

เอกสารเพิ่มเติม
โครงการพัฒนาและวิจัยระบบต้นแบบ Private Cloud
ด้วยซอฟต์แวร์โอเพนซอร์ส OpenStack สำหรับองค์กร



บริษัท ไอที เบเคอรี่ จำกัด

สารบัญ

OpenStack Telemetry.....	6
ลำดับการการทำงานของ Ceilometer.....	6
สถาปัตยกรรมของ Ceilometer.....	7
ขั้นตอนการติดตั้ง Ceilometer.....	9
กระบวนการทำงาน Workflow ภายในของ Ceilometer.....	10
1 การใช้งาน Ceilometer ด้วย OpenStack dashboard.....	10
2 การใช้งาน Ceilometer ด้วย OpenStack CLI.....	18
กระบวนการ Cloud Orchestration ใน Openstack.....	21
สถาปัตยกรรม Openstack Heat.....	21
องค์ประกอบภายใน Heat Component ประกอบด้วย.....	21
ขั้นตอนติดตั้งเพื่อการใช้งาน HEAT Template (Installation).....	21
การแก้ไขปัญหาเพื่อให้สามารถใช้ Heat ทำงาน.....	22
1. ปัญหานี้องมาจากการ authentication failed ของ head_admin.....	22
2. ปัญหาการไม่แสดงผล icon.....	26
การทดสอบกระทำการระหว่าง Heat Template และ Heat Engine.....	28
แนวทางการทดสอบ.....	28
เตรียมระบบให้พร้อมสำหรับการทดสอบ 1: พื้นฐานการสร้าง Instance ด้วย HOT (แบบ single instance).....	29
การทดสอบ 1: พื้นฐานการสร้าง Instance ด้วย HOT (แบบ single instance).....	37
คำอธิบายรายละเอียดแต่ละส่วนของ template stack1.yaml.....	37
ดำเนินการทดสอบ.....	38
วิธีการลบ stacks.....	44
การทดสอบ 2: สร้าง Database Server ด้วย การ run script ภายใน Heat Template.....	45
คำอธิบายรายละเอียดแต่ละส่วนของ template stack2.yaml.....	45
การทดสอบ template.....	47
ความสัมพันธ์ระหว่าง Horizon Form และ script.....	55
การทดสอบ 3: สร้าง instance Autoscale multinode (แบบไม่มี load balance).....	59
คำอธิบายรายละเอียดแต่ละส่วนของ template stack3.yaml.....	59
การดำเนินการทดสอบ.....	62
ขั้นตอนเพิ่ม float ip ให้แก่ instance.....	66
พื้นฐาน การทำงานของ service ชื่อ LBaaS (Load-Balancing-as-a-Service).....	71
กระบวนการกระจายโหลด Connection ด้วย Round robin algorithms.....	72
การทดสอบ 4: สร้าง instance Autoscale multinode (แบบมี load balance).....	73
การสร้าง Heat Stack Environment การทดสอบ 4.....	73
คำอธิบายรายละเอียดแต่ละส่วนของ template lb-env.yaml.....	74

ปรับลดเวลาเพื่อการทดสอบ ใน /etc/ceilometer/pipline.yaml.....	85
การทดสอบ 5 การสร้าง Auto scale สำหรับ Wordpress ให้รองรับ Cloud Application.....	89
คำอธิบายรายละเอียดแต่ละส่วนของ template.....	89
เริ่มต้นการทดสอบ.....	95
การทดสอบ 6 การสร้าง Auto scale สำหรับ Wordpress ร่วมกับ Mysql Galera Cluster.....	115
การติดตั้งการใช้งาน MaraiDB Galera Cluster.....	123
การติดตั้งการใช้งาน MaraiDB Galara Cluster สำหรับ database Master.....	123
การติดตั้งการใช้งาน MaraiDB Galara Cluster สำหรับ database server ลูก.....	125
การติดตั้ง webserver ด้วย script ชื่อ web.sh.....	126
การสร้าง Heat template.....	128
อ้างอิง: ข้อกำหนด Heat Orchestration Template (HOT) Specification.....	132
ความสำคัญของ Heat Template Version.....	132
version 2013-05-23.....	132
version 2014-10-16.....	133
version 2015-04-30.....	133
version 2015-10-15.....	133
รูปแบบการกำหนดค่าใน ส่วน Parameter Groups Section.....	134
รูปแบบการกำหนดค่าใน ส่วน Parameter Section.....	134
พักร์ชั้นภายใน hot (Intrinsic function).....	139
พักร์ชั้น get_param, get_attr, get_resource.....	146
สรุปการใช้งานคำสั่ง เพื่อการบริหารจัดการ Heat Template.....	150
1 แสดง stack ทั้งหมดที่ใช้งานอยู่.....	150
# heat stack-list.....	150
.....	150
2 แสดงรายละเอียดของ stack.....	150
#heat stack-show webfarm.....	150
3 แสดงรายละเอียดของ event ที่เกิดขึ้นภายใน stack.....	150
# heat event-list webfarm.....	150
.....	150
3 แสดงผลย่ออย่างเป็นแต่ละ event.....	150
# heat event-show webfarm cpu_alarm_low f5840c4a-7ce2-4c77-a644-53a0f5629ea1.....	150
4 แสดงผล resource ใน stack.....	150
# heat resource-list webfarm.....	150
# heat resource-show webfarm cpu_alarm_high.....	150
5 แสดงผล resource metadata ของ resource ใน stack.....	151
# heat resource-metadata webfarm pool.....	151

6 แสดงผล template ของ resource ใน stack.....	151
# heat template-show webfarm.....	151
.....	151
6 แสดงผล template ของ resource ใน stack.....	151
# heat stack-delete webfarm.....	151

จากความหมายของ cloud computing ที่ก่อตัวไว้โดย NIST

อ้างอิงจาก <http://www.nist.gov/itl/cloud/>

1. Rapid Elasticity

เป็นคุณลักษณะที่สามารถ เพิ่ม-ลด resource ได้ตามความต้องการ (OpenStack Orchestration ที่จะกล่าวถึงในหนังสือเล่มนี้)

2. Mesured Service

เป็นคุณลักษณะที่ก่อตัวไว้ว่าต้องสามารถวัดสิ่งใช้ผู้ใช้บริการใช้ไปได้ (OpenStack Telemetry)

3. On-Demand Self-Service

เป็นคุณลักษณะที่ผู้ใช้งานโดยเพียงพากาศจากผู้ให้บริการน้อยที่สุด (การ Access เข้าใช้งาน OpenStack ไม่ว่าจะเป็น WebUI CLI หรือ API)

4. Ubiquitous Network Access

เป็นคุณสมบัติที่ว่าด้วยการเข้าใช้งานที่เข้าใช้งานได้ทุกที่เมื่อมีการต่อ network

5. Location-Independent Resource Pooling

เป็นคุณสมบัติที่ว่าด้วยการเมื่อเข้าใช้งานโดยไม่อิงอยู่กับ location ของ infrastructure ที่ให้บริการ (ใน OpenStack นั้นได้นำเสนอการแยก infrastructure เอาไว้ 4 วิธี ได้แก่ Cells, Regions, Availability Zone, และ Host aggregates)

เราจะสังเกตเห็นได้ว่าสิ่งที่ยังไม่ได้กล่าวไว้ในเล่มก่อนได้แก่ OpenStack Orchestration(Rapid Elasticity) และ OpenStack Telemetry(Mesured Service)

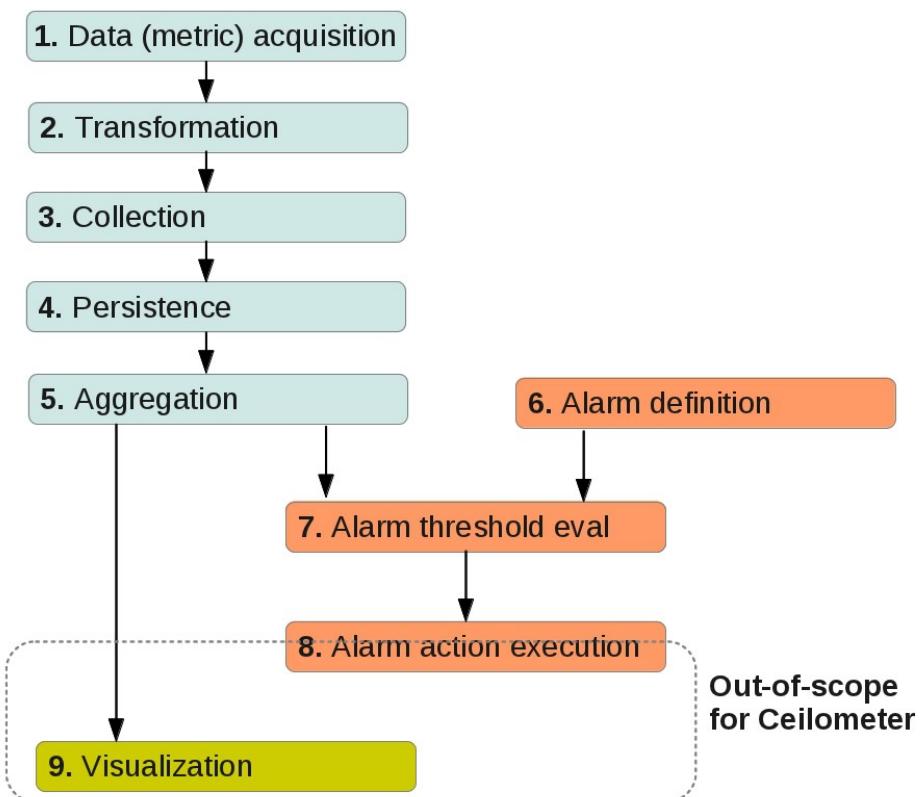
OpenStack Telemetry

มีอีกชื่อหนึ่งว่า Ceilometer เป็น project ที่เปรียบได้กับมาตราการวัดการทำงานใน OpenStack Environment โดยจะสามารถทำได้ทั้งการเก็บข้อมูลการใช้งาน การ monitor และการส่งสัญญาณเตือน เพื่อให้มีการทำอะไรบางอย่าง เช่น การทำ auto scale

ลำดับการการทำงานของ Ceilometer

ก่อนที่จะทำความเข้าใจเกี่ยวกับการทำงานของ Ceilometer นั้นจะต้องทำความเข้าใจในส่วนของ หน้าที่หลักของ Ceilometer โดยหน้าที่ดังกล่าว ประกอบด้วยกระบวนการ และขั้นตอน การรวมรวมข้อมูล (Aggregation) การวัด (Measurement) สร้างระบบเตือน (Alarm) และส่งข้อมูลไปยัง Dashboard เพื่อการแสดงผลแก่ผู้ใช้งาน

โดย Ceilometer สามารถวัดข้อมูลได้ทั้งที่เป็นส่วนของ physical Resource และส่วนที่เป็น Virtual Resource ของการทำงาน ใน Openstack เพื่อนำมาสร้างเป็นระบบบริหารจัดการระบบ และใช้ประโยชน์จากข้อมูล ดังกล่าว โดยมีวัตถุประสงค์สำหรับการ Deploy ระบบคลาวด์, การวิเคราะห์การทำงาน การติดตามการใช้งาน Resource ของระบบ และเริ่มต้นกระตุ้น (Trigger) เพื่อให้ให้กระบวนการทำงานอื่นๆ ต่อไปเมื่อถึงข้อกำหนด (Criteria) ที่ได้กำหนดไว้ภายในเงื่อนไขของ Ceilometer



จากรูปจะมีการทำงานทั้งหมด 9 ขั้นตอนดังนี้

ขั้นตอนที่ 1 Ceilometer จะทำการเก็บข้อมูลจาก OpenStack component ที่อยู่ในระบบ แต่ข้อมูลที่เก็บมาเป็นข้อมูลดิบที่สามารถทำความเข้าใจได้ยากมีปริมาณที่มากซึ่งข้อมูลบางอย่างก็ได้จำเป็นที่จะต้องใช้

ขั้นตอนที่ 2 เป็นขั้นตอนของการแปลงข้อมูลดิบเป็นข้อมูลที่ได้ถูกตั้งค่าเอาไว้เพื่อไว้สำหรับการเก็บอยู่ในที่

เก็บสำหรับการนำไปใช้ในส่วนอื่นๆ ต่อไป ยกตัวอย่าง เช่น ถ้าระบบได้ถูกตั้งค่าเอาไว้ให้เก็บเฉพาะค่า cpu utilization ก็จะเก็บแปลงข้อมูลเป็น format ที่จะใช้สำหรับการรวมและเก็บข้อมูลในขั้นตอนถัดไป

ขั้นตอนที่ 3 จะเป็นการรวมข้อมูลที่ได้จากการเก็บและแปลงข้อมูลให้อยู่ในรูปแบบที่สามารถเข้ามาใช้ได้โดย 2 ขั้นตอนที่ผ่านมาดังล่าวจะทำงานใกล้ชิดกับ OpenStack component อื่นๆ แต่สำหรับขั้นตอนนี้จะเป็นการเข้ามารวมกับเพื่อที่จะเตรียมสำหรับการเก็บเพื่อที่จะได้นำเอาไปใช้ต่อไป

ขั้นตอนที่ 4 จะเป็นการเก็บที่ได้ถูกรวมมาถูกเก็บอยู่ใน Database

ขั้นตอนที่ 5 เป็นการรวมข้อมูล จะจัดกลุ่มของข้อมูล

ขั้นตอนที่ 6 กำหนดนิยามและเงื่อนไขต่างๆ ของสัญญาณเตือน แต่ละชนิด

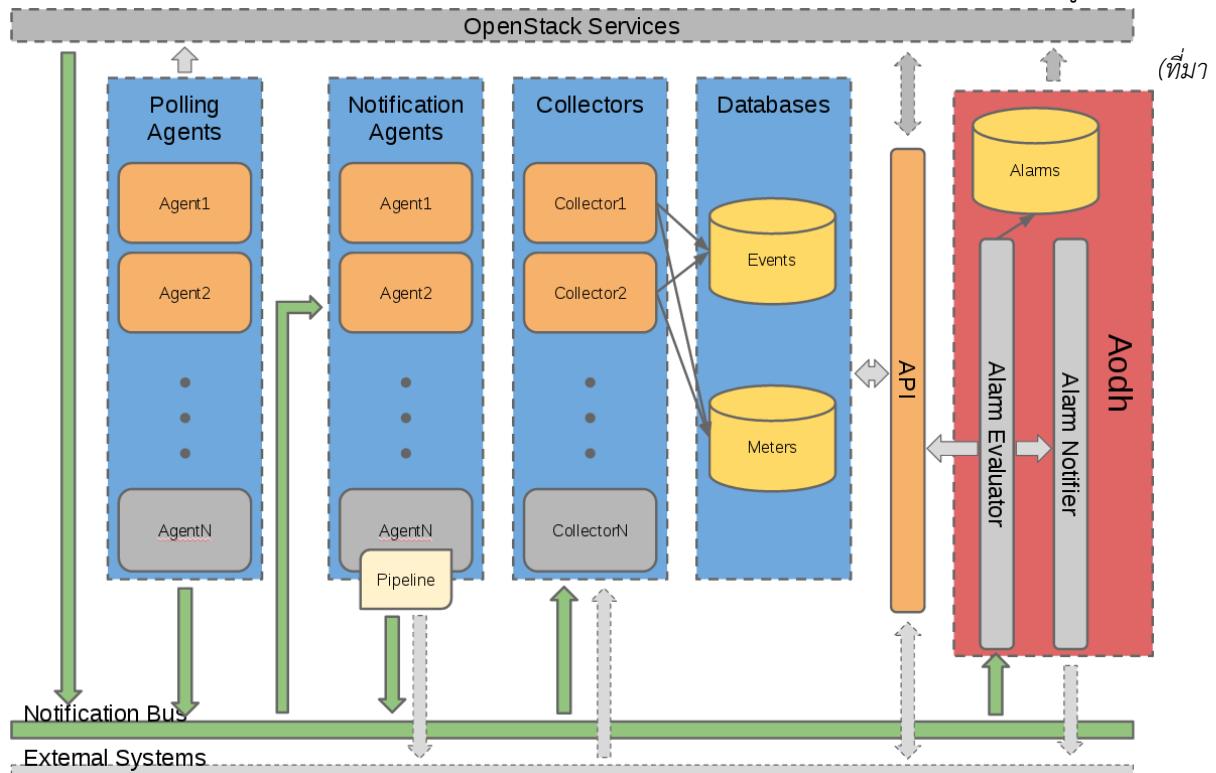
ขั้นตอนที่ 7 ขั้นตอนการประเมินผล โดยนำเอาข้อมูลที่เก็บไว้ในขั้นตอนที่ 5 มาเทียบกับนิยามที่สร้างในขั้นตอนที่ 6 ว่าสอดคล้องกับระดับของค่าที่กำหนดไว้หรือไม่ หากตรงตามเงื่อนไขของการสร้างสัญญาณเตือน ระบบก็ทำงานในขั้นตอนต่อไป

ขั้นตอนที่ 8 ดำเนินการตามขั้นตอนที่ขั้นตอนที่ได้กำหนดไว้ก่อนตามนิยามในขั้นตอนที่ 6 โดยการส่งสัญญาณไปยังโหนดอื่นๆ ต่อไปเพื่อให้รับทราบถึงสถานะดังกล่าว โดยที่่่ไป ก็จะส่งสัญญาณไปยัง services อื่นใน openstack ที่ได้กำหนดไว้

ขั้นตอนที่ 9 ทั้งนี้ก็สามารถนำเอาข้อมูลที่จัดกลุ่มสามารถนำเอาไปทำ virtualize เพื่อให้สามารถดูได้ง่ายขึ้น

สถาปัตยกรรมของ Ceilometer

ต่อมาเมื่อได้ทำความเข้าใจเกี่ยวกับขั้นตอนแต่ละขั้นตอนมาเบื้องกระบวนการทำงานทั้ง 9 ขั้นตอนแล้ว ก็จะมาทำความเข้าใจเกี่ยวกับขอบเขตตามสถาปัตยกรรมของ Ceilometer ซึ่งจะสามารถแสดง ให้เห็น ได้ตามรูปด้านล่าง



จะสามารถแบ่งออกได้เป็น 6 องค์ประกอบ

1 **Polling agent** คือ Daemon ทำหน้าที่เก็บมาเป็นข้อมูลติดจาก OpenStack component ยกตัวอย่าง เช่น compute agent จะถูกติดตั้งอยู่ใน compute node ก็จะทำการเก็บข้อมูลการทำงานของ instance และนำ มาสร้าง เป็น meter

2 **Notification Agents** คือ Daemon ที่ได้รับการออกแบบเพื่อที่จะทำหน้าที่ฟัง (Listen) สิ่งที่เรียกว่า Notifications ต่างๆ ที่ปรากฏขึ้นในส่วน message queue จากนั้นจะทำการแปลง (Convert) ให้กลายเป็น Events ก่อนที่จะส่งผ่านต่อไปยัง Pipeline Actions

3 **Pipeline** ทำหน้าที่แปลงข้อมูลและส่งข้อมูลลงไปใน message queue โดยจะทำงานอยู่ในขั้นตอนที่ 2 เนื่องจากข้อมูลที่ได้รับมาจาก Polling agent จะมีลักษณะสะสมเพิ่มไปเรื่อยๆ (cumulative) จึงจะต้องทำการแปลง ด้วย pipeline ให้มีลักษณะของข้อมูลเป็นลักษณะ gauge หรือมีค่าเป็นอัตราส่วนต่อเวลา

4 **Reiever** หรือ **Collector** ได้รับการออกแบบทำหน้าที่รวบรวมข้อมูลที่ถูกแปลงที่ส่งมาจาก pipeline สำหรับเพื่อเตรียมเก็บลงใน database ซึ่งจะอยู่ในขั้นตอนที่ 3

4 **Database** ทำหน้าที่เก็บข้อมูลที่ Collector ได้ทำการรวบรวม โดยจะอยู่ในขั้นตอนที่ 4

5 **API** ทำหน้าที่เป็นส่วนของ interface โดยรองรับเชื่อมต่อในการเข้าถึง ข้อมูล database สามารถที่ค้นหา (Query) และ แสดงผล (View) จาก component อื่นๆ ที่มีความต้องการใช้งานข้อมูลที่ Ceilometer ที่ได้ทำการเก็บ → แปลง → รวม → จัดเก็บ สำหรับการพัฒนาต่อยอดเพื่อที่จะระบบ billing ที่ทำการดึงข้อมูลจาก component นี้

6 **Alarm** ออกแบบให้ทำหน้าที่ในลักษณะของ Events Base หรือ Notify Base โดยดูจากนิยามที่ได้ระบุไว้ ใน Alarm Rule ในส่วนนี้จะมีหน้าที่สร้างและส่งสัญญาณเตือนโดยจะทำการดึงข้อมูลจาก database โดยผ่าน API เพื่อทำการจัดกลุ่มข้อมูลเพื่อประมวลผลสำหรับการการสร้างสัญญาณเตือนโดยงานอยู่ในขั้นตอน 5 6 7 และ 8

ขั้นตอนการติดตั้ง Ceilometer

1. การติดตั้ง Ceilometer service สามารถดำเนินการติดตั้งได้โดยผ่าน option ใน Answerfile

```
40 CONFIG_CEILOMETER_INSTALL=y  
1088 CONFIG_CEILOMETER_SECRET=b2f26c6e794248b7  
1092 CONFIG_CEILOMETER_KS_PW=7d1f50431e074343
```

2. หลังจากนั้นให้ทำการ run packstack อีกครั้ง

```
# packstack --answer-file answerfile.txt
```

3. ตรวจสอบสถานะการทำงานของ ceilometer

```
# openstack-status
```

ผลลัพธ์ที่ได้

```
== Ceilometer services ==  
openstack-ceilometer-api: active  
openstack-ceilometer-central: active  
openstack-ceilometer-compute: active  
openstack-ceilometer-collector: active  
openstack-ceilometer-alarm-notifier: active  
openstack-ceilometer-alarm-evaluator: active  
openstack-ceilometer-notification: active
```

การ restart services

```
export CEILO_SVCS='compute central collector api alarm-evaluator alarm-notifier'  
for svc in $CEILO_SVCS ; do sudo service openstack-ceilometer-$svc restart ; done  
for svc in $CEILO_SVCS ; do sudo grep ERROR /var/log/ceilometer/${svc}.log ; done
```

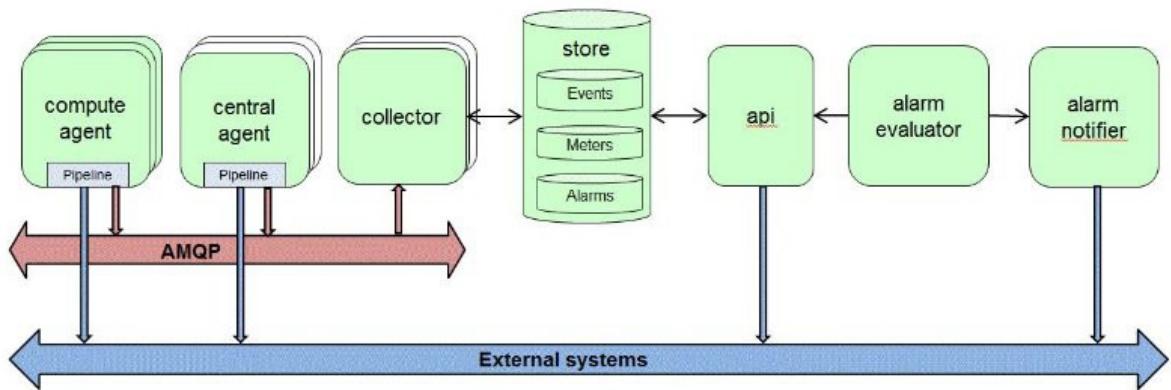
คำอธิบายเมื่อเปรียบเทียบ สถานะ กับ สถาปัตยกรรม

หลังจากที่เราได้ใช้คำสั่ง Openstack-status เพื่อทำการเรียกดู สถานะ ดังรูปด้านบน ทำให้เราสามารถเปรียบเทียบสถานะต่างๆ ของ ceilometer กับ สถาปัตยกรรมที่ได้กล่าวมาแล้วในข้างต้นได้ดังนี้

1. **openstack-ceilometer-api** เป็นส่วนของ API server
2. **openstack-ceilometer-central** เป็นส่วน Reciever ทำหน้าที่ร่วมกับ Collector แต่ component นี้จะเป็นการเก็บข้อมูลการใช้งาน API ของแต่ละ service ที่สื่อสารระหว่างกัน
3. **openstack-ceilometer-compute** เป็นส่วน Polling agent แต่สำหรับ service นี้เป็น compute agent
4. **openstack-ceilometer-collector** เป็นส่วนที่อยู่ใน Collector
5. **openstack-ceilometer-alarm-evaluator** เป็นส่วนที่อยู่ใน Alarm เป็นตัวที่ใช้ในการตรวจสอบค่า meter ที่ได้มากจาก database เพื่อตรวจสอบว่าถึงเกณฑ์สำหรับการสร้างสัญญาณเตือนเพื่อจะทำอะไรอย่าง
6. **openstack-ceilometer-alarm-notification** เมื่อข้อมูลสัญญาณเตือน ถ้าเก็บหรือlobออกไปจะมีการแจ้งข้อมูลออกมาจาก component นี้

กระบวนการทำงาน Workflow ภายในของ Ceilometer

เมื่อนำกระบวนการมาวิเคราะห์แล้ว จะสามารถเปลี่ยนแสดงความสัมพันธ์ในการทำงาน(WorkFlow) ได้ดังนี้



การใช้งาน Ceilometer

ผู้ใช้งาน ceilometer สามารถใช้งานผ่าน Dashboard โดยที่ผู้ใช้งานสามารถตรวจสอบคุณภาพตั้งค่าการแสดงผลผ่านทาง Horizon API ไปยัง Ceilometer และการใช้งานผ่านทาง Command Line เพื่อใช้คำสั่งโดยตรง โดยมีรายละเอียดการใช้งานแต่ละขั้นตอนดังนี้

1 การใช้งาน Ceilometer ด้วย OpenStack dashboard

1. ให้ login เข้าสู่ openstack ด้วย user admin



2. เมื่อ login ด้วย admin ให้เข้าไปที่ System → Resource Usage → Usage Report ตามลำดับ

Project	Service	Meter	Description	Day	Value (Avg)	Unit
admin	Nova	network.outgoing.packets.rate	Average rate per sec of outgoing packets on a VM network interface	2015-10-21	0.24294902118	packet/s
admin	Nova	disk.write.requests.rate	Average rate of write requests	2015-10-21	0.0135897503908	request/s
test	Glance	image	Image existence check	2015-10-21	1.0	image
Heat_Project	Glance	image	Image existence check	2015-10-21	1.0	image
admin	Glance	image	Image existence check	2015-10-21	1.0	image
admin	Nova	disk.read.bytes	Volume of reads	2015-10-21	182,475,776.0	B
admin	Nova	network.outgoing.bytes.rate	Average rate per sec of outgoing bytes on a VM network interface	2015-10-21	17.1287742651	B/s
test	Glance	image.size	Uploaded image size	2015-10-21	1,258,946,560.0	B
Heat_Project	Glance	image.size	Uploaded image size	2015-10-21	13,287,936.0	B

สำหรับหน้า dashboard ที่ใช้แสดงข้อมูลที่ได้จากการทำงานของ Ceilometer โดยดึงข้อมูลผ่าน API ซึ่งเป็นการแสดงข้อมูลอย่างง่ายโดยรายละเอียดของตารางเป็นดังนี้

1. Project คือ กลุ่มของผู้ใช้งานและ resource ในระบบ cloud โดยการกำหนด quota การใช้การใช้งาน resource จะถูกกำหนดจาก Project ยกตัวอย่างเช่น สามารถใช้ vcpu ได้ 4 core ใช้ RAM ได้ 4 GB เป็นต้น
2. Service สำหรับ Service ใน OpenStack อาจจะกล่าวได้ว่าเป็น Component ของ OpenStack โดยใน尼ยามของ OpenStack นั้น 1 component เท่ากับ 1 service ยกตัวอย่างเช่น Nova คือ Compute service Neutron คือ Network service เป็นต้น
3. Meter คือตัววัดที่ระบุได้ถึงประเภทของ data ที่จะทำการวัด และจัดเก็บจากค่าต่างๆ ในระบบ Cloud Openstack นั้น ข้อมูลทั้งหมด จะถูกรวบรวมไว้เป็นระบบ เพื่อใช้เป็นการควบคุม resource จนถึงพัฒนาเป็นระบบ billing เพื่อการให้การบริการได้สำหรับการใช้งาน Datacenter โดยที่ได้กำหนดไว้ใน specification เมื่อมีการออกแบบ

(<http://docs.openstack.org/admin-guide-cloud/telemetry-measurements.html>)

รายละเอียดของค่า meter

1. Day หมายถึงวันที่ได้ทำการเก็บข้อมูล
2. Value (Avg) สำหรับข้อมูลที่ได้ทำการเก็บลงใน database นั้น จะมี Time Stamp บ่งบอกถึงเวลาที่ได้ทำการเก็บข้อมูล แต่สำหรับหน่วยบางหน่วยที่มีความละเอียดมากนั้นอาจจะต้องใช้ค่าเฉลี่ยในการแสดงผล
3. Unit คือหน่วยที่ใช้สำหรับการเก็บข้อมูล
4. สามารถเลือกดูข้อมูลตามช่วงเวลาที่ต้องการได้โดยดังนี้

The screenshot shows the 'Resources Usage Overview' dashboard. On the left, there's a sidebar with dropdown menus for 'Project', 'Admin', and 'System'. Under 'System', 'Resource Usage' is selected, which further branches into 'Hypervisors', 'Host Aggregates', and 'Instances'. The main area has tabs for 'Usage Report' and 'Stats', with 'Usage Report' currently active. A red circle labeled '1' is over a button labeled 'Modify Usage Report Parameters'. Below it is a table with columns: Project, Service, Meter, Description, Day, Value (Avg), and Unit. One row is visible: 'services' under 'Service', 'Swift_meters' under 'Meter', 'storage.objects.containers' under 'Description', 'Number of containers' under 'Value (Avg)', '2015-10-15' under 'Day', and '0.0' under 'Unit'.

การเลือกช่วงเวลาให้ทำการกด drop-down ใน option ของ period

Modify Usage Report Parameters

This is a modal dialog titled 'Modify Usage Report Parameters'. It has a dropdown menu for 'Period *' with options like 'Last day', 'Last week', 'Month to date', etc., with 'Last day' selected. A red circle labeled '2' is over this dropdown. To the right, there's a 'Description:' section with a placeholder 'Select a pre-defined period or specify date.' and a 'View Usage Report' button (red circle labeled '3'). At the bottom, there are 'Cancel' and 'View Usage Report' buttons.

เมื่อกดปุ่ม View Usage Report

The screenshot shows the 'Resources Usage Overview' dashboard again. The sidebar and top navigation are identical to the first screenshot. The main table now shows different data: 'Heat_Project' under 'Project', 'Nova' under 'Service', 'network.outgoing.packets.rate' under 'Meter', 'Average rate per sec of outgoing packets on a VM network interface' under 'Description', '2015-10-15' under 'Day', '1.65722222222' under 'Value (Avg)', and 'packet/s' under 'Unit'.

4. ตัวอย่างการดูข้อมูลเป็นกราฟผ่านทาง dashboard

The screenshot shows a user interface for monitoring resource usage. On the left, there's a sidebar with navigation links: Project, Admin, and System (with Overview, Resource Usage, Hypervisors, Host Aggregates, Instances). The main area has tabs for Usage Report and Stats (the latter is highlighted with a red circle containing the number 1). Below the tabs are buttons for 'Modify Usage Report Parameters' and 'Download CSV Summary'. A table displays a single row of data:

Project	Service	Meter	Description	Day	Value (Avg)	Unit
Heat_Project	Nova	network.outgoing.packets.rate	Average rate per sec of outgoing packets on a VM network interface	2015-10-15	1.65722222222	packet/s

ตั้งค่า parameter สำหรับการเลือกแสดงผลข้อมูลออกมาเป็นกราฟ

The screenshot shows the same interface as above, but the 'Stats' tab is active. The configuration parameters are highlighted with red circles numbered 1 through 6:

- Metric:** cpu_util (highlighted with red circle 1)
- Group by:** Project (highlighted with red circle 2)
- Value:** Avg. (highlighted with red circle 3)
- Period:** Last day (highlighted with red circle 4)
- From:** (highlighted with red circle 5)
- To:** (highlighted with red circle 6)

จากรูปข้างต้นเป็นการปรับแต่งค่าเพื่อการใช้งาน resource โดยมีคำอธิบายแต่ละ parameter ดังต่อไปนี้
1. Metric เป็นการเลือกค่าตัววัด เพื่อที่จะแสดงผลออกมาเป็นกราฟ จากตัวอย่างจะเป็นการเลือกค่า cpu_util

Resources Usage Overview

Usage Report Stats

Metric: cpu_util ▾

Group by:

Value: Compute (Nova)

Period:

From:

To:

Statistics of all resources

The screenshot shows a user interface for monitoring cloud resource usage. At the top, there are tabs for 'Usage Report' and 'Stats'. Below them, there are several input fields: 'Metric' set to 'cpu_util', 'Group by' dropdown, 'Value' dropdown, 'Period' dropdown, 'From' and 'To' date range inputs, and a large button labeled 'Statistics of all resources'. A dropdown menu for 'Value' is open, listing various metrics under 'Compute (Nova)' such as instance, memory.usage, cpu, and disk.read.requests. The 'cpu_util' option is highlighted with a red background.

2. Group by เป็นเลือกกลุ่มของ resource ที่ต้องการแสดงผล โดยสามารถเลือกได้เฉพาะ project เท่านั้น

Resources Usage Overview

Usage Report Stats

Metric: cpu_util ▾

Group by: Project ▾

Value: Project

Period: Last day ▾

From:

To:

This screenshot shows the same interface as the previous one, but with a different 'Group by' selection. The 'Group by' dropdown is now set to 'Project', and the 'Value' dropdown also shows 'Project' as the selected item. The other fields like Metric, Period, and Date range remain the same.

3. Value การเลือกรูปแบบของผลลัพธ์ที่ต้องการแสดงผลสำหรับตัวอย่างนี้จะเลือกเป็นค่า Average หรือค่าเฉลี่ย

Resources Usage Overview

Usage Report Stats

Metric: cpu_util

Group by: Project

Value: Avg.

Period:

- Avg.
- Min.
- Max.
- Sum.

From:

To:

4. Period เป็นการเลือกช่วงเวลาสำหรับตัวอย่างนี้จะเลือกเป็นในช่วงระยะเวลา 1 วัน

Resources Usage Overview

Usage Report Stats

Metric: cpu_util

Group by: Project

Value: Avg.

