั้นมีผลมาจากโมดูล ที่เป็นโมดูลการทำงานใน openstack ชื่อว่า Heat มีคุณสมบัติ Auto scaling นั่นคือสามารถสร้างเครื่องเสมือน หรือ Instances ในระบบ พร้อมกันได้มากกว่า 1 เครื่อง

รูปที่ 4.19 การใช้ Heat สร้าง Instance ในระบบ



รูปที่ 4.20 Template การสร้าง Heat

4.7.2 ค่าใช้จ่ายการลงทุน

สำหรับค่าใช้จ่ายในการลงทุน เพื่อให้ง่ายต่อความเข้าใจจึงจำลองเหตุการณ์การลงทุนระหว่าง openstack กับตัว VMware โดยค่าใช้จ่ายในการลงทุนนั้นหาได้จาก ราคาของเครื่องเซิร์ฟเวอร์ ราคาของค่าลิขสิทธิ์ซอฟต์แวร์ โดยราคาเครื่องเซิร์ฟเวอร์ราคา 1,800 USD ถ้าใช้ openstack ที่เป็นซอฟต์แวร์โอเพ่นซอสจะไม่มีค่าลิขสิทธิ์ซอฟต์แวร์ จะเสียค่าใช้จ่ายในส่วนของเครื่องเซิร์ฟเวอร์อย่างเดียว แต่ถ้าใช้งาน VMware จะมีค่าใช้จ่ายในส่วนค่าลิขสิทธิ์ซอฟต์แวร์เพิ่มเข้ามาตามตารางค่าลิขสิทธิ์ซอฟต์แวร์ แสดงดังตารางที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงค่าใช้จ่ายเครื่องเสมือน

ลิขสิทธิ์ซอฟต์แวร์	ราคา (USD)
VMware vSphere Standard, 1 CPU	995
VMware vCenter Server Standard	4,995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการดำเนินการ

สหกิจศึกษานี้ มีจุดมุ่งหมายในการศึกษา และวิจัยความเป็นไปได้ในการใช้ซอฟต์แวร์โอเพ่นซอร์สมาพัฒนาระบบ IT Infrastructure บนระบบประมวลผลกลุ่มเมฆแบบส่วนตัว สำหรับในภายในองค์กร เพื่อเพิ่มความสามารถและประสิทธิภาพในการบริหารจัดการเครื่องเซิร์ฟเวอร์ให้มีประสิทธิภาพมากขึ้นทั้งในเรื่องของการแบ่งเนื้อที่หน่วยความจำ แบ่งเครือข่าย(Network) การแบ่งการประมวลผล การจัดเก็บข้อมูลที่ทำงานร่วมกับซอฟต์แวร์อื่น และการลดค่าใช้จ่ายทางด้านซอฟต์แวร์ของระบบ อีกทั้งผู้ใช้งานในองค์กรยังสามารถเข้าใช้งานในส่วนของตนเองในการสร้างเครื่องเสมือนด้วยตนเองตามความต้องการผ่านทาง แดชบอร์ด (dashboard) ของ openstack โดยการทำสหกิจศึกษาได้ใช้ซอฟต์แวร์โอเพ่นซอร์ส คือ Ubuntu เป็นระบบปฏิบัติการหลักของทุกเครื่องเสมือน และ openstack เป็นซอฟต์แวร์ที่จัดการระบบประมวลกลุ่มเมฆแบบส่วนตัว ในส่วนซอฟต์แวร์ที่ใช้ในการจัดเก็บข้อมูล คือ ceph ซึ่งคณะผู้จัดทำได้ศึกษาเกี่ยวกับโครงสร้างพื้นฐานส่วนประกอบต่างๆของซอฟต์แวร์และศึกษาความต้องการขององค์กร หลักการทำงานของซอฟต์แวร์ การเชื่อมโยงซอฟต์แวร์ทั้งสองให้ทำงานร่วมกัน หลักการเขียนสคริปเพื่อสร้างระบบประมวลผลแบบกลุ่มเมฆแบบส่วนตัว การเขียนสคริปเพื่อทำให้สามารถทำงานร่วมกับซอฟต์แวร์ที่ทำหน้าที่กับการจัดเก็บข้อมูลของระบบ การสร้างเครือข่าย (Network) เพื่อให้เครื่องเสมือนทำงานร่วมกัน และสามารถใช้งานผ่านหน้า แดชบอร์ด (dashboard) ของ openstack แทนการใช้งานผ่าน command line ที่ซับซ้อนและยุ่งยาก คณะผู้จัดทำจึงได้ศึกษาออกแบบ ทำการศึกษา และทดสอบกับการใช้งานในองค์กร ว่าสามารถนำ openstak ที่เป็นซอฟต์แวร์โอเพ่นซอร์สมาใช้ในการประมวลผล จัดการเครื่องเสมือนมาใช้ในองค์กรได้

5.2 สรุปผลการทดลอง

จากผลการทดสอบ High Availability Infrastructure และ Application โดยการปิดเครื่อง controller และ ceph ทีละเครื่อง ระบบสามารถทำงานได้ปกติ เมื่อปิดเครื่อง compute instance ที่ทำงานอยู่บน compute ตัวที่ปิด ไม่สามารถใช้งานได้ โดยแนวทางแก้ปัญหาคือต้องทำ Load balance เฉพาะ compute serverซึ่งจะทำให้ในการปิดเครื่อง compute server ระบบยังสามารถใช้งานได้ปกติ

5.3 แนวทางในการศึกษาต่อและข้อเสนอแนะ

1) เนื่องจาก openstack เป็นเทคโนโลยีที่ค่อนข้างใหม่สำหรับองค์กร ภายในองค์กรต้องหาบุคลากรที่ต้องดูแลและแก้ไข ซึ่งต้องมีความเชี่ยวชาญ openstack เพื่อรองรับการใช้งาน

2) นำระบบไปพัฒนาบนเครื่องเซิรฟเวอร์ที่มีทรัพยากรที่เพียงพอเพื่อการทำงานที่สมบูรณ์ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ปัญหาที่พบ

1) ใช้เวลาในการศึกษาข้อมูลเป็นเวลานาน เนื่องจากเป็นความรู้ใหม่ทั้งหมด

2) ในการติดตั้ง openstack ต้องดาวน์โหลด image ผ่านอินเทอร์เน็ตทุกครั้ง ซึ่ง bandwidth ในช่วงเวลาทำงานไม่พอที่จะสามารถติดตั้งได้ จะต้องใช้เวลาหลังเลิกงานในการติดตั้ง

3) คอมพิวเตอร์แม่ข่ายที่ใช้ในการพัฒนาระบบมีทรัพยากรฮาร์ดแวร์น้อย ได้แก่ หน่วยประมวลผล หน่วยความจำหลัก หน่วยความจำสำรอง และอินเทอร์เน็ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] วิกิพีเดีย. 2558.Cloud computingหรือการประมวลผลกลุ่มเมฆ. [Online].Avaliable:http://th.wikipedia.org/wiki/การประมวลผลแบบกลุ่มเมฆ. เข้าถึงเมื่อวันที่ 16 มกราคม 2560

[2] แปลโดยทีม blognone.2550.นิยาม Cloud computing บริษัท Gartner. [Online].Avaliable:http://lib.blognone.com/Cloud_Computing. เข้าถึงเมื่อวันที่ 19 มกราคม 2560

[3] แปลโดย ทีม blognone.2550.นิยาม Cloud computing ฟอเรสเตอร์กรุ๊ป. [Online].Avaliable:http://lib.blognone.com/Cloud_Computing. เข้าถึงเมื่อวันที่ 19 มกราคม 2560

[4] Javaboom.2551.Cloud computing. [Online].Avaliable:https://javaboom.wordpress.com/2008/07/23/whatiscloud computing/.เข้าถึงเมื่อวันที่ 20 มกราคม 2560

[5] เว็บไซต์ประกอบการเรียนวิทยาลัยเซาส์อีสบางกอก คณะบริหารธุรกิจ สาขาคอมพิวเตอร์.2555.CloudComputing. [Online].Avaliable:https://sites.google.com/site/korwten/home/fdgltru. เข้าถึงเมื่อวันที่ 20 มกราคม 2560

[6] rattanasak. 2555.รูปแบบการให้บริการของ Cloud Computing. [Online].Avaliable:http://rattanasak.jigsawoffice.com/content/content.php?mid=87&did=369&tid=1&0.เข้าถึงเมื่อวันที่ 30 มกราคม 2560

[7] Netbrigth. 2556.ประเภทการให้บริการ Cloud Computing. [Online].Avaliable:http://www.netbright.co.th/index.php?name=knowledge&file=readknowledge&id=19.เข้าถึงเมื่อวันที่ 31 มกราคม 2560

[8] Manoo Ordeedolchest.2555.คุณสมบัติของ cloud computing. [Online].Avaliable:http://ictandservices.blogspot.com/2012/05/cloudcomputing.html.เข้าถึงเมื่อวันที่ 31 มกราคม 2560

[9] เว็บไซต์ประกอบการเรียนวิทยาลัยเซาส์อีสบางกอก คณะบริหารธุรกิจ สาขาคอมพิวเตอร์.2555.องค์ประกอบของ CloudComputing. [Online].Avaliable:https://sites.google.com/site/korwten/home/fdgltru. เข้าถึงเมื่อวันที่ 9 กุมภาพันธ์ 2560