Period:

- Last day
- Last week
- Month to date
- Last 15 days
- Last 30 days
- Last year
- Other

From:

To:

5. From เมื่อเลือก other ใน Period จะต้องเลือกช่วงเวลาจาก parameter นี้ สำหรับตรงนี้จะเป็นเวลาเริ่ม

Resources Usage Overview

Usage Report Stats

« October 2015 »

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Metric:

Group by:

Value:

Period:

From:

To:

6. To เมื่อเลือก other ใน Period จะต้องเลือกช่วงเวลาจาก parameter นี้ สำหรับตรงนี้จะเป็นเวลาสิ้นสุด

Resources Usage Overview

Usage Report Stats

Metric:

Group by:

Value:

Period:

From:

To:

« October 2015 »

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

ตัวอย่างการเลือกกราฟ

Resources Usage Overview

Usage Report Stats

Metric:

Group by:

Value:

Period:

From:

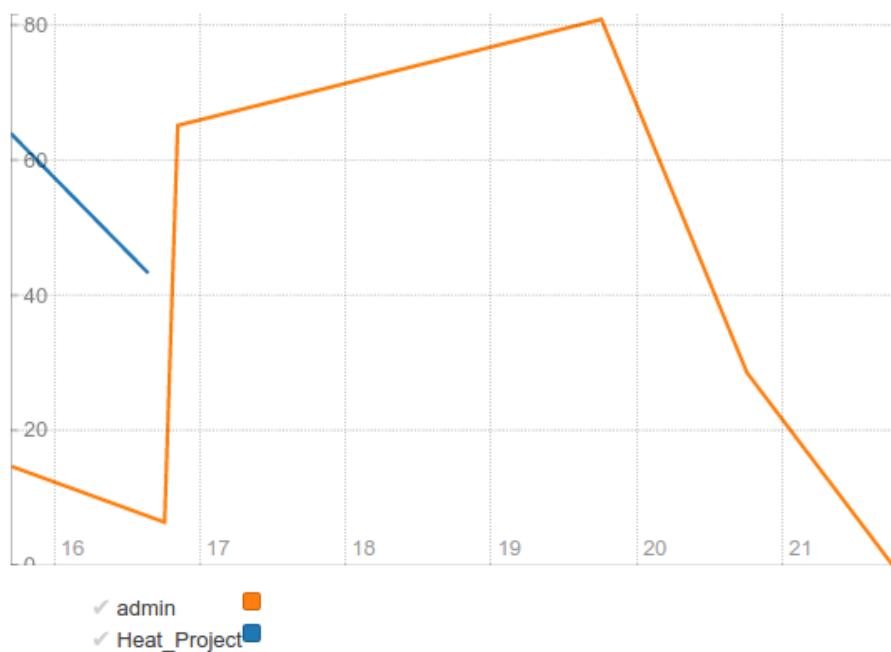
To:

เมื่อค่าที่ต้องการเขียนกราฟเป็นตาราง

Heat_Project	Nova	cpu_util	Average CPU utilization	2015-10-15	63.9522222222	%
Heat_Project	Nova	cpu_util	Average CPU utilization	2015-10-15	63.9522222222	%
admin	Nova	cpu_util	Average CPU utilization	2015-10-15	14.6423092943	%
admin	Nova	cpu_util	Average CPU utilization	2015-10-15	14.6423092943	%
admin	Nova	cpu_util	Average CPU utilization	2015-10-15	14.6423092943	%
admin	Nova	cpu_util	Average CPU utilization	2015-10-15	14.6423092943	%
admin	Nova	cpu_util	Average CPU utilization	2015-10-15	14.6423092943	%
admin	Nova	cpu_util	Average CPU utilization	2015-10-15	14.6423092943	%

เมื่อนำค่าตัวอย่างในตารางดังกล่าวมา plot เป็นกราฟ

Statistics of all resources



2 การใช้งาน Ceilometer ด้วย OpenStack CLI

1. เข้าไปที่ directory ที่ได้เก็บ RC file ซึ่งเป็น file ที่ใช้สำหรับการตั้งค่า Environment สำหรับการ Login

```
# cd /root
```

2. ทำการ set ค่าเพื่อทำการ login

```
# source ./keystonerc_admin
```

3. ทดสอบการใช้คำสั่งแสดงค่าใน คอลัมน์ name,type,unit ทั้งหมด

```
# ceilometer meter-list | cut -d'|' -f 2,3 | sort -n | uniq
```

NAME	Type	Unit
cpu	cumulative	ns
cpu_util	gauge	%
disk.allocation	gauge	B
disk.capacity	gauge	B
disk.device.allocation	gauge	B
disk.device.capacity	gauge	B
disk.device.read.bytes	cumulative	B
disk.device.read.bytes.rate	gauge	B/s
disk.device.read.requests	cumulative	request
disk.device.read.requests.rate	gauge	request/s
disk.device.usage	gauge	B
disk.device.write.bytes	cumulative	B
disk.device.write.bytes.rate	gauge	B/s
disk.device.write.requests	cumulative	request
disk.device.write.requests.rate	gauge	request/s
disk.read.bytes	cumulative	B
disk.read.bytes.rate	gauge	B/s
disk.read.requests	cumulative	request
disk.read.requests.rate	gauge	request/s
disk.usage	gauge	B
disk.write.bytes	cumulative	B
disk.write.bytes.rate	gauge	B/s
disk.write.requests	cumulative	request
disk.write.requests.rate	gauge	request/s
image.delete	delta	image
image.download	delta	B
image	gauge	image
image.serve	delta	B
image.size	gauge	B
image.update	delta	image
image.upload	delta	image
instance	gauge	instance
instance:m1.large	gauge	instance
instance:m1.medium	gauge	instance
instance:m1.small	gauge	instance
instance:m1.tiny	gauge	instance
instance:m1.xlarge	gauge	instance
instance:m2.small	gauge	instance
ip.floating.create	delta	ip
ip.floating	gauge	ip
ip.floating.update	delta	ip
memory.resident	gauge	MB
memory.usage	gauge	MB
Name	Type	Unit
network.create	delta	network
network	gauge	network
network.incoming.bytes	cumulative	B

network.incoming.bytes.rate	gauge	B/s
network.incoming.packets	cumulative	packet
network.incoming.packets.rate	gauge	packet/s
network.outgoing.bytes	cumulative	B
network.outgoing.bytes.rate	gauge	B/s
network.outgoing.packets	cumulative	packet
network.outgoing.packets.rate	gauge	packet/s
network.services.lb.active.connections	gauge	connection
network.services.lb.health_monitor.create	delta	health_monitor
network.services.lb.health_monitor	gauge	health_monitor
network.services.lb.incoming.bytes	cumulative	B
network.services.lb.member.create	delta	member
network.services.lb.member	gauge	member
network.services.lb.outgoing.bytes	cumulative	B
network.services.lb.pool.create	delta	pool
network.services.lb.pool	gauge	pool
network.services.lb.total.connections	cumulative	connection
network.services.lb.vip.create	delta	vip
network.services.lb.vip	gauge	vip
network.update	delta	network
port.create	delta	port
port	gauge	port
port.update	delta	port
router.create	delta	router
router	gauge	router
router.update	delta	router
storage.objects.containers	gauge	container
storage.objects	gauge	object
storage.objects.size	gauge	B
subnet.create	delta	subnet
subnet	gauge	subnet
subnet.update	delta	subnet

คำอธิบายประกอบตาราง

1. Name เป็นชื่อของ meter ที่ใช้แสดงแทน ID เพื่อให้สามารถอ่านและเข้าใจได้ง่าย เช่น cpu disk.allocation เป็นต้น ทำหน้าที่เป็นตัวบ่งบอกแหล่งกำเนิดของข้อมูลว่าส่งมาที่ใด
2. Type เป็นชนิดของ meter โดยแบ่งได้ออกเป็น 3 ประเภท ได้แก่
 - A) Cumulative เป็นชนิดของ meter ที่มีการเก็บค่าแบบการ “ค่าสะสม Total” ทำให้ค่าที่ปรากฏจะเพิ่มขึ้นตลอดเวลา เป็นการแสดงผลรวมทั้งหมดของการทำงาน ยกตัวอย่างเช่น จำนวน connection ที่มาเข้ามายังตัวบอร์ด load balance

network.services.lb.total.connections	cumulative connection	

- B) Gauge เป็นชนิดของ meter ที่มีการเก็บ “ค่าปัจจุบัน (Current Value)” ในขณะที่มีการอ่าน

subnet	gauge	subnet	10af489c-29d8-4e5e-978a-a930a6af6d23
subnet	gauge	subnet	1200f3b9-6c92-4905-8665-1b875d78407d
subnet	gauge	subnet	12107aa9-5ed8-495f-b8f7-e308f903dc3f
subnet	gauge	subnet	23b4ed15-9603-4281-b66a-8c4dea87d339
subnet	gauge	subnet	6bc0eaeb-ae69-46b5-9711-88915e1441c4
subnet	gauge	subnet	7d12c10d-d100-49d9-a15f-df6e9527a93d
subnet	gauge	subnet	813aa79c-6e08-4187-8d77-cded27279d93
subnet	gauge	subnet	86ad9179-d447-4a7a-ab2c-2f22c51ad578
subnet	gauge	subnet	9ebd69b2-cb57-420a-898d-3c11205d5647
subnet	gauge	subnet	b9db36c7-fbea-43b2-a394-002a5a9daeca
subnet	gauge	subnet	c14c8b8e-31c4-44bf-91a4-241b7837bb70
subnet	gauge	subnet	c2f655d0-988b-4912-9ac1-54fac7fd1941
subnet	gauge	subnet	c7102f22-2bda-420d-978f-573c878c9695
subnet	gauge	subnet	d3757d7a-4de1-460d-b614-77243605705f
subnet	gauge	subnet	df22a71a-31d7-47c4-bda3-024c3b028609
subnet	gauge	subnet	f2f1a1ef-6ca2-4ad9-af56-33f3d6e5de40

C) Delta เป็นชนิด meter ที่เป็นข้อมูลที่มีการเปลี่ยนแปลง ณ ขณะหนึ่ง ยกตัวอย่างเช่น เมื่อมีการสร้าง instance ข้อมูลของ port มีการเปลี่ยนแปลง

port.update	delta	port	a0584cb7-5899-4a39-9752-e9989a324af
port.update	delta	port	a854cd2a-7e1d-4046-91c8-1a8727dbfead
port.update	delta	port	a96e1e85-aac3-41f7-86bd-d19c5ab8cc33
port.update	delta	port	b3999456-01bc-471b-8a8d-866b8bad4963
port.update	delta	port	bd3204e5-1da0-455c-a79a-07da7e393b5b
port.update	delta	port	bf5ab0fd-c76e-4920-ac2e-0a50488acc22
port.update	delta	port	bf5df8a7-7a78-4b00-a399-ad1ca9c40f6a
port.update	delta	port	c4c7e588-a38e-44e4-a0ef-a4923774d7ad

3. Unit เป็นหน่วยของ meter

4. Resource ID เป็นหมายเลข ID ของ resource

5. User ID เป็นหมายเลข ID ของ user ที่ได้เข้ามาใช้ resource

6. Project ID เป็นหมายเลขของ project ที่ได้เข้ามาใช้ resource

หมายเหตุ

สำหรับข้อมูล เพิ่มเติมของการวัด Measurement สามารถติดตามเพิ่มเติมได้ที่

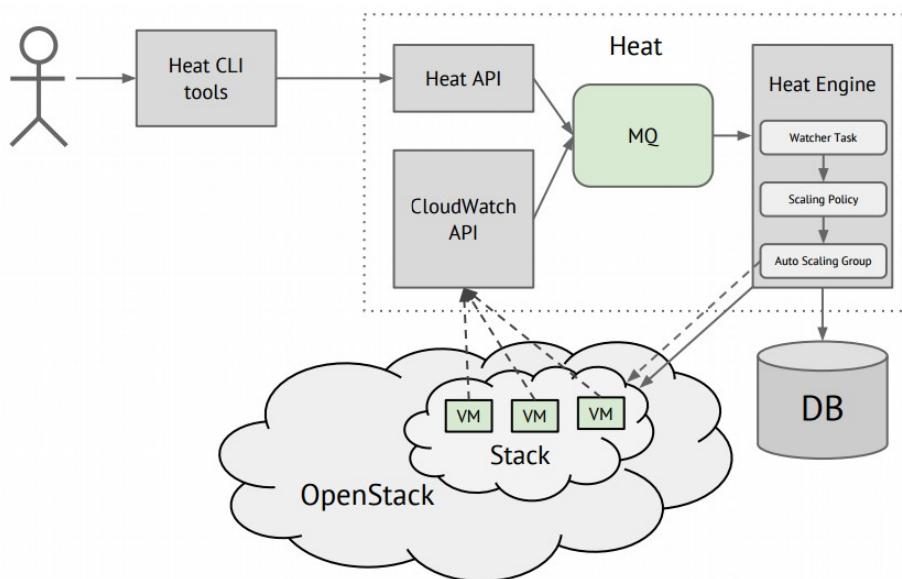
<http://docs.openstack.org/developer/ceilometer/measurements.html>

กระบวนการ Cloud Orchestration ใน Openstack

Orchestration และ Automation ที่อยู่ใน OpenStack เป็นสิ่งที่เป็นหัวใจของระบบ Cloud Service ใน OpenStack สำหรับ project ที่ทำหน้าที่นี้ใช้ codename ที่ชื่อว่า Heat โดยที่ Heat จะเป็น Service ที่อาศัยข้อมูลที่ส่งมาโดย Ceilometer มาทำงานต่อ

สิ่งที่ Heat ให้บริการคือการอ่าน process template ที่เป็น YAML file สำหรับใช้สร้าง cluster ตาม policy ที่ได้กำหนดไว้ซึ่งจุดเด่นที่สำคัญคือ การทำ autoscaling คือ การที่Heat จะเป็นผู้ที่ขยาย หรือ ลดขนาดจำนวนของ Instance ตามโหลดการทำงานของ ระบบ ซึ่งจะได้อธิบายและทำการทดลองเพื่อการศึกษาต่อไป

สถาปัตยกรรม Openstack Heat



องค์ประกอบภายใน Heat Component ประกอบด้วย

1. **heat-api** เป็น component ที่ทำหน้าที่เป็น REST API process ไว้ใช้สำหรับการติดต่อสำหรับ service อื่นๆ
2. **heat-api-cfn** เป็น component ที่ทำหน้าที่รองรับ AWS API โดยจะส่งไปยัง component อื่นๆ ใน Heat โดยผ่าน RPC (Remote Procedure Call)
3. **heat-engine** เป็น engine หลักสำหรับการประมวลผล HOT template
4. **heat-cli** เป็นเครื่องมือ client สำหรับการเข้าใช้งาน Heat

ขั้นตอนติดตั้งเพื่อการใช้งาน HEAT Template (Installation)

การติดตั้ง Heat Template สามารถดำเนินการติดตั้งได้โดยผ่าน option ใน Answerfile

```
42 # Specify 'y' to install OpenStack Orchestration (heat). ['y', 'n']
43 CONFIG_HEAT_INSTALL=y
1024:CONFIG_HEAT_CLOUDWATCH_INSTALL=y
1028:CONFIG_HEAT_CFN_INSTALL=y
```

หลังจากนั้นให้ทำการ run packstack อีกครั้ง

```
packstack --answer-file answerfile.txt
```

หลังจาก ได้ดำเนินการติดตั้งให้ login เข้าสู่ openstack ด้วย user admin

การแก้ไขปัญหาเพื่อให้สามารถใช้ Heat ทำงาน

หลังจากได้เตรียม Resource ต่างๆ เรียบร้อยแล้ว ยังมีขั้นตอนการแก้ปัญหา เนื่องจากว่า เนื่องด้วยการติดตั้ง ใน version ที่ซื้อว่า kilo สำหรับการใช้งานจะมีปัญหาดังต่อไปนี้ ดังต่อไปนี้

1. ปัญหานี้เองมาจาก authentication failed ของ head_admin

การตรวจสอบปัญหาได้ โดยการ upload heat template จากหน้า web หน้าเวปจะแสดงสถานะ ว่า create_failed ยืนยันด้วยการตรวจสอบ heat-engine.log

```
# tail -f /var/log/heat/heat-engine.log
```

log ที่เกิดขึ้นจะแสดงผลดังนี้

```
INFO heat.engine.service [req-99571f4c-47d7-42f9-a1d3-07736538d416 - admin]
validate_template
INFO heat.engine.service [req-2ef8767b-2de2-4661-8e8c-1913d990efff admin admin] Creating
stack stack1
INFO heat.engine.resource [req-2ef8767b-2de2-4661-8e8c-1913d990efff admin admin]
Validating Server "my_instance"
18496 ERROR heat.common.keystoneclient [-] Domain admin client authentication failed
18496 INFO heat.engine.stack [-] Stack CREATE FAILED (stack1): Authorization failed.
18496 INFO heat.engine.service [-] Stack create failed, status FAILED
```

เมื่อการทดสอบ heat การทำโดยใช้ heat template พบร่วมกับ

```
# heat stack-list
```

จะได้ผลของคำสั่งดังนี้

id	stack_name	stack_status	creation_time
99f930de-2510-465f-9974-b31ae9fc6fe0	stack1	CREATE_FAILED	2015-10-10T07:21:28Z

ทดสอบด้วยคำสั่งอีกชุดจะพบว่า

```
# heat event-list stack1
```

ผลลัพธ์ที่ได้

resource_name	id	resource_status	resource_status_reason
resource_status	event_time		
stack1	826c7e79-ed31-4c76-b0c0-16573867feb5	CREATE_FAILED	Authorization failed.
	2015-10-10T07:21:28Z		

เมื่อใช้คำสั่ง แสดงผล ก็จะแสดงข้อมูล stack_status ดูในค่าของ stack_status_reason

```
# heat stack-show stack1
```

ผลลัพธ์จะแสดงผลลักษณะดังนี้

```
[root@rdo heat(keystone_admin)]# heat stack-show stack1
+-----+
| Property | Value
+-----+
| capabilities | []
| creation_time | 2015-10-10T04:49:37Z
| description | Simple template to deploy a single compute instance
| disable_rollback | True
| id | 5395f673-c9e1-4bb3-8e70-55aca88d2c9d
| links | http://192.168.56.10:8004/v1/3de2de3356f14471bcc00a6042b803f0/stacks/stack1/5395f673-c9e1-4bb3-8e70-55aca88d2c9d (self)
| notification_topics | []
| outputs | []
| parameters | {
|   |   {
|   |     "output_value": null,
|   |     "description": "IP address of the instance",
|   |     "output_key": "instance_ip"
|   |   }
|   | }
|   | {
|   |   "OS::project_id": "3de2de3356f14471bcc00a6042b803f0",
|   |   "OS::stack_id": "5395f673-c9e1-4bb3-8e70-55aca88d2c9d",
|   |   "OS::stack_name": "stack1",
|   |   "image": "centos7",
|   |   "key": "itb_cloudkey2",
|   |   "private_network": "private_network",
|   |   "flavor": "m1.small"
|   | }
|   | }
| parent | None
| stack_name | stack1
| stack_owner | admin
| stack_status | CREATE_FAILED
| stack_status_reason | Authorization failed.
| stack_user_project_id | None
| template_description | Simple template to deploy a single compute instance
| timeout_mins | 60
| updated_time | None
+-----+
```

วิธีการแก้ปัญหา Authentication Fail ใน Openstack มีลำดับขั้นตอนดังต่อไปนี้

(อ้างอิงตาม <http://docs.openstack.org/admin-guide-cloud/orchestration-stack-domain-users.html>)

การแก้ปัญหางานสร้าง user ใหม่เขื่อว่า “heat_domain_admin” ภายใต้ heat domain และมีสิทธิเป็น admin และ แก้ไขรายละเอียดของ user ใน file config ของ heat “/etc/heat/heat.conf” แต่หากปัญหานี้ได้รับ การแก้ไขแล้วก็จะไม่จำเป็นต้องดำเนินการทำในขั้นตอนนี้

เมื่อติดตั้ง heat ด้วย rdo จะได้ user ชื่อ heat

1. ตรวจสอบ user ในระบบ

```
# openstack user list
+-----+-----+
| ID | Name |
+-----+-----+
| 04615fe95d4c4bdc8e23953ea31085be | glance |
| 1fef3c58e3f841c583e62be4ebae0344 | neutron |
| 47b93fce1a3e4467a647fb80fffc2a39b | nova |
| 6a4081fa36e64d88a082eb8dbc0ff13f | heat |
| 7634728a8b14471eb4c0f61917fef20f | demo |
| 7d4d3017fe4a458fadbf18d4ba95211 | cinder |
| ad8d08931d38433fbed1fc3cc4aa61d | ceilometer |
| debb32e1b02944b0bfbedb6d87b0b465 | swift |
| ee2cbcdf10424a2c8e39ed329d3b1fc2 | admin |
+-----+-----+
```

2. ตรวจสอบ role ในระบบ

```
# openstack role list
+-----+-----+
| ID | Name |
+-----+-----+
| 19f7bb0836084fd8a65096351514e85b | heat_stack_owner |
| 2ab3c3b66916406d8a376286200f479d | heat_stack_user |
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_ |
| b3bfa9da36fd4356ad6d381686778791 | SwiftOperator |
| c79acbf078034a8190fe14de7c3aefec | ResellerAdmin |
| d1a376238f54403bad1d673d5cee6ba0 | admin |
+-----+-----+
```

3. ทำการ login ด้วย admin ที่มีสิทธิสูงสุดในระบบ โดยใช้ admin_token ที่อยู่ที่ file /etc/keystone/keystone.conf ด้วยคำสั่งด้านล่างนี้

```
# grep admin_token /etc/keystone/keystone.conf
admin_token = 9ebef668232e43148ef2b8985fb90c22
```

เมื่อได้ค่าแล้ว ให้ทำการ export ค่า admin_token ไปอยู่ใน shell environment (เทียบเท่ากับการ login) ให้รอมัตระวังไม่ให้ซองว่าระหว่างระหว่าง เครื่องหมาย =

```
# export OS_TOKEN=9ebef668232e43148ef2b8985fb90c22
```

4. ตรวจสอบ heat domain ด้วยการใช้คำสั่ง openstack domain list ด้วย option --os-token , --os-url (คือ url ของ keystone version 3) และ --os-identity-api=3

```
# openstack --os-token $OS_TOKEN --os-url=http://192.168.1.2:5000/v3 --os-identity-api-version=3 domain list
+-----+-----+-----+-----+
| ID | Name | Enabled | Description |
+-----+-----+-----+-----+
| cfbed856ad1947f090117f4927262909 | heat | True | Contains users and |
| default | Default | True | Owns users and |
+-----+-----+-----+-----+
```

5. สร้าง admin user ขึ้นว่า heat_domain_admin ภายใต้ heat domain แทน admin user เดิม เนื่องจากในเวอร์ชันนี้ มีปัญหาของการ authentication

```
# openstack --os-token $OS_TOKEN --os-url=http://192.168.1.2:5000/v3 --os-identity-api-version=3 user create --password redhat --domain cfbed856ad1947f090117f4927262909
heat_domain_admin --description 'Manages users and projects created by heat'
+-----+-----+
| Field | Value |
+-----+-----+
| description | Manages users and projects created by heat |
| domain_id | cfbed856ad1947f090117f4927262909 |
| enabled | True |
| id | 0d56a00f520d4220abe99e9eb433a63e |
| name | heat_domain_admin |
+-----+-----+
```

ทำการตรวจสอบ

```
# openstack --os-token $OS_TOKEN --os-url=http://192.168.1.2:5000/v3 --os-identity-api-version=3 user list
+-----+-----+
| ID | Name |
+-----+-----+
| ... | ... |
| 0d56a00f520d4220abe99e9eb433a63e | heat_domain_admin |
+-----+-----+
```

ให้สิทธิ role แก่ user heat_domain_admin เป็น admin ของ domain

```
# openstack --os-token $OS_TOKEN --os-url=http://192.168.1.2:5000/v3 --os-identity-api-version=3 role add --user heat_domain_admin --domain cfbed856ad1947f090117f4927262909 admin
```

6. ทำการแก้ไข configuration ของ heat (/etc/heat/heat.conf)

ให้ทำการแก้ไขค่าดังต่อไปนี้

```
stack_user_domain_id= cfbed856ad1947f090117f4927262909
stack_domain_admin = heat_domain_admin
stack_domain_admin_password = redhat
```

ทำการ restart services

```
# openstack-service restart
```

7. ทำการกำหนด role ที่มีสิทธิ์สามารถในการใช้งาน heat ได้

user ที่จะมีสิทธิ์ใช้งาน heat นั้นจะต้องมี role ชื่อว่า heat_stack_owner ดังนั้น ให้กลับไปที่ horizon login ไปยัง user admin และเพิ่มเติม role ของ demo user ให้เป็น heat_stack_owner , heat_stack_user เพื่อให้สามารถ จะ upload heat template ได้

Project Details: demo

Project Overview

Information

Project Name	demo
Project ID	f8b0d8d0cdc342b388c57377547fc8ea
Enabled	True
Description	demo

เลือก role ให้กับ user demo (ได้สร้างไว้ก่อนแล้ว) เป็น heat_stack_owner (ไม่ต้องเลือก heat_stack_user)

Edit Project

Project Information * Project Members Quota *

All Users

nova	+
ceilometer	+
trove	+
neutron	+
heat	+
swift	+

Project Members

demo	heat_stack_owner, _member_	-
------	----------------------------	---

heat_stack_owner
heat_stack_user
member
SwiftOperator
ResellerAdmin
admin

และอย่างลีมกดปุ่ม save ที่อยู่ด้านล่าง เพื่อต้องการบันทึกการเปลี่ยนแปลง และ หากต้องการใช้งานผ่านทาง command line สามารถใช้คำสั่งดังล่าง

```
# keystone user-role-add --user demo --tenant demo --role heat_stack_owner
# keystone user-role-list --tenant demo --user demo
```

ดังรูป

	id	name	user_id	tenant_id
9fe2ff9ee4384b1894a90878d3e92bab	_member	3661912e4e7247718832775c02233099	df5ded99a7a74b6896647d11126dd7bd	
19f7bb0836084fd8a65096351514e85b	heat_stack_owner	3661912e4e7247718832775c02233099	df5ded99a7a74b6896647d11126dd7bd	

หลังจากนั้นให้ login ใหม่อีกครั้งด้วย user demo

2. ปัญหาการไม่แสดงผล icon

ปัญหาอีกประการหนึ่งที่เกิดขึ้นจะเกิดไม่สามารถแสดงผล icon ออกทางหน้าจอได้ จะแสดงเป็น icon เครื่องหมายคำถาม ดังรูปด้านล่าง

วิธีการแก้ปัญหา

ให้ทำการแก้ไขด้วยการ ssh ไปยัง server เพื่อแก้ไขค่าของ openstack-dashboard.conf

1. เข้าไปที่ directory ที่เป็น configuration ของ httpd service

```
# cd /etc/httpd/conf.d/
```

ทำการแก้ไข file ดังนี้ ด้วยการเพิ่ม บรรทัด ดังนี้

```
# vim openstack-dashboard.conf
```

```
Alias /static/dashboard /usr/share/openstack-dashboard/static/dashboard
```

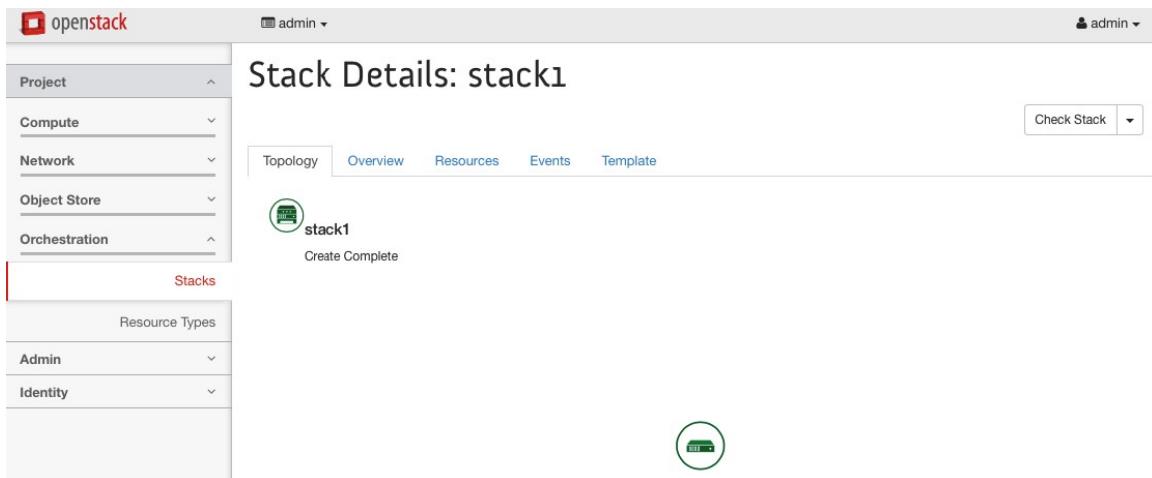
ดังรูป

```
#  
# This file has been cleaned by Puppet.  
#  
# OpenStack Horizon configuration has been moved to:  
# - 15-horizon_vhost.conf  
# - 15-horizon_ssl_vhost.conf  
#  
Alias /static/dashboard /usr/share/openstack-dashboard/static/dashboard
```

2. ทำการ restart service

```
systemctl restart httpd
```

reload อีกครั้ง จะเป็นว่า มี หน้าเวป จะแสดง icon ตามต้องการ



การทดสอบกระทำงานระหว่าง Heat Template และ Heat Engine

แนวทางการทดสอบ

วัตถุประสงค์ในการทดสอบเพื่อต้องการให้ผู้ใช้งาน private สามารถมีความเข้าใจการใช้งาน Heat ได้อย่างชัดเจนจึงได้แบ่งการทดสอบออกเป็น 3 การทดสอบ โดยมีลำดับการทดสอบจากขั้นพื้นฐาน ที่ใช้งาน Heat Template อย่างง่ายจนถึงระดับที่สามารถใช้งานได้จริงในองค์กร มีรายละเอียดดังนี้

ชื่อการทดสอบ	วัตถุประสงค์การทดสอบ
การทดสอบ 1: พื้นฐานการสร้าง Instance ด้วย HOT (แบบ single instance)	ต้องการทดสอบการสร้าง instance จำนวน 1 instance ด้วย Heat template เช้าใจการกำหนดค่าของตัวแปรที่จำเป็นในการสร้าง instance ได้แก่ image, network, subnet และ security group
การทดสอบ 2: สร้าง Database Server ด้วย การ run script ภายใน Heat Template	ต้องการทดสอบการ run script ภายใน instance ที่สร้างขึ้นด้วยโปรแกรม cloud-init โดยจะทำงานตาม script ที่อยู่ค่าของ template ที่อยู่ภายในค่าของ user_data: และ params: การทำงานจะเป็นทำงานเหมือนกับการติดตั้งปกติทั่วไปเมื่อมีการติดตั้งแบบ manual และ script ที่นำมาทดสอบการติดตั้งคือการการติดตั้งฐานข้อมูล
การทดสอบ 3: สร้าง instance Autoscale multinode (แบบไม่มี load balance)	ต้องการทดสอบการสร้าง instance ด้วย Heat template โดยมีการใช้งานฟังก์ชัน Autoscale (OS::Heat::AutoScalingGroup) ในการทดสอบผู้ใช้งานจะเข้าใจการทำงานร่วมกันระหว่าง Heat สำหรับการทำสิ่ง image โดยการสร้างแต่ละ image จะเป็นผลการการทำงานร่วมกันกับ Ceilometer เพื่อสร้าง Alert ในการทดสอบแนะนำให้ใช้ OS::Heat::CloudConfig เพื่อ run โปรแกรมสำหรับ สร้าง load cpu ให้แก่ instance คือ stress เพื่อให้เกิด alert ส่งไปยัง heat สำหรับการ monitor จะใช้ OS::Heat::ScalingPolicy เพื่อเริ่ม scale up และ scale down
การทดสอบ 4: สร้าง instance Autoscale multinode (แบบมี load balance)	ต้องการทดสอบ autoscale โดยทำงานร่วมกับ Load Balancer As-a-Service (LBAAS) ซึ่ง backend technology ที่ใช้นั้นคือ ha-proxy ในแต่ละ instace ที่สร้างขึ้นจะเป็นสมาชิก VIP ของ pool ใน loadbalance และได้ทำการติดตั้ง webserver ร่วมกับ php เพื่อใช้ทดสอบด้วย php ให้แสดงผลค่าของ hostname ออกมาที่หน้าจอเท่านั้น
การทดสอบ 5 การสร้าง Auto scale สำหรับ Wordpress เพื่อให้สามารถใช้งานในส่วนของ Cloud Application	ต้องการทดสอบการสร้าง wordpress ให้สามารถใช้งาน autoscale โดยการทดสอบจะทำการแยก ส่วนของ Web Application Tier ออกจาก ส่วนของ Database Tier ออกจากกัน โดย instance ของ Wordpress จะอยู่ในส่วนของสมาชิกของ Loadbalance เช่นเดียวกับการทดสอบ 4

เตรียมระบบให้พร้อมสำหรับการทดสอบ 1: พื้นฐานการสร้าง Instance ด้วย HOT (แบบ single instance)

ก่อนที่จะมีการสั่งให้ Heat Template ทำงานจะต้องมีการเตรียมสิ่งต่างๆ ให้พร้อมเสียก่อน เนื่องจาก Heat Template จะไม่สามารถติดตั้งหรือใช้งานได้ หากตรวจสอบแล้ว ว่าไม่มี Resource ที่ Template ต้องการ ได้แก่ image, network , subnet, flavor และ security group เป็นต้น

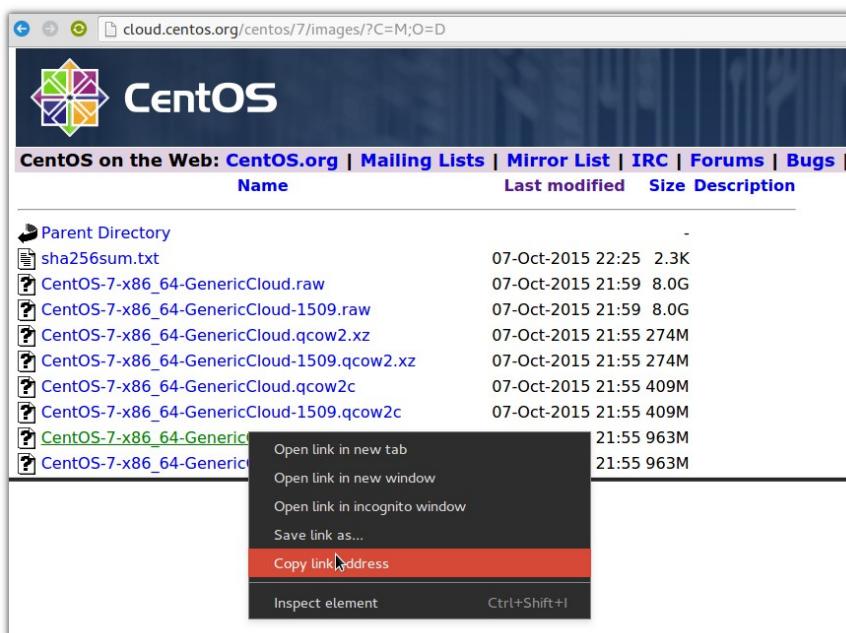
การเตรียม resource

1. เตรียม image ชื่อ CentOS-7-x86_64
2. เตรียม flavor ที่ต้องการใช้ m1.small
3. เตรียม key-pair สำหรับการ remote login
4. เตรียม private_network และ private_subnet 192.168.32.0/24
5. เตรียม router ชื่อ router1
6. ใช้งาน publick network ที่ได้สร้างไว้ก่อนหน้านี้

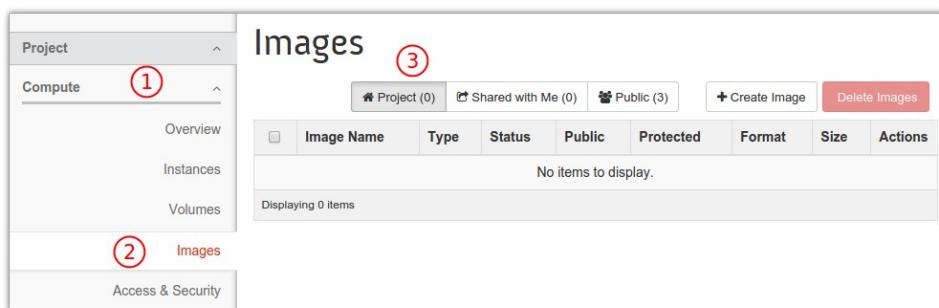
วิธีการดำเนินการ

1. เตรียม cloud image ที่มีชื่อว่า CentOS-7-x86_64

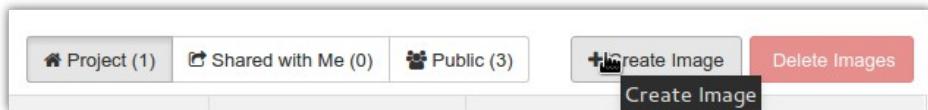
เปิด browser เช้าไปที่ <http://cloud.centos.org/centos/7/images>



จากนั้นให้เปิดหน้า dashboard และเข้าไปที่ เมนู Project > Compute > Image > Project



กดปุ่ม Create Image เพื่อสร้าง image ดังรูป



โดยลับชื่อของ image จะกำหนดไว้ดังนี้

Create An Image

Name *
CentOS-7.1-x86_64

Description
centos7.1 Cloud Image

Image Source
Image Location

Image Location ?
http://cloud.centos.org/centos/7/images/CentOS-7-x86_

Format *
QCOW2 - QEMU Emulator

Architecture
[empty field]

Minimum Disk (GB) ?
[empty field]

Minimum RAM (MB) ?
[empty field]

Copy Data ?
 Public
 Protected

Description:
Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

Cancel **Create Image**

จากนั้นให้กดปุ่ม Create Image เพื่อทำการ upload และ ลงทะเบียน image เมื่อทำการ upload และ ลงทะเบียน image เป็นที่เรียบร้อย

ดำเนินการผ่านทาง commandline

```
source ./keystone_demo
wget http://cloud.centos.org/centos/7/images/CentOS-7-x86_64-GenericCloud-1510.qcow2
glance image-create --name CentOS-7-x86_64 --disk-format qcow2 --container-format bare
--file ./CentOS-7-x86_64-GenericCloud-1510.qcow2
```

2. เตรียม Flavor ตรวจสอบว่าในระบบมี flavor ที่ต้องการหรือไม่ ในการทดสอบนี้จะใช้ flavor ชนิด m1.small มีขนาด 1 VCPUs, 2GB RAM, 20GB Root Disk

ให้เปิดหน้า dashboard และเข้าไปที่ Project → Compute → Instance → Launch Instance

ทำการตรวจสอบดูว่ามี m1.small ไว้สำหรับการทดสอบ

Launch Instance

Details *	Access & Security	Networking *	Post-Creation	Advanced Options												
Availability Zone nova	Specify the details for launching an instance. The chart below shows the resources used by this project in relation to the project's quotas.															
Instance Name * <input type="text"/>																
Flavor * (1) <input type="text" value="m1.small"/> m1.tiny m2.small m1.small m1.medium m1.large m1.xlarge	Flavor Details <table border="1"> <tr><td>Name</td><td>m1.small</td></tr> <tr><td>VCPUs</td><td>1</td></tr> <tr><td>Root Disk</td><td>20 GB</td></tr> <tr><td>Ephemeral Disk</td><td>0 GB</td></tr> <tr><td>Total Disk</td><td>20 GB</td></tr> <tr><td>RAM</td><td>2,048 MB</td></tr> </table>				Name	m1.small	VCPUs	1	Root Disk	20 GB	Ephemeral Disk	0 GB	Total Disk	20 GB	RAM	2,048 MB
Name	m1.small															
VCPUs	1															
Root Disk	20 GB															
Ephemeral Disk	0 GB															
Total Disk	20 GB															
RAM	2,048 MB															
Select source <input type="text"/>	Project Limits <table> <tr><td>Number of Instances</td><td>0 of 10 Used</td></tr> <tr><td>Number of VCPUs</td><td>0 of 20 Used</td></tr> <tr><td>Total RAM</td><td>0 of 51,200 MB Used</td></tr> </table>				Number of Instances	0 of 10 Used	Number of VCPUs	0 of 20 Used	Total RAM	0 of 51,200 MB Used						
Number of Instances	0 of 10 Used															
Number of VCPUs	0 of 20 Used															
Total RAM	0 of 51,200 MB Used															
<input type="button" value="Cancel"/> <input type="button" value="Launch"/>																

3. เตรียม key-pair สำหรับการ remote login

Access & Security

Project	Compute (1)	Security Groups	Key Pairs	Floating IPs	API Access	(3)
Overview	Instances	Filter <input type="text"/>	<input type="button" value="Create Key Pair"/> <input type="button" value="Import Key Pair"/>			
Volumes	Images		<input type="button" value="Create Key Pair"/>			
No items to display. Displaying 0 items						

ในการทดสอบสร้าง key ชื่อว่า cloudkey1 และ Download มาเก็บไว้ในเครื่อง และให้เปลี่ยน permission ก่อนนำไปใช้

```
# chmod 600 cloudkey1
```

Create Key Pair

Key Pair Name *	(4)
<input type="text" value="cloudkey1"/>	
Description:	
Key pairs are ssh credentials which are injected into images when they are launched. Creating a new key pair registers the public key and downloads the private key (a .pem file).	
Protect and use the key as you would any normal ssh private key.	
(5)	
<input type="button" value="Cancel"/> <input type="button" value="Create Key Pair"/>	

4. เตรียม private_network เพื่อใช้เป็น tenant network ให้แก่ instance ที่สร้างขึ้น

Networks

Name	Subnets Associated	Shared	Status	Admin State	Actions
No items to display.					
Displaying 0 items					

ชื่อว่า private_network

Create Network

Network	Subnet *	Subnet Details
Network Name	(4)	
Admin State *	<input type="button" value="UP"/> <input type="button" value="DOWN"/>	
Create a new network. In addition, a subnet associated with the network can be created in the next panel.		
(5)		
<input type="button" value="Cancel"/> <input type="button" value="« Back"/> <input type="button" value="Next »"/>		

ทำการสร้าง subnet ที่ใช้ชื่อว่า private_sub

Create Network

Network > Subnet > Subnet Details

Create Subnet

Subnet Name: private_sub 6

Network Address *: 192.168.32.0/24 7

IP Version *: IPv4

Gateway IP:

Disable Gateway 8

Cancel | « Back | Next »

ทำการตั้งค่า DNS Server เครื่องที่ได้รับ IP จาก DHCP

Create Network

Network > Subnet > Subnet Details

Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools:

DNS Name Servers: 8.8.8.8, 8.8.4.4 9

Host Routes:

Cancel | « Back | 10 Create

เมื่อกดปุ่ม Create จะได้ข้อมูลของ network แสดงออกมาดังรูป

Networks

Name	Subnets Associated	Shared	Status	Admin State	Actions
private_network	private_sub 192.168.32.0/24	No	Active	UP	Edit Network

สรุปค่าที่ต้องใช้ในการทดสอบ อาจตรวจสอบผ่านทาง GUI หรือ Command line เพื่อสร้างความคุ้นเคยกับการใช้งาน cloud admin จะตรวจสอบด้วยการใช้ command line

ตรวจสอบชื่อของ image

```
# source ./keystonerc_admin
# glance image-list
```

ตรวจสอบ หมายเลข id ของ network, subnet

```
# neutron net-list
# neutron subnet-list
```

5 ขั้นตอนสร้าง Router

หลังจากสร้าง network ต่อให้สร้าง router ชื่อ router1 เชื่อม private_network และ public_network

Create Router

Router Name *	router1	Description:
Admin State	UP	Creates a router with specified parameters.
External Network	public_network	

เพิ่ม interface ของ router1 ไปยัง private_network

Add Interface

Subnet *	private_network: 192.168.32.0/24 (private_sub)	Description:
IP Address (optional) ?		You can connect a specified subnet to the router.
Router Name *	router1	The default IP address of the interface created is gateway of the selected subnet. You can specify another IP address of the interface here. You must select a subnet to which the specified IP address belongs from the above list.
Router ID *	b031e9d5-ce1e-4af6-9281-2683506962ef	

ผลที่ได้แสดงดังรูปด้านล่าง

openstack demo

Router Details

Project: Compute

Network: Networks

Routers

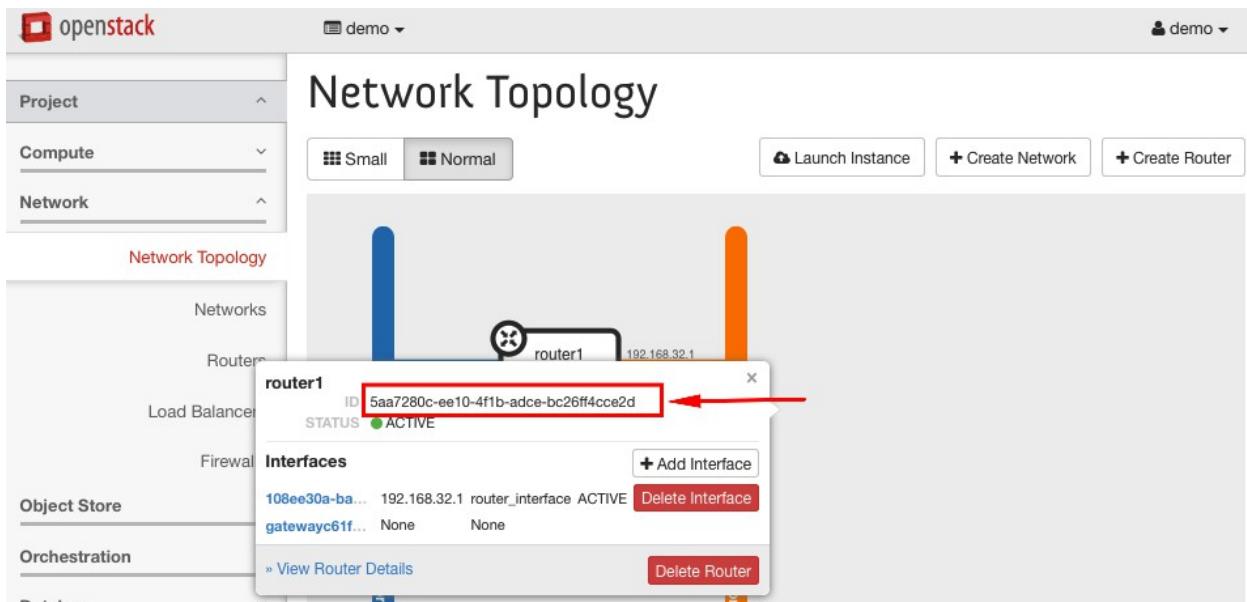
Load Balancers

Overview Interfaces

<input type="checkbox"/>	Name	Fixed IPs	Status	Type	Admin State	Actions
<input type="checkbox"/>	(108ee30a-ba00)	192.168.32.1	Down	Internal Interface	UP	<input type="button" value="Delete Interface"/>

Displaying 1 item

พิจารณาความของ network topology



ตรวจสอบ network namespace ของ router-<id> ที่มี id ตรงกับ id ด้านบน

```
[root@cloud ~(keystone_admin)]# ip netns
qrouter-5aa7280c-ee10-4f1b-adce-bc26ff4cce2d
qdhcp-07f908e4-8321-4d74-a2ed-5088f19c8bd6
qrouter-b031e9d5-ce1e-4af6-9281-2683506962ef
qdhcp-32f4ca4c-de2d-4be2-a7fb-c2d28bb8388d
```

ตรวจสอบ ip ที่อยู่ภายใน router

```
[root@cloud ~(keystone_demo)]# neutron net-list
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
29: qr-108ee30a-ba: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:e3:80:b1 brd ff:ff:ff:ff:ff:ff
        inet 192.168.32.1/24 brd 192.168.32.255 scope global qr-108ee30a-ba
            valid_lft forever preferred_lft forever
        inet6 fe80::f816:3eff:fe3:80b1/64 scope link
            valid_lft forever preferred_lft forever
30: qg-3901243a-03: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:e9:82:69 brd ff:ff:ff:ff:ff:ff
        inet 103.27.200.240/25 brd 103.27.200.255 scope global qg-3901243a-03
            valid_lft forever preferred_lft forever
        inet6 fe80::f816:3eff:fe9:8269/64 scope link
            valid_lft forever preferred_lft forever
```

ตรวจสอบการเชื่อมต่อไปยัง router ด้วยการ ping ไปยัง 103.27.200.240

```
[root@cloud ~(keystone_admin)]# ping -c4 103.27.200.240
PING 103.27.200.240 (103.27.200.240) 56(84) bytes of data.
64 bytes from 103.27.200.240: icmp_seq=1 ttl=64 time=0.224 ms
103.27.200.24064 bytes from 103.27.200.240: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 103.27.200.240: icmp_seq=3 ttl=64 time=0.047 ms
103.27.200.24064 bytes from 103.27.200.240: icmp_seq=4 ttl=64 time=0.045 ms
```

การทดสอบ 1: พื้นฐานการสร้าง Instance ด้วย HOT (แบบ single instance)

การทดสอบนี้เป็นการทดสอบการสร้าง และการใช้งาน heat template แบบง่าย ไม่มีความซับซ้อน และจะเป็นการทดสอบการทำงานของการอ่านค่าใน parameters ไปใช้ในส่วนของ resource การทำงานของ heat engine จะทำการอ่านและตรวจสอบ template ก่อนที่จะมีการสั่งให้ Heat Template ทำงาน จะต้องมีการเตรียมค่าตัวแปรต่างๆ ให้พร้อมเสียก่อน เนื่องจาก Heat Template จะไม่สามารถติดตั้งหรือใช้งานได้ หากตรวจสอบแล้ว ว่าค่าตัวแปรไม่ครบถ้วนตามที่ได้ระบุไว้ใน Template ต้องการ ได้แก่ image, network , subnet, flavor และ security group เป็นต้น ค่าตัวแปรหรือ parameters นี้จะถูกนำไปใช้ในการสร้าง Resource

ทำการทดสอบ upload template เชื่อ stack1.yaml ด้วยการสร้าง file ชื่อ stack1 มีวัตถุประสงค์เพื่อสร้าง instance โดยการใช้ Heat Template แทนการสร้างแบบ manual โดยให้สามารถไฟล์ตามตัวอย่างด้านล่าง และทำการ upload ให้ทำการ Clone git ด้านล่าง

```
git clone git@github.com:thaiopen/privatecloud.git
cd privatecloud
vi stack1.yaml
```

คำอธิบายรายละเอียดแต่ละส่วนของ template stack1.yaml

โครงสร้างจะประกอบด้วยส่วนของ parameters ที่หน้าที่ รองรับค่าที่ผู้ใช้ป้อนมาหรือกำหนดให้เป็นค่า default หากไม่มีการป้อนค่า และส่วนของ resources ที่จะส่งต่อไปยัง heat engine ในไปสร้างองค์ประกอบต่างๆใน openstack

```
1  heat_template_version: 2015-04-30
2
3  description: Simple template to deploy a single compute instance
4
5  ► parameters: ●●●
6
7  ► resources: ●●●
```

จากรูปด้านล่างจะเห็นว่า มีกำหนดตัวแปรทั้งหมด 4 ตัวแปร คือ

image คือ ชื่อของ image ที่จะนำไปสร้าง instance และมีค่า default เป็น CentOS-7-x86_64

flavor คือ ค่าของ profile ของ instance ที่จะสร้าง มีค่า default เป็น m1.small

key คือ ค่าของ keypair ที่จะนำไปใช้เพื่อการ ssh เนื่องจาก instance ที่สร้างรองรับการใช้งาน ssh private network คือ ชื่อของ tenant network สำหรับ รองรับในการสร้าง instance

```

1 heat_template_version: 2015-04-30
2
3 description: Simple template to deploy a single compute instance
4
5 parameters:
6   image:
7     type: string
8     label: Image name or ID
9     description: Image to be used for compute instance
10    default: CentOS-7-x86_64
11   flavor:
12     type: string
13     label: Flavor
14     description: Type of instance (flavor) to be used
15    default: m1.small
16   key:
17     type: string
18     label: Key name
19     description: Name of key-pair to be used for compute instance
20    default: cloudkey1
21   private_network:
22     type: string
23     label: Private network name or ID
24     description: Network to attach instance to.
25    default: private_network
26
27 resources:
28   my_instance:
29     type: OS::Nova::Server
30   properties:
31     image: { get_param: image }
32     flavor: { get_param: flavor }
33     key_name: { get_param: key }
34   networks:
35     - network: { get_param: private_network }

```

ในส่วนของ source เป็นการกำหนดให้มีสร้าง instance ชื่อ my_instance เนื่องจากเป็นการสร้าง instance จึงกำหนดให้มีการใช้ type OS::Nova::Server และมีการกำหนดค่า property ให้แก่ OS::Nova::Server คือ

image คือ ค่า image โดยใช้คำสั่ง get_param เพื่อดึงค่า ตัวแปรชื่อ image ในส่วนของ parameters

flavor คือ ค่า flavor โดยใช้คำสั่ง get_param เพื่อดึงค่า ตัวแปรชื่อ flavor ในส่วนของ parameters

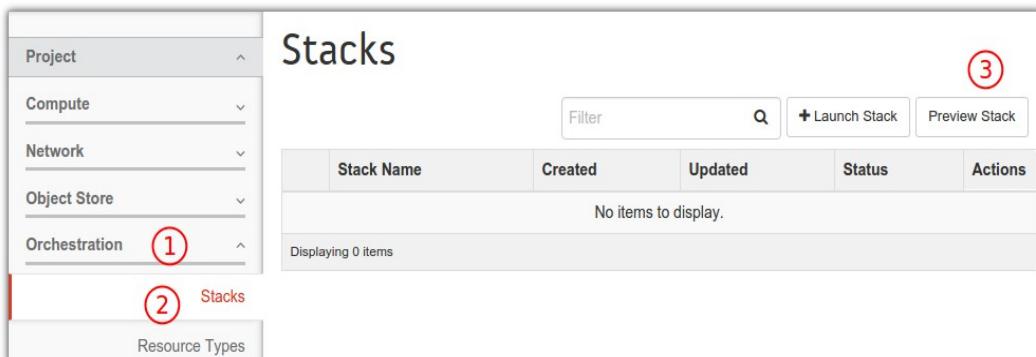
key คือ ค่า keypair โดยใช้คำสั่ง get_param เพื่อดึงค่า ตัวแปรชื่อ key ในส่วนของ parameters

network คือ ค่า flavor โดยใช้คำสั่ง get_param เพื่อดึงค่า ตัวแปรชื่อ network ในส่วนของ

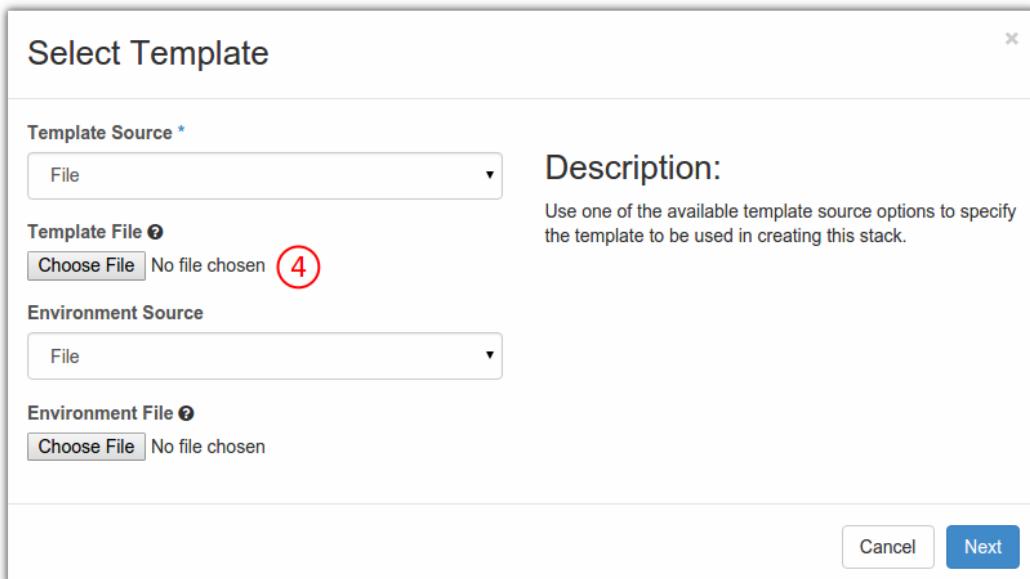
parameters

ดำเนินการทดสอบ

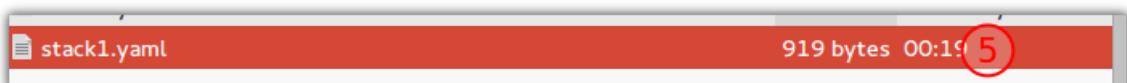
ต่อมาให้เลือก Launch Stack เพื่อทำการ upload ผ่านทาง horizone เลือก Template file เพื่อ upload file stack1 หลังจากเลือกแล้วก็ให้ดำเนินการ กด next เพื่อเข้าสู่กระบวนการต่อไป



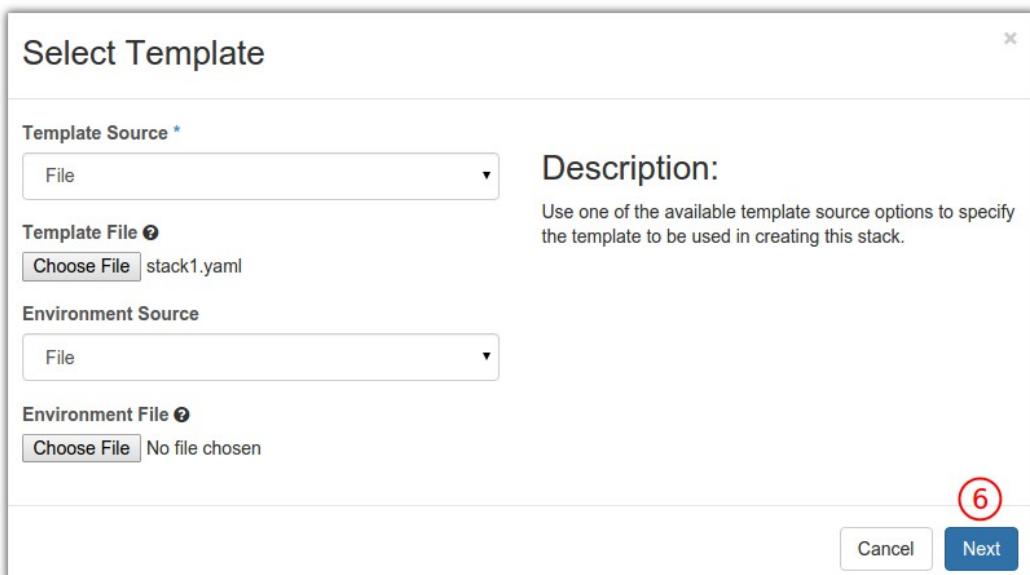
ทำการ upload heat template file



ทำการเลือก file ที่จะทำการ upload ใน lab นี้จะเลือก file ที่ชื่อว่า stack1.yaml



จากนั้นให้กด next สำหรับขั้นตอนถัดไป



หากทำการ upload สำเร็จ Horizon จะแสดงแบบ form ให้กรอกให้เรากรอกข้อมูลให้ครบถ้วน

Launch Stack

Stack Name * 1 Description:
Create a new stack with the provided values.

Creation Timeout (minutes) * 2

Rollback On Failure

Password for user "demo" * 3

Flavor * 4

Image name or ID * 5

Key name * 6

Private network name or ID * 7

8

สำหรับรายละเอียดของแต่ละช่องกรอกข้อมูลนั้นเป็นดังนี้

1 Stack Name ชื่อของ stack โดยจะเป็นชื่อเดียวกับที่ได้สำหรับ lab นี้จะตั้งชื่อว่า lab1

2 Creation Timeout (minutes) เนื่องจาก heat เมื่อทำงานจะต้องไปติดต่อกับ component อื่นๆ ใน OpenStack ค่าดังกล่าวจะหมายถึงเวลาสูงสุดที่จะรอได้ก่อนที่เปลี่ยน state ไปเป็น fail

3 Password for user "demo" เป็น password ที่ใช้สำหรับการยืนยันตัวตนสำหรับการเริ่มทำงานกับ heat สำหรับการทดสอบนี้จะใช้ user ที่ชื่อว่า demo โดยถูกสร้างจาก admin

4 Flavor ขนาดของ flavor ที่จะใช้ใน stack ซึ่งดังกล่าวถูกสร้างจาก parameter ใน hot template ดังนี้

```
flavor:
  type: string
  label: Flavor
  description: Type of instance (flavor) to be used
  default: m1.small
```

5 Image name or ID ชื่อหรือหมายเลข UUID ของ Image ที่จะใช้ใน stack ซึ่งดังกล่าวถูกสร้างจาก parameter ใน hot template ดังนี้

```
image:
  type: string
  label: Image name or ID
  description: Image to be used for compute instance
  default: CentOS-7.1-x86_64
```

6 Key name ชื่อ key ที่ใช้สำหรับ instance ที่สร้างจาก template ซึ่งดังกล่าวถูกสร้างจาก parameter ใน hot template ดังนี้

```

key:
  type: string
  label: Key name
  description: Name of key-pair to be used for compute instance
  default: cloudkey1

```

7 Private network name or ID ชื่อหรือหมายเลข UUID ของ network ที่จะใช้ใน stack ซึ่งดังกล่าวถูกสร้างจาก parameter ใน hot template ดังนี้

```

private_network:
  type: string
  label: Private network name or ID
  description: Network to attach instance to.
  default: private_network

```

จากนั้นให้กด Launch เมื่อสำเร็จจะแสดงผลลัพธ์ใน Stacks menu ดังรูป

The screenshot shows the OpenStack Stacks interface. On the left, there's a sidebar with 'Project' dropdown and 'Compute', 'Network', 'Object Store', and 'Orchestration' sections. The main area is titled 'Stacks' with a table header: Stack Name, Created, Updated, Status, Actions. A single row for 'lab1' is listed with '0 minutes' for Created, 'Never' for Updated, and 'Create In Progress' for Status. There's a 'Check Stack' button in the Actions column.

เมื่อเสร็จแล้วจะเปลี่ยนสถานะเป็น create complete

This screenshot is identical to the one above, except the status for stack 'lab1' has changed from 'Create In Progress' to 'Create Complete'.

ผลที่แสดงใน Instances menu

The screenshot shows the OpenStack Instances interface. On the left, there's a sidebar with 'Project' dropdown and 'Compute' section containing 'Overview', 'Instances' (which is selected), 'Volumes', 'Images', and 'Access & Security'. The main area is titled 'Instances' with a table header: Instance Name, Image Name, IP Address, Size, Key Pair, Status, Availability Zone, Task, Power State, Time since created, Actions. A single row for 'lab1-my_instance-t23qjvnbtu' is listed with 'CentOS-7-x86_64' for Image Name, '192.168.32.3' for IP Address, 'm1.small' for Size, 'cloudkey1' for Key Pair, 'Active' for Status, 'nova' for Availability Zone, 'None' for Task, 'Running' for Power State, and '2 minutes' for Time since created. A red arrow points to a context menu for the instance row with options 'Associate Floating IP' and 'Disassociate Floating IP'.

ให้ทำการเพิ่ม floating ip

Instances											
	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
	lab1-my_instance-t23qjvnbui	CentOS-7-x86_64	192.168.32.3 Floating IPs: 103.27.200.241	m1.small	cloudkey1	Active	nova	None	Running	4 minutes	<button>Create Snapshot</button>

Displaying 1 item

ตรวจสอบ security group

Instances											
	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
	lab1-my_instance-t23qjvnbui	CentOS-7-x86_64	192.168.32.3 Floating IPs: 103.27.200.241	m1.small	cloudkey1	Active	nova	None	Running	4 minutes	<button>Create Snapshot</button>

Displaying 1 item

จะได้ว่า instance มี security group ชื่อ default โดยค่าตั้งต้นนั้น security group default นี้จะไม่สามารถ ping (icmp protocol) หรือ ssh ได้ดังนั้นให้ทำการเพิ่ม role ให้แก่

Edit Instance

Information *	Security Groups
Add and remove security groups to this project from the list of available security groups.	
<input type="button" value="All Security Groups"/> Filter <input type="text"/>	<input type="button" value="Instance Security Groups"/> Filter <input type="text"/>
No security groups found.	<div style="border: 1px solid red; padding: 2px; margin-right: 10px;">default</div> <div style="border: 1px solid blue; padding: 2px;">-</div>

การเพิ่ม role สามารถดำเนินการผ่าน horizon

Access & Security											
Security Groups				Key Pairs				Floating IPs API Access			
	Name	Description	Actions								
	default	Default security group	<button>Manage Rules</button>								

Displaying 1 item

เพิ่ม rule

icmp เพื่อให้สามารถ ping และ ssh เพื่อให้สามารถ ssh ได้

สามารถตรวจสอบผ่านทาง commandline ได้ดังนี้

```
[root@cloud ~]# nova secgroup-list-rules default
+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+
| tcp          | 22        | 22      | 0.0.0.0/0 | default      |
| icmp         | -1        | -1      | 0.0.0.0/0 | default      |
+-----+-----+-----+-----+
```

หลังจากได้เพิ่มติม icmp และ ssh role แล้ว ให้ทำการทดสอบการ ping และ ssh สามารถใช้คำสั่ง nova list

เพื่อดูรายละเอียดของ instance ของ userdemo

```
[root@cloud ~]# nova list
+-----+-----+-----+-----+-----+
| ID           | Name            | Status | Task State | Power State | Networks   |
+-----+-----+-----+-----+-----+
| d187a81b-062e-4e5e-8e40-d9fd81bbff13 | lab1-my_instance-t23qiqvnbtu | ACTIVE | -          | Running     | private_network=192.168.32.3, 103.27.200.241 |
+-----+-----+-----+-----+-----+
[root@cloud ~]# ping -c4 103.27.200.241
PING 103.27.200.241 (103.27.200.241) 56(84) bytes of data.
64 bytes from 103.27.200.241: icmp_seq=1 ttl=63 time=0.345 ms
64 bytes from 103.27.200.241: icmp_seq=2 ttl=63 time=0.192 ms
64 bytes from 103.27.200.241: icmp_seq=3 ttl=63 time=0.188 ms
64 bytes from 103.27.200.241: icmp_seq=4 ttl=63 time=0.185 ms
```

ทดสอบการ ssh ด้วย key-pair

```
[root@cloud ~](keystone_demo)]# chmod 600 cloudkey1.pem
[root@cloud ~](keystone_demo)]# ssh -i cloudkey1.pem centos@103.27.200.241
The authenticity of host '103.27.200.241 (103.27.200.241)' can't be established.
ECDSA key fingerprint is 25:77:ab:a3:84:40:88:6d:f2:7d:4a:9a:b1:af:74:a1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '103.27.200.241' (ECDSA) to the list of known hosts.
[centos@lab1-my-instance-t23qiqvnbiu ~]$ 
```

การตรวจสอบ heat ผ่านทาง command line ด้วยคำสั่ง heat stack-list

```
# heat stack-list
```

จะได้ผลลัพธ์รูป ให้สังเกตุว่าในช่องของ stack_status จะแสดงผลว่า CREATE_COMPLETE

```
[root@cloud ~](keystone_admin)]# heat stack-list
```

id	stack_name	stack_status	creation_time
508a61dc-e2b0-4de2-a713-23c906d0bb16	lab1	CREATE_COMPLETE	2015-12-02T15:35:26Z

ส่วน log ที่เกิดขึ้นจากการทำงานจะอยู่ที่ /var/log/heat

```
[root@cloud heat]# pwd
/var/log/heat
[root@cloud heat]# ls -l
total 5400
-rw-r--r--. 1 heat heat 4766 Dec 2 23:36 heat-api-cfn.log
-rw-r--r--. 1 heat heat 300 Nov 30 22:22 heat-api-cloudwatch.log
-rw-r--r--. 1 heat heat 71221 Dec 3 01:17 heat-api.log
-rw-r--r--. 1 heat heat 5386989 Dec 3 01:59 heat-engine.log
```

ตรวจสอบ log การทำงาน ในการสร้าง lab1

```
# grep lab1 heat-engine.log
```

ผลที่ได้จะได้รูปด้านล่าง

```
2015-12-02 22:48:10.531 4395 INFO heat.engine.stack [-] Stack DELETE COMPLETE (lab1): Stack DELETE completed successfully
2015-12-03 01:17:19.463 20466 INFO heat.engine.service [req-db2108e9-7f33-4d69-b73d-fb07040effbf demo demo] Creating stack lab1
2015-12-03 01:17:19.862 20466 INFO heat.engine.stack [-] Stack CREATE IN_PROGRESS (lab1): Stack CREATE started
2015-12-03 01:17:19.864 20466 INFO heat.engine.resource [-] creating Server "my_instance" Stack "lab1" [3efe5e36-928b-4ec0-ab30-68d73eeef51ac]
2015-12-03 01:17:30.115 20466 INFO heat.engine.stack [-] Stack CREATE COMPLETE (lab1): Stack CREATE completed successfully
```

วิธีการลบ stacks

การ stacks จะส่งผลให้ instance ที่สร้างขึ้นจะถูกทำลายไปด้วย การลบนั้นสามารถทำได้ โดยเลือก stack name และ กดปุ่ม Delete Stacks

<input checked="" type="checkbox"/>	Stack Name	Created	Updated	Status	Actions
<input checked="" type="checkbox"/>	lab1	48 minutes	Never	Create Complete	<button>Check Stack</button> <button>Delete Stacks</button>

การลบ stack ด้วย command line

```
[root@cloud ~](keystone_demo)]# heat stack-delete lab1
+-----+-----+-----+
| id | stack_name | stack_status | creation_time |
+-----+-----+-----+
| 3efe5e36-928b-4ec0-ab30-68d73eeef51ac | lab1 | DELETE_IN_PROGRESS | 2015-12-02T18:17:19Z |
+-----+-----+-----+
```

ตรวจสอบ log

```
[root@cloud heat]# tail -f heat-engine.log
2015-12-03 02:08:55.772 20468 INFO heat.engine.service [req-93b817bd-c078-429a-bac5-b46106caec91 - -] Service c88d4d18-5d69-4ace-
2015-12-03 02:08:55.785 20465 INFO heat.engine.service [req-253d61b8-d298-41f4-808b-87599520351a - -] Service 1fc37371-62fb-423d-
2015-12-03 02:08:55.791 20471 INFO heat.engine.service [req-fd85e508-3eab-41c6-8417-ceffccfc35f13 - -] Service f53c1238-4dfe-4b0e-
2015-12-03 02:08:55.791 20467 INFO heat.engine.service [req-ea2ebe3c-75e6-4afb-abe3-e30d62bd9af8 - -] Service 1b1f35cb-f72a-49b7-
2015-12-03 02:08:55.834 20472 INFO heat.engine.service [req-0525f465-1c70-4690-aa95-1f037c99ee18 - -] Service d8b7139d-3fee-4c62-
2015-12-03 02:08:55.835 20473 INFO heat.engine.service [req-a31dde3a-5034-48b3-984d-7de782949c2d - -] Service 3c127362-adc7-4716-
2015-12-03 02:09:16.709 20471 INFO heat.engine.service [req-91233c67-8dee-40a4-9542-c0c515c0e4dc - -] demol Deleting stack lab1
2015-12-03 02:09:16.723 20471 INFO heat.engine.stack [-] Stack DELETE IN_PROGRESS (lab1): Stack DELETE started
2015-12-03 02:09:16.736 20471 INFO heat.engine.resource [-] deleting Server "my_instance" [d187a81b-062e-4e5e-8e40-d9fd81bfb13]
```

ตรวจสอบ instance จะพบว่าถูกลบไปแล้วเช่นกัน (ไม่มีรายการ)

```
[root@cloud ~(keystone_demo)]# nova list
+-----+-----+-----+-----+
| ID | Name | Status | Task State | Power State | Networks |
+-----+-----+-----+-----+
|-----+-----+-----+-----+
```

การทดสอบ 2: สร้าง Database Server ด้วย การ run script ภายใน Heat Template

ตัวอย่าง HOT Template สำหรับการสร้าง Database Server (MariaDB) เพื่อศึกษาการใช้งาน Heat Template รวมกับการใช้งาน Cloud init script เพื่อการติดตั้ง package MariaDB ทำการ Clone git ด้านล่าง

```
git clone git@github.com:thaiopen/privatecloud.git
cd privatecloud
vi stack2.yaml
```

คำอธิบายรายละเอียดแต่ละส่วนของ template stack2.yaml

หากดูโครงสร้างของ template จะแบ่งโครงสร้างออกเป็น 4 ส่วนได้แก่ heat_template_version, description, parameters, resources

```
1  heat_template_version: 2015-04-30
2
3  description: >
4    HOT template to deploy database servers(mariadb) with init script
5
6  parameters: <>
83
84  resources: <>
```

ในส่วนของ parameters ประกอบ ตัวแปร 11 ตัว ที่จะถูกเรียก ด้วย function get_params ในส่วน resources

```

1 ▼ heat_template_version: 2015-04-30
2 ▾
3 ▼ description: >
   HOT template to deploy database servers(mariadb) with init script
4 ▾
5
6 ▼ parameters:
7 ▶   key_name: ●●●
8 ▶   image: ●●●
9 ▶   flavor: ●●●
10 ▶  db_name: ●●●
11 ▶  db_username: ●●●
12 ▶  db_password: ●●●
13 ▶  db_root_password: ●●●
14 ▶  public_net_id: ●●●
15 ▶  private_net_id: ●●●
16 ▶  private_subnet_id: ●●●
17 ▶  key_name: ●●●
18
19
20 ▼ resources:
21 ▶   dbserver: ●●●
22 ▶   dbserver_port: ●●●
23 ▶   dbfloating_ip: ●●●
24 ▶   db_security_group: ●●● |
```

parameter แต่ละตัวมีโครงสร้างที่เหมือนกัน ประกอบด้วยคุณสมบัติ ได้แก่ type, description, default, constraints ยกตัวอย่าง โครงสร้าง parameter ของ db_name, db_username, db_password

```

23 ▼   db_name:
24     type: string
25     description: Database name
26     default: databasetest
27     ▼ constraints:
28       - length: { min: 1, max: 64 }
29         description: db_name must be between 1 and 64 characters
30       - allowed_pattern: '[a-zA-Z][a-zA-Z0-9]*'
31         description: >
32           db_name must begin with a letter and contain only alphanumeric
33             characters
```

หมายความว่าเป็นการกำหนด parameter ชื่อ db_name เพื่อสร้าง ฐานข้อมูลชื่อ databasetest เป็นค่า default และมีการกำหนด constraint มีขนาดอย่างน้อย 1 ตัวอักษร และมีขนาดไม่เกิน 64 ตัวอักษร และกำหนดให้มี pattern ด้วย regular express ให้เป็นตัวอักษร หรือตัวเลข template จะอยู่ในรูปแบบของ yaml format หากต้องการดูในรูปแบบของ json สามารถแปลงได้ ทดสอบการแปลงด้วย <http://yamltojson.com/>

YML

Paste your YAML here

```

db_name:
  type: string
  description: Database name
  default: databasetest
  constraints:
    - length: { min: 1, max: 64 }
      description: db_name must be between
      1 and 64 characters
    - allowed_pattern: '[a-zA-Z][a-zA-Z0-9]*'
      description: >
        db_name must begin with a letter and
        contain only alphanumeric
        characters
  
```

JSON

See the JSON output here

```

{
  "db_name": {
    "type": "string",
    "description": "Database name",
    "default": "databasetest",
    "constraints": [
      {
        "length": {
          "min": 1,
          "max": 64
        },
        "description": "db_name must be
        between 1 and 64 characters"
      },
      {
        "allowed_pattern": "[a-zA-Z][a-zA-Z0-9]*",
        "description": "db_name must begin with
        a letter and contain only alphanumeric
        characters\n"
      }
    ]
  }
}
  
```

การเรียกใช้งาน parameter จะเป็นเกิดขึ้นใน resources เพื่อนำไปสร้าง องค์ประกอบ heat template จะดูว่าผู้ใช้งานต้องการสร้างองค์ประกอบอะไรนั้น จะดูจาก คุณสมบัติ type และนำเอา parameter ไปกำหนดค่าให้แก่ properties แต่ละ type จะมี properties ที่ไม่เหมือนกัน เช่น type: OS::Nova::Server เป็น type ที่ระบุให้มีการสร้าง instance และ properties ที่จำเป็นสำหรับ OS::Nova::Server เพื่อนำไปสร้าง instance ประกอบด้วย name, image, flavor, key_name, networks, user_data ตัวอย่าง

```

83  resources:
84    dbserver:
85      type: OS::Nova::Server
86      properties:
87        name: wp_dbserver
88        image: { get_param: image }
89        flavor: { get_param: flavor }
90        key_name: { get_param: key_name }
91        networks:
92          - port: { get_resource: dbserver_port }
93        user_data:
94          str_replace:
95            template: |
96              #!/bin/bash -v
97              yum -y install mariadb mariadb-server
98              systemctl start mariadb.service
99              systemctl enable mariadb.service
100             mysqladmin -u root password db_rootpassword
101             cat << EOF | mysql -u root --password=db_rootpassword
102             CREATE DATABASE db_name;
103             GRANT ALL PRIVILEGES ON db_name.* TO "db_user"@"%" IDENTIFIED BY "db_password";
104             GRANT ALL PRIVILEGES ON db_name.* TO "db_user"@"localhost" IDENTIFIED BY "db_password";
105             FLUSH PRIVILEGES;
106             EXIT
107             EOF
108           params:
109             db_rootpassword: { get_param: db_root_password }
110             db_name: { get_param: db_name }
111             db_user: { get_param: db_username }
112             db_password: { get_param: db_password }
  
```

การทดสอบ template

หลังจาก clone มาแล้วยังไม่สามารถใช้งานได้ทันที ให้ทำการแก้ไขค่าให้สอดคล้องกับ ระบบที่ต้องการทดสอบ ได้แก่ค่าดังต่อไปนี้

ค่าตัวแปร parameter ที่จะต้องแก้ไข	ค่าที่ใช้งาน
image:	default: CentOS-7-x86_64
flavor:	default: m1.small
public_net_id:	default: c61fe76f-25ba-4fa6-a3f7-37d59fe17999
private_net_id:	default: 07f908e4-8321-4d74-a2ed-5088f19c8bd6
private_subnet_id:	default: e75af332-c726-420b-9c35-a69dc6b202ad
key_name:	default: cloudkey1

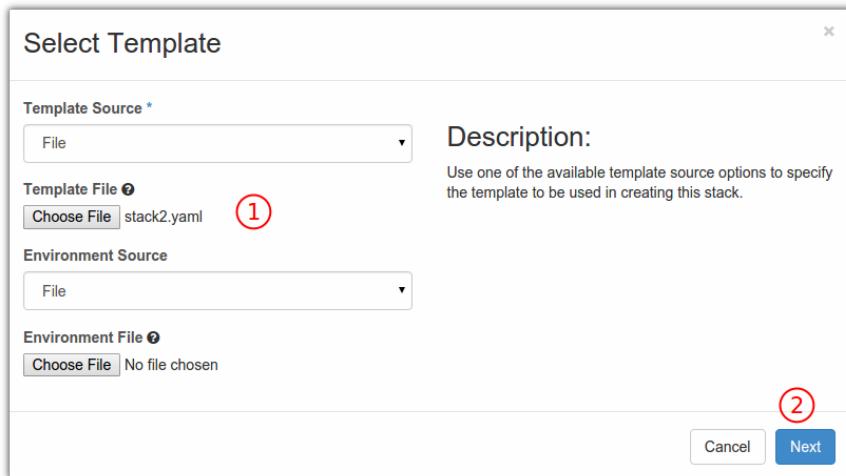
ค่าทั้งหมดสามารถที่ใช้คำสั่ง ดังนี้

```
# glance image-list
# neutron net-list
# nova keypair-list
```

ดูรูป

```
[root@cloud ~ (keystone_demo)]# glance image-list
+-----+-----+-----+-----+-----+
| ID      | Name       | Disk Format | Container Format | Size   | Status |
+-----+-----+-----+-----+-----+
| 8d2bb342-b70b-4845-8318-5c44cede66b2 | CentOS-7-x86_64 | qcow2        | bare            | 859111424 | active |
| 4747ffbf-0132-41c9-8e02-01322b067889 | Fedora23     | qcow2        | bare            | 234363392 | active |
| 53d7b5b4-b180-44d4-a444-8951a4a35bae | manila-service-image | qcow2        | bare            | 250218    | active |
| a3bb6f59-8b26-445b-aa18-a44054b8ee43 | Ubuntu-trusty | qcow2        | bare            | 258540032 | active |
+-----+-----+-----+-----+-----+
[root@cloud ~ (keystone_demo)]# neutron net-list
+-----+-----+-----+
| id      | name       | subnets          |
+-----+-----+-----+
| 07f908e4-8321-4d74-a2ed-5088f19c8bd6 | private_network | e75af332-c726-420b-9c35-a69dc6b202ad 192.168.32.0/24 |
| c61fe76f-25ba-4fa6-a3f7-37d59fe17999 | public_network  | 26291621-6ade-4a3a-bc35-d2606037fcfd 103.27.200.128/25 |
+-----+-----+-----+
[root@cloud ~ (keystone_demo)]# nova keypair-list
+-----+-----+
| Name      | Fingerprint |
+-----+-----+
| Cloudkey1 | 23:5e:1e:c0:46:64:da:fa:94:0f:1e:7b:30:68:8b:80 |
+-----+-----+
```

upload ไปยัง โดยการไปที่ เมนู Project > Orchestration > Stack



หลังจากนั้นให้กดปุ่ม Next หากไม่มีการผิดพลาดจะได้แบบ form เพื่อรับค่าจากผู้ใช้งานเพิ่มเติม form ที่แสดงผลนั้นสามารถแบ่งออกได้ 3 ส่วน ดังนี้

ส่วนที่ 1 การตั้งชื่อและ password ในการสร้าง stack

Stack Name * ? lab2

Description: Create a new stack with the provided values.

Creation Timeout (minutes) * ? 60

Rollback On Failure ?

Password for user "demo" * ?

รายละเอียดดังนี้

Stack Name ชื่อของ stack โดยจะเป็นชื่อใดก็ได้ก็ได้สำหรับ lab นี้จะตั้งชื่อว่า lab2

Creation Timeout (minutes) เนื่องจาก heat เมื่อทำงานจะต้องไปติดต่อกับ component อื่นๆ ใน OpenStack ค่าดังกล่าวจะหมายถึงเวลาสูงสุดที่จะรอได้ก่อนที่เปลี่ยน state ไปเป็น fail

Password for user "demo" เป็น password ที่ใช้สำหรับการยืนยันตัวตนสำหรับการเริ่มทำงานกับ heat สำหรับการทดสอบนี้จะใช้ user ที่ชื่อว่า demo โดยถูกสร้างจาก admin

ส่วนที่ 2 parameters สำหรับการสร้าง ฐานข้อมูล ประกอบด้วย db_name, db_password, db_root_password และ db_username

The screenshot shows a configuration interface for database credentials. On the left, there's a sidebar with sections like 'Data Proce' and 'Identity'. The main area contains four input fields:

- db_name ***: A text input field containing 'databasetest'.
- db_password ***: A password input field with an eye icon to the right.
- db_root_password ***: A password input field with an eye icon to the right.
- db_username ***: A text input field containing 'admin'.

รายละเอียด

db_name ชื่อ database ที่ต้องการสร้าง

db_password ตั้งค่า password ของ user ที่กำหนดในช่อง **db_username**

db_root_password ตั้งค่า password ให้แก่ user root

db_username ชื่อ username ที่เป็นเจ้าของ ฐานข้อมูล

ส่วนที่ 3 parameters ส่วนที่นำไปสร้าง instance

The screenshot shows a configuration interface for creating a new instance. On the left, there's a sidebar with sections like 'Identity'. The main area contains several input fields:

- flavor**: A dropdown menu showing 'm1.small'.
- image**: A dropdown menu showing 'CentOS-7-x86_64'.
- Key name**: A text input field containing 'cloudkey1'.
- private_net_id**: A text input field containing '07f908e4-8321-4d74-a2ed-5088f19c8bd6'.
- private_subnet_id**: A text input field containing 'e75af332-c726-420b-9c35-a69dc6b202ad'.
- public_net_id**: A text input field containing 'c61fe76f-25ba-4fa6-a3f7-37d59fe17999'.

At the bottom right, there are two buttons: 'Cancel' and 'Launch'.

รายละเอียด

flavor ขนาดของ flavor ที่ใช้

image ชื่อ image ที่ต้องการสร้าง instance

key_name ชื่อ key-pair

private_net_id ค่า tenant network id สำหรับ instance

private_subnet_id ค่า tenant subnet id ของ tenant network

public_net_id ค่า public network id ทำหน้าที่เป็น external network ที่เชื่อมต่อกับภายนอก

ก่อนจะกดปุ่ม lunch ให้ทำการ tail log ของ /var/log/heat/heat-engine.log เมื่อพร้อมแล้วก็ให้กด launch

The screenshot shows the OpenStack Horizon interface with the 'demo' project selected. On the left, a sidebar shows 'Compute', 'Network', 'Object Store', and 'Orchestration' sections. The main area is titled 'Stacks' and displays a table with one item: 'lab2'. The table columns are 'Stack Name', 'Created', 'Updated', 'Status', and 'Actions'. The 'Status' for 'lab2' is 'Create In Progress'. The 'Actions' column for 'lab2' contains a single button labeled 'Check Stack'.

ตรวจสอบ log

```
#tail -f /var/log/heat/heat-engine.log
```

ผลลัพท์ดังรูป

```
INFO heat.engine.service [req-95a1f07e-3e35-4676-8249-339c48a8a037 demo demo] Creating stack lab2
INFO heat.engine.resource [req-95a1f07e-3e35-4676-8249-339c48a8a037 demo demo] Validating SecurityGroup "db_security_group"
INFO heat.engine.resource [req-95a1f07e-3e35-4676-8249-339c48a8a037 demo demo] Validating Port "dbserver_port"
INFO heat.engine.resource [req-95a1f07e-3e35-4676-8249-339c48a8a037 demo demo] Validating Server "dbserver"
INFO heat.engine.resource [req-95a1f07e-3e35-4676-8249-339c48a8a037 demo demo] Validating FloatingIP "dbfloating_ip"
INFO heat.engine.stack [-] Stack CREATE IN_PROGRESS (lab2): Stack CREATE started
INFO heat.engine.resource [-] creating SecurityGroup "db_security_group" Stack "lab2" [17c9a4e3-f966-40c4-80c2-d1df8423ec8b]
INFO heat.engine.resource [-] creating Port "dbserver_port" Stack "lab2" [17c9a4e3-f966-40c4-80c2-d1df8423ec8b]
INFO heat.engine.resource [-] creating Server "dbserver" Stack "lab2" [17c9a4e3-f966-40c4-80c2-d1df8423ec8b]
INFO heat.engine.resource [-] creating FloatingIP "dbfloating_ip" Stack "lab2" [17c9a4e3-f966-40c4-80c2-d1df8423ec8b]
INFO heat.engine.stack [-] Stack CREATE COMPLETE (lab2). STACK CREATE completed successfully
```

ตรวจสอบ instance ที่สร้าง

```
#nova list
```

```
[root@cloud ~(keystone_demo)]# nova list
+-----+-----+-----+-----+-----+
| ID   | Name    | Status | Task State | Power State | Networks          |
+-----+-----+-----+-----+-----+
| e46da161-88cd-4b4e-9749-c12b94c37b95 | wp_dbserver | ACTIVE | -           | Running        | auto_floating_ip |
+-----+-----+-----+-----+-----+
```

ตรวจสอบรายละเอียดของ instance ที่สร้าง ด้วย heat

```
# nova show wp_dbserver
```

จะได้ผลดังรูปด้านล่าง

```
[root@cloud ~ (keystone_demo)]# nova show wp_dbserver
+-----+-----+
| Property | Value |
+-----+-----+
| OS-DCF:diskConfig | MANUAL |
| OS-EXT-AZ:availability_zone | nova |
| OS-EXT-STS:power_state | 1 |
| OS-EXT-STS:task_state | - |
| OS-EXT-STS:vm_state | active |
| OS-SRV-USG:launched_at | 2015-12-02T20:26:35.000000 |
| OS-SRV-USG:terminated_at | - |
| accessIPv4 | |
| accessIPv6 | |
| config_drive | |
| created | 2015-12-02T20:26:28Z |
| flavor | m1.small (2) |
| hostId | 3f4fd80408e217d615e1390e2da2fbfacd78f2fe4230f865d5a551ce |
| id | e46da161-88cd-4b4e-9749-c12b94c37b95 |
| image | CentOS-7-x86_64 (8d2bb342-b70b-4845-8318-5c44cede66b2) |
| key_name | cloudkey1 |
| metadata | {} |
| name | wp_dbserver |
| os-extended-volumes:volumes_attached | [] |
| private_network | 192.168.32.5, 103.27.200.243 |
| progress | 0 |
| security_groups | db_security_group |
| status | ACTIVE |
| tenant_id | df5ded99a7a74b6896647d11126dd7bd |
| updated | 2015-12-02T20:26:35Z |
| user_id | 3661912e4e7247718832775c02233099 |
+-----+-----+
```

ตรวจสอบ security group ที่สร้างจาก heat template แบบ อัตโนมัติ

```
[root@cloud ~ (keystone_demo)]# nova secgroup-list-rules db_security_group
+-----+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+-----+
| tcp | 3306 | 3306 | 0.0.0.0/0 | |
| icmp | -1 | -1 | 0.0.0.0/0 | |
| tcp | 22 | 22 | 0.0.0.0/0 | |
+-----+-----+-----+-----+-----+
```

ทดสอบการ ssh ด้วย key-pair ชื่อ cloudkey1.pem ไปยัง instance ที่สร้างขึ้น โดยจะทำการ ssh ไปยัง username centos

```
[root@cloud ~ (keystone_demo)]# ssh -i cloudkey1.pem centos@103.27.200.243
The authenticity of host '103.27.200.243 (103.27.200.243)' can't be established.
ECDSA key fingerprint is 7c:1d:75:78:df:e2:ff:b1:2d:7e:fb:7c:54:d2:43:c7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '103.27.200.243' (ECDSA) to the list of known hosts.
[centos@wp-dbserver ~]$
```

นอกจากจะสร้าง instance แล้วยังมีได้ script ที่เรียกใช้ด้วยโปรแกรมชื่อ cloud-init เพื่อทำการติดตั้งฐานข้อมูล ด้วย ดังนั้นสามารถทดสอบการเข้าถึง database server ด้วย user: admin และ password: 123456

```
[centos@wp-dbserver ~]$ mysql -uadmin -p123456
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 5.5.44-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

สามารถระบุค่าเพื่อให้ cloud-init ดึงรูปด้านล่าง จะต้องระบุใน template ภายใต้ส่วนของ user-data: ดังรูป

```

93 user_data:
94 str_replace:
95   template: |
96     #!/bin/bash -v
97     yum -y install mariadb mariadb-server
98     systemctl start mariadb.service
99     systemctl enable mariadb.service
100    mysqladmin -u root password db_rootpassword
101    cat << EOF | mysql -u root -p$password=db_rootpassword
102    CREATE DATABASE db_name;
103    GRANT ALL PRIVILEGES ON db_name.* TO "db_user"@"%" IDENTIFIED BY "db_password";
104    GRANT ALL PRIVILEGES ON db_name.* TO "db_user"@"localhost" IDENTIFIED BY "db_password";
105    FLUSH PRIVILEGES;
106    EXIT
107  EOF
108 params:
109   db_rootpassword: { get_param: db_root_password }
110   db_name: { get_param: db_name }
111   db_user: { get_param: db_username }
112   db_password { get_param: db_password }

```

การทำงานของ cloud-init สามารถที่รับค่าตัวแปรด้วยคำสั่ง get_param เพื่อกำหนดให้กับตัวแปรที่จะใช้ใน script ดังรูปด้านบนจะเห็นว่าใน script นั้นมีตัวแปรทั้งหมด 4 ค่าคือ db_rootpassword, db_name, db_user, db_password

ตรวจสอบการทำงานของ cloud-init ด้วยการตรวจสอบ log จะแสดงผลการทำงานช่วงที่มีการทำงานขณะเริ่มต้นการทำงานสร้าง image โดยไปที่ เมนู instance เลือก instance ด้วยการกดซื้อ

Overview	Instances	Volumes	Images	Access & Security																						
Displaying 1 item																										
	<table border="1"> <thead> <tr> <th>Instance Name</th> <th>Image Name</th> <th>IP Address</th> <th>Size</th> <th>Key Pair</th> <th>Status</th> <th>Availability Zone</th> <th>Task</th> <th>Power State</th> <th>Time since created</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>wp_dbserver</td> <td>CentOS-7-x86_64</td> <td>192.168.32.5 Floating IPs: 103.27.200.243</td> <td>m1.small</td> <td>cloudkey1</td> <td>Active</td> <td>nova</td> <td>None</td> <td>Running</td> <td>20 hours, 6 minutes</td> <td><button>Create Snapshot</button></td> </tr> </tbody> </table>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions	wp_dbserver	CentOS-7-x86_64	192.168.32.5 Floating IPs: 103.27.200.243	m1.small	cloudkey1	Active	nova	None	Running	20 hours, 6 minutes	<button>Create Snapshot</button>			
Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions																
wp_dbserver	CentOS-7-x86_64	192.168.32.5 Floating IPs: 103.27.200.243	m1.small	cloudkey1	Active	nova	None	Running	20 hours, 6 minutes	<button>Create Snapshot</button>																

หลังจากนั้นเลือก tab ที่ 2 จะพบ log ของ cloud-init

The screenshot shows the 'Log' tab for the 'wp_dbserver' instance in the OpenStack Horizon interface. The log output is as follows:

```

cloud-init[3608]: warning: /var/cache/yum/x86_64/7/base/packages/libaio-0.3.109-12.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 0xf4a08eb5: Importing GPG key 0xf4a08eb5:
cloud-init[3608]: Userid : "CentOS 7 Official Signing Key <security@centos.org>"
cloud-init[3608]: Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
cloud-init[3608]: Package : centos-release-7-1.1503.el7.centos.2.8.x86_64 (installed)
cloud-init[3608]: From : /etc/pki/rpm-GPG/RPM-GPG-KEY-CentOS-7
cloud-init[3608]: systemctl start mariadb.service
cloud-init[3608]: systemctl enable mariadb.service
cloud-init[3608]: mysqladmin -u root password 123456
cloud-init[3608]: cat << EOF | mysql -u root --password=123456
cloud-init[3608]: CREATE DATABASE databasetest;
cloud-init[3608]: GRANT ALL PRIVILEGES ON databasetest.* TO "admin"@"%" IDENTIFIED BY "123456";
cloud-init[3608]: GRANT ALL PRIVILEGES ON databasetest.* TO "admin"@"localhost" IDENTIFIED BY "123456";
cloud-init[3608]: FLUSH PRIVILEGES;
cloud-init[3608]: EXIT
cloud-init[3608]: EOF
cloud-init[3608]: Provision done: 2015-12-02 20:30:31.195393
ec2:

```

เมื่อตรวจสอบผ่านทาง horizon เมื่อไปที่ เมนู instance จะเห็นว่า heat ได้ทำการสร้าง instance มีชื่อว่า wp_dbserver และมีการกำหนดค่าของ floating ip แบบอัตโนมัติ

openstack demo

Instances

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	wp_dbserver	CentOS-7-x86_64	192.168.32.5 Floating IPs: 103.27.200.243	m1.small	cloudkey1	Active	nova	None	Running	22 minutes	<button>Create Snapshot</button>

Displaying 1 item

ตรวจสอบผ่านทาง stack topology ให้ทดสอบด้วยการนำ mouse มาวางที่ icon จะเห็นรายละเอียดของแต่ละ resource

openstack demo

Stack Details: lab2

- Topology
- Overview
- Resources
- Events
- Template

lab2
Create Complete

```

graph TD
    A(( )) --- B(( ))
    A --- C(( ))
    B --- C
  
```

ตรวจสอบผ่านทาง stack resources ใน tab ที่ 3

openstack demo

Stack Details: lab2

- Topology
- Overview
- Resources
- Events
- Template

Check Stack

Stack Resource	Resource	Stack Resource Type	Date Updated	Status	Status Reason
dbfloating_ip	1ff61046-316f-4590-aa49-c1269cb5309c	OS::Neutron::FloatingIP	29 minutes	Create Complete	state changed
db_security_group	aee893e7-1425-4579-8948-3da3d18d7c68	OS::Neutron::SecurityGroup	29 minutes	Create Complete	state changed
dbserver_port	f6797ba1-1f17-48d5-8cb7-4f771062d19a	OS::Neutron::Port	29 minutes	Create Complete	state changed
dbserver	e46da161-88cd-4b4e-9749-c12b94c37b95	OS::Nova::Server	29 minutes	Create Complete	state changed

Displaying 4 items

ตรวจสอบผ่านทาง commandline ด้วยคำสั่ง heat สามารถตรวจสอบ resource ได้ เช่น กัน

```
# heat stack-list
# heat resource-list lab2
```

ผลลัพธ์

```
[root@cloud ~]# heat stack-list
+-----+-----+-----+
| id | stack_name | stack_status | creation_time |
+-----+-----+-----+
| 17c9a4e3-f966-40c4-80c2-d1df8423ec8b | lab2 | CREATE_COMPLETE | 2015-12-02T20:26:26Z |
+-----+-----+-----+
[root@cloud ~]# heat resource-list lab2
+-----+-----+-----+-----+-----+
| resource_name | physical_resource_id | resource_type | resource_status | updated_time |
+-----+-----+-----+-----+
| db_security_group | aee893e7-1425-4579-8948-3da3d18d7c68 | OS::Neutron::SecurityGroup | CREATE_COMPLETE | 2015-12-02T20:26:26Z |
| dbfloating_ip | 1ff61046-316f-4590-aa49-c1269cb5309c | OS::Neutron::FloatingIP | CREATE_COMPLETE | 2015-12-02T20:26:26Z |
| dbserver | e46da161-88cd-4b4e-9749-c12b94c37b95 | OS::Nova::Server | CREATE_COMPLETE | 2015-12-02T20:26:26Z |
| dbserver_port | f6797ba1-1f17-48d5-8cb7-4f771062d19a | OS::Neutron::Port | CREATE_COMPLETE | 2015-12-02T20:26:26Z |
+-----+-----+-----+-----+
```

คำสั่งเพิ่มเติมหากต้องการรายละเอียดของแต่ละ resource

```
# heat resource-show lab2 db_security_group
# heat resource-show lab2 dbfloating_ip
# heat resource-show lab2 dbserver
# heat resource-show lab2 dbserver_port
```

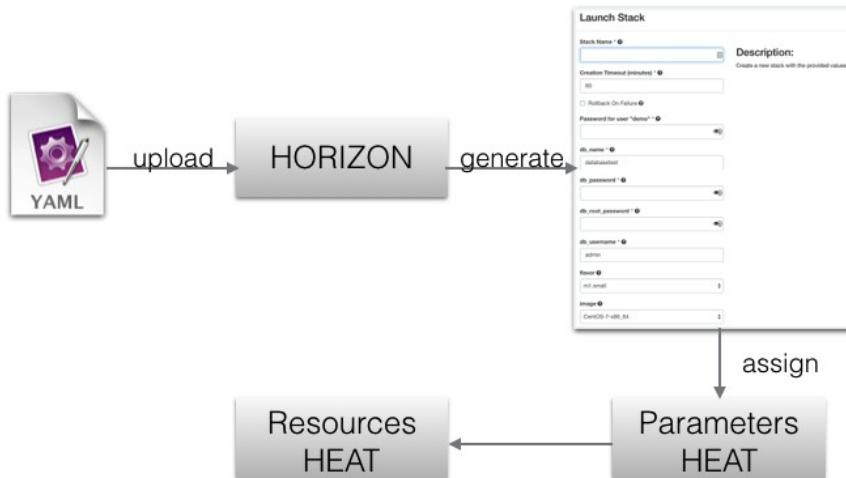
ยกตัวอย่าง การแสดง resource floating ip

```
[root@cloud ~]# heat resource-show lab2 dbfloating_ip
+-----+
| Property | Value
+-----+
| attributes | {
|   "router_id": "5aa7280c-ee10-4f1b-adce-bc26ff4cce2d",
|   "status": "ACTIVE",
|   "tenant_id": "df5ded99a7a74b6896647d11126dd7bd",
|   "floating_network_id": "c61fe76f-25ba-4fa6-a3f7-37d59fe17999",
|   "fixed_ip_address": "192.168.32.5",
|   "floating_ip_address": "103.27.200.243",
|   "port_id": "f6797ba1-1f17-48d5-8cb7-4f771062d19a",
|   "id": "1ff61046-316f-4590-aa49-c1269cb5309c"
| }
| description
| links
| logical_resource_id
| physical_resource_id | dbfloating_ip
| required_by
| resource_name | dbfloating_ip
| resource_status | CREATE_COMPLETE
| resource_status_reason | state changed
| resource_type | OS::Neutron::FloatingIP
| updated_time | 2015-12-02T20:26:26Z
+-----+
```

สำหรับการตั้งค่าในการสร้าง Database Instance โดย template ที่ใช้ใน

ความลับพันธ์ระหว่าง Horizon Form และ script

จะพบว่าเมื่อมีการ upload template ไปยัง horizon ผลที่ได้คือ horizon จะสร้างแบบฟอร์มจาก script เพื่อรับค่าจาก user ค่าที่กรอกในแบบฟอร์ม จะนำไปกำหนดให้ของตัวแปรของ parameter ที่ความสอดคล้องกับ script โดยเมื่อเป็นการกำหนดค่าผ่านตัวแปร ในส่วนของ parameter ดังนี้



db_name ชื่อของ database ที่สร้างเมื่อได้ทำการสร้าง instance จาก heat ซึ่งดังกล่าวถูกสร้างจาก parameter ใน hot template ดังนี้

```

db_name:
  type: string
  description: Database name
  default: database1
  constraints:
    - length: { min: 1, max: 64 }
      description: db_name must be between 1 and 64 characters
    - allowed_pattern: '[a-zA-Z][a-zA-Z0-9]*'
      description: >
        db_name must begin with a letter and contain only alphanumeric characters

```

-db_username เป็นชื่อของ username ซึ่งดังกล่าวถูกสร้างจาก parameter ใน hot template ดังนี้

```

db_username:
  type: string
  description: The Database admin account username
  default: admin
  hidden: true
  constraints:
    - length: { min: 1, max: 16 }
      description: db_username must be between 1 and 64 characters
    - allowed_pattern: '[a-zA-Z][a-zA-Z0-9]*'
      description: >
        db_username must begin with a letter and contain only alphanumeric characters

```

-db_password เป็นช่องกรอกค่า password ของฐานข้อมูล ซึ่งดังกล่าวถูกสร้างจาก parameter ใน hot template ดังนี้

```

db_password:
  type: string
  description: The Database admin account password
  default: admin
  hidden: true
  constraints:
    - length: { min: 1, max: 41 }
      description: db_username must be between 1 and 64 characters
    - allowed_pattern: '[a-zA-Z0-9]*'
      description: db_password must contain only alphanumeric characters

```

-db_root_password เป็นช่องกรอกค่า password ของ root user ใน MariaDB ซึ่งดังกล่าวถูกสร้าง

จาก parameter ใน hot template ดังนี้

```
db_root_password:  
  type: string  
  description: Root password for MariaDB  
  default: admin  
  hidden: true  
  constraints:  
    - length: { min: 1, max: 41 }  
      description: db_username must be between 1 and 64 characters  
    - allowed_pattern: '[a-zA-Z0-9]*'  
      description: db_password must contain only alphanumeric characters
```

-flavor ขนาดของ flavor ที่จะใช้ใน stack ถูกสร้างจาก parameter ใน hot template ดังนี้

```
flavor:  
  type: string  
  description: Flavor to use for servers  
  default: m1.small  
  constraints:  
    - allowed_values: [m1.small, m1.medium]
```

-image ชื่อหรือหมายเลข UUID ของ Image ที่จะใช้ใน stack ถูกสร้างจาก parameter ใน hot template ดังนี้

```
image:  
  type: string  
  description: Name of image to use for servers  
  default: CentOS-7-x86_64  
  constraints:  
    - allowed_values: [CentOS-7.1-x86_64]  
      description: Image Name must be CentOS-7-x86_64
```

-Key name ชื่อ key ที่ใช้สำหรับ instance ที่สร้างจาก template ถูกสร้างจาก parameter ใน hot template ดังนี้

```
key_name:  
  type: string  
  description: Name of keypair to assign to servers
```

-public_net_id: ค่า id ของ public network

```
public_net_id:  
  type: string  
  description: ID of public network for which floating IP addresses will be allocated  
  default: c61fe76f-25ba-4fa6-a3f7-37d59fe17999
```

-private_net_id ค่า id ของ private network

```
private_net_id:  
  type: string  
  description: ID of private network into which servers get deployed  
  default: 07f908e4-8321-4d74-a2ed-5088f19c8bd6
```

-private_subnet_id ค่า id ของ private subnetwork

```
private_subnet_id:  
  type: string  
  description: ID of private sub network into which servers get deployed  
  default: e75af332-c726-420b-9c35-a69dc6b202ad
```

private_net_id ชื่อหรือหมายเลข UUID ของ network ที่จะใช้ใน stack ถูกสร้างจาก parameter ใน hot template ดังนี้

```
private_net_id:  
  type: string  
  description: ID of private network into which servers get deployed  
  default: db9e5ff1-47ed-4acc-bb36-40123e6a647f
```

private_subnet_id ชื่อหรือหมายเลข UUID ของ subnet network ที่จะใช้ใน stack ถูกสร้างจาก parameter ใน hot template ดังนี้

```
private_subnet_id:  
  type: string  
  description: ID of private sub network into which servers get deployed  
  default: cdbee95e-653b-4c82-a0d1-7915a1339e18
```

public_net_id ชื่อหรือหมายเลข UUID ของ public network ที่จะใช้ใน stack ซึ่งหมายเลข IP นี้จะเป็น floating ip สำหรับใช้ในการเข้าถึง instance ถูกสร้างจาก parameter ใน hot template ดังนี้

```
public_net_id:  
  type: string  
  description: ID of public network for which floating IP addresses will be allocated  
  default: 2f9bf474-606c-4037-a276-54f5fa23e898
```

การทดสอบ 3: สร้าง instance Autoscale multinode (แบบไม่มี load balance)

ตัวอย่าง Template ที่ใช้สำหรับอธิบายหลักการทำงานของ การ ทำงานของ heat template เพื่อควบคุม การ scale up และ scale down โดยการควบคุม จากโหลดการทำงานของ CPU ทำการ Clone git ด้านล่าง

```
git clone git@github.com:thaiopen/privatecloud.git
cd privatecloud
vi stack3.yaml
```

คำอธิบายรายละเอียดแต่ละส่วนของ template stack3.yaml

ในการทดสอบบทนี้ template จะมีโครงสร้างเหมือนกับในการทดสอบ 2 แต่จะเพิ่มส่วนของ outputs: เพิ่มเติม ดัง รูปด้านล่างเป็นการแสดงผล code แบบ fold

```
1  heat_template_version: 2015-04-30
2
3  description: >
4      This is a very simple template that illustrates automatic scaling up and down
5      using OS::Heat::AutoScalingGroup. CPU load is generated for
6      15 mins after each instance is spawned, triggering a scale-up event.
7      Once the max number of instances has been reached and the CPU
8      load generation has finished, the number of instances will be scaled
9      back down to 1.
10 > parameters: <>
11
12 > resources: <>
13
14 > outputs: <>
```

พิจารณา script ในแต่ละส่วน เริ่มจากส่วนของ parameters จะมีตัวแปรที่จะใช้ในการสร้าง instance 4 ค่า ได้แก่ key_name, flavor, image, public_net_id

```
10 > parameters:
11   key_name:
12     type: string
13     description: Keypair for authentication
14     default: cloudkey1
15   flavor:
16     type: string
17     description: Flavor to use for servers
18     default: m1.small
19   constraints:
20     - allowed_values: [m1.small, m1.medium]
21   image:
22     type: string
23     description: Name of image to use for servers
24     default: CentOS-7-x86_64
25   constraints:
26     - allowed_values: [CentOS-7-x86_64]
27     description: Image Name must be CentOS-7-x86_64
28   public_net_id:
29     type: string
30     label: Public network ID
31     description: ID of the public network to use
32     default: c61fe76f-25ba-4fa6-a3f7-37d59fe17999
```

resource ที่ต้องการสร้างใน การทดสอบนี้ มีอยู่ 6 resource ด้วยกัน คือ private_net, private_subnet, router, router_gw, router_interface และ cloud_config_stress การทดสอบในครั้งนี้เราจะให้ heat engine ทำหน้าที่ สร้าง private_network จาก type OS::Neutron::Net , จากนั้นจะนำเอา private_subnet จาก type OS::Neutron::Subnet (บรรทัดที่ 38) นอกจากนั้นการสร้าง private_subnet ยังสามารถกำหนดค่าของ cidr 192.168.10.0/24 dns_nameservers [8.8.8.8, 4.4.4.4] ดังรูปด้านล่าง และ private_network ยังนำไปสร้าง resource ชื่อ asg: จาก type OS::Heat::AutoScalingGroup (ในบรรทัดที่ 77)

```

34   resources:
35     private_network:
36       type: OS::Neutron::Net
37       properties:
38         admin_state_up: true
39         name: private_network
40     private_subnet:
41       type: OS::Neutron::Subnet
42       properties:
43         name: private_subnet
44         cidr: 192.168.10.0/24
45         dns_nameservers: [8.8.8.8, 8.8.4.4]
46         enable_dhcp: true
47         network_id: { get_resource: private_network }

```

Heat engine จะสร้าง router จาก type OS::Router แต่การสร้าง router จะต้องประกอบด้วย router gateway เชื่อมกับ public_net_id โดยสร้าง router gateway จาก type OS::RouterGateway และสร้าง router interface เชื่อมกับ private_subnet จาก type OS::RouterInterface

```

48   router:
49     type: OS::Neutron::Router
50     properties:
51       name: router
52       admin_state_up: true
53     router_gw:
54       type: OS::Neutron::RouterGateway
55       properties:
56         network_id: { get_param: public_net_id }
57         router_id: { get_resource: router }
58     router_interface:
59       type: OS::Neutron::RouterInterface
60       properties:
61         router_id: { get_resource: router }
62         subnet_id: { get_resource: private_subnet }
63

```

resource ตัวมาที่สร้างจะเป็น resource ที่ใช้สำหรับการสร้าง cloud-init เพื่อสร้าง script ให้ทำงานในขณะสร้าง instance โดยสร้างจาก type OS::Heat::CloudConfig และถูกนำไปใช้ user_data: get_resource

```

64   cloud_config_stress:
65     type: OS::Heat::CloudConfig
66     properties:
67       cloud_config:
68         runcmd:
69           - yum update -y
70           - rpm -ivh https://dl.fedoraproject.org/pub/epel/6Server/x86_64/stress-1.0.4-4.el6.x86_64.rpm
71           - stress --cpu 1 --timeout 10m
72
73   asg:
74     type: OS::Heat::AutoScalingGroup
75     depends_on: private_subnet
76     properties:
77       min_size: 1
78       max_size: 10
79     resource:
80       type: OS::Nova::Server
81       properties:
82         key_name: {get_param: key_name}
83         flavor: {get_param: flavor}
84         image: {get_param: image}
85       networks:
86         - network: { get_resource: private_network }
87         metadata: {"metering.stack": {get_param: "OS::stack_id"}}
88         user_data_format: RAW
89       user_data:
90         get_resource: cloud_config_stress
91

```

จาก script ด้านบน จะสร้าง instance ด้วย type OS::Nova::Server แต่การสร้างจะสร้างภายใต้ resource ชื่อ asg ที่สร้างจาก OS::Heat::AutoScalingGroup ทำให้ instance ที่สร้างมีคุณสมบัติ autoscale โดย resource asg นั้นจะถูกเรียกใช้งาน จากเงื่อนไข (policy) resource ชื่อ scale_up_policy, scale_down_policy ทั้งสอง resource สร้างจาก type OS::Heat::ScalingPolicy

```

92    scale_up_policy:
93      type: OS::Heat::ScalingPolicy
94      properties:
95        adjustment_type: change_in_capacity
96        auto_scaling_group_id: {get_resource: asg}
97        cooldown: 60
98        scaling_adjustment: 1
99    scale_down_policy:
100      type: OS::Heat::ScalingPolicy
101      properties:
102        adjustment_type: change_in_capacity
103        auto_scaling_group_id: {get_resource: asg}
104        cooldown: 60
105        scaling_adjustment: '-1'
106

```

ต่อมาจะเป็นการสร้าง resource ที่ทำหน้าที่สร้าง event เพื่อให้ไปรегистูนให้ policy ด้วย alarm_actions: โดย resource ที่ทำหน้าที่ตั้งนี้ สร้างด้วย type OS::Ceilometer::Alarm โดยให้สังเกตว่า Alarm นั้นจะไปทำหน้าที่ ตรวจจับค่า meter ชนิดที่ทำหน้าที่วัดค่าการใช้งาน cpu เรียกว่า cpu_util ในบรรทัดที่ระบุ meter_name: cpu_util ใน script มีการกำหนดให้วัดการทำงานดังนี้

cpu_alarm_high ถ้าหาก cpu มีการทำงานมากกว่า 50% นานกว่าระยะเวลา 1 นาที (period: 60 มีหน่วยเป็นวินาที)

cpu_alarm_low ถ้าหาก cpu มีการทำงานน้อยกว่า 15% เป็นระยะเวลา 10 นาที

```

107    cpu_alarm_high:
108      type: OS::Ceilometer::Alarm
109      properties:
110        description: Scale-up if the average CPU > 50% for 1 minute
111        meter_name: cpu_util
112        statistic: avg
113        period: 60
114        evaluation_periods: 1
115        threshold: 50
116      alarm_actions:
117        - {get_attr: [scale_up_policy, alarm_url]}
118        matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
119        comparison_operator: gt
120    cpu_alarm_low:
121      type: OS::Ceilometer::Alarm
122      properties:
123        description: Scale-down if the average CPU < 15% for 10 minutes
124        meter_name: cpu_util
125        statistic: avg
126        period: 600
127        evaluation_periods: 1
128        threshold: 15
129      alarm_actions:
130        - {get_attr: [scale_down_policy, alarm_url]}
131        matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
132        comparison_operator: lt

```

ส่วน output มีวัตถุประสงค์เพื่อจะแสดงผลลัพธ์ของการทำงาน ได้แก่ค่า scale_up_url, scale_dn_url และ server_list (ip ของ server ที่สร้างขึ้น)

```

134    outputs:
135      scale_up_url:
136        description: >
137          Webhook URL to scale up the group.
138          You can invoke the scale-up operation by doing an HTTP POST to this URL.
139          value: {get_attr: [scale_up_policy, alarm_url]}
140      scale_dn_url:
141        description: >
142          Webhook URL to scale down the group.
143          You can invoke the scale-down operation by doing an HTTP POST to this URL.
144          value: {get_attr: [scale_down_policy, alarm_url]}
145      server_list:
146        description: >
147          List of server names that are part of the group.
148          value: {get_attr: [asg, outputs_list, name]}

```

การดำเนินการทดสอบ

ก่อนหน้าทำการ upload ให้ทำการลบ private_network, private_subnet ก่อนจากการทดสอบ 2 เนื่องจากในบทนี้เราจะให้ heat engine เป็นคนสร้างให้เอง

The screenshot shows the 'Networks' page in the OpenStack dashboard. On the left, there's a sidebar with 'Project', 'Compute', 'Network' (selected), 'Network Topology', 'Routers', and 'Load Balancers'. The main area is titled 'Networks' with a table below it. The table has columns: Name, Subnets Associated, Shared, Status, Admin State, and Actions. Two rows are listed: 'private_network' (selected with a checked checkbox) and 'public_network'. The 'private_network' row shows 'private_subnet 192.168.32.0/24' under 'Subnets Associated'. The 'Actions' column for 'private_network' contains a red box around the 'Delete Network' button. A red arrow points from the text above to this button.

และจะต้องทำการลบ router ออกเช่นกัน

The screenshot shows the 'Routers' page in the OpenStack dashboard. On the left, there's a sidebar with 'Project', 'Compute', 'Network' (selected), 'Network Topology', and 'Routers'. The main area is titled 'Routers' with a table below it. The table has columns: Name, Status, External Network, Admin State, and Actions. One row is listed: 'router1' (selected with a checked checkbox). The 'router1' row shows 'Active' under 'Status' and 'public_network' under 'External Network'. The 'Actions' column for 'router1' contains a red box around the 'Delete Router' button. A red arrow points from the text above to this button.

ต่อมา ให้แก้ไขค่าของ public_net_id

```
[root@cloud ~]# neutron net-list
+-----+-----+-----+
| id      | name    | subnets |
+-----+-----+-----+
| c61fe76f-25ba-4fa6-a3f7-37d59fe17999 | public_network | 26291621-6ade-4a3a-bc35-d2606037fcfd 103.27.200.128/25 |
+-----+-----+-----+
```

เมื่อดำเนินการเรียบร้อยก็ให้ดำเนินการ upload ตัวซึ่ง stack เป็น lab3 พร้อมกับใส่ รหัสของ user

Launch Stack

Stack Name * lab3

Creation Timeout (minutes) * 60

Rollback On Failure

Password for user "demo" * [REDACTED]

flavor m1.small

image CentOS-7-x86_64

key_name cloudkey1

Public network ID c61fe76f-25ba-4fa6-a3f7-37d59fe17999

Description:
Create a new stack with the provided values.

Cancel **Launch**

หลังจาก กรอกข้อมูล เรียบร้อย ให้กด launch รอสักครู่ ระหว่าง กระบวนการสร้าง และให้ตรวจสอบการทำงานผ่านทาง log ที่

```
# tail -f /var/log/heat/heat-engine.log
```

จะสามารถตรวจสอบการทำงานผ่านทาง log จะเห็นการทำงานจนสิ้นสุด