[10] เว็บไซต์ประกอบการเรียนวิทยาลัยเซาส์อีสบางกอก คณะบริหารธุรกิจ สาขาคอมพิวเตอร์.2555.ข้อดีของ Cloud Computing. [Online].Avaliable:https://sites.google.com/site/korwten/home/fdgltru. เข้าถึงเมื่อวันที่ 9 กุมภาพันธ์ 2560

[11] เว็บไซต์ประกอบการเรียนวิทยาลัยเซาส์อีสบางกอก คณะบริหารธุรกิจ สาขาคอมพิวเตอร์.2555.ข้อเสียของ Cloud Computing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Online].Avaliable:https://sites.google.com/site/korwten/home/fdgltru.
เข้าถึงเมื่อวันที่ 9 กุมภาพันธ์ 2560

[12] เว็บไซต์ประกอบการเรียนวิทยาลัยเซาส์อีสบางกอก คณะบริหารธุรกิจ สาขาคอมพิวเตอร์.2555.จุดเด่นของ Cloud Computing.
[Online].Avaliable:https://sites.google.com/site/korwten/home/fdgltru.
เข้าถึงเมื่อวันที่ 9 กุมภาพันธ์ 2560

[13] Openstack.2560.Openstack.
[Online].Avaliable:https://www.openstack.org/software/.
เข้าถึงเมื่อวันที่ 15 กุมภาพันธ์ 2560

[14] บริษัท นิภา เทคโนโลยี จำกัด.2558.ความหมาย Openstack.
[Online]. Avaliable: http://www.ispio.com/openstack/.
เข้าถึงเมื่อวันที่ 23 กุมภาพันธ์ 2560

[15] บริษัท นิภา เทคโนโลยี จำกัด.2558.Enterprise Cloud by OpenStack.
[Online]. Avaliable: http://www.ispio.com/openstack/.
เข้าถึงเมื่อวันที่ 23 กุมภาพันธ์ 2560

[16] TcktalkThai.2559.5 เหตุผลที่เหล่าองค์กรและผู้ดูแลระบบ ควรเริ่มต้นใช้ OpenStack สร้าง Private Cloud ภายในองค์กร.
[Online]. Avaliable: https://www.techtalkthai.com/5-reasons-enterprises-and-system-engineers-should-start-trying-openstack/.
เข้าถึงเมื่อวันที่ 27 กุมภาพันธ์ 2560

[17] Nutthadol Chaiparinya.2560.แนะนำ Ceph Storage.
[Online].Avaliable: https://www.throughwave.co.th/2017/04/10/แนะนำ-ceph-storage-distributed-storage-รูปแบบใหม่ที่/.เข้าถึงเมื่อวันที่ 15 มีนาคม 2560

[18] Mr. Charnsilp Chinprasert.2560.Ceph Ansible.กรุงเทพมหานคร.เข้าถึงเมื่อวันที่ 28 มีนาคม 2560

[19] Z-plus. 2559. Virtualization.
[Online].Avaliable: http://www.zplus.co.th/virtualization.htm.
เข้าถึงเมื่อวัน 30 มีนาคม 2560

[20] ฐิติพร เกรียงศักดิ์พงศ์.2558.โครงสร้างพื้นฐานทางไอที (IT Infrastructure).
[Online].Avaliable: https://sites.google.com/site/buufisclass/materials/class7-khorngsrangphunthanthangxithiitinfrastructure. เข้าถึงเมื่อวัน 19 เมษายน 2560

[21] Ubuntu.2559.Ubuntu. [Online].Avaliable: https://saixiii.com/linux-ubuntu/.
เข้าถึงเมื่อวัน 28 เมษายน 2560

[22] ITBakery.2559. Ansible Configuration Management.
[Online].Avaliable:https: http://itbakery.net/2016/08/22/ansible-configuration-management/. เข้าถึงเมื่อวัน 28 เมษายน 2560

[23] พันตรี อนันต์ สมไร่ขิง.2556.“การเปรียบเทียบการทำงานของเซิร์ฟเวอร์การจัดเก็บข้อมูลแบบคลาวด์ระหว่างฮาร์ดแวร์ทางกายภาพกับฮาร์ดแวร์เสมือน”. สารนิพนธ์วิทยาศาสตร์

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

มหาบัณฑิต สาขาวิชาเทคโนโลยีคอมพิวเตอร์และ การสื่อสาร คณะวิศวกรรมศาสตร์ , มหาวิทยาลัยธุรกิจบัณฑิตย์.

[24] สุชาติ คุ้มมะณี .2556. “ความมั่นคงของเทคโนโลยีการประมวลผลกลุ่มเมฆ”. วิทยานิพนธ์วิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ , มหาวิทยาลัยมหาสารคาม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
คู่มือการใช้งาน

การใช้งานเบื้องต้น

1) ผู้ใช้งานจะต้องมี Public IP Address เพื่อเข้าถึง Openstack Dashboard กรอกชื่อผู้ใช้ (User Name) และรหัสผ่าน (Password)
2) เปิดใช้ Browser ที่มีการติดตั้ง JavaScript และ cookies
3) ใส่ IP Address ในแถบ บน Browser ยกตัวอย่างเช่น https://192.168.1.96
4) ใส่ ชื่อผู้ใช้ (User Name) และรหัสผ่าน (Password) แล้วกด Connect
5) ด้านขวาบนสุดของหน้าจอ สามารถใช้งาน Settings tab เพื่อตั้งค่าและออกจากระบบ

หมายเหตุ

1) ถ้าผู้ใช้ เข้าสู่ระบบแบบ End user จะแสดง Project tab และ Identity tab
2) ถ้าผู้ใช้ เข้าสู่ระบบแบบ Administrator จะแสดง Project tab Admin tab และ Identity tab

ก.1 หน้าจอเข้าสู่ระบบ openstack

ก่อนที่จะเข้าใช้งานระบบจะต้องทำการล็อคอินเข้าสู่ระบบก่อน โดยกรอก username และ password จากนั้นกดปุ่ม login โดยสามารถขอ username และ password จากผู้ดูแลระบบ

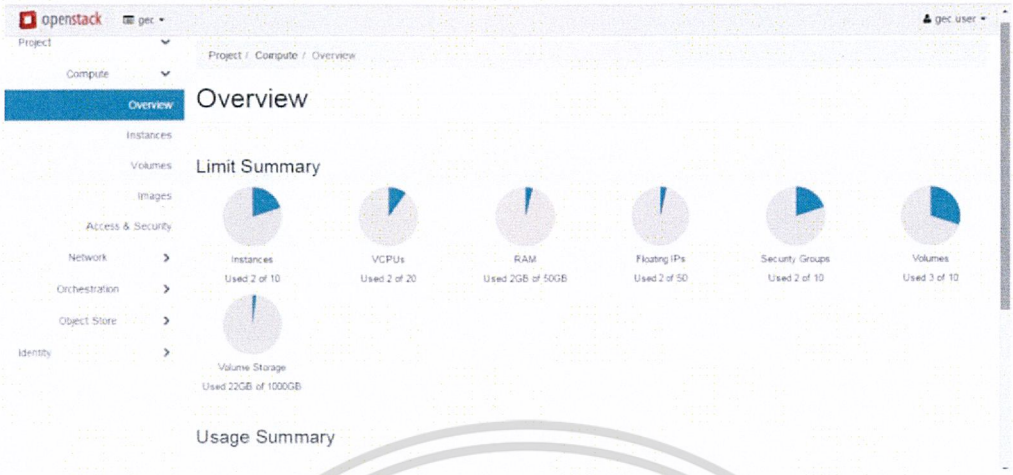
รูปที่ ก.1 หน้าจอเข้าสู่ระบบ

ก.2 เมนูหลัก

ก.2.1 Project tab

openstack มองงานที่อยู่บนระบบประมวลผลแบบกลุ่มเมฆเป็น project ผู้ใช้แต่ละคนเป็นสมาชิกได้หนึ่งหรือหลาย Project โดยภายในโปรเจคสามารถ จัดการทรัพยากรรวมถึงการจัดการเครื่องเสมือนและอิมเมจ แสดงดังรูปที่ ก.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 หน้าจอภาพรวมของ project tab

Project tab สามารถใช้งานตามเมนูต่อไปนี้

1) Compute

Overview		แสดงภาพรวมการใช้งานของ project
Instances		แสดง instance ทั้งหมดที่มี ใช้สร้างและจัดการกับ instance รวมถึงการทำ Snapshot
Volumes		สร้าง,แก้ไข และลบ Volumes ,การทำ Volumes Snapshot รวมถึงแสดง volumes ทั้งหมด
Images		สร้าง,แก้ไข และลบ Images สามารถสร้าง Instance จาก Images ได้ แบ่งการใช้งานทั้งหมดออกเป็น
Access & Security	Security Groups	แสดง, สร้าง, แก้ไข, และลบ security groups และ security group rules
	Key Pairs	แสดง, สร้าง, แก้ไข, นำเข้า, และลบ key pairs
	Floating IPs	การจัดสรร IP address ให้แต่ละ Project
	API Access	แสดง API endpoints