```

[root@cloud ~(keystone_demo)]# tail -f /var/log/heat/heat-engine.log
2015-12-04 01:51:53.583 20471 INFO heat.engine.resource [req-aee5ca40-2a07-4043-9758-047606f74a73 demo demo] Validating CeilometerAlarm "cpu_alarm_high"
2015-12-04 01:51:53.776 20471 INFO heat.engine.stack [-] Stack CREATE_IN_PROGRESS (lab3): Stack CREATE started
2015-12-04 01:51:53.779 20471 INFO heat.engine.resource [-] creating CloudConfig "cloud_config_stress" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:51:53.814 20471 INFO heat.engine.resource [-] creating Router "router" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:51:53.854 20471 INFO heat.engine.resource [-] creating Net "private_network" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:51:53.972 20471 INFO heat.engine.resource [-] creating Subnet "private_subnet" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:51:55.079 20471 INFO heat.engine.resource [-] creating RouterInterface "router_interface" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:51:55.250 20469 INFO heat.engine.service [req-aee5ca40-2a07-4043-9758-047606f74a73 demo demo] Creating stack lab3-asg-k55g4hkmr3hi
2015-12-04 01:51:55.251 20469 INFO heat.engine.environment [req-aee5ca40-2a07-4043-9758-047606f74a73 demo demo] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:51:55.265 20469 INFO heat.engine.resource [req-aee5ca40-2a07-4043-9758-047606f74a73 demo demo] Validating Server "u2tuh5zhsm3q"
2015-12-04 01:51:55.484 20469 INFO heat.engine.stack [-] Stack CREATE_IN_PROGRESS (lab3-asg-k55g4hkmr3hi): Stack CREATE started
2015-12-04 01:51:55.487 20469 INFO heat.engine.resource [-] creating Server "u2tuh5zhsm3q" Stack "lab3-asg-k55g4hkmr3hi" [9af5f8dab-6b5b-41ee-8071-db6c008c559b]
2015-12-04 01:51:55.934 20468 INFO heat.engine.service [req-227fb012-c211-4880-977f-1c519949b00d] [-] Service 08ab0adact-20c2-4c89-914d-5823b203042c is updated
2015-12-04 01:51:55.934 20471 INFO heat.engine.service [req-3667fd40-d74e-4034-a13e-fc31886e2d19] [-] Service f53c1238-40fe-4b0e-98b7-7d626385cd89 is updated
2015-12-04 01:51:55.939 20465 INFO heat.engine.service [req-969e4d08-02b9-49cd-8331-4b34fb02c339] [-] Service 1fc37371-62fb-423d-9a94-f60d6eb07a01 is updated
2015-12-04 01:51:55.939 20469 INFO heat.engine.service [req-9bc24a1a-149b-4699-aef0-991818c7c45] [-] Service 384a4fa7-5ef00-4f12-82c5-3db0117478a3 is updated
2015-12-04 01:51:55.939 20469 INFO heat.engine.service [req-9bc24a1a-149b-4699-aef0-991818c7c45] [-] Service 38804d18-5e04-4ce8-ac67-ba0dc78b3d34 is updated
2015-12-04 01:51:55.942 20467 INFO heat.engine.service [req-1f124300-73b3-4205-99f8-4f35ec50747b] [-] Service 1b1f35cb-f72a-49b7-a484-3130c814673f is updated
2015-12-04 01:51:55.942 20472 INFO heat.engine.service [req-00dc2a2a-fe04-46e8-a51f-5b5effb2140a] [-] Service d8b7139d-3fee-4c62-bbea-8fa53f04e176 is updated
2015-12-04 01:51:55.971 20473 INFO heat.engine.service [req-8a14621a-ba0a-4ea9-81ff-9d12bc8e8c1c4] [-] Service 3c127362-adc7-4716-b786-c1dc1ed0b6b5 is updated
2015-12-04 01:51:56.265 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:51:56.514 20471 INFO heat.engine.resource [-] creating RouterGateway "router_gw" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:51:57.832 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:51:58.821 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:51:59.830 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:00.838 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:01.847 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:02.859 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:03.868 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:04.876 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:05.885 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:06.893 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:07.840 20469 INFO heat.engine.stack [-] Stack CREATE_COMPLETE (lab3-asg-k55g4hkmr3hi): Stack CREATE completed successfully
2015-12-04 01:52:07.966 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:07.966 20471 INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
2015-12-04 01:52:08.024 20471 INFO heat.engine.resource [-] creating AutoscalingPolicy "scale_up_policy" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:52:09.132 20471 INFO heat.engine.resource [-] creating AutoscalingPolicy "scale_down_policy" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:52:09.176 20471 INFO heat.engine.resource [-] creating CeilometerAlarm "cpu_alarm_high" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:52:09.176 20471 INFO heat.engine.resource [-] creating CeilometerAlarm "cpu_alarm_low" Stack "lab3" [6cc7e02f-c28d-4861-bc89-b9a815eafc55]
2015-12-04 01:52:10.240 20471 INFO heat.engine.stack [-] Stack CREATE_COMPLETE (lab3): Stack CREATE completed successfully

```

เมื่อเสร็จเรียบร้อยจะเข้าสู่ complete

<input type="checkbox"/>	Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/>	lab3	0 minutes	Never	Create Complete	<button>Check Stack</button>

เมื่อตรวจสอบในส่วนของ horizon ในเมนู instance จะพบว่ามีการสร้าง instance

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	lab3-asg-k55g4hkmr3hi-u2tuh5zhsm3g-elyencwc2mus	CentOS-7-x86_64	192.168.10.3	m1.small	cloudkey1	Active	nova	None	Running	3 minutes	<button>Create Snapshot</button>

ตรวจสอบ log ของ instance

openstack

demo

Project

Compute

Overview

Instances

Volumes

Images

Access & Security

Network

Object Store

Orchestration

Database

Data Processing

Identity

demo

Instance Details: lab3-asg-k55g4hkmr3hi-u2tuh5zhsm3g-elyencwc2mus

Create Snapshot

Overview Log Console Action Log

Log Length: 35 Go View Full Log

Instance Console Log

```

cloud-init[3603]: Verifying : nss-tools-3.19.1-5.el7_1.x86_64 23/30
cloud-init[3603]: Verifying : nspr-4.10.8-1.el7_1.x86_64 24/30
cloud-init[3603]: Verifying : nss-3.19.1-5.el7_1.x86_64 25/30
cloud-init[3603]: Verifying : libnl3-cli-3.2.21-8.el7.x86_64 26/30
cloud-init[3603]: Verifying : kernel-tools-3.10.0-229.14.1.el7.x86_64 27/30
cloud-init[3603]: Verifying : nss-sysinit-3.19.1-5.el7_1.x86_64 28/30
cloud-init[3603]: Verifying : rsyslog-7.4.7-7.el7_0.x86_64 29/30
cloud-init[3603]: Verifying : libnl3-3.2.21-8.el7.x86_64 30/30
cloud-init[3603]: Installed:
cloud-init[3603]: kernel.x86_64 0:3.10.0-229.20.1.el7
cloud-init[3603]: Dependency Installed:
cloud-init[3603]: linux-firmware.noarch 0:20140911-0.1.git365e80c.el7
cloud-init[3603]: Updated:
cloud-init[3603]: gmp.x86_64 1:6.0.0-12.el7_1
cloud-init[3603]: kernel-tools.x86_64 0:3.10.0-229.20.1.el7
cloud-init[3603]: kernel-tools-libs.x86_64 0:3.10.0-229.20.1.el7
cloud-init[3603]: kpartx.x86_64 0:0.4.9-77.el7_1
cloud-init[3603]: libnl3.x86_64 0:3.2.21-9.el7_1
cloud-init[3603]: libnl3-cli.x86_64 0:3.2.21-9.el7_1
cloud-init[3603]: nspr.x86_64 0:4.10.8-2.el7_1
cloud-init[3603]: nss.x86_64 0:3.19.1-7.el7_1.2
cloud-init[3603]: nss-sysinit.x86_64 0:3.19.1-7.el7_1.2
cloud-init[3603]: nss-tools.x86_64 0:3.19.1-7.el7_1
cloud-init[3603]: nss-util.x86_64 0:3.19.1-4.el7_1
cloud-init[3603]: rsyslog.x86_64 0:7.4.7-7.el7_1
cloud-init[3603]: selinux-policy.noarch 0:3.13.1-23.el7_1.21
cloud-init[3603]: selinux-policy-targeted.noarch 0:3.13.1-23.el7_1.21
cloud-init[3603]: Complete!
cloud-init[3603]: warning: /var/tmp/rpm-tmp.KSAGBM: Header V3 RSA/SHA256 Signature, key ID 0608b895: NOKEY
cloud-init[3603]: Retrieving https://dl.fedoraproject.org/pub/epel/6Server/x86_64/stress-1.0.4-4.el6.x86_64.rpm
cloud-init[3603]: Preparing...
cloud-init[3603]: Updating / installing...
cloud-init[3603]: stress-1.0.4-4.el6 #####
cloud-init[3603]: stress: info: [20572] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd

```

ขั้นตอนเพิ่ม float ip ให้แก่ instance

Instances

		Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	lab3-asg-k55g4hkrmr3hi-u2tuh5zhsm3g-elyencwc2mus	CentOS-7-x86_64	192.168.10.3	m1.small	cloudkey1	Active	nova	None	Running	5 minutes	<button>Create Snapshot</button>	<button>Associate Floating IP</button>
Displaying 1 item												<button>Disassociate Floating IP</button>

เลือก

demo ▾

Manage Floating IP Associations

IP Address *

IP Address *
103.27.200.241

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

lab3-asg-k55g4hkrmr3hi-u2tuh5zhsm3g-elyencwc2n

Cancel Associate

ตรวจสอบ floating ip ที่สร้าง

Instances

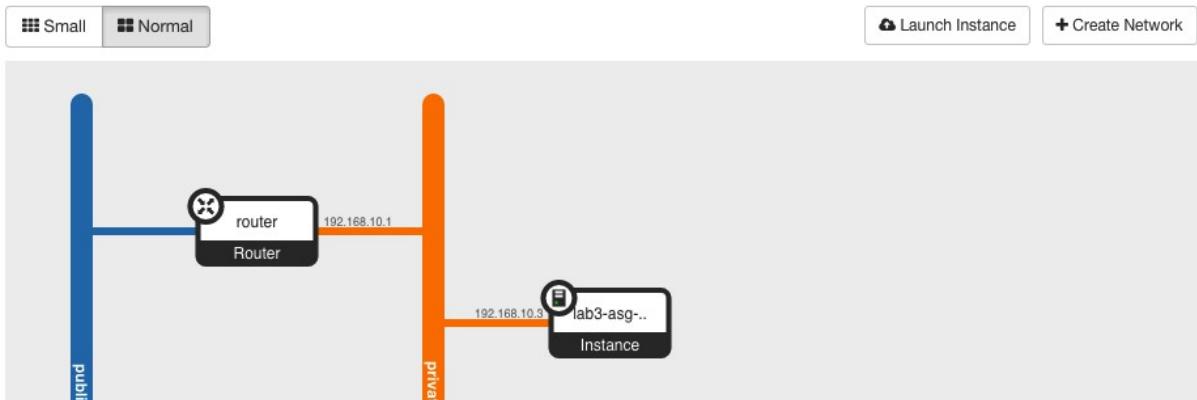
		Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	lab3-asg-k55g4hkrmr3hi-u2tuh5zhsm3g-elyencwc2mus	CentOS-7-x86_64	192.168.10.3	Floating IPs: 103.27.200.241	m1.small	cloudkey1	Active	nova	None	Running	7 minutes	<button>Create Snapshot</button>

Heat engine สร้าง private network ให้

```
[root@cloud ~(keystone_demo)]# neutron net-list
+-----+-----+
| id | name | subnets |
+-----+-----+
| c61fe76f-25ba-4faf-a3f7-37d59fe17999 | public_network | 26291621-6ade-4a3a-bc35-d2606037fcfd 103.27.200.128/25 |
| 4f53028d-73cd-4d62-9500-37d4186e99a9 | private_network | ab79cf10-6fbf-49f2-8582-47e846049fb6 192.168.10.0/24 |
+-----+-----+
```

ตรวจสอบ network topology

Network Topology



ให้ทำการ shell เข้าไปยังเครื่อง instance โดยการใช้งาน key

```
[root@cloud ~(keystone_demo)]# ssh -i cloudkey1.pem centos@103.27.200.241
The authenticity of host '103.27.200.241 (103.27.200.241)' can't be established.
ECDSA key fingerprint is 2d:a5:6c:ff:5e:62:d6:f6:5f:a5:f2:98:60:6d:2f:af.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '103.27.200.241' (ECDSA) to the list of known hosts.
[centos@lab3-asg-k55g4hkmr3hi-u2tuh5zhsm3g-elyencwc2mus ~]$
```

ตรวจสอบการทำงาน CPU

```
[centos@lab3-asg-k55g4hkmr3hi-u2tuh5zhsm3g-elyencwc2mus ~]$ top
```

จะเป็นว่า cpu จะถูกใช้งาน

```
top - 19:01:58 up 9 min, 1 user, load average: 1.00, 0.91, 0.52
Tasks: 77 total, 3 running, 74 sleeping, 0 stopped, 0 zombie
%Cpu(s):100.0 us, 0.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Kib Mem : 1864544 total, 1191304 free, 89316 used, 603924 buff/cache
Kib Swap: 0 total, 0 free, 0 used. 1620108 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR %CPU %MEM     TIME+ COMMAND
20573 root      20   0    7256    80    0 R 99.9 0.0 0:25.46 stress
  1 root      20   0  56748  6580  3896 S 0.0 0.3 0:00.90 systemd
  2 root      20   0      0    0 S 0.0 0.0 0:00.00 kthreadd
  3 root      20   0      0    0 S 0.0 0.0 0:00.00 ksftirqd/0
  5 root      0 -20      0    0 S 0.0 0.0 0:00.00 kworker/0:0H
  7 root      rt  0      0    0 S 0.0 0.0 0:00.00 migration/0
  8 root      20   0      0    0 S 0.0 0.0 0:00.00 rcu_bh
  9 root      20   0      0    0 S 0.0 0.0 0:00.00 rcuob/0
 10 root      20   0      0    0 S 0.0 0.0 0:00.10 rcu_sched
 11 root      20   0      0    0 R 0.0 0.0 0:00.11 rcuos/0
```

ใน Heat template มีการระบุให้มีการสร้าง cpu load เพื่อสร้าง alert โดยจะอยู่ในส่วนของ cloud_config_stress ข้างล่าง

```
64  cloud_config_stress:
65    type: OS::Heat::CloudConfig
66    properties:
67      cloud_config:
68        runcmd:
69          - yum update -y
70          - rpm -ivh https://dl.fedoraproject.org/pub/epel/6Server/x86_64/stress-1.0.4-4.el6.x86_64.rpm
71          - stress --cpu 1 --timeout 10m
72
```

คำสั่งที่จะทำหลังจากมีการสร้าง instance

```
yum update -y  
rpm -ivh https://dl.fedoraproject.org/pub/epel/6Server/x86_64/stress-1.0.4-  
4.el6.x86_64.rpm  
stress --cpu 1 --timeout 10m
```

สรุปผลการสร้าง stack lab3 ได้ที่ horizon ในเมนู stack overview

Stack Details: lab3

Topology Overview Resources Events Template

Stack Overview

Information

Name	lab3
ID	6cc7e02f-c28d-4861-bc89-b9a815eafc55
Description	This is a very simple template that illustrates automatic scaling up and down using OS::Heat::AutoScalingGroup. CPU load is generated for 15 mins after each instance is spawned, triggering a scale-up event. Once the max number of instances has been reached and the CPU load generation has finished, the number of instances will be scaled back down to 1.

Status

Created	36 minutes
Last Updated	Never
Status	Create_Complete: Stack CREATE completed successfully

Outputs

scale_dn_url	Webhook URL to scale down the group. You can invoke the scale-down operation by doing an HTTP POST to this URL. http://192.168.1.2:8000/v1/signal/arm%3Aopenstack%3Aheat%3A%3Adaf5f5ded99a7a74b6896647d11126dd7bd%3Astacks%2Flab3%2F6cc7e02f-c28d-4861-bc89-b9a815eafc55%2Fresources%2Fscale_down_policy?Timestamp=2015-12-03T18%3A51%3A53Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=2e2434c9eed4a009486c1edca7118fd&SignatureVersion=2&Signature=M7Ky3Zw
scale_up_url	Webhook URL to scale up the group. You can invoke the scale-up operation by doing an HTTP POST to this URL. http://192.168.1.2:8000/v1/signal/arm%3Aopenstack%3Aheat%3A%3Adaf5f5ded99a7a74b6896647d11126dd7bd%3Astacks%2Flab3%2F6cc7e02f-c28d-4861-bc89-b9a815eafc55%2Fresources%2Fscale_up_policy?Timestamp=2015-12-03T18%3A51%3A53Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=f9bd2cde0854576998c216c2e56ffbb&SignatureVersion=2&Signature=TvhW7r67D
server_list	List of server names that are part of the group. ["lab3-asg-k55g4hkmr3hi-7mhratomjqat-desvnabfzmyc"]

Stack Parameters

OS::project_id	d5f5ded99a7a74b6896647d11126dd7bd
OS::stack_id	6cc7e02f-c28d-4861-bc89-b9a815eafc55
OS::stack_name	lab3
key_name	cloudkey1
image	CentOS-7-x86_64

ให้รอผลของการทำงานของ คำสั่ง stress ซักครู่ เพื่อรอให้ เกิดการ scale up policy ที่เกิดจาก cpu load จากนั้น ให้เข้าไปดูที่ tab ของ Instances จะพบว่ามี instance ใหม่ๆ สร้างขึ้นมา

Instances

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	lab3-asg-k55g4hkrm3hi-7mhratomjqat-desvnabfzmyc	CentOS-7-x86_64	192.168.10.4	m1.small	cloudkey1	Active	nova	None	Running	2 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	lab3-asg-k55g4hkrm3hi-u2tuh5zhsm3g-elyencwc2mus	CentOS-7-x86_64	192.168.10.3 Floating IPs: 103.27.200.241	m1.small	cloudkey1	Active	nova	None	Running	17 minutes	<button>Create Snapshot</button>

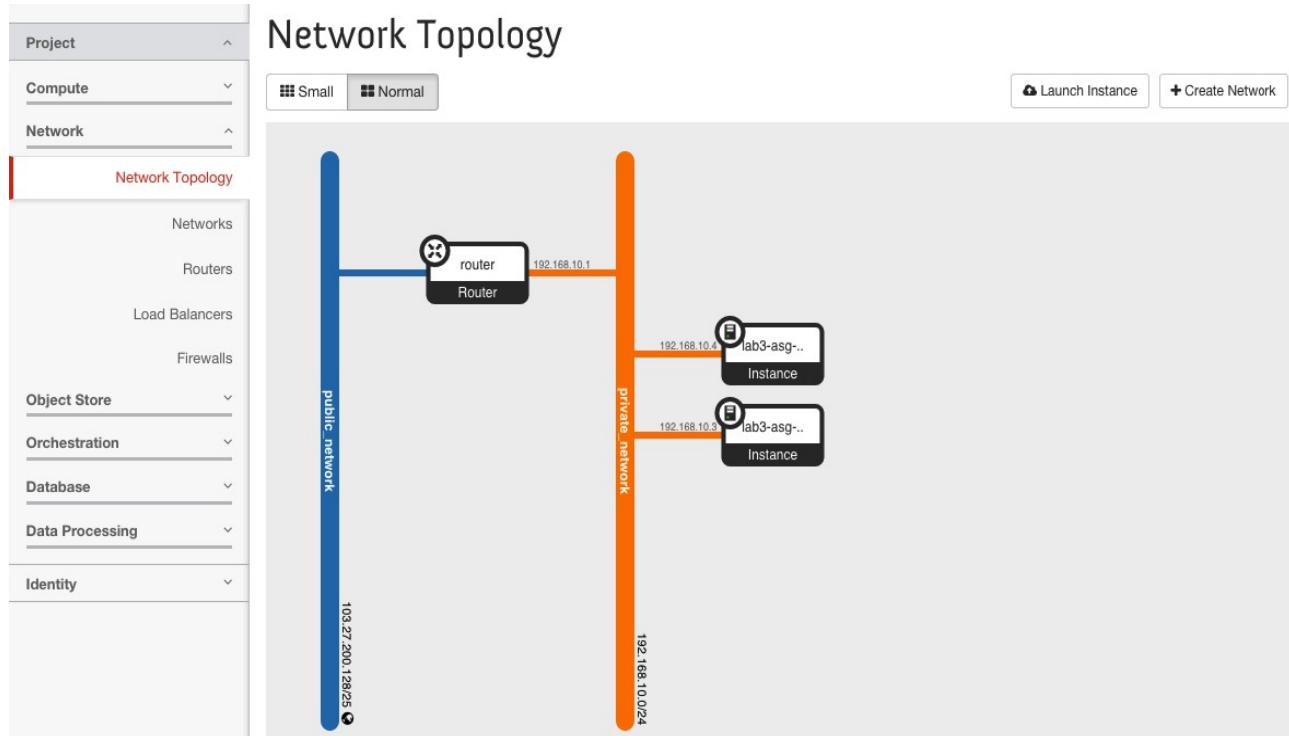
Displaying 2 items

เมื่อตรวจสอบ log ของ alarm จะพบมีเกิด Notifying alarm lab3-cpu_alarm_high

```
# tail -f /var/log/ceilometer/alarm-notifier.log
```

```
[root@cloud ~(keystone demo)]# tail -f /var/log/ceilometer/alarm-notifier.log
2015-12-01 02:03:38.125 5616 INFO oslo.messaging._drivers.impl_rabbit [-] Connecting to AMQP server on 192.168.1.2:5672
2015-12-01 02:03:38.237 5616 INFO oslo.messaging._drivers.impl_rabbit [-] Connected to AMQP server on 192.168.1.2:5672
2015-12-01 03:02:58.728 5616 INFO ceilometer.openstack.common.service [-] Caught SIGTERM, exiting
2015-12-01 03:07:25.300 1372 INFO oslo.messaging._drivers.impl_rabbit [-] Connecting to AMQP server on 192.168.1.2:5672
2015-12-01 03:07:25.336 1372 INFO oslo.messaging._drivers.impl_rabbit [-] Connected to AMQP server on 192.168.1.2:5672
2015-12-02 23:36:48.016 1372 INFO ceilometer.openstack.common.service [-] Caught SIGTERM, exiting
2015-12-02 23:36:50.640 19710 INFO oslo.messaging._drivers.impl_rabbit [-] Connecting to AMQP server on 192.168.1.2:5672
2015-12-02 23:36:50.783 19710 INFO oslo.messaging._drivers.impl_rabbit [-] Connected to AMQP server on 192.168.1.2:5672
2015-12-04 02:07:01.026 19710 INFO ceilometer.alarm.notifier.rest [-] Notifying alarm lab3-cpu_alarm_high-jagbqca7126f 4d6279ae-eb65-439e-b8ac-92c63092b4bf from insufficient data to alarm with action SplitResult(scheme=u'http', netloc=u'192.168.1.2:8000', path=u'/v1/signal/arm%3Aopenstack%3Aheat%3A%3Adf5ded99a7a74b6896647d11126dd7bd%3astacks%2flab3%2F6cc7e02f-c28d-4861-bc89-b9a815eafc55%2Fresources%2Fscale_up_policy', query=u'Timestamp=2015-12-03T18%3A51%3A53Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=f79bd2cd0e0854576998c216c2e56fbba&SignatureVersion=2&Signature=TVhW7r67DNExw4AVX3rzPw3rxrXFawm9Jlu1xWhfss%3D', fragment='') because Transition to alarm due to 1 samples outside threshold, most recent: 66.7866666667. request-id: req-38abc644-20bc-4e1f-bf3d-7810375fdef2
2015-12-04 02:08:01.006 19710 INFO ceilometer.alarm.notifier.rest [-] Notifying alarm lab3-cpu_alarm_high-jagbqca7126f 4d6279ae-eb65-439e-b8ac-92c63092b4bf from alarm to alarm with action SplitResult(scheme=u'http', netloc=u'192.168.1.2:8000', path=u'/v1/signal/arm%3Aopenstack%3Aheat%3A%3Adf5ded99a7a74b6896647d11126d7bd%3astacks%2flab3%2F6cc7e02f-c28d-4861-bc89-b9a815eafc55%2Fresources%2Fscale_up_policy', query=u'Timestamp=2015-12-03T18%3A51%3A53Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=fff9bd2cd0e0854576998c216c2e56fbba&SignatureVersion=2&Signature=TVhW7r67DNExw4AVX3rzPw3rxrXFawm9Jlu1xWhfss%3D', fragment='') because Remaining as alarm due to 1 samples outside threshold, most recent: 66.7866666667. request-id: req-38113729-a859-4ce5-9d71-ece3ebc4008b
■
```

ตรวจสอบที่ Network Topology



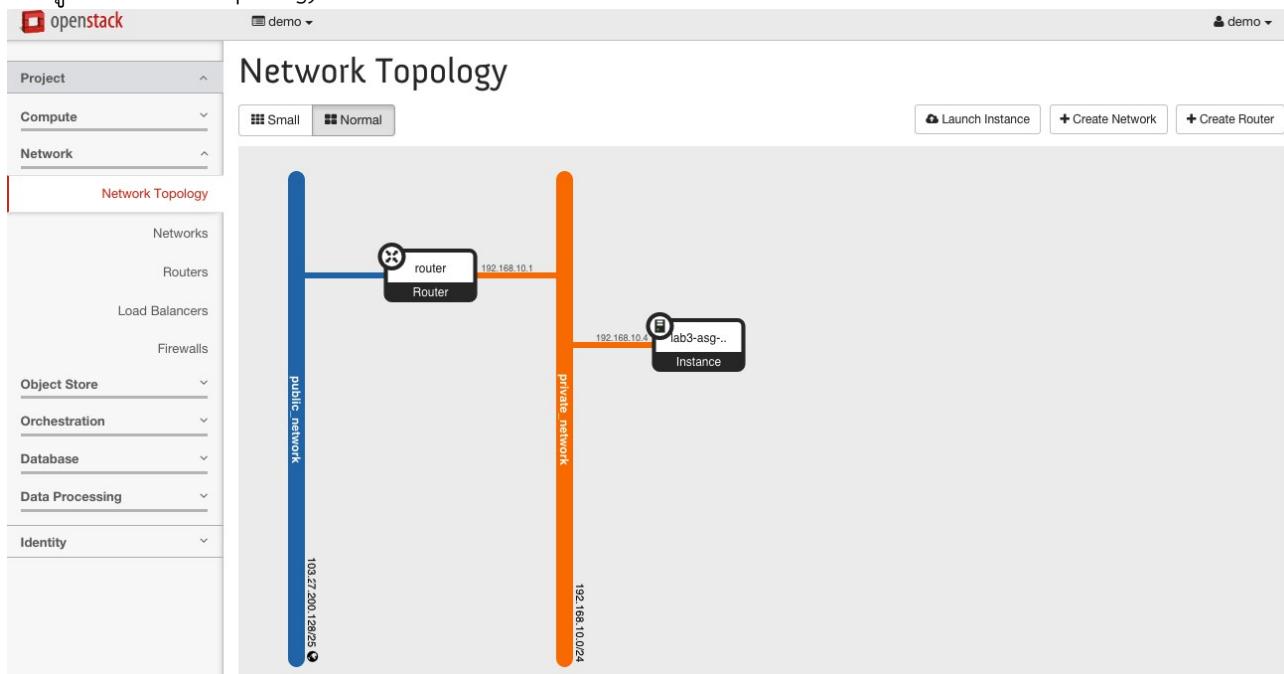
เมื่อครบเวลา การ load cpu จะเกิด notification Notifying alarm lab3-cpu_alarm_low

```
# tail -f /var/log/ceilometer/alarm-notifier.log
```

จะสังเกตเห็นได้ว่ามีการ เปลี่ยนสถานะจาก alarm-high > alarm-low

```
INFO ceilometer.alarm.notifier.rest [-] Notifying alarm lab3-cpu_alarm_high-jagbqca7i26f 4d6279ae-eb65-439e-b8ac  
ult(scheme='http', netloc='192.168.1.2:8000', path='/v1/signals/arn%3Aopenstack%3Aheat%3A%3Adf5ded99a7a74b6896  
:89-b9a815eafc55%2Fscale_up_policy', query='Timestamp=2015-12-03T18%3A51%3A53Z&SignatureMethod=HmacSHA  
SignatureVersion=2&Signature=TVW7767NEXW4AVX3rzPw3rXrXFawm9JUlxWhfSS%3D', fragment='') because Remaining as  
: 66.786666666667. request-id: req-38113729-a259-4ce5-9d71-ece3ebc0008h  
INFO ceilometer.alarm.notifier.rest [-] Notifying alarm lab3-cpu_alarm_low-rustje2z2xf bb835e95-55c7-41a8-a823-  
scheme='http', netloc='192.168.1.2:8000', path='/v1/signals/arn%3Aopenstack%3Aheat%3A%3Adf5ded99a7a74b6896647d  
:9a815eafc55%2Fresources%2Fscale_down_policy', query='Timestamp=2015-12-03T18%3A51%3A53Z&SignatureMethod=HmacSHA  
SignatureVersion=2&Signature=M7Ky3ZwWhJ3%2BWUzbFOLcxn99QvcYYvh%2BZ11jyThWSI%3D', fragment='') because Transition  
nt: 0.035. request-id: req-26d60937-8d3d-4cf4-a540-84613ed91cf6  
INFO ceilometer.alarm.notifier.rest [-] Notifying alarm lab3-cpu_alarm_low-rustje2z2xf bb835e95-55c7-41a8-a823-
```

เมื่อดูที่ network topology อีกครั้ง



ตรวจสอบ ใน horizon instance

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	lab3-asg-k55g4hkmr3hi-7mrat0mjqat-desvnabfzmyc	CentOS-7-x86_64	192.168.10.4	m1.small	cloudkey1	Active	nova	None	Running	15 minutes	<button>Create Snapshot</button>

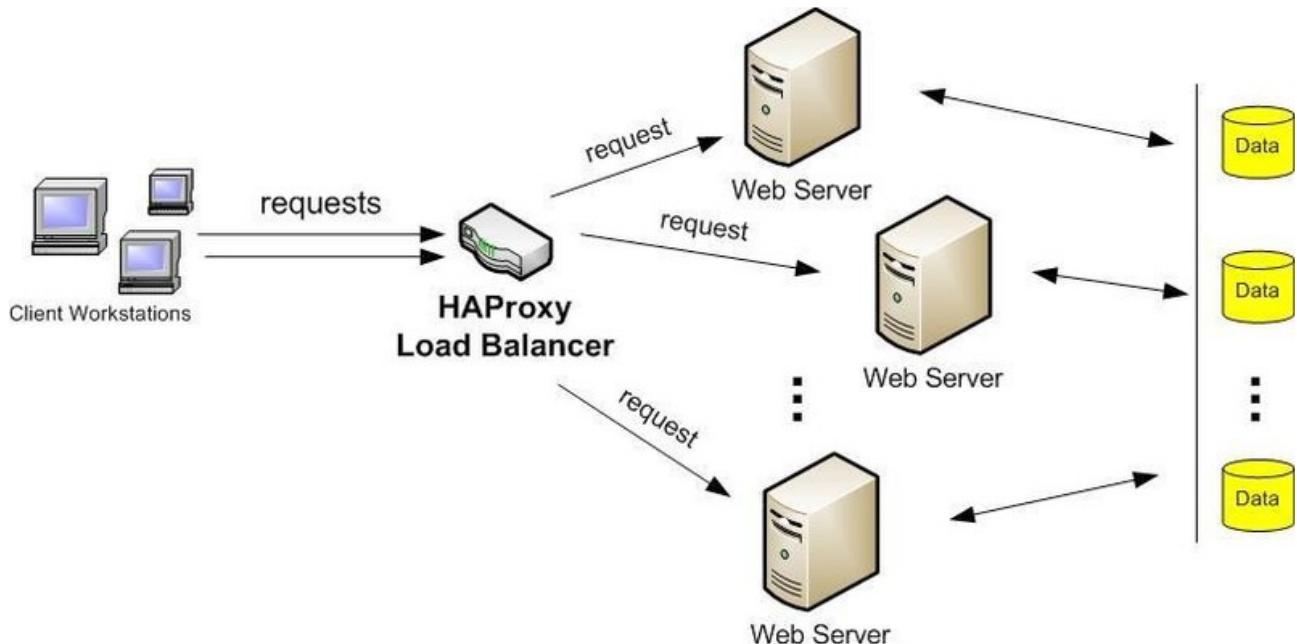
พื้นฐาน การทำงานของ service ชื่อ LBaaS (Load-Balancing-as-a-Service)

LBaaS เป็น service ที่เป็นส่วนเสริมของ Neutron เพื่อรับความสามารถของ Loadbalance โดยพื้นฐานที่สำคัญเพื่อที่จะนำไปสู่การเข้าใจ LBaaS ใน Neutron มีดังนี้

- Virtual IPs (VIP) Pool และ Member
- Load balance algorithm
- Monitor
- Persistence
- network กับ load balance

สำหรับ LBaaS นั้นสามารถใช้ back-end technology เป็น software load balance หรือ hardware load balance โดย default จะใช้ software load balance ที่ชื่อว่า HAProxy

HAProxy เป็น OpenSource Software สำหรับการทำ TCP/HTTP load balance และสามารถใช้งานทั้งใน Linux Solaris และ FreeBSD



สำหรับ load balance ที่ทำงานอยู่ใน Neutron ประกอบไปด้วย 3 ส่วนได้แก่

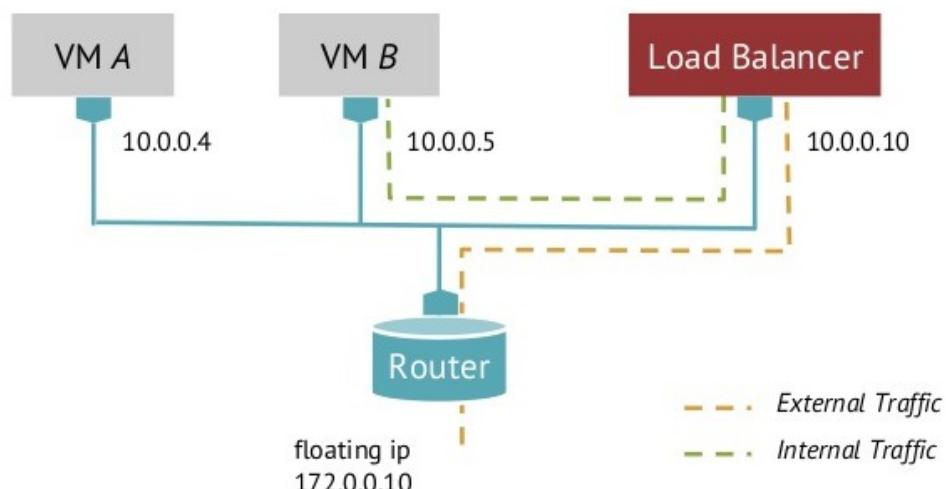
Pool Member คือ instance ใดก็ register มาเพื่อที่จะใช้งานใน pool

Pool คือ กลุ่มของ instance ที่ใช้สำหรับการรับ load เมื่อ traffic วิ่งมาที่ตัว load balance

Virtual IP คือ เป็นหมายเลข IP ที่ตั้งค่าอยู่ที่ load balance เพื่อใช้สำหรับการรับ connection จาก client โดยทั่วไปจะใช้สำหรับการ mapping กับ name

การบวนการกระจายโหลด Connection ด้วย Round robin algorithms

สำหรับ LBaaS ใน OpenStack อาศัยการทำงานของ HA Proxy ดังนี้ได้ก่อความแล้วข้างตัน ทุกรอบที่มี connection ใหม่ เข้ามาที่ load balancer ดังรูปด้านล่าง load balancer จะทำหน้าที่กระจาย connection เหล่านั้นไปยัง instance ที่เป็นสมาชิกของ pool โดยแต่ละ instance จะมี ip แตกต่างกัน การกระจาย connection จะเป็นการกระจาย (Distribute) ไปยัง instance อย่างเท่าเทียมกัน หลักการการกระจายที่คำนึงถึงความเท่ากันของปริมาณการใช้งาน resource (resource-intensive) ได้แก่ ความเร็วในการ process (processing speed) ความเร็วในการเชื่อมต่อ (connection speed) และการใช้งานหน่วยความจำ (memory) เรียกหลักการนี้ Round Robin Algorithms



จากรูปด้านบนเมื่อมี connection เข้าเข้าที่ Router นั้น traffice จะถูกส่งไปยัง Load Balancer ก่อนที่จะถูกส่งต่อไปยัง instance อีกต่อหนึ่ง

การทดสอบ 4: สร้าง instance Autoscale multinode (แบบมี load balance)

อย่างที่ได้กล่าวไปในหัวข้อก่อนหน้าที่เกี่ยวข้องการทำ autoscale เพียงแค่ server (instance) นอกจากนี้เรายังสามารถทำ scale ในระดับ application โดยทำงานร่วมกับ Load Balancer As-a-Service (LBAAS) ซึ่ง backend technology ที่ใช้นั้นคือ ha-proxy โดยมีลำดับขั้นตอนการสร้างดังนี้

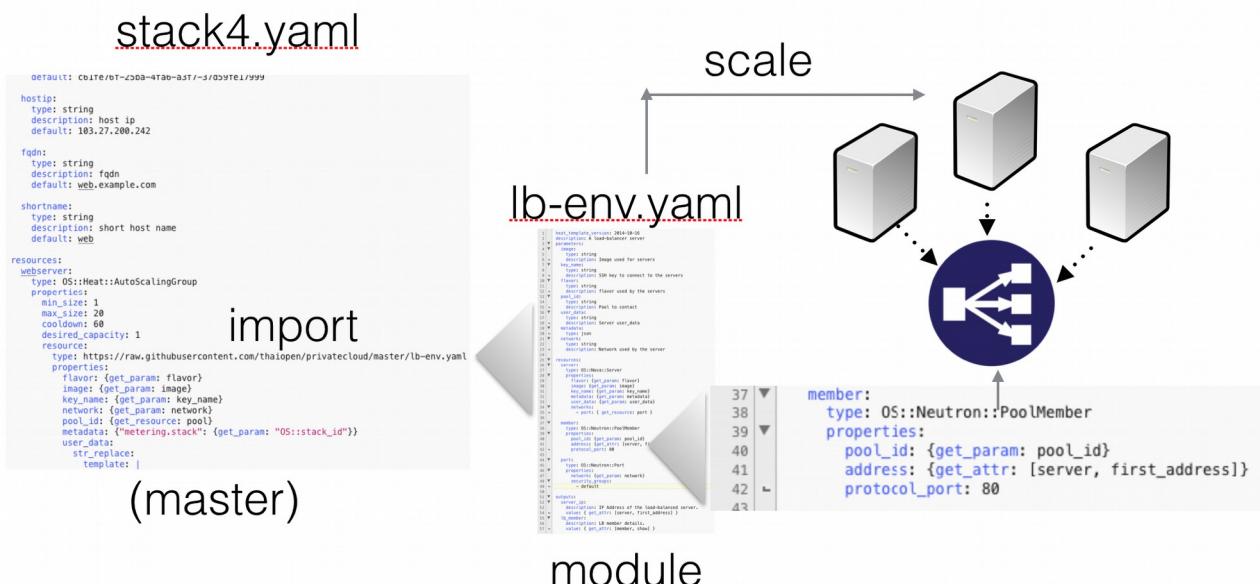
ทำการ Clone git ด้านล่าง

```
git clone git@github.com:thaiopen/privatecloud.git
cd privatecloud
vi stack4.yaml
```

การสร้าง Heat Stack Environment การทดสอบ 4

ตาม Specification รองรับการอ้างอิงจาก template หลัก หรือ master template ไปยัง template อื่นๆ ทำให้สามารถที่จะแบ่ง Template ใหญ่ให้กลายเป็น module ย่อยๆ และสามารถนำกลับมาใช้ใหม่ได้ เรียกว่า คุณสมบัตินี้ว่า nested template ทำให้สามารถบริหารจัดการ Template ได้อย่างมีประสิทธิภาพและรักษาความถูกต้องได้ แต่ละส่วนที่มีการแยกออกมานั้นก็มีโครงสร้างเหมือนกับ Template เพียงแต่จะถูกเรียกไปใช้งานจาก Template หลัก จะเรียก module เหล่านั้นว่า Heat Stack Environment

ในการทดสอบนี้ จะสร้าง Heat Stack Environment เพื่อใช้สำหรับสร้าง instance ที่จะเข้าไปสร้างใน member pool ของ cluster โดยทั่วไปประกอบไปด้วยความสัมพันธ์ของที่ใช้สำหรับการสร้าง instance กับ loadbalance



คำอธิบายรายละเอียดแต่ละส่วนของ template lb-env.yaml

ทำหน้าที่สร้าง server ให้กับ loadbalance ประกอบด้วย ส่วนต่างๆ ดังรูปเหมือนกันการทดสอบที่ผ่านมา