2) Network

Network Topology	ใช้แสดงภาพ Network โดยสามารถดูได้สองแบบคือ แบบ Topology และ Graph
Networks	สร้างและจัดการ public and private networks
Routers	สร้างและจัดการ routers
Load Balancers	สร้างและจัดการ load balancers

3) Orchestration

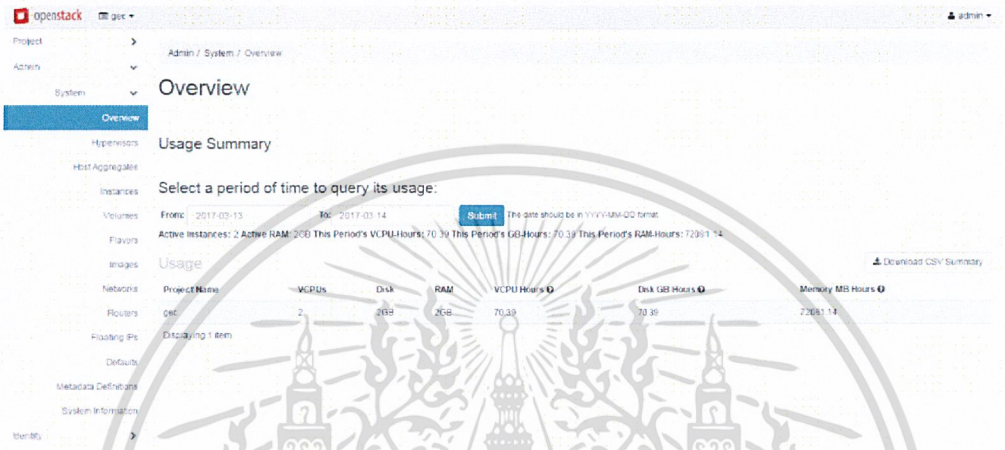
Stacks	ใช้ REST API ควบคุม multiple composite
Resource Types	cloud applications
Template Versions	แสดงรายการของทรัพยากรที่สนับสนุน HOT templates

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

4) Object Store

Containers	สร้างและจัดการ containers และ objects

ก.2.2 Admin Tab



รูปที่ ก.3 หน้าจอผู้ดูแลระบบ

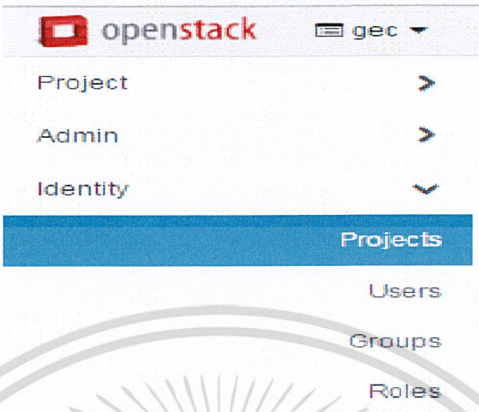
Admin tab สามารถใช้งานตามเมนูต่อไปนี้

1) System

Overview	แสดงข้อมูลพื้นฐาน
Hypervisors	แสดง hypervisor โดยย่อ
Host Aggregates	แสดง สร้าง และแก้ไข host aggregates แสดงรายการของ availability zones
Instances	แสดง หยุด เริ่มต้น พัก, ย้าย , รีบูต , และ ลบ instances ที่เป็นของผู้ใช้งานบางส่วน แต่ไม่ใช่ทั้งหมด ของโปรเจค ใช้แสดง log ของ instances หรือเข้าถึง instances ผ่าน VNC
Volumes	Volumes แสดง, สร้าง, จัดการ, และลบ volumes. Volume Types แสดง, สร้าง, จัดการ, และลบ volume types Volume Snapshots แสดง,จัดการ, และลบ volume snapshots แสดงรายละเอียดของข้อจำกัด สร้าง แก้ไข และลบ flavors.
Flavors	flavor คือขนาดของ instance
Images	แสดง, สร้าง, แก้ไขสิทธิ์ของผู้ใช้งาน, และลบ images
Networks	แสดง, สร้าง, แก้ไขสิทธิ์ของผู้ใช้งาน, และลบnetworks
Routers	แสดง, สร้าง, แก้ไขสิทธิ์ของผู้ใช้งาน, และลบ routers
Floating IPs	การจัดสรร IP address ให้แต่ละ Project
Defaults	แสดงขอบเขตของจำนวนทรัพยากรที่สามารถใช้ได้สูงสุด
Metadata Definitions	การนำเข้า namespace และแสดงข้อมูลของ metadata

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา หรือนำไปเผยแพร่ต่อ ในรูปแบบอื่นๆ ที่มีการนำไปใช้

ก.2.3 Identity Tab



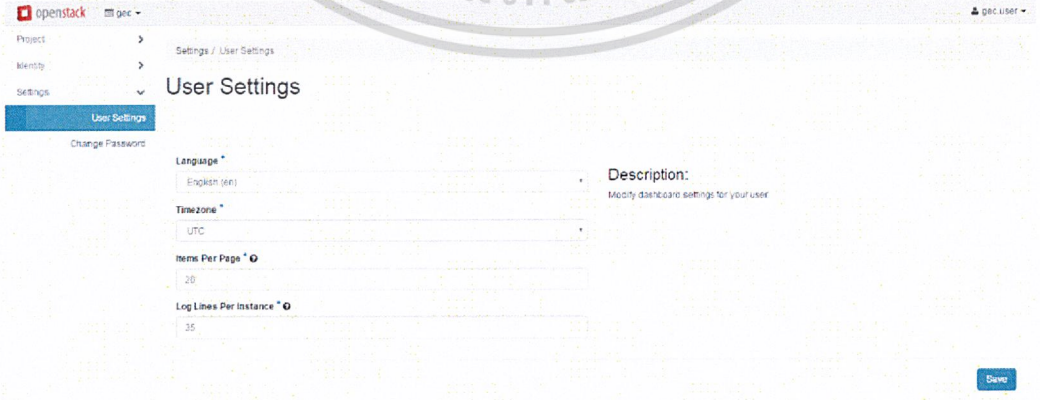
รูปที่ ก.4 ฟังก์ชัน Identity

Identity tab สามารถใช้งานตามเมนูต่อไปนี้ แสดงดังตารางที่ ก.1

ตารางที่ ก.1 ฟังก์ชันของ Identity

Projects	แสดง,สร้าง, กำหนดผู้ใช้, ลบผู้ใช้งานและโปรเจค
*Users	แสดง,สร้าง, และลบผู้ใช้งาน
*Groups	แสดง,สร้าง, และลบกลุ่ม
*Roles	แสดง,สร้าง, และลบ Roles
	*หมายเหตุ สิทธิ์ของ admin

ก.2.4 Settings Tab



รูปที่ ก.5 ฟังก์ชัน Setting Tab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

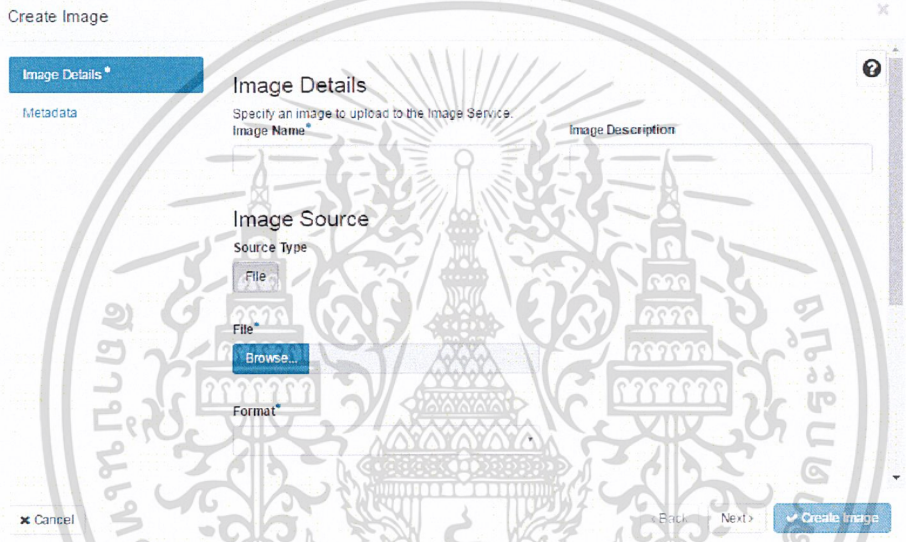
Settings tab สามารถใช้งานตามเมนูต่อไปนี้ แสดงดังตารางที่ ก.2

ตารางที่ ก.2 ฟังก์ชัน Setting

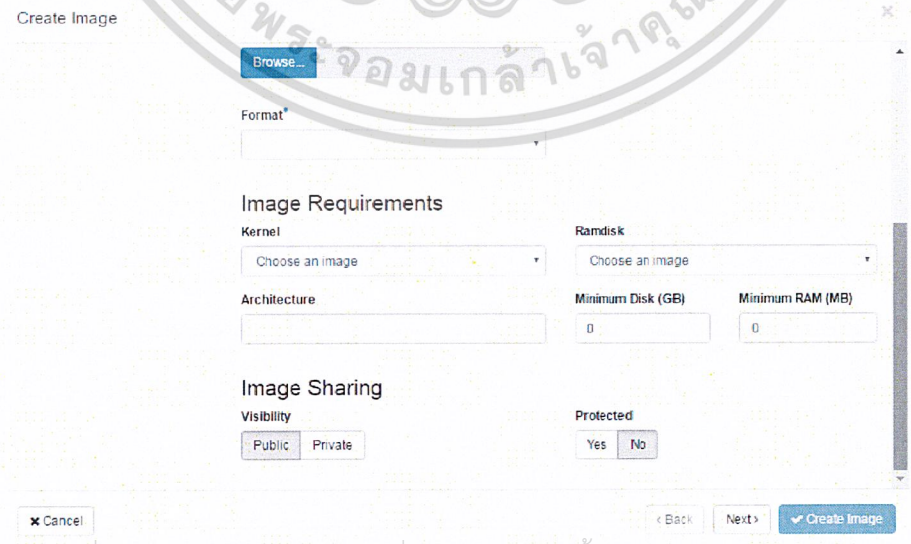
UserSettings	ตั้งค่าการใช้งาน เช่น ภาษา และ เวลา
ChangePassword	การเปลี่ยนรหัสผ่านของผู้ใช้งาน

ก.3 ฟังก์ชันการทำงานของอิมเมจ

เมื่อต้องการใช้สร้างอิมเมจให้คลิกปุ่ม Create image แล้วกรอกรายละเอียดในการสร้างอิมเมจ โดยที่สามารถนำอิมเมจมาจากเครื่องคอมพิวเตอร์ หรือนำมาจากข้างนอกระบบได้ ไฟล์อิมเมจจะนำมาใช้สำหรับบูตเครื่องเสมือน เปรียบเสมือนระบบปฏิบัติการ แสดงดังรูป ก.6-ก.7



รูปที่ ก.6 หน้าจอ Create image (1)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.7 หน้าจอ Create image (2)

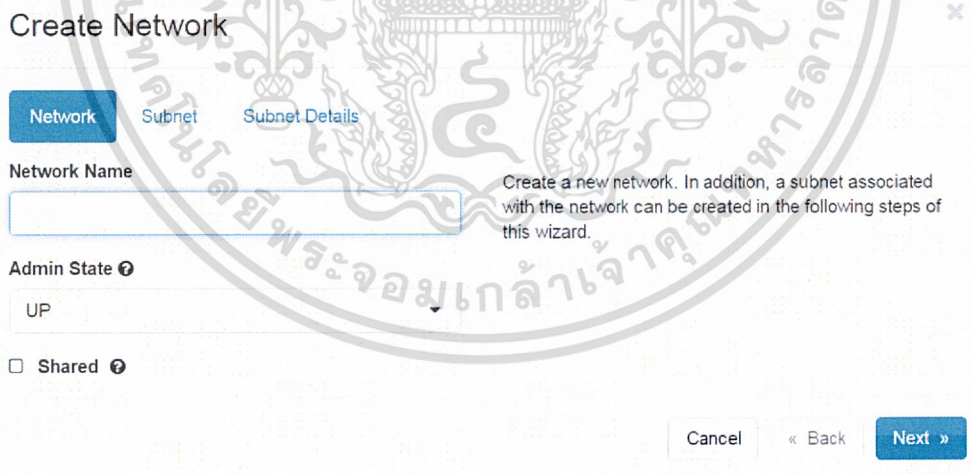
รายละเอียดที่ต้องกรอกในช่องว่างนั้น แสดงดังตารางที่ ก.3

ตารางที่ ก.3 แสดงรายละเอียดการสร้างอิมเมจ

Image Name	ใส่ชื่อ Image
Image Description	ใส่คำอธิบาย เพื่อ
Image Source	เลือก Image โดยกดปุ่ม Browse เพื่อเลือก File image ที่ต้องการ
Format	เลือก Format เช่น QCOW2
Architecture	กำหนดขนาดของ Architecture
Minimum Disk (GB)	กำหนดขนาดของ Disk (GB)
Minimum RAM (MB)	กำหนดขนาดของ RAM (MB)
Visibility	กำหนดสิทธิในการเข้าถึง Public or Private
Protected	เลือกช่องนี้เพื่อให้ทราบว่าผู้ใช้มีสิทธิ์ ลบ image ใช่หรือไม่

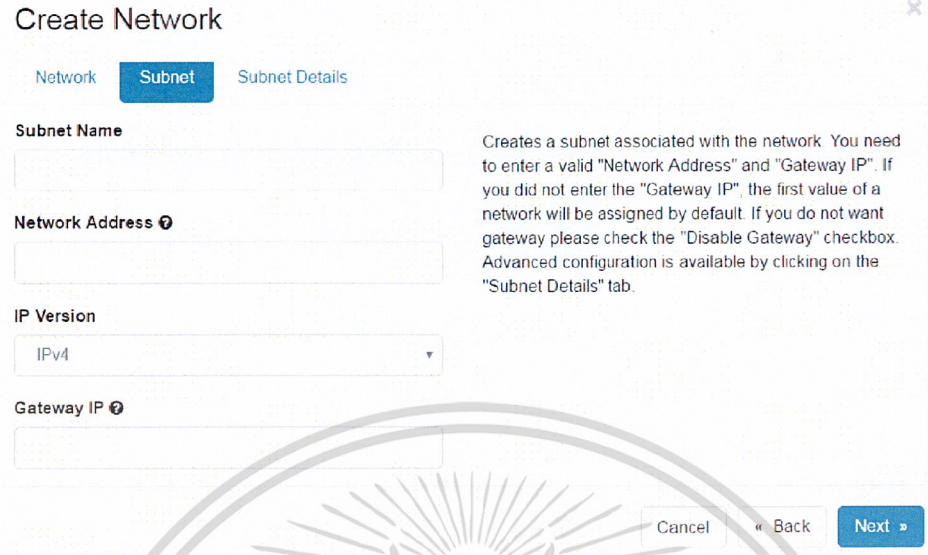
ก.4 ฟังก์ชันการทำงานของเครือข่าย (Network)

ฟังก์ชันเครือข่าย (Network) เป็นฟังก์ชันในการจัดระบบเครือข่ายทั้งระบบ ทั้งเครือข่ายภายนอก (External Network) เครือข่ายภายใน (Internal Network) โดยสามารถสร้าง subnet ขึ้นได้ มีวิธีและรายละเอียดการสร้างโดยกดที่ Create Network แสดงดังรูปที่ ก.8-ก.10



รูปที่ ก.8 หน้าจอ Create Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

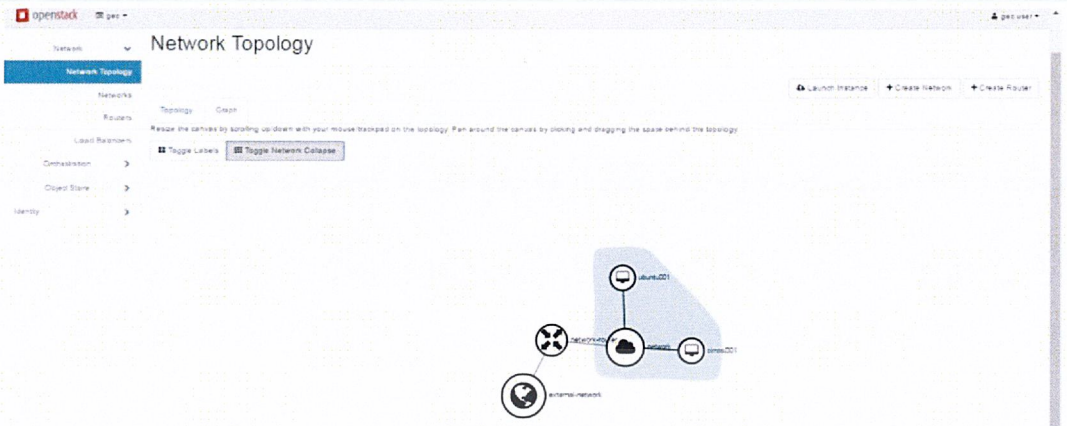


รูปที่ ก.9 หน้าจอ Create Subnet

รูปที่ ก.10 หน้าจอ Subnet Details

ข้อมูลและรายละเอียดที่ต้องกรอกลงในการสร้างเครือข่าย แสดงดังตารางที่ ก.6 ดังนี้ แล้วเมื่อสร้างเครือข่ายเรียบร้อยแล้วจะได้แผนผังเครือข่าย แสดงดังรูป ก.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.11 แผนผังเครือข่าย Network Topology