```
1  heat_template_version: 2014-10-16
2  description: A load-balancer server
3  parameters: ...
4
5  resources: ...
6
7  outputs: ... }
```

ส่วน parameters ประกอบด้วย image, key_name, flavor, pool_id, user_data, metadata, network

```
3  parameters:
4    image:
5      type: string
6      description: Image used for servers
7    key_name:
8      type: string
9      description: SSH key to connect to the servers
10   flavor:
11     type: string
12     description: flavor used by the servers
13   pool_id:
14     type: string
15     description: Pool to contact
16   user_data:
17     type: string
18     description: Server user_data
19   metadata:
20     type: json
21   network:
22     type: string
23     description: Network used by the server
```

ส่วนของ resource ประกอบด้วย server, member, port

```
25  resources:
26    server:
27      type: OS::Nova::Server
28      properties:
29        flavor: {get_param: flavor}
30        image: {get_param: image}
31        key_name: {get_param: key_name}
32        metadata: {get_param: metadata}
33        user_data: {get_param: user_data}
34        networks:
35          - port: { get_resource: port }
36
37    member:
38      type: OS::Neutron::PoolMember
39      properties:
40        pool_id: {get_param: pool_id}
41        address: {get_attr: [server, first_address]}
42        protocol_port: 80
43
44    port:
45      type: OS::Neutron::Port
46      properties:
47        network: {get_param: network}
48        security_groups:
49          - web_security_group
50
```

ส่วน output ประกอบด้วย

```
51  outputs:
52    server_ip:
53      description: IP Address of the load-balanced server.
54      value: { get_attr: [server, first_address] }
55
56    lb_member:
57      description: LB member details.
58      value: { get_attr: [member, show] }
```

คำอธิบายรายละเอียดแต่ละส่วนของ template stack4.yaml

ส่วน parameters ประกอบด้วย

```
1  heat_template_version: 2014-10-16
2  description: AutoScaling CentOS 7.1 Web Application
3  parameters:
4    image:
5      type: string
6      description: Image used for servers
7      default: CentOS-7-x86_64
8    key_name:
9      type: string
10     description: SSH key to connect to the servers
11     default: cloudkey1
12   flavor:
13     type: string
14     description: flavor used by the web servers
15     default: m1.small
16   network:
17     type: string
18     description: Network used by the server
19     default: 2e526792-10f2-43e4-aab5-8f3912571e76
20   subnet_id:
21     type: string
22     description: subnet on which the load balancer will be located
23     default: 7bae7b87-4e8c-be6f-6b5ea112bbdd
24   external_network_id:
25     type: string
26     description: UUID of a Neutron external network
27     default: c61fe76f-25ba-4fa6-a3f7-37d59fe17999
28
29   hostip:
30     type: string
31     description: host ip
32     default: 103.27.200.242
33
34   fqdn:
35     type: string
36     description: fqdn
37     default: web.example.com
38
39   shortname:
40     type: string
41     description: short host name
42     default: web
43
```

ส่วน resource ประกอบด้วย

```
44  resources:
45    webserver:
46      type: OS::Heat::AutoScalingGroup
47      properties:
48        min_size: 1
49        max_size: 20
50        cooldown: 60
51        desired_capacity: 1
52        resource:
53          type: https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
54          properties:
55            flavor: {get_param: flavor}
56            image: {get_param: image}
57            key_name: {get_param: key_name}
58            network: {get_param: network}
59            pool_id: {get_resource: pool}
60            metadata: {"metering.stack": {get_param: "OS::stack_id"}}
61            user_data:
62              str_replace:
63                template: |
64                  #!/bin/bash -
65                  echo root:centos | chpasswd
66                  echo "$hostip $fqdn $shortname" >> /etc/hosts
67                  yum -y install httpd php
68                  systemctl enable httpd
69                  systemctl start httpd
70                  cat <<EOF > /var/www/html/hostname.php
71                  <?php echo "Hello, My name is " . php_uname('n'); ?>
72                  EOF
73        params:
74          $hostip: {get_param: hostip}
75          $fqdn: {get_param: fqdn}
76          $shortname: {get_param: shortname}
```

ส่วนที่ทำหน้าที่เกี่ยวกับ policy

```

78 ▼   web_server_scaleup_policy:
79     type: OS::Heat::ScalingPolicy
80     properties:
81       adjustment_type: change_in_capacity
82       auto_scaling_group_id: {get_resource: webserver}
83       cooldown: 60
84       scaling_adjustment: 1
85
86 ▼   web_server_scaledown_policy:
87     type: OS::Heat::ScalingPolicy
88     properties:
89       adjustment_type: change_in_capacity
90       auto_scaling_group_id: {get_resource: webserver}
91       cooldown: 60
92       scaling_adjustment: -1
93

```

ส่วนการสร้าง event action จาก load CPU

```

94 ▼   cpu_alarm_high:
95     type: OS::Ceilometer::Alarm
96     properties:
97       description: Scale-up if the average CPU > 50% for 1 minute
98       meter_name: cpu_util
99       statistic: avg
100      period: 60
101      evaluation_periods: 1
102      threshold: 50
103      alarm_actions:
104        - {get_attr: [web_server_scaleup_policy, alarm_url]}
105      matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
106      comparison_operator: gt
107
108 ▼   cpu_alarm_low:
109     type: OS::Ceilometer::Alarm
110     properties:
111       description: Scale-down if the average CPU < 15% for 60 minutes
112       meter_name: cpu_util
113       statistic: avg
114       period: 600
115       evaluation_periods: 1
116       threshold: 15
117      alarm_actions:
118        - {get_attr: [web_server_scaledown_policy, alarm_url]}
119      matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
120      comparison_operator: lt

```

ส่วนการบริหารจัดการ Load Balance

```

122 ▼   monitor:
123     type: OS::Neutron::HealthMonitor
124     properties:
125       type: TCP
126       delay: 5
127       max_retries: 5
128       timeout: 5
129
130 ▼   pool:
131     type: OS::Neutron::Pool
132     properties:
133       protocol: HTTP
134       monitors: [{get_resource: monitor}]
135       subnet_id: {get_param: subnet_id}
136       lb_method: ROUND_ROBIN
137       vip:
138         protocol_port: 80
139
140 ▼   lb:
141     type: OS::Neutron::LoadBalancer
142     properties:
143       protocol_port: 80
144       pool_id: {get_resource: pool}
145
146 ▼   lb_floating:
147     type: OS::Neutron::FloatingIP
148     properties:
149       floating_network_id: {get_param: external_network_id}
150       port_id: {get_attr: [pool, vip, port_id]}
151

```

ส่วน outputs

```

152 ▼ outputs:
153 ▼   scale_up_url:
154 ▼     description: >
155       This URL is the webhook to scale up the autoscaling group. You
156       can invoke the scale-up operation by doing an HTTP POST to this
157       URL; no body nor extra headers are needed.
158     value: {get_attr: [web_server_scaleup_policy, alarm_url]}
159 ▼   scale_dn_url:
160 ▼     description: >
161       This URL is the webhook to scale down the autoscaling group.
162       You can invoke the scale-down operation by doing an HTTP POST to
163       this URL; no body nor extra headers are needed.
164     value: {get_attr: [web_server_scaledown_policy, alarm_url]}
165 ▼   pool_ip_address:
166     value: {get_attr: [pool, vip, address]}
167   description: The IP address of the load balancing pool
168 ▼   website_url:
169     value:
170       str_replace:
171         template: http://serviceip/hostname.php
172       params:
173         serviceip: { get_attr: [lb_floating, floating_ip_address] }
174   description: >
175     This URL is the "external" URL that can be used to access the
176     website.
177 ▼   ceilometer_query:
178     value:
179       str_replace:
180         template: >
181           ceilometer statistics -m cpu_util
182             -q metadata.user_metadata.stack=stackval -p 600 -a avg
183         params:
184           stackval: { get_param: "OS::stack_id" }
185   description: >
186     This is a Ceilometer query for statistics on the cpu_util meter
187     Samples about OS::Nova::Server instances in this stack. The -q
188     parameter selects Samples according to the subject's metadata.
189     When a VM's metadata includes an item of the form metering.X=Y,
190     the corresponding Ceilometer resource has a metadata item of the
191     form user_metadata.X=Y and samples about resources so tagged can
192     be queried with a Ceilometer query term of the form
193     metadata.user_metadata.X=Y. In this case the nested stacks give
194     their VMs metadata that is passed as a nested stack parameter,
195     and this stack passes a metadata of the form metering.stack=Y,
196     where Y is this stack's ID.

```

การดำเนินการเตรียม environment เพื่อการทดสอบ

เนื่องด้วย ใน heat template จะมีการอ้างอิง environment file ในส่วน OS::Heat::AutoScalingGroup เพื่อสร้าง load balance ดังรูป ด้านล่าง ในส่วนของ type ได้มีการทำหนด url scheme เป็น http หรือ https ดังนั้น จึงนำเอาไฟล์ไปวางไว้บน github และอ้างอิงเป็น https (หากองค์กรมี webserver เองก็สามารถนำว่าไว้บน webserver ขององค์กรและเปลี่ยนเป็น url ที่ถูกต้อง)

```

44 ▼ resources:
45 ▼   webserver:
46     type: OS::Heat::AutoScalingGroup
47   properties:
48     min_size: 1
49     max_size: 20
50     cooldown: 60
51     desired_capacity: 1
52   resource:
53     type: https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
54   properties:
55     flavor: {get_param: flavor}
56     image: {get_param: image}
57     key_name: {get_param: key_name}
58     network: {get_param: network}
59     pool_id: {get_resource: pool}
60     metadata: {"metering.stack": {get_param: "OS::stack_id"}}

```

โดยคุณสมบัติการกำหนดจะกำหนดไว้ที่ /usr/lib/python2.7/sitepackages/heat/common/urlfetch.py ดังตัวอย่าง code ด้านล่าง

```

35 def get(url, allowed_schemes=('http', 'https')):
36     """Get the data at the specified URL.
37
38     The URL must use the http: or https: schemes.
39     The file: scheme is also supported if you override
40     the allowed_schemes argument.
41     Raise an IOError if getting the data fails.
42     """
43     LOG.info(_LI('Fetching data from %s'), url)
44
45     components = urlparse.urlparse(url)
46
47     if components.scheme not in allowed_schemes:
48         raise URLFetchError(_('Invalid URL scheme %s') % components.scheme)
49

```

ต่อมาใน lb-env.yaml ส่วนของ resources ที่มีชื่อว่า port ชนิด OS::Neutron::Port มีการเรียกใช้งานส่วนของ security-group ชื่อ wp_security_group ซึ่งจะต้องเตรียมไว้ก่อนแล้ว แบบ manual โดยให้เปิด port 80,22

```

44 port:
45     type: OS::Neutron::Port
46     properties:
47         network: {get_param: network}
48         security_groups:
49             - wp_security_group

```

ดังนั้นจึงต้องสร้าง security group นี้ก่อน

```
# nova secgroup-list
# nova secgroup-create web_security_group "Allow Web Traffic and allow ssh"
```

ดังรูป

```
[root@cloud ~(keystone_demo)]# nova secgroup-list
+-----+-----+-----+
| Id      | Name      | Description      |
+-----+-----+-----+
| 3de4b02a-f437-492b-b305-0f6a5c8f753c | default | Default security group |
+-----+-----+-----+
[root@cloud ~(keystone_demo)]# nova secgroup-create web_security_group "Allow Web Traffic and allow ssh"
+-----+-----+-----+
| Id      | Name      | Description      |
+-----+-----+-----+
| e91693c0-434d-49fa-9dff-07f9b391e31c | web_security_group | Allow Web Traffic and allow ssh |
+-----+-----+-----+
```

เพิ่ม rule

```
# nova secgroup-add-rule web_security_group tcp 80 80 0.0.0.0/0
# nova secgroup-add-rule web_security_group tcp 22 22 0.0.0.0/0
# nova secgroup-add-rule web_security_group tcp 443 443 0.0.0.0/0
```

ดังรูป

```
[root@cloud ~(keystone_demo)]# nova secgroup-add-rule web_security_group tcp 80 80 0.0.0.0/0
+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+
| tcp          | 80        | 80      | 0.0.0.0/0 |           |
+-----+-----+-----+-----+
[root@cloud ~(keystone_demo)]# nova secgroup-add-rule web_security_group tcp 22 22 0.0.0.0/0
+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+
| tcp          | 22        | 22      | 0.0.0.0/0 |           |
+-----+-----+-----+-----+
[root@cloud ~(keystone_demo)]# nova secgroup-add-rule web_security_group tcp 443 443 0.0.0.0/0
+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+
| tcp          | 443       | 443     | 0.0.0.0/0 |           |
+-----+-----+-----+-----+
```

อีกประการหนึ่งที่มีความจำเป็นในการที่จะให้ instance ออก internet ได้คือ ค่า public dns ของ subnet จะต้อง เป็น 8.8.8.8, 8.8.4.4 หรือจะเป็น dns ขององค์กร

```
# neutron subnet-list
# neutron subnet-show 7bae7b87-4e7c-4e8e-be6f-6b5ea112bbdd
```

ดังรูปด้านล่าง

```
[root@clouc ~ (keystone_demo)]# neutron subnet-list
+-----+-----+-----+
| id | name | cidr | allocation_pools |
+-----+-----+-----+
| 7bae7b87-4e7c-4e8e-be6f-6b5ea112bbdd | private_subnet | 192.168.10.0/24 | {"start": "192.168.10.2", "end": "192.168.10.254"} |
| 26291621-6ade-4a3a-bc35-d2606637fcfd | public_subnet | 103.27.200.128/25 | {"start": "103.27.200.237", "end": "103.27.200.248"} |
+-----+-----+-----+
[root@clouc ~ (keystone_demo)]# neutron subnet-show 7bae7b87-4e7c-4e8e-be6f-6b5ea112bbdd
+-----+
| Field | Value |
+-----+
| allocation_pools | [{"start": "192.168.10.2", "end": "192.168.10.254"}] |
| cidr | 192.168.10.0/24 |
| dns_nameservers | 8.8.4.4 |
| dns_nameservers | 8.8.8.8 |
| enable_dhcp | True |
| gateway_ip | 192.168.10.1 |
| host_routes |
| id | 7bae7b87-4e7c-4e8e-be6f-6b5ea112bbdd |
| ip_version | 4 |
| ipv6_address_mode |
| ipv6_ra_mode |
| name | private_subnet |
| network_id | 2e526792-10f2-43e4-aab5-8f3912571e76 |
| subnetpool_id |
| tenant_id | df5ded99a7a74b6896647d11126dd7bd |
+-----+
```

หากไม่มีค่าดังที่เห็นให้ใช้คำสั่ง neutron subnet-update เพื่อปรับปรุงค่า

```
# neutron subnet-update 7bae7b87-4e7c-4e8e-be6f-6b5ea112bbdd --dns-nameservers
list=true 8.8.8.8 8.8.4.4
```

เมื่อเตรียมระบบแล้ว การทดสอบโดยการ upload stack4.yaml ดังรูป

Select Template

Template Source *

File

Description:

Use one of the available template source options to specify the template to be used in creating this stack.

Template File ?

Choose File | stack4.yaml

Environment Source

File

Environment File ?

Choose File | No file chosen

Cancel **Next**

กรอกข้อมูลในฟอร์ม

หลังจากการอักข้อมูลเรียบร้อยแล้ว ให้ทำการเปิดดู log /var/log/heat/heat-engine.log เพื่อดูขึ้นตอนการทำงาน เมื่อพิจารณา log ที่เกิดขึ้นสามารถเห็นการทำงานเป็นขั้นตอน ดังรูปด้านล่างถ้าเป็นการแบ่งส่วนของ log ทั้งหมดออกมายังส่วนของตัวอย่างที่แสดง

1 ตรวจสอบความถูกต้องของ template และทดสอบ Dowload file lb-env.html ที่อยู่บน github มาพร้อมตรวจสอบความถูกต้อง

```
[req-b7093bc9-8d6c-4a74-bbe3-d13338085ee1 - demo] validate_template
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Creating stack lab4
e [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating HealthMonitor "monitor"
e [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating Pool "pool"
e [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating AutoScalingResourceGroup "webserver"
ment [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
ment [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
ment [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Registering https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml -> https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
n [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Fetching data from https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
h [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Fetching data from https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
e [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating TemplateResource "ksqrly7zq7d"
```

ตรวจสอบดูว่าจะต้องสร้าง resource อะไรบ้าง

```
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating Port "port"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating Server "server"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating PoolMember "member"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating AutoScalingPolicy "web_server_scaledown_policy"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating CeilometerAlarm "cpu_alarm_low"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating AutoScalingPolicy "web_server_scaleup_policy"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating CeilometerAlarm "cpu_alarm_high"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating LoadBalancer "lb"
[req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating FloatingIP "lb_floating"
] Stack CREATE IN_PROGRESS (lab4): Stack CREATE started
[-] creating HealthMonitor "monitor" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
[-] creating Pool "pool" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
[-] creating AutoScalingResourceGroup "webservice" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
```

ขั้นตอนการสร้าง resource

```
.stack [-] Stack CREATE IN_PROGRESS (lab4-webservice-lpqcooxksqr): Stack CREATE started
.resource [-] creating LoadBalancer "lb" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
.resource [-] creating TemplateResource "jsxnz3qp6mm4" Stack "lab4-webservice-lpqcooxksqr" [ac484f30-309f-42fe-b961-700d8709ba7a]
.resource [-] creating FloatingIP "lb_floating" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
.environment [-] Registering https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml -> https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml

.service [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Creating stack lab4-webservice-lpqcooxksqr-jsxnz3qp6mm4-mugysrz4dgh7
.environment [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
.resource [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating Port "port"
.resource [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating Server "server"
.resource [req-a3ada686-cfe2-4975-b8d8-9485384bd064 demo demo] Validating PoolMember "member"
.stack [-] Stack CREATE IN_PROGRESS (lab4-webservice-lpqcooxksqr-jsxnz3qp6mm4-mugysrz4dgh7): Stack CREATE started
.resource [-] creating Port "port" Stack "lab4-webservice-lpqcooxksqr-jsxnz3qp6mm4-mugysrz4dgh7" [ef3a3e9b-733c-41f6-96b3-d72a79597665]
.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
.resource [-] creating Server "server" Stack "lab4-webservice-lpqcooxksqr-jsxnz3qp6mm4-mugysrz4dgh7" [ef3a3e9b-733c-41f6-96b3-d72a795976]
.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
.environment [-] Registering https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml -> https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
```

สร้าง instance ชื่อ lab4-webservicexxxxxx และนำไปเป็นสมาชิกของ pool loadbalance

```
nt [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
[-] creating PoolMember "member" Stack "lab4-webservice-lpqcooxksqr-jsxnz3qp6mm4-mugysrz4dgh7" [ef3a3e9b-733c-41f6-96b3-d72a79597665]
nt [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
nt [-] Registering https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml -> https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml

nt [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
Stack CREATE COMPLETE (lab4-webservice-lpqcooxksqr-jsxnz3qp6mm4-mugysrz4dgh7): Stack CREATE completed successfully
nt [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
nt [-] Registering https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml -> https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml

nt [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
Stack CREATE COMPLETE (lab4-webservice-lpqcooxksqr): Stack CREATE completed successfully
nt [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
nt [-] Registering https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml -> https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
```

ขั้นตอนสุดท้ายคือการสร้าง AutoScalingPolicy ทั้ง scaleup และ scaledown พร้อมทั้ง CeilometerAlarm ทั้ง cpu_alarm_low และ cpu_alarm_high

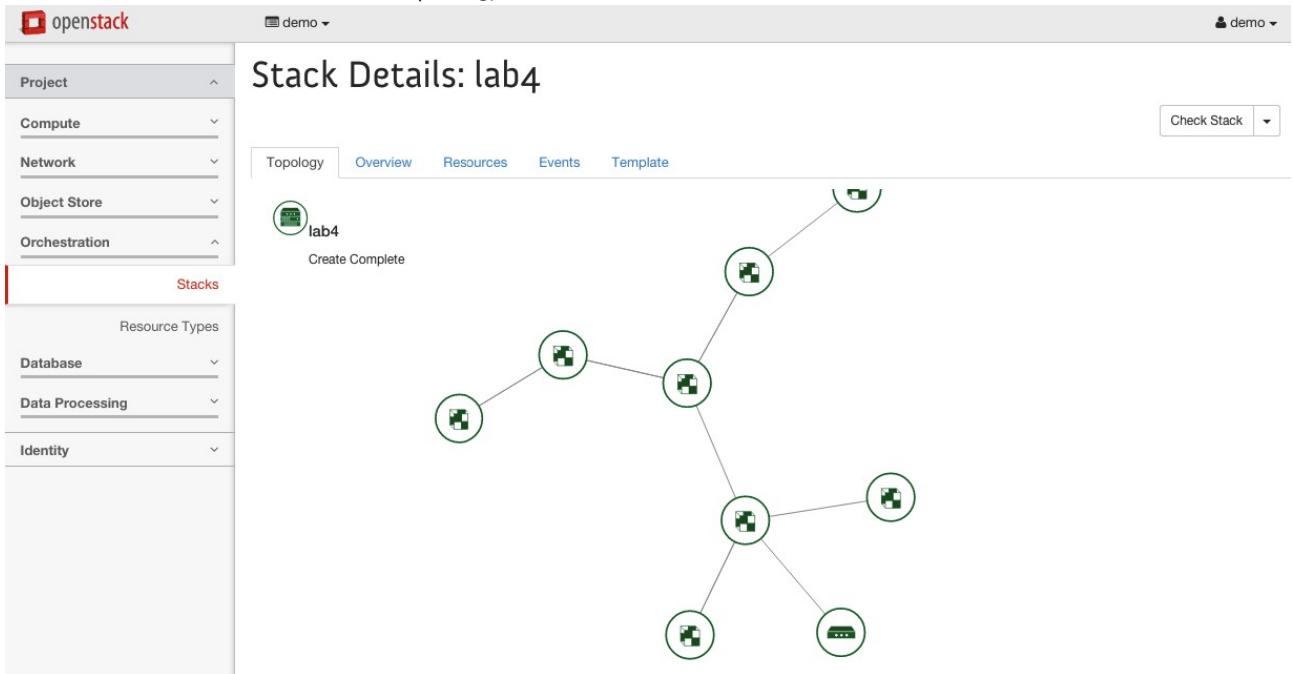
```
.urlfetch [-] Fetching data from https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-env.yaml
.resource [-] creating AutoScalingPolicy "web_server_scaledown_policy" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
.resource [-] creating AutoScalingPolicy "web_server_scaleup_policy" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
.resource [-] creating CeilometerAlarm "cpu_alarm_low" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
.resource [-] creating CeilometerAlarm "cpu_alarm_high" Stack "lab4" [6abb0df-2fb4-46ba-aa13-384722dd914]
.stack [-] Stack CREATE COMPLETE (lab4): Stack CREATE completed successfully
.service [req-64d31224-2045-4745-8a04-7dd5bbdfa9b - -] Service f53c1238-40fe-4b0e-98b7-7d620385cd89 is updated
.service [req-cff23a4e-6daa-4814-abf2-38b0358fa459 - -] Service ifc37371-62fb-423d-9a94-f60d6eb07a01 is updated
```

ตรวจสอบที่ horizon ในเมนูของ stack

<input type="checkbox"/>	Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/>	lab4	25 minutes	Never	Create Complete	<button>Check Stack</button>

Displaying 1 item

เมื่อ กดที่ชื่อของ stack ตรวจสอบ topology



ตรวจสอบ overview

The screenshot shows the OpenStack Horizon interface with the project set to 'demo'. The left sidebar is identical to the previous screenshot. The main content area is titled 'Stack Details: lab4'. The 'Overview' tab is active. It displays a 'Stack Overview' section with 'Information' and 'Status' tables. The 'Information' table includes columns for Name (lab4), ID (6abbb0df-2fb4-46ba-aa13-3847222dd914), and Description (AutoScaling CentOS 7.1 Web Application). The 'Status' table includes columns for Created (26 minutes ago), Last Updated (Never), and Status (Create_COMPLETE: Stack CREATE completed successfully). Below these, there's an 'Outputs' section listing several URLs with their descriptions:

Output Name	Description
pool_ip_address	The IP address of the load balancing pool 192.168.10.6
scale_dn_url	This URL is the webhook to scale down the autoscaling group. You can invoke the scale-down operation by doing an HTTP POST to this URL; no body nor extra headers are needed. http://192.168.1.2:8000/v1/signal/arm%3Aopenstack%3Aheat%3Aheat%3A%3Adf5ded99a7a74b6896647d11126dd7bd%3Astacks%2Flab4%2F6abbb2fb4-46ba-aa13-3847222dd914%2Fresources%2Fweb_server_scaledown_policy?Timestamp=2015-12-04T10%3A52%3A31Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=90a076d4b1ae43d6a28cbb01b4c2fb53&SignatureVersion=2&Signature
scale_up_url	This URL is the webhook to scale up the autoscaling group. You can invoke the scale-up operation by doing an HTTP POST to this URL; no body nor extra headers are needed. http://192.168.1.2:8000/v1/signal/arm%3Aopenstack%3Aheat%3Aheat%3A%3Adf5ded99a7a74b6896647d11126dd7bd%3Astacks%2Flab4%2F6abbb2fb4-46ba-aa13-3847222dd914%2Fresources%2Fweb_server_scaleup_policy?Timestamp=2015-12-04T10%3A52%3A31Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=b89ec5e9dfc24f60bbb95b9fa89c7d&SignatureVersion=2&Signature
website_url	This URL is the "external" URL that can be used to access the website. http://103.27.200.240/hostname.php

จะเห็น Floating ip จาก Output เพื่อเข้าไปใช้งาน สามารถทำการทดสอบด้วยการเปิด browser ไปยัง ip <http://103.27.200.240/hostname.php> ซึ่งเป็น ip ของ load balance เมื่อเรียก มายัง ip นี้ traffic members



ตรวจสอบที่ horizon สร้าง load balance pool ขึ้นมา ชื่อ lab4-pool-wtt3uznpze2q และเมื่อคลิกชื่อดูจะเห็นรายละเอียด

<input type="checkbox"/>	Name	Description	Provider	Subnet	Protocol	Status	VIP	Actions
<input type="checkbox"/>	lab4-pool-wtt3uznpze2q		haproxy	192.168.10.0/24	HTTP	Active	pool.vip	<button>Edit Pool</button>

รายละเอียด

ID	878a4972-9066-4371-841d-a37763fc213d
Name	lab4-pool-wtt3uznpze2q
Description	-
Project ID	df5ded99a7a74b6896647d11126dd7bd
VIP	pool.vip
Provider	haproxy
Subnet	private_subnet 192.168.10.0/24
Protocol	HTTP
Load Balancing Method	ROUND_ROBIN
Members	192.168.10.7:80
Health Monitors	TCP delay:5 retries:5 timeout:5
Admin State Up	Yes
Status	ACTIVE

หากกดที่ instance จะเห็นว่า heat engine สร้าง instance ดังรูป

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	la-sqcr-jxnz3qp6mm4-mugysrz4dgh7-server-xpgmd3kfcfq	CentOS-7-x86_64	192.168.10.7	m1.small	cloudkey1	Active	nova	None	Running	22 minutes	<button>Create Snapshot</button>

ตรวจสอบ log เพื่อดูการทำงานของ cloud init

Log	
Instance Console Log	
Log Length	35
<input type="button" value="Go"/> <input type="button" value="View Full Log"/>	

```

cloud-init[3610]: Cannot find a valid baseurl for repo: base/7/x86_64
cloud-init[3610]: systemctl enable httpd
cloud-init[3610]: Failed to issue method call: Access denied
cloud-init[3610]: systemctl start httpd
cloud-init[3610]: Failed to issue method call: Unit httpd.service failed to load: No such file or directory.
cloud-init[3610]: cat <<EOF > /var/www/html/hostname.php
cloud-init[3610]: <?php echo "Hello, My name is " . php_uname('n'); ?>
cloud-init[3610]: EOF

```

กดที่ console เพื่อทดสอบ การ ssh ไปยัง instance ด้วย password เนื่องจาก ใน cloud init ได้มีคำสั่งให้ตั้งค่า password

```

61
62
63
64
65 echo root:centos | chpasswd
66
67
68
69
70
71
72
73
74
75
76
77

```

ทดสอบการ login user: root ด้วย password: centos

หลังจาก ทำการ login เรียบร้อยด้วย password แล้ว จะทำการเพิ่ม โหลดให้กับ cpu ด้วยคำสั่ง ด้วยการทำ stress test ด้วยคำสั่ง

```
# yes > /dev/null
```

ให้เปิด terminal ขึ้นมา เพื่อ ตรวจสอบ log /var/log/ceilometer/alarm-notifier.log จะเห็นว่า ก่อนที่สั่ง stress test จะเห็นว่า เป็น Notifying alarm มีค่าเป็น low

```
[root@cloud ~]# tail -f /var/log/ceilometer/alarm-notifier.log
```

```
2015-12-04 19:19:06.936 19710 INFO ceilometer.alarm.notifier.rest [-] Notifying alarm lab4-cpu_alarm_low-btnbbjwrwek7 ea222dd9-ff47-4b93-8711-3bdcc228cc433 from alarm to alarm with action SplitResult(scheme=u'http', netloc=u'192.168.1.2:5000', path=u'/v1/signal/arn%3Aopenstack%3Aheat%3A%3Adff5ded99a7a74b6896647d11126dd7bd%3Astacks%2Flab4%2F6abbb0df-2fb4-46ba-aa13-3847222dd914%2Fresources%2Fweb_server_scaledown_policy', query=u'Timestamp=2015-12-04T10%3A52%3A31Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=90a076d4b1ae43d6a28c8b01b4c2fb53&SignatureVersion=2&Signature=xZTFYphCpCJOHHeEHeRPQqFFrb%2FsfifXm3l497Umbg%3D', fragment='') because Remaining as alarm due to 1 samples outside threshold, most recent: 0.065. request-id: req-336b61c9-f3e3-46cf-ac00-dcc293608579
```

ปรับลดเวลาเพื่อการทดสอบ ใน /etc/ceilometer/pipeline.yaml

เนื่องจากเป็นการทดสอบการทำงาน โดย default แล้ว Ceilometer กำหนดค่าของรอบเวลาที่ ceilometer จะดึงข้อมูลมาประมวลผล (polling period) อยู่ที่ 600 วินาที (10 นาที) ให้ทำการแก้ไขค่า interval ใน file /etc/ceilometer/pipeline.yaml ให้เป็น 60 วินาที (1 นาที) หลังจากแก้ไขแล้วจะได้ผลดังรูปด้านล่าง

```

---
sources:
  - name: meter_source
    interval: 60
    meters:
      - "cpu"
    sinks:
      - meter_sink
  - name: cpu_source
    interval: 60
    meters:
      - "cpu"
    sinks:
      - cpu_sink
  - name: disk_source
    interval: 60
    meters:
      - "disk.read.bytes"
      - "disk.read.requests"
      - "disk.write.bytes"
      - "disk.write.requests"
      - "disk.device.read.bytes"
      - "disk.device.read.requests"
      - "disk.device.write.bytes"
      - "disk.device.write.requests"
    sinks:
      - disk_sink
  - name: network_source
    interval: 60
    meters:
      - "network.incoming.bytes"
      - "network.incoming.packets"
      - "network.outgoing.bytes"
      - "network.outgoing.packets"
    sinks:
      - network_sink

```

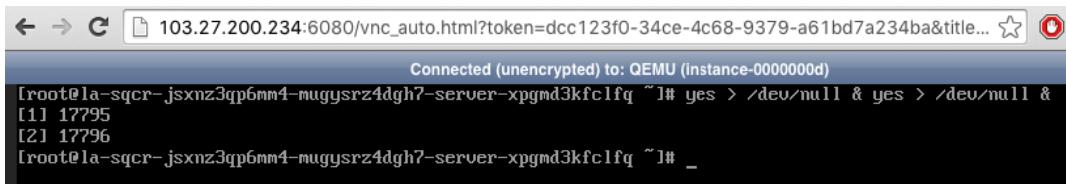
และทำการ restart ceilometer services

```

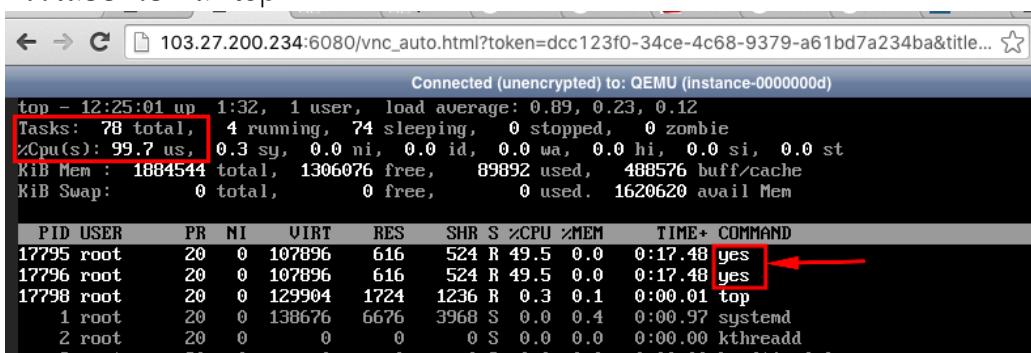
# export CEILO_SVCS='compute central collector api alarm-evaluator alarm-notifier'
# for svc in $CEILO_SVCS ; do sudo service openstack-ceilometer-$svc status ; done
# for svc in $CEILO_SVCS ; do sudo service openstack-ceilometer-$svc restart ; done

```

หลังจาก ทำ stress



ตรวจสอบด้วยคำสั่ง top



ให้รอสักครู่ และตรวจสอบ

The screenshot shows the OpenStack Instances dashboard. On the left, there's a sidebar with navigation links like Project, Compute, Network, Object Store, Orchestration, Database, Data Processing, and Identity. The Compute section is expanded, showing sub-links for Volumes, Images, and Access & Security. The Network link is also expanded. The main area is titled 'Instances' and displays a table with three rows. Each row represents an instance with columns for Instance Name, Image Name, IP Address, Size, Key Pair, Status, Availability Zone, Task, Power State, Time since created, and Actions. The first two instances are 'la-sqcr-' and the third is 'la-sqcr-j'. All instances are in the 'Running' state. A button for 'Create Snapshot' is visible next to each instance.

ตรวจสอบที่ load balancer

The screenshot shows the OpenStack Load Balancer dashboard. The sidebar has links for Project, Compute, Network, and Object Store. The Network link is expanded, showing sub-links for Network Topology, Networks, Routers, and Load Balancers. The Load Balancers link is selected and highlighted in red. The main area is titled 'Load Balancer' and shows a table with three rows under the 'Pools' tab. Each row represents a pool with columns for IP Address, Protocol Port, Weight, Pool, Status, and Actions. The pools are 'lab4-pool-wtt3uznpze2q' for ports 80 and 'lab4-pool-wtt3uznpze2q' for port 80. All members are in the 'Active' state. Buttons for '+ Add Member' and 'Delete Members' are at the top right.

หรือทดสอบด้วยคำสั่ง nova list

```
# nova list
```

ได้ผลดังรูป

```
[root@cloud ~]# source ./keystonerc_demo
[root@cloud ~]# nova list
+-----+-----+-----+-----+-----+
| ID   | Name      | Status | Task State | Power State | Networks          |
+-----+-----+-----+-----+-----+
| e8a258e0-32ed-4610-a005-b4a870b3d9d2 | la-sqcr-cpwxeuvwt4il-ttun7epwyfua-server-166aznnm3yuv | ACTIVE | -           | Running        | private_network=192.168.10.9 |
| 6e561a6a-02c3-4989-b1f8-0d879d5d3039 | la-sqcr-jeizlmsp2wdc-o3ays3nt4z14-server-pmvtnsojfu26 | ACTIVE | -           | Running        | private_network=192.168.10.8 |
| ecfacc42-8620-462f-92f2-cd166be5e8e2 | la-sqcr-jxnz3qp6mm4-mugysrz4dgh7-server-xpgmd3kfc1fq | ACTIVE | -           | Running        | private_network=192.168.10.7 |
+-----+-----+-----+-----+-----+
```

ต่อมาให้ทำการเปิด browser ขึ้นมา 3 tab จะเห็นได้ว่าค่าที่ได้จากการแสดงผล ที่แตกต่างกัน ค่าที่แสดงผลของมา เป็นชื่อของ instance ที่เกิดขึ้น

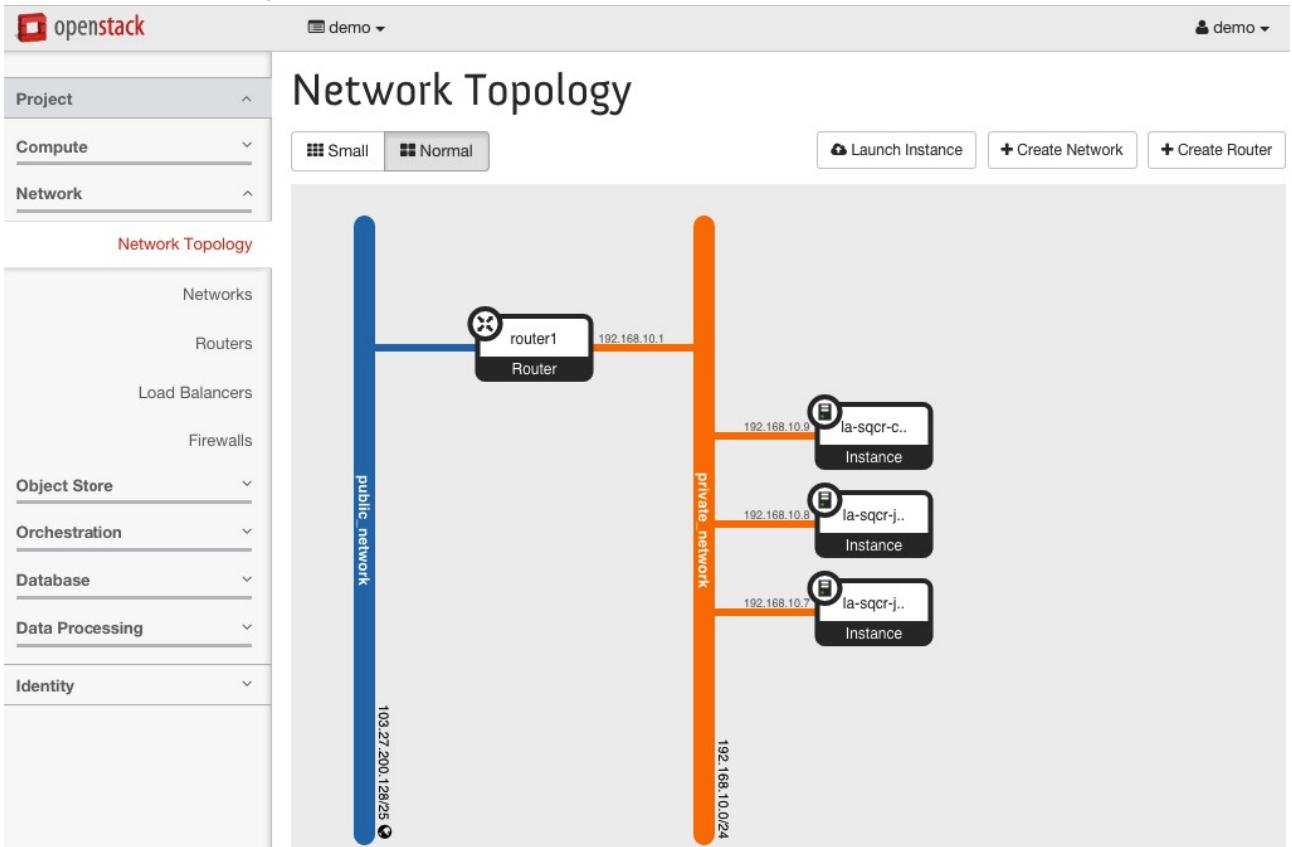
Hello, My name is la-sqr-cpwxeuvwt4il-ttun7epwyfua-server-i66aznnm3yuv

Hello, My name is la-sqr-jeizlmsp2wdc-o3ays3nt4zl4-server-pmvtnsojfu26

Hello, My name is la-sqr-jxnz3qp6mm4-mugysrz4dgh7-server-xpgmd3kfclfq