ตาราง ก.4 รายละเอียดการสร้างเครือข่าย

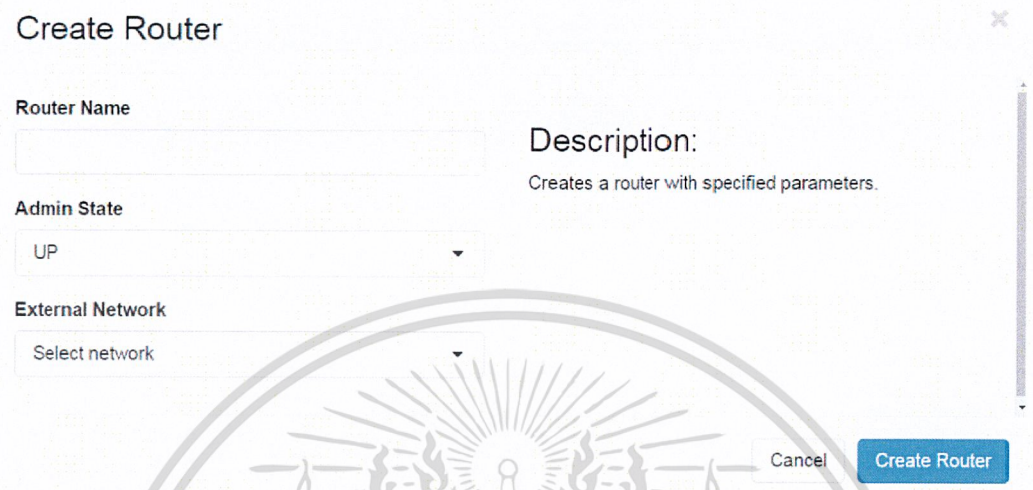
Network Name	**กำหนดชื่อของ Network สำหรับการเรียกใช้**
Shared	แชร์ network ให้แต่ละ project โดยจำกัดสิทธิ์การตั้งค่าให้ใช้ได้เพียง admin
Admin State	กำหนดสถานะเริ่มต้นของ Network
Create Subnet	การเลือกสร้าง subnet * ไม่จำเป็นต้องระบุ subnet เมื่อสร้าง network แต่ถ้าไม่ได้กำหนด subnet จะไม่สามารถเชื่อมต่อกับ instances ได้
Subnet tab	
Subnet Name	กำหนดชื่อของ subnet
Network Address	กำหนด IP address ของ subnet
IP Version	เลือกใช้ระหว่าง Pv4 หรือ IPv6
Gateway IP	กำหนด IP address ของ gateway โดยเลือกใช้ Gateway หรือไม่ใช้ก็ได้
Disable Gateway	เลือก Disable Gateway เพื่อไม่ใช้ Gateway
Allocation Pools	กำหนด IP address pools
DNS Name Servers	กำหนดชื่อ DNS server
Host Routes	กำหนด IP address ของ host routes

ก.5 ฟังก์ชันการทำงานของ Router

การเชื่อม network ที่สร้างไว้กับ Router ทำได้โดยคลิกไปที่ชื่อของ Router เมื่อมีรายละเอียดของ Router ให้กดเมนู Interfaces แล้วกด Add Interface จะมีหน้าจอขึ้นมาให้ใส่รายละเอียด แสดงรูปที่ ก.12 ให้ตั้งค่า IP Address สำหรับ Router interface เพื่อเลือก subnet.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* ถ้าเลือก NO SUBNETS AVAILABLE ค่าเริ่มต้นของ IP Address จะใช้ IP Address แรกใน Subnet เพิ่มลงในRouter Name และ Router ID จะได้รับการอัปเดตโดยอัตโนมัติ



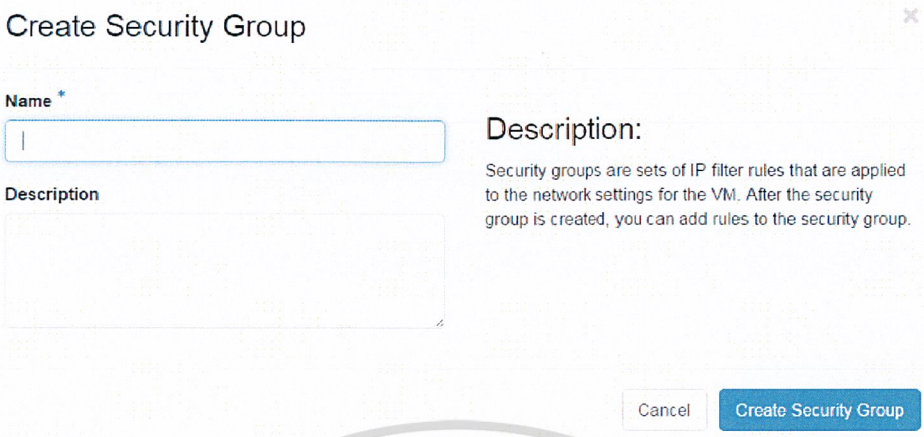
รูปที่ ก.12 หน้าจอ Create Router

รูปที่ ก.13 หน้าจอ Add Interface

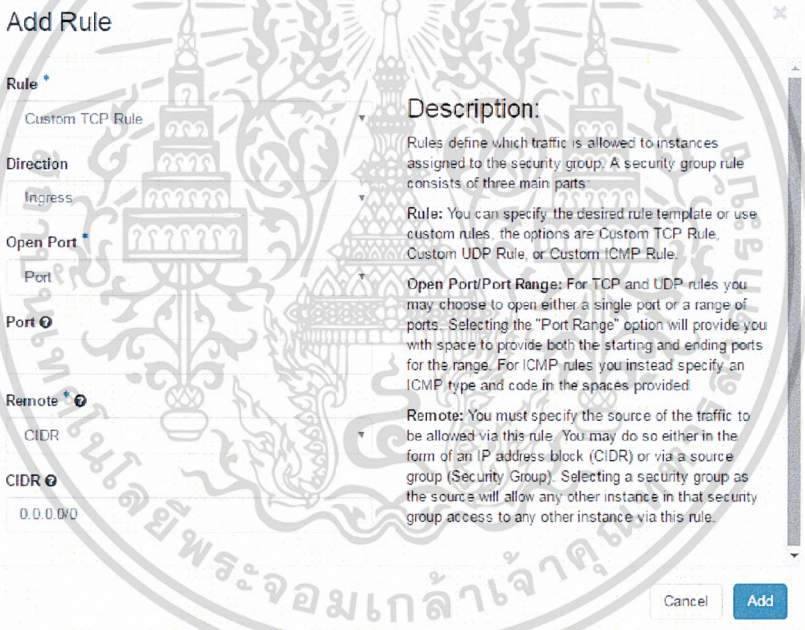
ก.6 ฟังก์ชันการทำงานของ Access & Security

ก่อนที่จะสร้าง เครื่องเสมือน หรือ Instance ควรเพิ่มกฎของ Security groups เพื่อให้ผู้ใช้สามารถ ping และใช้ SSH เพื่อเชื่อมต่อกับ Instance ได้ Security groups คือชุดของการตรวจสอบ IP ที่กำหนดการเข้าถึงเครือข่ายและใช้กับ Instance ทั้งหมดภายใน Project ซึ่งในการทำคุณสามารถเพิ่มกฎ (Add Rule) ลงใน Security groups แสดงดังรูป ก.14 และ ก.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.14 หน้าจอ Create Security Group

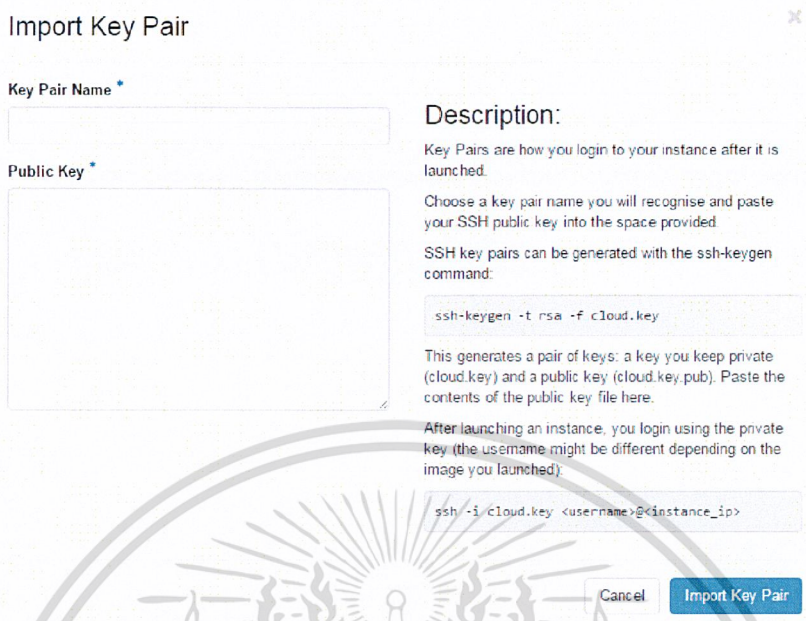


รูปที่ ก.15 หน้าจอ Add Rule

ก.7 ฟังก์ชันการทำงานของ Key Pair

Key Pair คือข้อมูลรับรอง SSH ที่ถูกส่งเข้าไปใน Instance เมื่อมีการ Launched Instance แต่ละ Instance ควรมี Key Pair อย่างน้อยหนึ่งคีย์ หากผู้ใช้สร้าง Key Pair จากที่อื่น สามารถนำเข้าไปใช้ใน OpenStack ได้ โดย Key Pair สามารถใช้กับ Instance ได้หลาย Instance ที่อยู่ใน Project เดียวกัน การสร้าง key Pair แสดงดังรูป ก.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.16 หน้าจอ Import Key Pair

ก.8 ฟังก์ชันการทำงานของ Allocate a floating IP address to an instance

เมื่อ Instance ถูกสร้างขึ้นใน OpenStack จะมีการ fixed IP address ในเครือข่ายที่มีการกำหนด Instanceไว้ IP address นี้มีการเชื่อมโยงกับ Instance จนกระทั่ง Instance ถูกยกเลิก

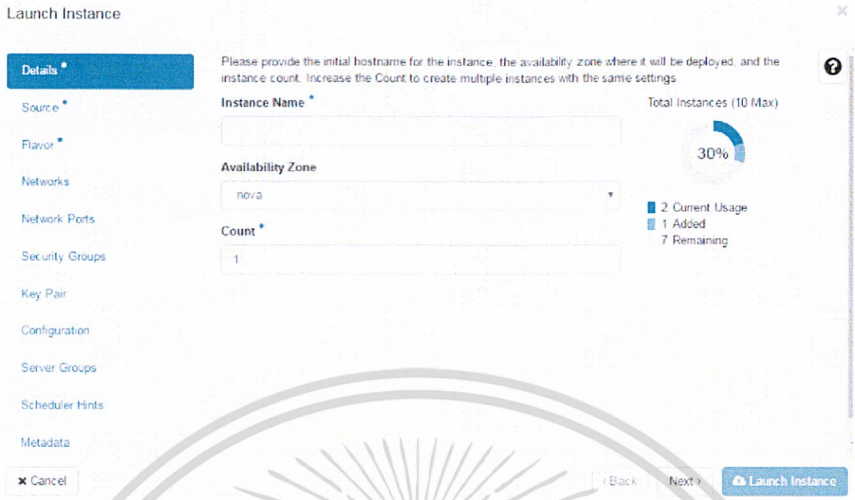
อย่างไรก็ตามนอกเหนือจาก IP address แล้วยังมี floating IP address สามารถแนบไปกับ Instance ได้เช่นกัน floating IP address สามารถแก้ไขได้ตลอดเวลาโดยไม่คำนึงถึงสถานะของ Instance แสดงดังรูปที่ ก.17

รูปที่ ก.17 หน้าจอ Allocate a floating IP

ก.9 ฟังก์ชันการสร้างเครื่องเสมือน (Instance)

การสร้างเครื่องเสมือนให้คลิกที่ปุ่ม Launch Instance จะแสดงหน้าจอดังรูปที่ ก.18 จากนั้นกรอกรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.18 หน้าจอการสร้างเครื่องเสมือน

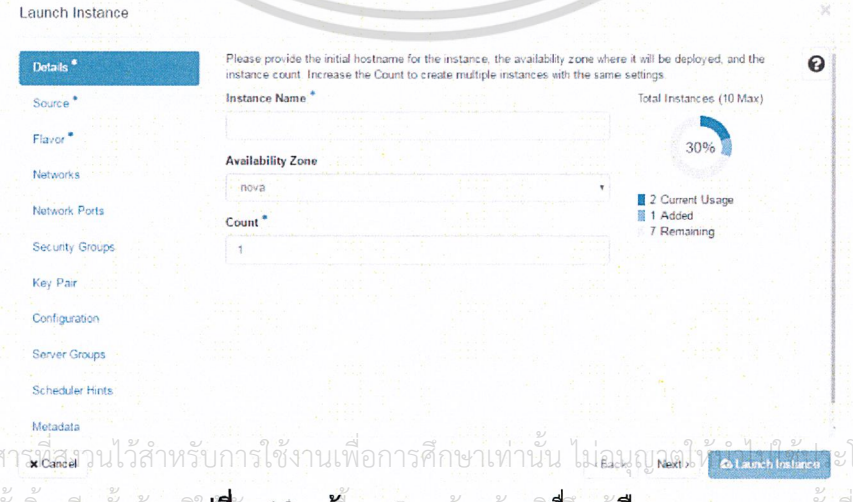
รายละเอียดทางด้านซ้ายมือมีดังตารางที่ ก.3

ตารางที่ ก.5 Details tab

Instance Name	ระบุชื่อของ เครื่องเสมือน
Availability Zone	ค่านี้ถูกตั้งค่าเป็น Availability Zone ใช้งานโดยผู้ให้บริการระบบคลาวด์ (เช่น us-west หรือ apac-south) สำหรับบาง กรณีอาจเป็น nova
Count	ในการสร้างหลาย instances ให้ป้อนค่าที่มากกว่า 1 ค่าเริ่มต้นคือ 1

1) Source tab

เลือกตามรายการที่ผู้ใช้ต้องการและปรากฏ

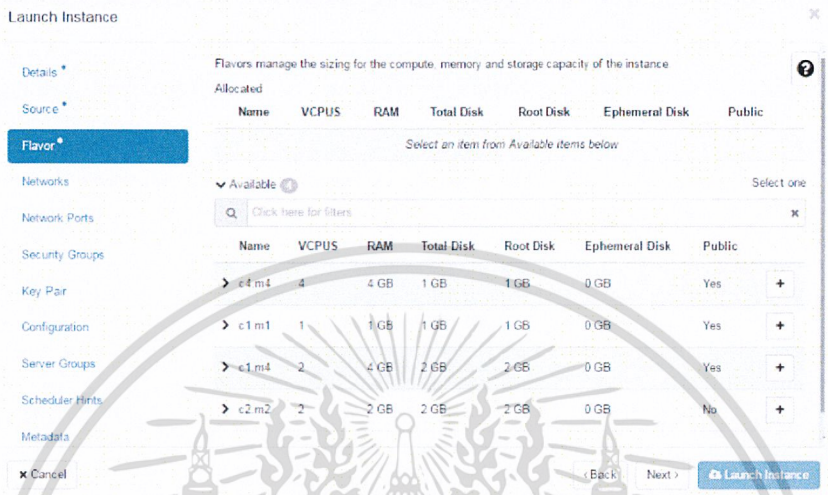


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.19 หน้าจอ Source ของเครื่องเสมือน

2) Flavor tab

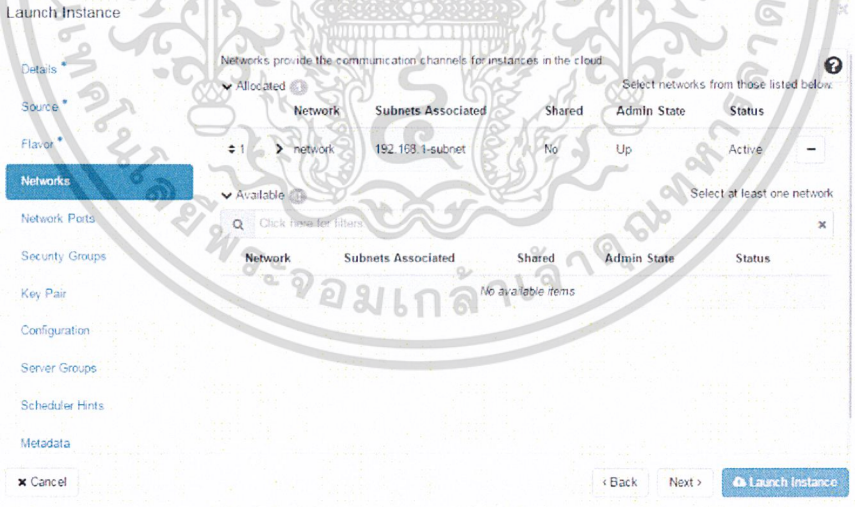
กำหนดขนาดของ instance โดยกดเครื่องหมายบวกเพื่อเลือกขนาดที่ต้องการ