```
[root@cloud ceilometer]# cd ~
[root@cloud ~(keystone_admin)]# source ./keystonerc_demo
[root@cloud ~(keystone_demo)]# nova list
+---+-----+-----+
| ID | Name |
+---+-----+-----+
| e8a250e0-32ed-4610-aa05-b4a870b3d9d2 | la-sqr-cpwxeuvwt411-ttun7epwyfua-server-166aznnm3yuv |
| 66561a6a-62c3-4989-b1f8-0d879d5d3039 | la-sqr-jeizlmsp2wdc-o3ays3nt4zl4-server-pmvtnsojfu26 |
| ecfacc42-8628-462f-92f2-cd166be5e8e2 | la-sqr-jxnz3qp6mm4-mugysrz4dgh7-server-xpgmd3kfclfq |
+---+-----+
```

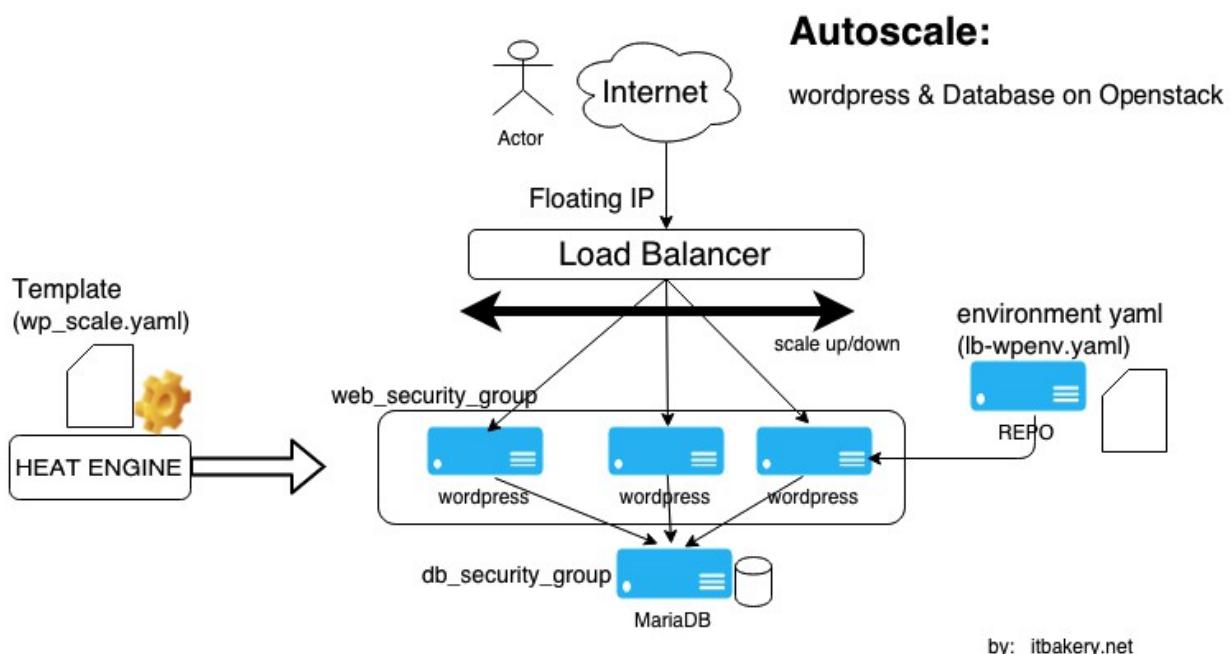
ຕរវាគសូប network topology



การทดสอบ 5 การสร้าง Auto scale สำหรับ Wordpress ให้รองรับ Cloud Application

HOT template ทำหน้าที่สำหรับ deploy server 3 เครื่องโดยทำหน้าที่เป็น Repository Server, Database server และ Web server (ใช้ Wordpress เป็นการทดสอบ) บน private network หรือ tenant network ให้เป็นสมาชิกของ loadbalancer กำหนดให้ loadbalancer ได้รับค่า floating ip เพื่อเข้าถึงได้จาก internet และจะทำการ redirect ไปยัง wordpress ที่เป็นสมาชิกของ loadbalancer อีกทีหนึ่ง

นอกจากนั้นยัง มีการทำงานร่วมกับ ceilometer เพื่อสร้าง alarm จากอัตราการใช้งาน CPU Utilization Rate ส่งสัญญาณ scaleup หรือ scaledown แก่ webserver ในการทดสอบครั้งนี้เพิ่มเติมความรู้สำหรับการเพิ่ม security group ให้กับ instance โดยกำหนด resource ให้เป็นชนิด OS::Neutron::SecurityGroup



ทำการ Clone git ด้านล่าง

```
git clone git@github.com:thaiopen/privatecloud.git
cd privatecloud
vi stack4.yaml
vi createrepo.sh
vi lb-wpenv.yaml
```

คำอธิบายรายละเอียดแต่ละส่วนของ template

เริ่มต้นจาก script createrepo.sh เป็น script ที่ทำหน้าที่เป็น postscript เพื่อสร้าง repo ให้กับระบบในการติดตั้ง package wordpress โดย repo server นั้นจะอยู่ใน tenant network เดียวกันกับ instance ที่จะสร้างขึ้น วิธีการใช้งานนั้นจะทำการ upload ไปยังส่วนของ post-creation ไปพร้อมๆ กับการสร้าง instance เป็นของหลังการทำงานนั้น cloud-init จะนำ script ไปทำงานในระหว่างการสร้าง instance

ต่อมาคือ Environment Template ชื่อ lb-wpenv.yaml ทำหน้าที่สร้าง resource ชื่อ server ชนิด

OS::Nova::Server เพื่อสร้าง instance แต่การสร้างได้กำหนดให้ไปสร้างภายใต้ resource ชื่อ member ชนิด OS::Neutron::PoolMember หลังจากนั้นก็จะกำหนด ipaddress และเปิดใช้งาน port 80 resource อีกอย่างหนึ่ง ที่สร้างขึ้นโดย neutron คือ port โดยการสร้าง port จะไปเรียกใช้ security_groups ชื่อ wp_security_group (จะต้องสร้างรอไว้ก่อน ก่อนการทดสอบ)

```

1 heat_template_version: 2014-10-16
2 description: A load-balancer wordpress server
3 parameters:
4   image:
5     type: string
6     description: Image used for servers
7   key_name:
8     type: string
9     description: SSH key to connect to the servers
10  flavor:
11    type: string
12    description: flavor used by the servers
13  pool_id:
14    type: string
15    description: Pool to contact
16  user_data:
17    type: string
18    description: Server user_data
19  metadata:
20    type: json
21  network:
22    type: string
23    description: Network used by the server
24
25 resources:
26   server:
27     type: OS::Nova::Server
28     properties:
29       flavor: {get_param: flavor}
30       image: {get_param: image}
31       key_name: {get_param: key_name}
32       metadata: {get_param: metadata}
33       user_data: {get_param: user_data}
34     networks:
35       - port: {get_resource: port }
36
37   member:
38     type: OS::Neutron::PoolMember
39     properties:
40       pool_id: {get_param: pool_id}
41       address: {get_attr: [server, first_address]}
42       protocol_port: 80
43
44   port:
45     type: OS::Neutron::Port
46     properties:
47       network: {get_param: network}
48       security_groups:
49         - wp_security_group
50
51 outputs:
52   server_ip:
53     description: IP Address of the load-balanced server.
54     value: { get_attr: [server, first_address] }
55   lb_member:
56     description: LB member details.
57     value: { get_attr: [member, show] }

```

ต่อมาเป็น template หลักเพื่อการทดสอบชื่อ wp_scale.yaml จะมีหน้าที่ในการสร้างระบบ โดยมี Architecture สำหรับการสร้างระบบ autoscale ให้แก่ wordpress เป็นการสร้างระบบให้มีลักษณะ 2 tier คือ web tier และ database tier โดยจะไปเรียกใช้งาน environment template ในส่วนที่ผ่านมา พร้อมกับใช้งาน repository ที่สร้างโดย shell script ที่ได้อธิบายก่อนหน้านี้ จากเนื้อหา template ในส่วนของ parameters นั้น จะมีการสร้างตัวแปร ได้แก่ image, key, flavor, database_flavor, database_flavor, network (อ้างอิงแบบ id), subnet_id (อ้างอิงแบบ id), database_name, database_user, public_network (อ้างอิงแบบ id) ดังรูปด้านล่าง ก่อนการทดสอบจะต้องแก้ไขให้ตรงกับระบบที่ทดสอบ

```

1 heat_template_version: 2015-04-30
2 description: AutoScaling Wordpress
3 ▼ parameters:
4   ▼ image:
5     type: string
6     description: Image used for servers
7     default: CentOS7-x86_64
8   ▼ key:
9     type: string
10    description: SSH key to connect to the servers
11    default: cloudkey1
12   ▼ flavor:
13     type: string
14     description: flavor used by the web servers
15     constraints:
16       - allowed_values: [m1.small, m1.medium]
17   ▼ database_flavor:
18     type: string
19     description: flavor used by the db server
20     constraints:
21       - allowed_values: [m1.small, m1.medium]
22   ▼ network:
23     type: string
24     description: Network used by the server
25     default: 6a328376-be26-4be6-a511-110ca0abf35a
26   ▼ subnet_id:
27     type: string
28     description: subnet on which the load balancer will be located
29     default: 4721b82d-b004-45ad-aca8-265f162b7d53
30   ▼ database_name:
31     type: string
32     description: Name of the wordpress DB
33     default: wordpress
34   ▼ database_user:
35     type: string
36     description: Name of the wordpress user
37     default: wordpress
38   ▼ public_network:
39     type: string
40     description: UUID of a Neutron external network
41     default: c61fe76f-25ba-4fa6-a3f7-37d59fe17999
42

```

ส่วน resources ของ wp_scale.yaml นั้น แบ่งเป็นการสร้าง server 2 แบบ คือ สร้าง server (resource ชื่อ db:) สำหรับ Database Server (Database Tier) เพื่อให้งานรองรับการทำงานเป็นฐานของมูลให้แก่ wordpress (Web Tier) ที่ทำการ scale การทำงานเบื้องหลังของ heat engine ของ db จะทำหน้าสร้าง instance โดยมีการใช้งาน security_groups ชื่อ db_security_group ที่จะต้องมีการสร้างไว้ก่อนเพื่อเดียวกัน ความสำคัญ ใน resource db คือ cloud-init script ในส่วน user_data จะเป็นการทำหนด local repo ให้ instance จะต้องแก้ไขให้ตรงกับ ip ของ local repo ดังตัวอย่างจะกำหนดที่

```

cat << REPO > /etc/yum.repos.d/local.repo
[local]
name=local
baseurl=http://192.168.10.3/repos/wordpress
enabled=1
gpgcheck=0
REPO

```

หรือสามารถใช้งานร่วมกับ localrepo ที่อยู่ในองค์กรเองก็ได้ หลังจากกำหนด local repo แล้วก็จะเป็นการติดตั้งฐานข้อมูล คือ mariadb-server ที่มีการสร้าง รหัสให้แก่ฐานข้อมูลแบบ random

```

yum -y install mariadb mariadb-server
systemctl enable mariadb.service
systemctl start mariadb.service
firewall-cmd --permanent --add-port=3306/tcp
firewall-cmd --reload

mysqladmin -u root password $db_rootpassword

cat << EOF | mysql -u root --password=$db_rootpassword
CREATE DATABASE $db_name;
GRANT ALL PRIVILEGES ON $db_name.* TO "$db_user"%""
EOF

```

```

IDENTIFIED BY "$db_password";
FLUSH PRIVILEGES;
EXIT
EOF

```

ดังรูปด้านล่าง

```

43 | resources:
44 |   database_password:
45 |     type: OS::Heat::RandomString
46 |   database_root_password:
47 |     type: OS::Heat::RandomString
48 | db:
49 |   type: OS::Nova::Server
50 | properties:
51 |   security_groups:
52 |     - db_security_group
53 |   flavor: {get_param: database_flavor}
54 |   image: {get_param: image}
55 |   key_name: {get_param: key}
56 |   networks: [{network: {get_param: network} }]
57 |   user_data_format: RAW
58 |   user_data:
59 |     str_replace:
60 |       template: |
61 |         #!/bin/bash -v
62 |         setenforce 0
63 |         cat << REPO > /etc/yum.repos.d/local.repo
64 |         [local]
65 |         name=local
66 |         baseurl=http://192.168.10.3/repos/wordpress
67 |         enabled=1
68 |         gpgcheck=0
69 |         REPO
70 |         yum -y install mariadb mariadb-server
71 |         systemctl enable mariadb.service
72 |         systemctl start mariadb.service
73 |         firewall-cmd --permanent --add-port=3306/tcp
74 |         firewall-cmd --reload
75 |
76 |         mysqladmin -u root password $db_rootpassword
77 |
78 |         cat << EOF | mysql -u root --password=$db_rootpassword
79 |           CREATE DATABASE $db_name;
80 |           GRANT ALL PRIVILEGES ON $db_name.* TO "$db_user"@"%"
81 |           IDENTIFIED BY "$db_password";
82 |           FLUSH PRIVILEGES;
83 |           EXIT
84 |           EOF
85 |         params:
86 |           $db_rootpassword: {get_attr: [database_root_password, value]}
87 |           $db_name: {get_param: database_name}
88 |           $db_user: {get_param: database_user}
89 |           $db_password: {get_attr: [database_password, value]}

```

ส่วนต่อมาคือ resource ชื่อ asg: ที่รับผิดชอบสำหรับการสร้าง instance ที่ทำหน้าเป็น web server ให้กับ wordpress โดยจะสร้างอยู่ภายใต้ resource ชนิด OS::Heat::AutoScalingGroup โดยการสร้างนั้นจะเรียกใช้ environment ที่ <https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-wpenv.yaml> ที่ได้กล่าวไปแล้ว มีการสร้าง cloud-init เพื่อที่สร้าง local repo และมีการแก้ไข configuration ของ wordpress ให้หันมาใช้งาน database server ด้วยการอ้างอิงด้วย ip ในตัวแปร \$db_host: {get_attr: [db, first_address]}

ดังรูป

```

90 ▼      asg:
91       type: OS::Heat::AutoScalingGroup
92       properties:
93         min_size: 1
94         max_size: 10
95       resource:
96         type: https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-wpenv.yaml
97       properties:
98         flavor: {get_param: flavor}
99         image: {get_param: image}
100        key_name: {get_param: key}
101        network: {get_param: network}
102        pool_id: {get_resource: pool}
103        metadata: {"metering.stack": {get_param: "OS::stack_id"}}
104      user_data:
105        str_replace:
106          template: |
107            #!/bin/bash -v
108
109            setenforce 0
110            cat << REPO > /etc/yum.repos.d/local.repo
111            [local]
112            name=local
113            baseurl=http://192.168.10.3/repos/wordpress
114            enabled=1
115            gpgcheck=0
116            REPO
117            yum -y install httpd wordpress
118            systemctl enable httpd.service
119            systemctl start httpd.service
120            setsebool -P httpd_can_network_connect_db=1
121
122            sed -i "/Deny from All/d" /etc/httpd/conf.d/wordpress.conf
123            sed -i "s/Require local/Require all granted/" /etc/httpd/conf.d/wordpress.conf
124            sed -i s/database_name_here/$db_name/ /etc/wordpress/wp-config.php
125            sed -i s/username_here/$db_user/ /etc/wordpress/wp-config.php
126            sed -i s/password_here/$db_password/ /etc/wordpress/wp-config.php
127            sed -i s/localhost/$db_host/ /etc/wordpress/wp-config.php
128
129            systemctl restart httpd.service
130            firewall-cmd --permanent --add-port=80/tcp
131            firewall-cmd --reload
132      params:
133        $db_name: {get_param: database_name}
134        $db_user: {get_param: database_user}
135        $db_password: {get_attr: [database_password, value]}
136        $db_host: {get_attr: [db, first_address]}
137

```

ต่อมาเป็นการกำหนด Scaleup/ScaleDown policy กำหนดให้เป็นการ ตรวจสอบที่ auto_scaling_group_id:

```

{get_resource: asg}
138 ▼      web_server_scaleup_policy:
139       type: OS::Heat::ScalingPolicy
140       properties:
141         adjustment_type: change_in_capacity
142         auto_scaling_group_id: {get_resource: asg}
143         cooldown: 60
144       scaling_adjustment: 1
145
146 ▼      web_server_scaledown_policy:
147       type: OS::Heat::ScalingPolicy
148       properties:
149         adjustment_type: change_in_capacity
150         auto_scaling_group_id: {get_resource: asg}
151         cooldown: 60
152       scaling_adjustment: -1
153

```

และเป็นการสร้าง cpu_alarm_high หากมีการใช้งาน cpu มากกว่า 50% เป็นเวลา 1 นาที และ cpu_alarm_low หากมีการลดการใช้งาน cpu น้อยกว่า 15% เป็นเวลา 15% โดยการเปลี่ยนแปลง นั้นจะมีการสร้าง instance ด้วย การเปลี่ยนแปลงนั้น ให้อยู่ภายใต้การติดตามการเปลี่ยนแปลงโดย neutron ด้วยเช่นเดียวกัน ด้วย resource ชื่อ monitor ชนิด OS::Neutron:Pool

```

154    cpu_alarm_high:
155      type: OS::Ceilometer::Alarm
156      properties:
157        description: Scale-up if the average CPU > 50% for 1 minute
158        meter_name: cpu_util
159        statistic: avg
160        period: 60
161        evaluation_periods: 1
162        threshold: 50
163        alarm_actions:
164          - {get_attr: [web_server_scaleup_policy, alarm_url]}
165          matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
166          comparison_operator: gt
167    cpu_alarm_low:
168      type: OS::Ceilometer::Alarm
169      properties:
170        description: Scale-down if the average CPU < 15% for 10 minutes
171        meter_name: cpu_util
172        statistic: avg
173        period: 60
174        evaluation_periods: 1
175        threshold: 15
176        alarm_actions:
177          - {get_attr: [web_server_scaledown_policy, alarm_url]}
178          matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
179          comparison_operator: lt
180    monitor:
181      type: OS::Neutron::HealthMonitor
182      properties:
183        type: TCP
184        delay: 5
185        max_retries: 5
186        timeout: 5

```

ส่วน resource ที่เป็นหัวใจคือการสร้าง loadbalance ชื่อ lb ชนิด OS::Neutron::LoadBalancer และ resource ชื่อ lb_floating ชนิด OS::Neutron::FloatingIP

```

196    lb:
197      type: OS::Neutron::LoadBalancer
198      properties:
199        protocol_port: 80
200        pool_id: {get_resource: pool}
201
202      # assign a floating ip address to the load balancer
203      # pool.
204      lb_floating:
205        type: OS::Neutron::FloatingIP
206        properties:
207          floating_network_id: {get_param: public_network}
208          port_id: {get_attr: [pool, vip, port_id]}
209

```

สามารถผลลัพธ์ได้จากการทำงาน โดยเฉพาะ url ของ wordpress ที่สร้าง ขึ้นเพื่อจะใช้งานต่อได้

```

210    outputs:
211      scale_up_url:
212        description: >
213          This URL is the webhook to scale up the autoscaling group. You
214          can invoke the scale-up operation by doing an HTTP POST to this
215          URL; no body nor extra headers are needed.
216        value: {get_attr: [web_server_scaleup_policy, alarm_url]}
217      scale_dn_url:
218        description: >
219          This URL is the webhook to scale down the autoscaling group.
220          You can invoke the scale-down operation by doing an HTTP POST to
221          this URL; no body nor extra headers are needed.
222        value: {get_attr: [web_server_scaledown_policy, alarm_url]}
223      pool_ip_address:
224        value: {get_attr: [pool, vip, address]}
225        description: The IP address of the load balancing pool
226      website_url:
227        value:
228          str_replace:
229            template: http://host/wordpress/
230            params:
231              host: { get_attr: [lb_floating, floating_ip_address] }
232        description: >
233          This URL is the "external" URL that can be used to access the
234          Wordpress site.
235      ceilometer_query:
236        value:
237          str_replace:
238            template: >
239              ceilometer statistics -m cpu_util
240              -q metadata.user_metadata.stack=stackval -p 60 -a avg
241            params:
242              stackval: { get_param: "OS::stack_id" }
243        description: >
244          This is a Ceilometer query for statistics on the cpu_util meter

```

เริ่มต้นการทดสอบ

เริ่มต้นการทดสอบให้ทำการสร้าง instance แบบ manual เพื่อที่สร้าง repository server ในที่นี่จะใช้ centos7 เป็น image บน private_network

	Image Name	Type	Status	Public	Protected	Format	Size	Actions
<input type="checkbox"/>	Fedora23	Image	Active	Yes	No	QCOW2	223.5 MB	<button>Launch Instance</button>
<input type="checkbox"/>	Ubuntu-trusty	Image	Active	Yes	No	QCOW2	246.6 MB	<button>Launch Instance</button>
<input type="checkbox"/>	CentOS-7-x86_64	Image	Active	Yes	No	QCOW2	819.3 MB	<button>Launch Instance</button>
<input type="checkbox"/>	manila-service-image	Image	Active	Yes	No	QCOW2	244.4 KB	<button>Launch Instance</button>

ตั้งชื่อว่า Repo_server

Launch Instance

Availability Zone: nova

Instance Name: Repo_server

Flavor: m1.small

Instance Count: 1

Instance Boot Source: Boot from image

Image Name: CentOS-7-x86_64 (819.3 MB)

Flavor Details:

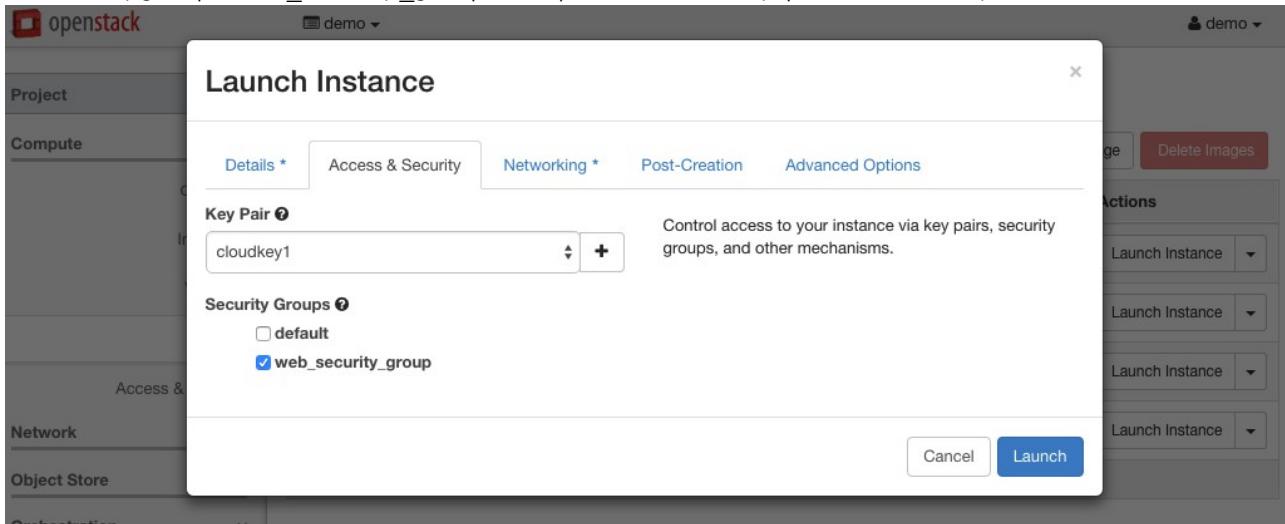
Name	m1.small
VCPUs	1
Root Disk	20 GB
Ephemeral Disk	0 GB
Total Disk	20 GB
RAM	2,048 MB

Project Limits:

Number of Instances	0 of 10 Used
Number of VCPUs	0 of 20 Used
Total RAM	0 of 51,200 MB Used

Cancel Launch

ใช้ securitygroup web_security_group (เปิด port 80, 20) , key pair ใช้ cloudkey1

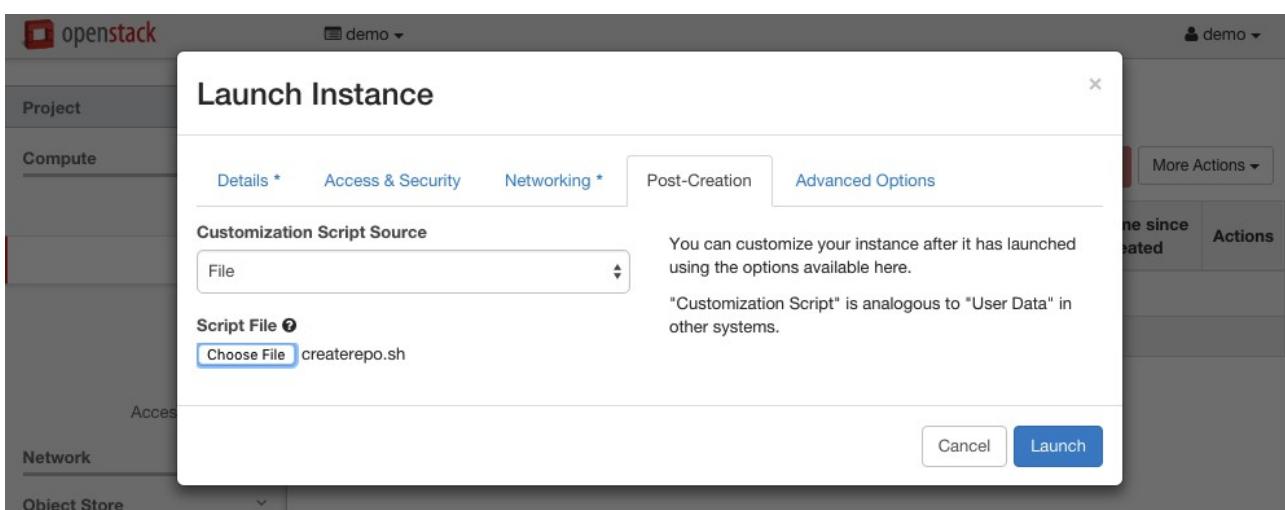


แต่ในการสร้าง repo นี้จะทำการ upload script createrepo.sh (ได้จาก clonerepo) ไปยัง ส่วน Post-Creation เพื่อทำหน้าที่เป็น Cloud-init script โดย post script มีหน้าที่ในการติดตั้ง local repo จะมีรายละเอียดดังนี้

```
#!/bin/bash -v
# Create Repo Script server

# add password to root
echo root:centos | chpasswd

yum update -y
yum -y install httpd
yum -y install vim
yum -y install createrepo
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
yum -y install http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
yum -y --enablerepo=remi install wordpress --downloadonly
--downloaddir=/var/www/html/repos/wordpress
yum -y install mariadb mariadb-server --downloadonly
--downloaddir=/var/www/html/repos/wordpress
createrepo /var/www/html/repos/wordpress
systemctl start httpd
```



ติดตามการทำงานของ cloud-init script

Instance Details: Repo_server

Overview Log Action Log

Log Length 35 Go View Full Log

Instance Console Log

```
cloud-init[3605]: exiting because "Download Only" specified
cloud-init[3605]: crearepo /var/www/html/repos/wordpress
cloud-init[3605]: Spawning worker 0 with 38 pkgs
cloud-init[3605]: Workers Finished
cloud-init[3605]: Saving Primary metadata
cloud-init[3605]: Saving file lists metadata
cloud-init[3605]: Saving other metadata
cloud-init[3605]: Generating sqlite DBs
cloud-init[3605]: Sqlite DBs complete
cloud-init[3605]: systemctl start httpd
Starting The Apache HTTP Server...
[[2m OK [0m] Started The Apache HTTP Server.
ci-info: ++++++Authorized keys from /home/centos/.ssh/authorized_keys for user centos+++++
ci-info: +-----+
ci-info: | Keypair | Fingerprint (md5) | Options | Comment |
ci-info: +-----+
ci-info: | ssh-rsa | 23:5e:1e:c0:46:64:da:fa:94:0f:le:7b:30:68:8b:80 | - | Generated-by-Nova |
ci-info: +-----+
```

วิธีการทดสอบว่า repo ที่สร้างขึ้นนั้นพร้อมที่จะทำงานหรือไม่ จะทดสอบด้วยการ เพิ่ม floating ip และ ทดสอบด้วยการเปิด ไปยัง server

Instances

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
Repo_server	CentOS-7-x86_64	192.168.10.3	m1.small	cloudkey1	Active	nova	None	Running	2 minutes	<button>Create Snapshot</button> <button>Associate Floating IP</button> <button>Disassociate Floating IP</button> <button>Edit Instance</button>

Displaying 1 item

เลือก floatingip

Manage Floating IP Associations

IP Address *

IP Address *

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

Repo_server: 192.168.10.3

Cancel Associate

ผลการดำเนินงาน

Instances

		Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	<input type="checkbox"/>	Repo_server	CentOS-7-x86_64	192.168.10.3 Floating IPs: 103.27.200.245	m1.small	cloudkey1	Active	nova	None	Running	7 minutes	<button>Create Snapshot</button>

Displaying 1 item

เปิด browser ทดสอบ <http://103.27.200.245/repos/>



Index of /repos

Name	Last modified	Size	Description
Parent Directory		-	
wordpress/	2015-12-04 18:30	-	

จากแผนภูมินี้มีไฟล์ที่ต้องใช้ในการทดสอบนี้ 2 ไฟล์คือ lb-wpenv.yaml ทำหน้าที่เป็น environment template สำหรับนำไปใช้สร้าง instance ภายใต้ OS::Heat::AutoScalingGroup และ wp_scale.yaml ทำหน้าที่เป็น template ซึ่งจะได้อธิบายรายละเอียดของแต่ละส่วนอย่างละเอียดต่อไป ก่อนการ upload ให้ทำการ แก้ไข network id ด้วยคำสั่ง neutron net-list

```
[root@cloud ~]# neutron net-list
+-----+-----+
| id | name | subnets |
+-----+-----+
| 6a328376-be26-4be6-a511-110ca0abf35a | private_network | 4721b82d-b004-45ad-aca8-265f162b7d53 192.168.10.0/24 |
| c61fe76f-25ba-4fa6-a3f7-37d59fe17999 | public_network | 26291621-6ade-4a3a-bc35-d2606037fcfd 103.27.200.128/25 |
+-----+-----+
```

หลังจากแก้ไขแล้วจะได้ดังรูป

```
22 |     network:
23 |       type: string
24 |       description: Network used by the server
25 |       default: 2e526792-10f2-43e4-aab5-8f3912571e76
26 |       subnet_id:
27 |         type: string
28 |         description: subnet on which the load balancer will be located
29 |         default: 7bae7b87-4e7c-4e8e-be6f-6b5ea112bbdd
30 |       database_name:
31 |         type: string
32 |         description: Name of the wordpress DB
33 |         default: wordpress
34 |       database_user:
35 |         type: string
36 |         description: Name of the wordpress user
37 |         default: wordpress
38 |       public_network:
39 |         type: string
40 |         description: UUID of a Neutron external network
41 |         default: c61fe76f-25ba-4fa6-a3f7-37d59fe17999
```

ใน script มีการเรียกใช้ security group ชื่อ db_security_group , wp_security_group

```
[root@cloud ~]# neutron security-group-list
+-----+-----+-----+
| id | name | description |
+-----+-----+-----+
| 3de4b02a-f437-492b-b305-0f6a5c8f753c | default | Default security group |
| d45135f6-9ea5-4688-9033-974a7d818fde | db_security_group | database security group |
|dbe09001-f034-45e6-9721-abf7975a833a | wp_security_group | wp_security_group |
| e91693c0-434d-49fa-9dff-07f9b391e31c | web_security_group | Allow Web Traffic and allow ssh |
+-----+-----+-----+
```

หรือจะดูผ่านทาง horizon

Access & Security

The screenshot shows the 'Security Groups' tab selected in the navigation bar. Below it is a search bar with a filter icon and a 'Create Security Group' button. A red box highlights the 'Actions' column where each row has a 'Manage Rules' button.

<input type="checkbox"/>	Name	Description	Actions
<input type="checkbox"/>	default	Default security group	<button>Manage Rules</button>
<input type="checkbox"/>	db_security_group	database security group	<button>Manage Rules</button>
<input type="checkbox"/>	wp_security_group	wp_security_group	<button>Manage Rules</button>
<input type="checkbox"/>	web_security_group	Allow Web Traffic and allow ssh	<button>Manage Rules</button>

Displaying 4 items

หลังจากนั้น

Select Template

The dialog has two main sections: 'Template Source *' and 'Description:'. The 'Template Source *' section includes a dropdown menu with 'File' selected, a 'Template File' input field containing 'Choose File wp_scale.yaml' (with a red box around it), and an 'Environment Source' section with a dropdown menu and a 'Choose File' input field containing 'no file selected'.

Template Source *

File

Template File ?
Choose File wp_scale.yaml

Description:

Use one of the available template source options to specify the template to be used in creating this stack.

Environment Source

File

Environment File ?
Choose File no file selected

Cancel **Next**

กรอกข้อมูล stack โดยกรอกข้อมูล stack name ชื่อว่า lab5 และรหัส 'demo' ส่วนค่าต่างๆ ที่เห็นนั้นเป็นค่าที่กำหนดเป็นค่า default ใน template

Description:
Create a new stack with the provided values.

ก่อนการกด launch ให้ทำการ tail -f /var/log/heat/heat-engine.log ไว้ด้วย แต่ครั้งนี้จะเกิด log ขึ้นมากกว่าทุกๆครั้งเนื่องจากมีการทำงานหลายส่วน ให้ลองสังเกตดู
ข้อมูล log ในส่วนนี้ เป็นการ Download environment template จาก Github ทำการ verify และสร้าง resource ต่างๆ

ตรวจสอบและสร้าง port, Server, Loadbalance Member

```
INFO heat.engine.service [req-4c71030e-bc9d-4297-8b19-115ede3b4e95 demo demo] Creating stack lab5-asg-vauy15k6grst-tkgvrqlnuni
INFO heat.engine.environment [req-4c71030e-bc9d-4297-8b19-115ede3b4e95 demo demo] Registering OS::Heat::ScaledResource -> AWS::
INFO heat.engine.resource [req-4c71030e-bc9d-4297-8b19-115ede3b4e95 demo demo] Validating Port "port"
INFO heat.engine.resource [req-4c71030e-bc9d-4297-8b19-115ede3b4e95 demo demo] Validating Server "server"
INFO heat.engine.resource [req-4c71030e-bc9d-4297-8b19-115ede3b4e95 demo demo] Validating PoolMember "member"
INFO heat.engine.stack [-] Stack CREATE_IN_PROGRESS (lab5-asg-vauy15k6grst-tkgvrqlnunit-07s222wndtx4): Stack CREATE started
INFO heat.engine.resource [-] creating Port "port" Stack "lab5-asg-vauy15k6grst-tkgvrqlnunit-07s222wndtx4" [75aa4168-d47c-4426
INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
INFO heat.engine.resource [-] creating Server "server" Stack "lab5-asg-vauy15k6grst-tkgvrqlnunit-07s222wndtx4" [75aa4168-d47c-
```

สร้าง policy และ alarm