รูปที่ ก.20 หน้าจอ flavor ของเครื่องเสมือน

3) Network Ports tab

เปิดใช้พอร์ตที่ต้องการกำหนดให้กับ instance

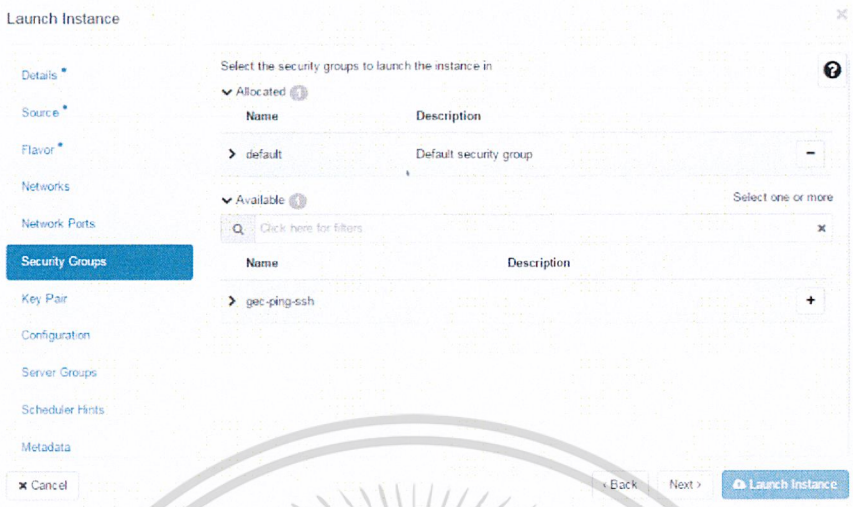


รูปที่ ก.21 หน้าจอ Network ของเครื่องเสมือน

4) Security Groups tab

เปิดใช้ Security groups ที่ต้องการกำหนดให้กับ instance

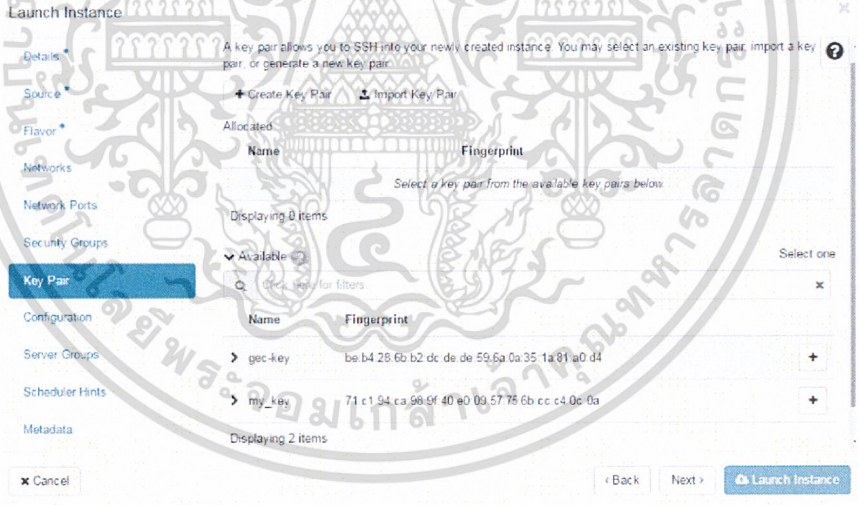
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.22 หน้าจอ Security Groups tab

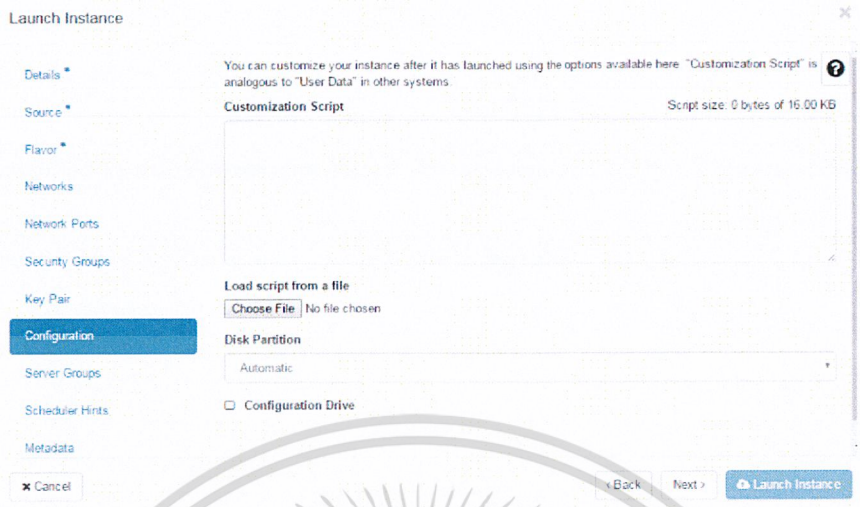
5) Key Pair tab

กำหนด Key Pair โดยคลิกเครื่องหมายบวกเพื่อเลือก Key Pair ที่ต้องการใช้งาน



รูปที่ ก.23 หน้าจอ Key Pair ของเครื่องเสมือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.24 หน้าจอ Configuration ของเครื่องเสมือน

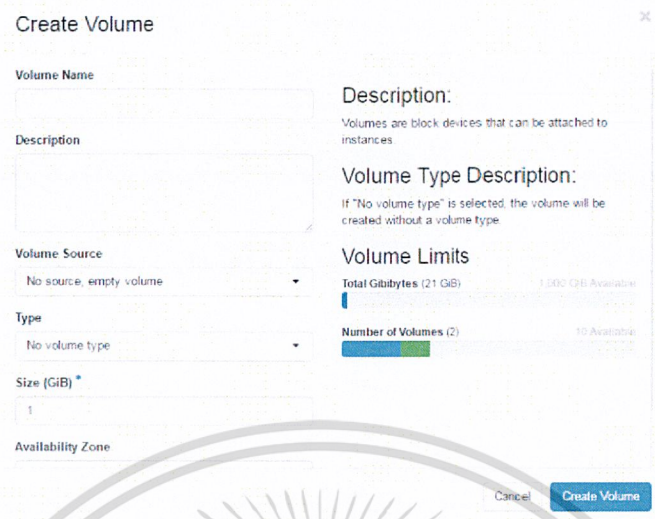
เป็นการกำหนด script ที่จะรันหลังจากสร้าง instance

รูปที่ ก.25 หน้าจอ Metadata ของเครื่องเสมือน

ก.10 ฟังก์ชันการสร้าง Volume

เป็นอุปกรณ์จัดเก็บข้อมูลแบบ block storage ที่อยู่ใน Instance เพื่อเปิดใช้ persistent storage คุณสามารถนำ Volume and Instance ที่กำลังทำงานอยู่หรือนำ Volume ออกและแนบไฟล์กับ Instance อื่นได้ทุกเมื่อ นอกจากนี้คุณยังสามารถ create a snapshot หรือ delete a volume ได้ เฉพาะผู้ดูแลระบบ(Admin) เท่านั้นที่สามารถสร้าง Volume ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.26 หน้าจอ Volume ของเครื่องเสมือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

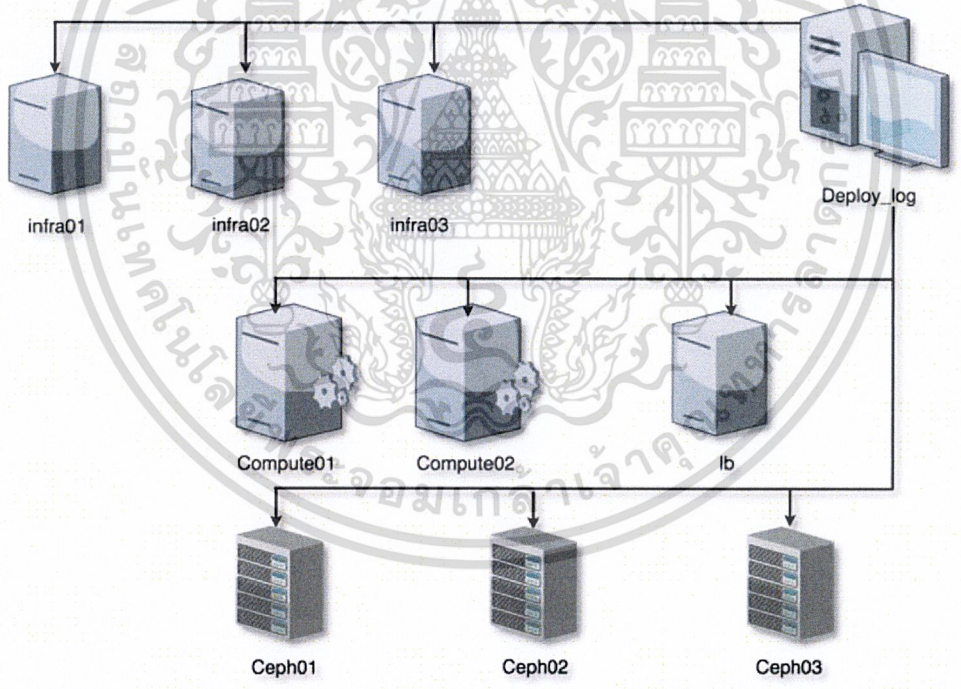
รายละเอียดการพัฒนาระบบประมวลผลของกลุ่มเมฆแบบส่วนตัวด้วย openstack และ ceph

ข.1 การเตรียมสภาพแวดล้อมของระบบ openstack

Opensatck เป็นซอฟต์แวร์ที่ทำงานบนระบบปฏิบัติการลินุกซ์ ในการทดลองเลือกใช้ระบบปฏิบัติการ Ubuntu server 16.04 ติดตั้งลงบนเครื่องเซิร์ฟเวอร์ เพื่อเป็นระบบโครงสร้างพื้นฐานของ openstack และ ceph โดยมีรายละเอียดการติดตั้งดังนี้

ข.1.1 บูตไฟล์อิมเมจของระบบปฏิบัติการ Ubuntu

ให้เครื่องเซิร์ฟเวอร์บูตไฟล์อิมเมจของระบบปฏิบัติการ Ubuntu 16.04 ที่เตรียมไว้ด้วยวิธีต่างๆ ซึ่งวิธีในการบูตไฟล์อิมเมจมีความแตกต่างกันในแต่ละยี่ห้อและรุ่นของเครื่องเซิร์ฟเวอร์ ให้ศึกษาหาวิธีการใช้เครื่องเซิร์ฟเวอร์นั้นๆ โดยเมื่อบูตไฟล์อิมเมจเรียบร้อยแล้วให้ทำการสร้างเครื่องเสมือน จำนวนทั้งหมด 10 เครื่อง แสดงดังรูปที่ ข.1

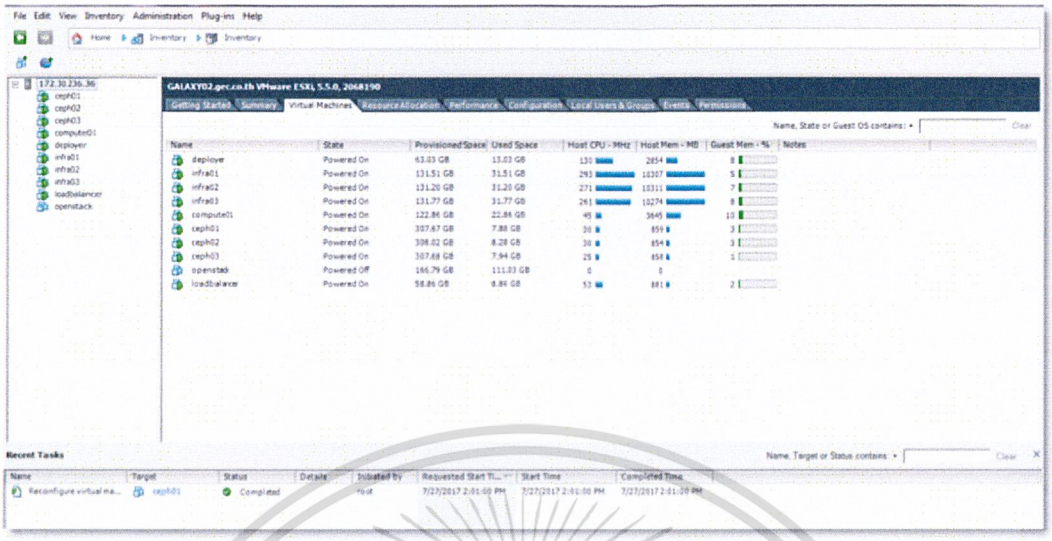


รูปที่ ข.1 สถาปัตยกรรมของระบบ openstack

ข.1.1 สภาพแวดล้อมและเครือข่าย (Network) ของระบบ

โดยการสร้างเครื่องเสมือนจะใช้ VMware vSphere client รีโมทเข้าไปในเครื่องเซิร์ฟเวอร์เพื่อทำการสร้างเครื่องเสมือน แสดงดังรูปที่ ข.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 สภาพแวดล้อมและทรัพยากรโดยรวมของระบบ

จากนั้นทำงานตั้งค่าเครือข่าย (Network) ของแต่ละเครื่องโดยกำหนด ip address ของแต่ละเครื่องแสดงตารางดังตาราง ข.1

ตารางที่ ข.1 แสดงการตั้งค่าเครือข่าย (Network)

การตั้งค่า Network			
เครื่อง	ชื่อเครื่อง	IP ในเครื่อง	ชื่อเครือข่าย
1	Deploy_log	172.30.100.245 172.30.236.10 10.10.3.10 10.10.2.10	 (br-mgmt) (br-storage) (br-vxlan)
2	Infra01	172.30.100.247 172.30.236.11	 (br-mgmt)
3	Infra02	172.30.100.248 172.30.236.12	 (br-mgmt)
4	Infar03	172.30.100.249 172.30.236.13	 (br-mgmt)
5	Compute01	172.30.100.222 172.30.236.14 10.10.3.14 10.10.2.14	 (br-mgmt) (br-storage) (br-vxlan)
6	Compute02	172.30.100.223 172.30.236.18 10.10.3.18	 (br-mgmt) (br-storage)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

		10.10.2.18	(br-vxlan)
7	Ceph01	172.30.100.217 172.30.236.15 10.10.3.15	 (br-mgmt) (br-storage)
8	Ceph02	172.30.100.217 172.30.236.16 10.10.3.16	 (br-mgmt) (br-storage)
9	Ceph03	172.30.100.219 172.30.236.17 10.10.3.17	 (br-mgmt) (br-storage)
10	lb	172.30.100.181 172.30.236.9	 (br-mgmt)

ข.2 การติดตั้งซอฟต์แวร์ openstack , ceph และเขียนค่าการทำงาน

ข.2.1 การติดตั้งเครื่อง Deploy_log

1) เข้าใช้งานเครื่อง Deploy_log ทำการ update packages และติดตั้งโปรแกรมดังต่อไปนี้

```
# apt-get update
# apt-get dist-upgrade
# apt-get install aptitude build-essential git ntp ntpdate \
openssh-server python-dev sudo -y
```

2) ติดตั้ง Ansible ด้วยคำสั่ง

```
# apt-get install software-properties-common -y && \
apt-add-repository ppa:ansible/ansible -y && \
apt-get update -y && \
apt-get install ansible -y
```

3) ติดตั้ง NTP

```
# apt-get install -y ntp ntpstat ntpdate && \
# systemctl enable ntp && \
# systemctl start ntp
```

ข.2.1 การติดตั้ง ceph

1) เข้าใช้งานเครื่อง Deploy_log ทำการ clone repository ลงมาที่เครื่อง Deploy_log โดยใช้คำสั่ง

```
# git clone -b v2.2.11 https://github.com/ceph/ceph-ansible.git
```

2) สร้างไฟล์ inventory

```
# cd ceph-ansible
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# vi inventory
```

*แก้ไขไฟล์ inventory

```
[mons]
ceph01 ansible_host=10.10.3.15
ceph02 ansible_host=10.10.3.16
ceph03 ansible_host=10.10.3.17
[osds]
ceph01 ansible_host=10.10.3.15
ceph02 ansible_host=10.10.3.16
ceph03 ansible_host=10.10.3.17
# ansible -m ping -i inventory all
```

3) Run playbook โดยเข้าใช้งาน Deploy_log
สร้างโฟลเดอร์และคัดลอกไฟล์ตามนี้

```
#  cp site.yml.sample site.yml
# cp group_vars/all.yml.sample group_vars/all.yml
# cp group_vars/mons.yml.sample group_vars/mons.yml
```

4) Run playbook

```
# ansible-playbook site.yml -i inventory -u root
```

ข.2.2 การติดตั้ง openstack

1) ทำการ clone repository ลงมาที่เครื่อง Deploy_log โดยใช้คำสั่'

```
# git clone -b stable/ocata
https://git.openstack.org/openstack/openstack-ansible \
/opt/openstack-ansible
```

2) ทำการรัน Scripts เพื่อติดตั้ง Bootstrap โดยใช้คำสั่ง

```
# cd /opt/openstack-ansible
# scripts/bootstrap-ansible.sh
```

3) Prepare Target Host คำสั่งต่อไปนี้ จะทำการติดตั้งในเครื่องดังต่อไปนี้
infra01, infra02, infra03, compute01, lb01, Deploy_log

```
# apt-get update && \
 apt-get dist-upgrade -y && \
apt-get install bridge-utils debootstrap ifenslave ifenslave-2.6 \
 lsof lvm2 ntp ntpdate openssh-server sudo tcpdump vlan -y
# echo 'bonding' | tee --append /etc/modules
# echo '8021q' | tee --append /etc/modules
```

ข.2.3 การตั้งค่าเครื่อง Deployment

1) เข้าใช้งาน Deploy_log สร้างโฟลเดอร์และคัดลอกไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# mkdir -p /etc/openstack_deploy
# cp -R /opt/openstack-ansible/etc/openstack_deploy/*
/etc/openstack_deploy/
# cd /etc/openstack_deploy
# cp openstack_user_config.yml.prod.example
openstack_user_config.yml
# cp user_variables.yml.prod user_variables.yml
# vi openstack_user_config.yml
```

*แก้ไขไฟล์ openstack_user_config.yml และตั้งเครือข่าย (Network) ตามตารางที่ ข.1

2) จากนั้นแก้ไข

```
# vi user_variables.yml เพื่อให้ ceph ทำงานร่วมกับ openstack
haproxy_keepalived_external_vip_cidr: "172.30.100.0/24"
haproxy_keepalived_internal_vip_cidr: "172.30.236.0/24"
haproxy_keepalived_external_interface: ens160
haproxy_keepalived_internal_interface: br-mgmt
horizon_time_zone: Asia/Bangkok
horizon_images_upload_mode: legacy
horizon_session_timeout: 28800
glance_default_store: rbd
glance_notification_driver: noop
glance_ceph_client: glance
glance_rbd_store_pool: images
glance_rbd_store_chunk_size: 8
nova_libvirt_images_rbd_pool: vms
cinder_ceph_client: cinder
cephx: true
ceph_mons:
- 10.10.3.15
- 10.10.3.16
- 10.10.3.17
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.3 ทำการ run scripts

1) เข้าใช้งานเครื่อง lb Setup Hosts

```
# openstack-ansible haproxy-install.yml
# openstack-ansible setup-hosts.yml
```

2) เข้าใช้งานเครื่อง Deploy_log Configuration Service Credentials

```
# cd /opt/openstack-ansible/scripts
# python pw-token-gen.py --file
/etc/openstack_deploy/user_secrets.yml
```

3) เข้าใช้งานเครื่อง Deploy_log Setup Infrastructure

```
# cd /opt/openstack-ansible/playbooks
# openstack-ansible setup-infrastructure.yml --syntax-check
```

4) เข้าใช้งานเครื่อง Deploy_log

```
# openstack-ansible setup-infrastructure.yml
# ansible galera_container -m shell \
 -a "mysql -h localhost -e 'show status like
\"%wsrep_cluster_%\";'"
> echo '8021q' |  tee --append /etc/modules
```

5) เข้าใช้งานเครื่อง Deploy log Install Openstack

```
# openstack-ansible setup-openstack.yml
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้