```

.common.urlfetch [-] Fetching data from https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-wpenv.yaml
.engine.resource [-] creating AutoScalingPolicy "web_server_scaleup_policy" Stack "lab5" [488fd0ad-ff2d-4dc6-a769-b089f703]
.engine.resource [-] creating AutoScalingPolicy "web_server_scaledown_policy" Stack "lab5" [488fd0ad-ff2d-4dc6-a769-b089f71]
.engine.resource [-] creating CeilometerAlarm "cpu_alarm_high" Stack "lab5" [488fd0ad-ff2d-4dc6-a769-b089f7039aac]
.engine.resource [-] creating CeilometerAlarm "cpu_alarm_low" Stack "lab5" [488fd0ad-ff2d-4dc6-a769-b089f7039aac]
.engine.stack [-] Stack CREATE COMPLETE (lab5): Stack CREATE completed successfully

```

หากไม่มีความผิดพลาด ใน log จะปรากฏ ดังรูปด้านบน

Stack CREATE COMPLETE (lab5): Stack CREATE completed successfully

ตรวจสอบผลการทำงานผ่านทาง Horizon ให้เลือกชื่อ lab5 ที่สร้างขึ้น

Stack Name	Created	Updated	Status	Actions
lab5	12 minutes	Never	Create Complete	<button>Check Stack</button>

ให้เปิดในส่วนของ stack overview ค่าของ external url ตามรูปด้านล่าง จะเป็น ว่ามีค่าของ floating ip ที่สร้างโดยอัตโนมัติ จาก template <http://103.27.200.247/wordpress/>

Outputs

pool_ip_address	The IP address of the load balancing pool 192.168.10.4
scale_dn_url	This URL is the webhook to scale down the autoscaling group. You can invoke the scale-down operation by doing an HTTP POST to this URL; no body nor extra headers are needed. http://192.168.1.2:8000/v1/signal/arm%3Aopenstack%3Aheat%3A%3Ad5fed99a7a74b6896647d11126dd7bd%3Astacks%2Flab5%ff2d-4dc6-a769-b089f7039aac%2Fresources%2Fweb_server_scaledown_policy?Timestamp=2015-12-04T18%3A41%3A52Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=e141e038bf184049bb169f29209fb6c6&SignatureVersion=4
scale_up_url	This URL is the webhook to scale up the autoscaling group. You can invoke the scale-up operation by doing an HTTP POST to this URL; no body nor extra headers are needed. http://192.168.1.2:8000/v1/signal/arm%3Aopenstack%3Aheat%3A%3Ad5fed99a7a74b6896647d11126dd7bd%3Astacks%2Flab5%ff2d-4dc6-a769-b089f7039aac%2Fresources%2Fweb_server_scaleup_policy?Timestamp=2015-12-04T18%3A41%3A52Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=de292b57bc524dbfa781d972c6dc9055&SignatureVersion=4
website_url	This URL is the "external" URL that can be used to access the Wordpress site. http://103.27.200.247/wordpress/
ceilometer_query	This is a Ceilometer query for statistics on the cpu_util meter Samples about OS::Nova::Server instances in this stack. The -q parameter selects Samples according to the subject's metadata. When a VM's metadata includes an item of the form metering.X=Y, the corresponding Ceilometer resource has a metadata item of the form user_metadata.X=Y and samples about resources so tagged can be queried with a Ceilometer query term of the form metadata.user_metadata.X=Y. In this case the nested stacks give their VMs metadata that is passed as a nested stack parameter, and this stack passes a metadata of the form metering.stack=Y, where Y is this stack's name.

ทดสอบด้วยการเปิด browser ไปยัง หมายเลข ip จะพบหน้า install wordpress ตามที่ต้องการ และให้ทำการติดตั้ง wordpress ตามปกติ

← → C 103.27.200.247/wordpress/wp-admin/install.php



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title Change

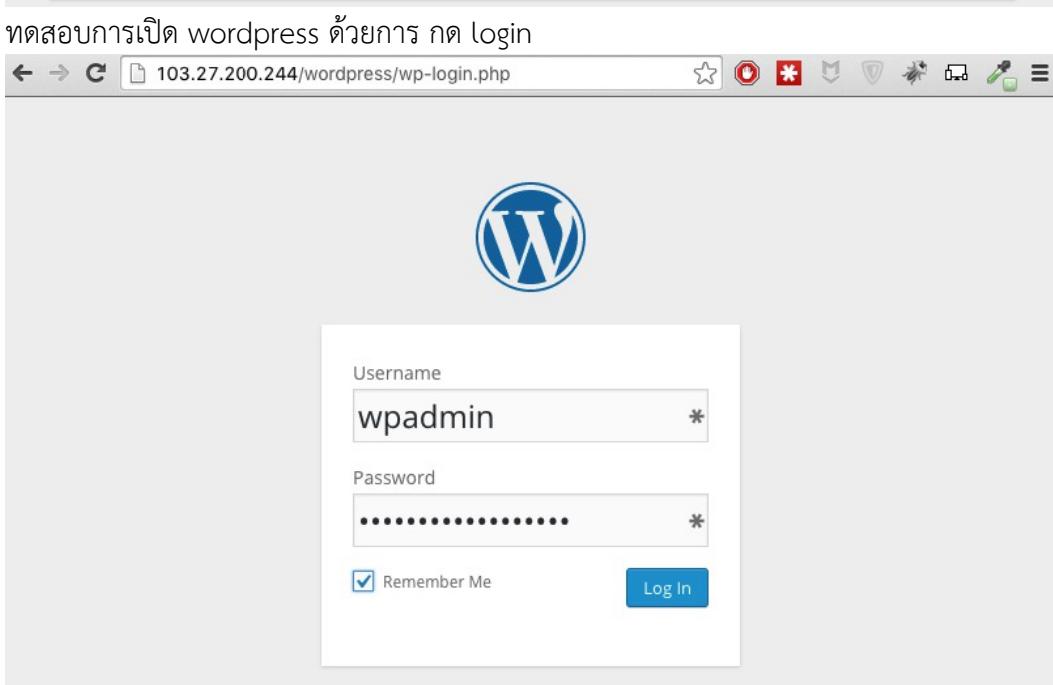
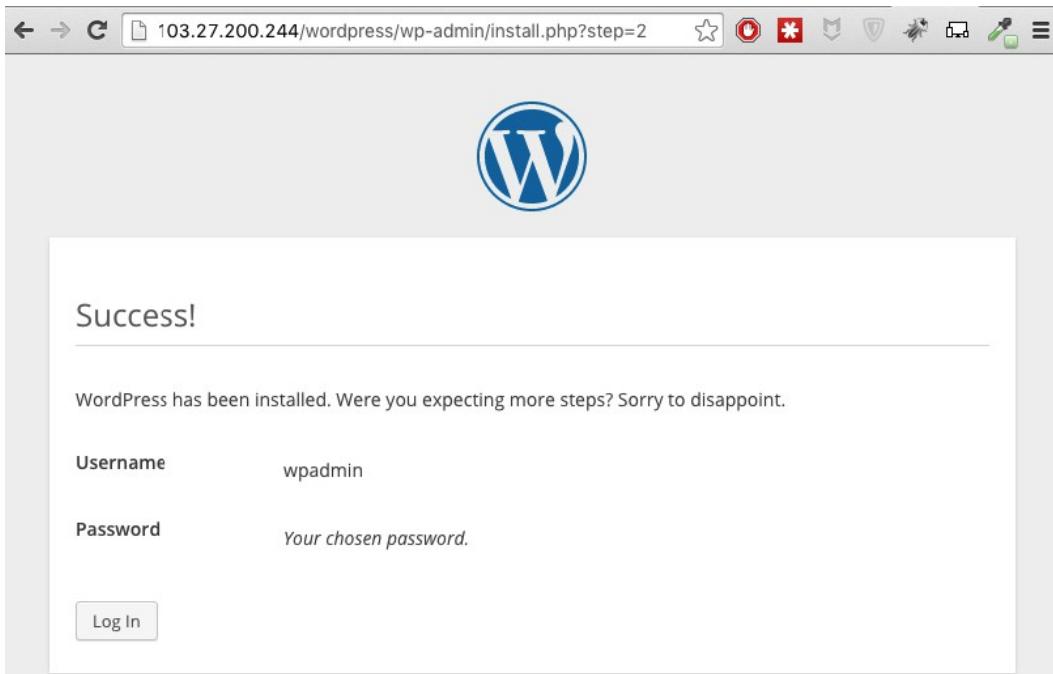
Username *
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password Change Hide
Strong
Important: You will need this password to log in. Please store it in a secure location.

Your E-mail *
Double-check your email address before continuing.

Privacy Allow search engines to index this site

ผลการติดตั้ง



หลังจากการ login จะเข้าสู่ หลังบ้านของ wordpress

Welcome to WordPress!

We've assembled some links to get you started:

- Get Started**
 - Customize Your Site
 - or, change your theme completely
- Next Steps**
 - Write your first blog post
 - Add an About page
 - View your site
- More Actions**
 - Manage widgets or menus
 - Turn comments on or off
 - Learn more about getting started

หน้าเวป

Wordpress AutoScale

Just another WordPress site

Search ...

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

RECENT POSTS

รหัส password ของ Database server

เนื่องจากการสร้าง database server กำหนดให้มีการสร้าง password จากการสุ่ม ดังนั้น เราจึงจำเป็นต้องทราบว่า password ที่สุ่มคืออะไร เพื่อไปยัง log instance ของ stack5-db ที่เป็น Database node เพื่อเก็บ password ของ root database

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions	
la-grst-tkgvrglnunit-07s222wntdx4-server-xo2mjllyfddpd	CentOS-7-x86_64	192.168.10.6	m1.small	cloudkey1	Active	nova	None	Running	6 minutes	<button>Create Snapshot</button>	
lab5-db-df5umz7xmdbe	CentOS-7-x86_64	192.168.10.5	m1.small	cloudkey1	Active	nova	None	Running	6 minutes	<button>Create Snapshot</button>	
Repo_server	CentOS-7-x86_64	192.168.10.3	Floating IPs: 103.27.200.245	m1.small	cloudkey1	Active	nova	None	Running	20 minutes	<button>Create Snapshot</button>

Displaying 3 items

หน้า 104/151

ดู log --password=l2loA5M3ezNoIJOrlVtOpUiJDV3zDoQP

```

cloud-init[4267]: firewall-cmd --permanent --add-port=3306/tcp
cloud-init[4267]: /var/lib/cloud/instance/scripts/part-001: line 13: firewall-cmd: command not found
cloud-init[4267]: firewall-cmd --reload
cloud-init[4267]: /var/lib/cloud/instance/scripts/part-001: line 14: firewall-cmd: command not found
cloud-init[4267]: mysqladmin -u root password l2loA5M3ezNoIJOrlVtOpUiJDV3zDoQP
cloud-init[4267]: cat << EOF | mysql -u root --password=l2loA5M3ezNoIJOrlVtOpUiJDV3zDoQP
cloud-init[4267]: CREATE DATABASE wordpress;
cloud-init[4267]: GRANT ALL PRIVILEGES ON wordpress.* TO "wordpress"@"%"
cloud-init[4267]: IDENTIFIED BY "gPV5k5v0bfmB1fW7jam3MTUSC1CucleX";
cloud-init[4267]: FLUSH PRIVILEGES;
cloud-init[4267]: EXIT
cloud-init[4267]: EOF
ci-info =====+Authorized Keys from /home/centos/.ssh/authorized_keys for user centos=====
ci-info +-----+
ci-info | Keytype | Fingerprint (md5) | Options | Comment |
ci-info +-----+
ci-info | ssh-rsa | 23:5e:1e:c0:46:64:da:fa:94:0f:1e:7b:30:68:8b:80 | - | Generated-by-Nova |
ci-info +-----+

```

Horizontal จะทำหน้าที่ในการสร้าง topology ให้เห็นดังรูป

เมื่อตรวจสอบ Instance ผ่านทางเมนูของ compute > Instances

openstack demo demo

Instances

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	la-grst-tkgvqrlnunit-o7s222wndtx4-server-xo2mijyfddpd	CentOS-7-x86_64	192.168.10.6	m1.small	cloudkey1	Active	nova	None	Running	10 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	lab5-db-df5umz7xmdbe	CentOS-7-x86_64	192.168.10.5	m1.small	cloudkey1	Active	nova	None	Running	10 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	Repo_server	CentOS-7-x86_64	192.168.10.3 Floating IPs: 103.27.200.245	m1.small	cloudkey1	Active	nova	None	Running	23 minutes	<button>Create Snapshot</button>

Displaying 3 items

ตรวจสอบ load balancer ที่สร้างขึ้น

openstack demo demo

Load Balancer

	Name	Description	Provider	Subnet	Protocol	Status	VIP	Actions
<input type="checkbox"/>	lab5-pool-vmf4iv6fp7pv		haproxy	192.168.10.0/24	HTTP	Active	pool.vip	<button>Edit Pool</button>

Displaying 1 item

ตรวจสอบ load balance และ network namespace

openstack demo demo

ID	dae6ad37-0ea6-4e99-a2c3-5736eeb9e6c0
Name	lab5-pool-vmf4iv6fp7pv
Description	-
Project ID	df5ded99a7a74b6896647d11126dd7bd
VIP	pool.vip
Provider	haproxy
Subnet	private_subnet 192.168.10.0/24
Protocol	HTTP
Load Balancing Method	ROUND_ROBIN
Members	192.168.10.6:80
Health Monitors	TCP delay:5 retries:5 timeout:5
Admin State Up	Yes
Status	ACTIVE

```
root@cloud ~(keystone_demo)# ip netns
qlbaas-dae6ad37-0ea6-4e99-a2c3-5736eeb9e6c0
qdhcp-6a328376-be26-4be6-a511-110ca0abf35a
qrouter-b031e9d5-ce1e-4af6-9281-2683506962ef
qrouter-5f771cd3-9b4c-4bde-8870-b84813c7618e
qlbaas-178019ba-b4cf-40cf-b5c5-e07713a036cf
qdhcp-32f4ca4c-de2d-4be2-a7fb-c2d28bb8388d
[root@cloud ~(keystone_demo)]# 
```

การทดสอบการ scale up ของ wordpress จะยึดถือแนวทางในการทดสอบ โดยจะทำการเพิ่ม floating ip ให้แก่ webserver ตัวแรก หลังจากนั้นให้ทำการ ssh เข้าไปยัง webserver ด้วยการใช้ key (ชื่อ key ที่กำหนดไว้ใน template คือ cloudkey1 มี ip 192.168.10.6

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjiyfddpd	CentOS-7-x86_64	192.168.10.6	m1.small	cloudkey1	Active	nova	None	Running	15 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	lab5-db-df5umz7xmdbe	CentOS-7-x86_64	192.168.10.5	m1.small	cloudkey1	Active	nova	None	Running	15 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	Repo_server	CentOS-7-x86_64	192.168.10.3 Floating IPs: 103.27.200.245	m1.small	cloudkey1	Active	nova	None	Running	28 minutes	<button>Create Snapshot</button>

ให้ทำการ ssh ไปยัง wordpress server ด้วยการใช้ keypair ชื่อ cloudkey1 ผ่านทาง network namespace ของ router จากเครื่อง host ดังนี้

```
# ip netns
qrouter-5f771cd3-9b4c-4bde-8870-b84813c7618e
# ls cloudkey1.pem
cloudkey1.pem
# ip netns exec qrouter-5f771cd3-9b4c-4bde-8870-b84813c7618e /bin/bash
# ls cloudkey1.pem
# ssh -i cloudkey1.pem centos@192.168.10.6
The authenticity of host '192.168.10.6 (192.168.10.6)' can't be established.
ECDSA key fingerprint is 95:c0:84:18:32:0d:2d:56:10:f8:77:d8:c4:f5:0b:99.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.6' (ECDSA) to the list of known hosts.
[centos@la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjiyfddpd ~]$ sudo su -
[root@la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjiyfddpd ~]#
```

หลังจากนั้น ทำการทดสอบ stress test และ ตรวจสอบด้วยคำสั่ง top

```
# yes > /dev/null & yes > /dev/null &
[root@la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjiyfddpd ~]# yes > /dev/null & yes > /dev/null &
[1] 10993
[2] 10994
[root@la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjiyfddpd ~]#
```

ตรวจสอบด้วย top

```

top - 19:03:41 up 21 min, 1 user, load average: 1.20, 0.35, 0.16
Tasks: 88 total, 4 running, 84 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.3 us, 0.7 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1884544 total, 1204804 free, 209256 used, 470484 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 1498308 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
10994 root 20 0 107896 616 524 R 50.2 0.0 0:28.90 yes
10993 root 20 0 107896 616 524 R 49.5 0.0 0:28.89 yes
1 root 20 0 138676 6664 3956 S 0.0 0.4 0:01.02 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0
5 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:0H
6 root 20 0 0 0 0 S 0.0 0.0 0:00.05 kworker/u2:0
7 root rt 0 0 0 0 S 0.0 0.0 0:00.00 migration/0
8 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_bh
9 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuob/0
10 root 20 0 0 0 0 S 0.0 0.0 0:00.11 rcu_sched
11 root 20 0 0 0 0 R 0.0 0.0 0:00.11 rcuos/0
12 ----

```

ให้กลับไปที่เครื่อง host ด้วยการเปิด terminal ใหม่ เพื่อตรวจสอบดูค่า ด้วยคำสั่ง ceilometer alarm-list จะพบว่า มี alarm ชื่อ lab5-cpu_alarm_high-w4qjt4725cnz มีสถานะ state (alarm)

```

[root@cloud ~]# ceilometer alarm-list
+-----+-----+-----+-----+
| Alarm ID | Name | State | Severity |
+-----+-----+-----+-----+
| 58158b9a-2634-4183-a385-221b5037e62a | lab5-cpu_alarm_low-f6eedrw7colv | ok | low |
| 6edb4aea-8aba-4893-a58e-c9bbf626bf63 | lab5-cpu_alarm_high-3t72orqufx3u | alarm | low |
+-----+-----+-----+-----+

```

```
# tail -f /var/log/ceilometer/alarm-evaluator.log
```

```

[root@cloud ~]# tail -f /var/log/ceilometer/alarm-evaluator.log
2015-12-05 01:59:35.981 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:00:35.982 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:01:35.984 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:02:35.982 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:03:07.1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:03:36.017 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:02:35.982 1390 INFO ceilometer.alarm.evaluator [-] alarm 58158b9a-2634-4183-a385-221b5037e62a transitioning to ok because Transit
2015-12-05 02:03:36.073 1390 INFO ceilometer.alarm.evaluator [-] alarm 6edb4aea-8aba-4893-a58e-c9bbf626bf63 transitioning to alarm because Trar
2015-12-05 02:04:35.999 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:05:35.993 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:06:36.040 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms
2015-12-05 02:07:36.277 1390 INFO ceilometer.alarm.service [-] initiating evaluation cycle on 2 alarms

```

ผลที่ได้จากการ scale Alarm web server scaleup_policy, new state alarm

```

INFO heat.engine.service [req-57ead746-9eb4-44a2-9f97-669395336464 -] Service c4b01498-0ca7-40a2-ab82-02a5e82907ea is updated
INFO heat.engine.resources.openstack.heat.scaling_policy [-] Alarm web_server_scaleup_policy, new state alarm
INFO heat.engine.resources.openstack.heat.scaling_policy [-] web_server_scaleup_policy Alarm, adjusting Group asg with id lab5-asg-vauyi5k6grst
INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
INFO heat.engine.environment [-] Registering https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-wpenv.yaml -> https://raw.github
INFO heat.common.urlfetch [-] Fetching data from https://raw.githubusercontent.com/thaiopen/privatecloud/master/lb-wpenv.yaml
INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance
INFO heat.engine.environment [-] Registering OS::Heat::ScaledResource -> AWS::EC2::Instance

```

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	la-grst-z5pq5ucrqge4-yoknd2qj06rk-server-6b4tbddm7q4o	CentOS-7-x86_64	192.168.10.8	m1.small	cloudkey1	Active	nova	None	Running	1 minute	<button>Create Snapshot</button>
<input type="checkbox"/>	la-grst-ylwftxd67rcz-xzvmurcniwpc-server-2lizmvfxzh5y	CentOS-7-x86_64	192.168.10.7	m1.small	cloudkey1	Active	nova	None	Running	3 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjijyfddpd	CentOS-7-x86_64	192.168.10.6	m1.small	cloudkey1	Active	nova	None	Running	25 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	lab5-db-df5umz7xmdbe	CentOS-7-x86_64	192.168.10.5	m1.small	cloudkey1	Active	nova	None	Running	25 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	Repo_server	CentOS-7-x86_64	192.168.10.3 Floating IPs: 103.27.200.245	m1.small	cloudkey1	Active	nova	None	Running	38 minutes	<button>Create Snapshot</button>

Displaying 5 items

หลังจากนั้น ให้หยุด stress test

```
[root@la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjijyfddpd ~]# kill -9 10994
[root@la-grst-tkgvrqlnunit-o7s222wndtx4-server-xo2mjijyfddpd ~]# kill -9 10993
[2]+  Killed                      yes > /dev/null
```

ทำให้เกิด alarm web_server_scaledown_policy

```
heat.engine.service [req-71d58f09-6bc3-4776-a602-059eee7a7098 - -] Service 3fc9765a-d120-4287-ba2a-908c08700c07 is updated
heat.engine.service [req-6d92c095-9116-4008-bb4f-4c771cef7b7 - -] Service c4b01498-0ca7-40a2-ab82-02a5e82907ea is updated
heat.engine.service [req-e3e04b43-792d-42c0-9b9e-f13bf7af795 - -] Service 40671e15-738f-4bdb-b31e-608172577b4f is updated
heat.engine.resources.openstack.heat.scaling_policy [-] Alarm web_server_scaledown_policy, new state alarm
heat.engine.resources.openstack.heat.scaling_policy [-] web_server_scaledown_policy NOT performing scaling action, cooldown 60
heat.engine.service [req-212183ab-92bf-4cdc-9b87-e42b4f258cd4 - -] Service 523cdeb9-29b8-4666-ab16-4fd9525a6ce0 is updated
heat.engine.service [req-cd217393-3ee1-419b-ba21-92b2713d8654 - -] Service 8cebd83a-9b46-4817-9e39-083c7fc0288c is updated
heat.engine.service [req-8a8b899f-88ac-462a-adb0-1244b055b17c - -] Service 36d554df-efff-4b4d-a811-5fff8de21556 is updated
```

openstack demo

Project Compute Overview Instances

Volumes Images Access & Security Network Object Store Orchestration Database Data Processing

Instances

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	la-grst-z5pq5ucrqge4-yoknd2qj06fk-server-6b4tbdm7q4o	CentOS-7-x86_64	192.168.10.8	m1.small	cloudkey1	Active	nova	None	Running	6 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	lab5-db-df5umz7xmdbe	CentOS-7-x86_64	192.168.10.5	m1.small	cloudkey1	Active	nova	None	Running	29 minutes	<button>Create Snapshot</button>
<input type="checkbox"/>	Repo_server	CentOS-7-x86_64	192.168.10.3 Floating IPs: 103.27.200.245	m1.small	cloudkey1	Active	nova	None	Running	43 minutes	<button>Create Snapshot</button>

Displaying 3 items

การทดสอบการทำงานระหว่างการ scaleup และ scaledown

เป้าหมายการทดสอบอีกประการหนึ่งนอกเหนือจากการที่จะมีความสามารถในการ scale up หรือ scale down เท่านั้น ดังนั้นการทดสอบนี้จะแสดงให้เห็นถึงการใช้งาน cloud application มีรูปแบบการทดสอบดังรูป



1 Add content
2 Change Theme

Load Balance

Add stress test



หลังจากที่เพิ่ม Content เรียบร้อยแล้ว ก็ให้ยกเลิก Stress Test รองรับทั้งลดลงเหลือ 1 instance และตรวจสอบดูว่า content ที่เพิ่มขึ้น และ theme ยังคงอยู่หรือไม่



Check Content

Load Balance

remove
stress test



ตรวจสอบความพร้อม

1 ตรวจสอบ network namespace ของ load balanace

```
[root@cloud ~]# ip netns
qlbaas-dae6ad37-0ea6-4e99-a2c3-5736eeb9e6c0
qdhcp-6a328376-be26-4be6-a511-110ca0abf35a
qrouter-b031e9d5-ce1e-4af6-9281-2683506962ef
qrouter-5f771cd3-9b4c-4bde-8870-b84813c7618e
qlbaas-178019ba-b4cf-40cf-b5c5-e07713a036cf
qdhcp-32f4ca4c-de2d-4be2-a7fb-c2d28bb8388d
[root@cloud ~]# ip netns exec qrouter-5f771cd3-9b4c-4bde-8870-b84813c7618e /bin/bash
[root@cloud ~]#
[root@cloud ~]# ssh -i cloudkey1.pem centos@192.168.10.8
The authenticity of host '192.168.10.8 (192.168.10.8)' can't be established.
ECDSA key fingerprint is 15:5b:ca:28:01:39:10:6b:8a:3d:30:30:45:11:6b:37.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.8' (ECDSA) to the list of known hosts.
[centos@la-grst-z5pq5ucrque4-yoknd2qjo6rk-server-6b4tbddm7q4o ~]$ sudo su -
[root@la-grst-z5pq5ucrque4-yoknd2qjo6rk-server-6b4tbddm7q4o ~]# yes > /dev/null & yes > /dev/null &
[1] 10959
[2] 10960
[root@la-grst-z5pq5ucrque4-yoknd2qjo6rk-server-6b4tbddm7q4o ~]#
```

2 ทดสอบจำนวน instance ที่เกิดขึ้น

```
[root@cloud ~]# nova list
+-----+-----+-----+-----+
| ID      | Name          | Status | Task State | Power State |
+-----+-----+-----+-----+
| c2569fb2-58b8-4c83-9679-5e044f268cd6 | Repo_server   | ACTIVE | -           | Running    |
| 90a7dfe5-955d-4de5-bcdf-a4b844be5bd2 | la-grst-gpebps6b6fqc-xi67hla7pqph-server-ezy6smd4ebiy | BUILD  | spawning   | NOSTATE   |
| 45bedbc2-4787-465e-87a8-80de9954d5c4 | la-grst-kqnjodj37dz-7fucvgkvypeo-server-okxm5riwkhc2 | ACTIVE | -           | Running    |
| 28becf2a-c4ee-40c4-a3ac-7c08ac3a3d8b | la-grst-z5pq5ucrque4-yoknd2qjo6rk-server-6b4tbddm7q4o | ACTIVE | -           | Running    |
| 793f3d5e-3b12-48cc-ba7b-1b3e0e2ca3f5 | lab5-db-df5umz7xmdbe   | ACTIVE | -           | Running    |
+-----+-----+-----+-----+
```

3 ทดสอบด้วยการเปลี่ยน Themes

The screenshot shows the WordPress Admin Themes page at <http://103.27.200.247/wordpress/wp-admin/themes.php>. The left sidebar is visible with 'Appearance' selected. The main area displays three themes: 'Twenty Fifteen' (active), 'Twenty Fourteen' (highlighted with a red arrow), and 'Twenty Thirteen'. Each theme has a preview image and a 'Theme Details' button.

4 ทดสอบด้วยการเปลี่ยน content เดิม

The screenshot shows the WordPress Admin Edit Post page at <http://103.27.200.247/wordpress/wp-admin/post.php?post=1&action=edit>. The left sidebar is visible with 'Posts' selected. The main area shows a post titled 'Hello world! < ทดสอบการเปลี่ยนแปลง เนื้อหาระหว่าง การ scale >'. The content editor shows the text 'ทดสอบการเปลี่ยนแปลง |'. The right sidebar shows publishing options: Status: Published, Visibility: Public, Published on: Dec 4, 2015 @ 18:46, Move to Trash, and Update.

5 ทดสอบด้วยการ post เพิ่มเติม post

The screenshot shows the WordPress Admin Add New Post page at <http://103.27.200.247/wordpress/wp-admin/post-new.php>. The left sidebar is visible with 'Posts' selected. The main area shows a new post titled 'ทดสอบการเพิ่มนื้อหาใหม่'. The content editor shows the text 'ทดสอบการเพิ่มนื้อหาใหม่ระหว่างการ scale'. The right sidebar shows publishing options: Status: Draft, Visibility: Public, Publish immediately, Move to Trash, and Publish.

6 การทดสอบ

The screenshot shows a WordPress website at the URL 103.27.200.247/wordpress/. The sidebar contains a search bar, recent posts (including 'Hello world!'), recent comments (including 'Mr WordPress on Hello world!'), archives (December 2015), and categories (Uncategorized). The main content area displays two posts:

- ทดสอบการเพิ่มเนื้อหาใหม่** (December 4, 2015) - This post contains the text 'Hello world! < ทดสอบการเปลี่ยนแปลงเนื้อหาระหว่างการ scale'.
- Hello world! < ทดสอบการเปลี่ยนแปลงเนื้อหาระหว่างการ scale** (December 4, 2015) - This post contains the text 'Welcome to WordPress. This is your first post. Edit or delete it, then start writing!'.

7 ยกเลิกการ stress test

```
[root@la-grst-z5pq5ucrqge4-yoknd2qjo6rk-server-6b4tbddm7q4o ~]# yes > /dev/null & yes > /dev/null &
[1] 10959
[2] 10960
[root@la-grst-z5pq5ucrqge4-yoknd2qjo6rk-server-6b4tbddm7q4o ~]# kill -9 10959
[root@la-grst-z5pq5ucrqge4-yoknd2qjo6rk-server-6b4tbddm7q4o ~]# kill -9 10960
[1]- Killed yes > /dev/null
[root@la-grst-z5pq5ucrqge4-yoknd2qjo6rk-server-6b4tbddm7q4o ~]# ]
```

จะเกิด Alarm scale down จาก ceilometer มายัง heat engine

```
i22 INFO heat.engine.service [req-ddde6063-c64e-496e-9278-91e9f0474fa0 - -] Service 40671e15-738f-4bdb-b31e-608172577b4f is updated
i24 INFO heat.engine.service [req-5cb59371-07a6-4108-b441-dd794c06daze - -] Service c4001498-0ca7-40a2-a082-02a5e8290/ea is updated
i23 INFO heat.engine.resources.openstack.heat.scaling_policy [!] Alarm web_server_scaledown_policy, new state alarm
i23 INFO heat.engine.resources.openstack.heat.scaling_policy [!] web_server_scaledown_policy NOT performing scaling action, cooldown 60
i25 INFO heat.engine.service [req-1e7c3d53-00ca-434c-929d-1b69fe21d21b - -] Service 8cebd83a-9b46-4817-9e39-083c7fc0288c is updated
i19 INFO heat.engine.service [req-aba08bc7-195c-40db-a2df-39ff58c412b3 - -] Service 523cae09-29b8-4566-a016-4fd9525a6ce0 is updated
i21 INFO heat.engine.service [req-fc35ae58-8ea2-43ef-a822-37f684685665 - -] Service 36d554df-efff-4b4d-a811-5fff8de21556 is updated
```

รอให้ cool down รอนานเหลือ 1 instance ทดสอบด้วย nova list อีกครั้ง

ID	Name	Status	Task State
c2569f82-58b8-4c83-9679-5e044f268cd6	Repo_server	ACTIVE	-
90a7dfe5-955d-4de5-bcdf-a4b844be5bd2	la-grst-gpebps6b6fqc-xi67hla7pqph-server-ezy6smd4ebiy	ACTIVE	-
793f3d5e-3b12-48cc-ba7b-1b3e0e2ca3f5	lab5-db-df5umz7xmdbe	ACTIVE	-

ทดสอบอีกครั้ง จะได้ว่าสิ่งที่แก้ไขไปในการทดสอบยังคงรักษาไว้ เช่นเดิมในระหว่างกระบวนการ scale up และ scale down

The screenshot shows a WordPress dashboard with the URL 103.27.200.247/wordpress/. The top navigation bar includes links for Customize, New, SAMPLE PAGE, and a search icon.

Recent Posts:

- ทดสอบการเพิ่มนื้อหาใหม่
- Hello world! < ทดสอบการเปลี่ยนแปลง เนื้อหาระหว่าง การscale>

Recent Comments:

- Mr WordPress on Hello world!

Archives:

- December 2015

Categories:

- Uncategorized

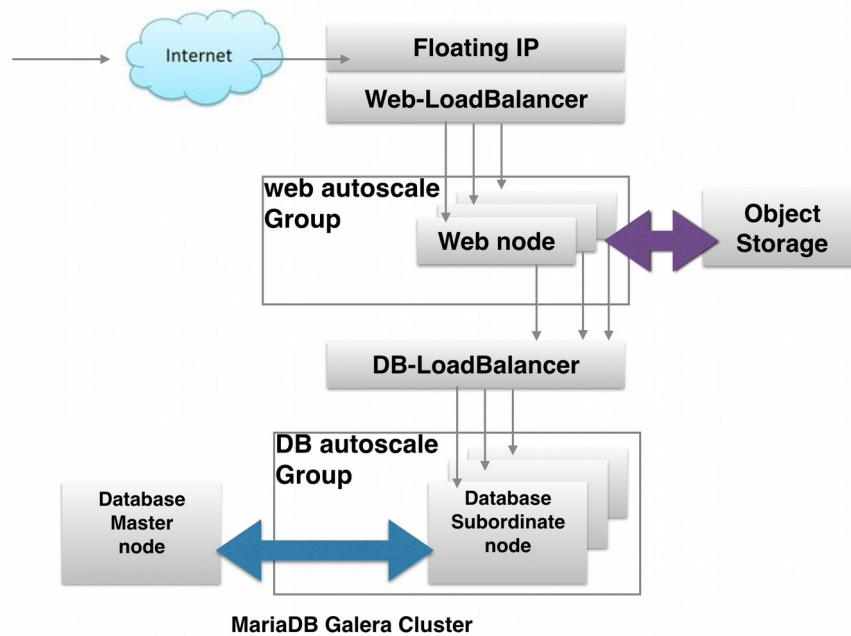
Post Details:

ทดสอบการเพิ่มนื้อหาใหม่
December 4, 2015 | Leave a Comment | Edit
ทดสอบการเพิ่มนื้อหาใหม่ระหว่างการ scale

Hello WORLD! < ทดสอบการเปลี่ยนแปลง เนื้อหาระหว่าง การSCALE>
December 4, 2015 | 1 Comment | Edit
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!
ทดสอบการเปลี่ยนแปลง

การทดสอบ 6 การสร้าง Auto scale สำหรับ Wordpress ร่วมกับ Mysql Galera Cluster

HOT template ยังสามารถนำมาสร้าง MariaDB Galera Cluster เพื่อให้สามารถรองรับการทำ Autoscale ทางด้านของ database สำหรับการทดสอบในบทที่ 6 จะเป็นการทดสอบอ้างอิงตาม Architecture ดังรูปด้านล่าง



อ้างอิง https://www-01.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liaca_bp_wordpress.html

จาก architecture นี้ จะประกอบด้วย โหนดด้วยกัน 3 ชนิดได้แก่

1 **Webnode** ทำหน้า เป็น webserver (Apache) สำหรับ wordpress อยู่ภายใต้ web loadbalance แบบ RoundRobin และมีความสามารถที่จะ web-autoscale group อัตโนมัติเมื่อ มีการใช้งาน CPU เกินกว่า 80% นอกจากนั้น แต่ละโหนดมีความสามารถที่จะ sync static Data ไปยัง Object Storage swift ภายใต้ user ที่กำหนด

2 **Database master node** (Database แม่) ทำหน้าเป็น database หลัก ติดตั้ง maria galera cluster เช่นเดียวกับ database ลูก และรองรับการ syncs data ให้แก่ database ลูกที่อยู่ภายใต้ db-autoscale group

3 **Database Subordinate node** (Database ลูก) ติดตั้ง mariadb galera cluster และมี config กำหนดให้ syncs data ไปยัง master เมื่อมี autoscale แบบอัตโนมัติเช่นเดียวกับ Webnode

โดย file ที่เกี่ยวข้องในการทดสอบนี้จะอยู่ใน workshop6 หลังที่ได้ clone repo มาแล้ว

```
$ git clone git@github.com:thaiopen/privatecloud.git
$ cd privatecloud/workshop6
$ ls
db-master.sh    db_server.yaml   web.sh      web_sql_as.yaml  db-slave.sh
example-deploy-sequence.yaml web_server.yaml
```

ไฟล์หลักที่จะใช้งานคือ web_sql_as.yaml จะทำหน้าที่เรียกใช้งานไฟล์ต่างๆ ตามลำดับ ให้เปิดไฟล์

web_sql_as.yaml และพิจารณาองค์ประกอบดังนี้ ระหว่างบรรทัดที่ 1-50 คือส่วนที่ทำหน้าเป็น parameter เหมือนกับ การทดสอบบทที่ผ่านมา

```
1 heat_template_version: 2013-05-23
2 description: AutoScaling Wordpress
3 parameters:
4   web-image:
5     type: string
6     description: Image used for web servers
7     default: CentOS-7-x86_64-heat
8   db-image:
9     type: string
10    description: Image used for db servers
11    default: CentOS-7-x86_64-heat
12   key:
13     type: string
14     description: SSH key to connect to the servers
15     default: cloudkey1
16   web-flavor:
17     type: string
18     description: flavor used by the web servers
19     default: m1.small
20   db-flavor:
21     type: string
22     description: flavor used by the db server
23     default: m1.small
24   subnet_id:
25     type: string
26     description: subnet on which the load balancer will be located
27     default: private_subnet
28   network:
29     type: string
30     description: subnet on which the load balancer will be located
31     default: private_network
32   external:
33     type: string
34     description: subnet on which the load balancer will be located
35     default: public_network
36   db_name:
37     type: string
38     description: Name of the wordpress DB
39     default: wordpress
40   db_user:
41     type: string
42     description: Name of the wordpress user
43     default: wordpress
44   # db_root_password:
45   #   type: string
46   #   default: Passw0rd
47   db_password:
48     type: string
49     default: Passw0rd
50
```

ส่วนนี้ให้ทำการกำหนดค่าต่างๆให้ตรงกับค่าที่อยู่ใน environment สังเกตว่าค่าของ subnet_id, network, external จะใช้เป็นการกำหนดด้วยชื่อ แทนการกำหนดด้วย id เพื่อความสะดวก

ส่วนต่อมาในบรรทัดที่ 51 – 80 เป็นการกำหนด Web-loadbalance (web-lb) ให้กับ web-pool โดยกำหนดเป็นการกระจาย traffic ให้แก่ webserver ที่เป็นสมาชิกใน web-pool ผ่านทาง port 80 กำหนดให้ web-pool สร้างบน subnet_id ทำให้ได้ ip ของ web จะเป็น ip เดียวกับ subnet network ของ subnet_id และมีการกำหนดให้สร้าง floating ip (บรรทัด 76) ให้แก่ web-pool เพื่อให้สามารถใช้งานจาก internet ได้

```

51 resources:
52
53   web-monitor:
54     type: OS::Neutron::HealthMonitor
55     properties:
56       type: TCP
57       delay: 5
58       max_retries: 5
59       timeout: 5
60   web-pool:
61     type: OS::Neutron::Pool
62     properties:
63       protocol: HTTP
64       monitors: [{get_resource: web-monitor}]
65       subnet_id: {get_param: subnet_id}
66       lb_method: ROUND_ROBIN
67     vip:
68       protocol_port: 80
69   web-lb:
70     type: OS::Neutron::LoadBalancer
71     properties:
72       protocol_port: 80
73       pool_id: {get_resource: web-pool}
74
75
76   wb-floating:
77     type: OS::Neutron::FloatingIP
78     properties:
79       floating_network_id: {get_param: external}
80       port_id: {get_attr: [web-pool, vip, port_id]}
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

ต่อมา จะเป็นการสร้าง webserver โดยคุณสมบัติของ web ที่สร้างนั้นจะกำหนดโดยใน file ชื่อ web_server.yaml

```

81
82   web_server_group:
83     type: OS::Heat::AutoScalingGroup
84     properties:
85       min_size: 1
86       max_size: 2
87     resource:
88       type: web_server.yaml
89     properties:
90       flavor: {get_param: web-flavor}
91       image: {get_param: web-image}
92       key_name: {get_param: key}
93       network: {get_param: network}
94       pool_id: {get_resource: web-pool}
95       metadata: {"metering.stack": {get_param: "OS::stack_id"}}
96       db_host: {get_attr: [db-pool, vip, address]}
97       db_name: {get_param: db_name}
98       db_user: {get_param: db_user}
99       db_password: {get_param: db_password}
100

```

โดย file หลักจะทำหน้าที่ส่งผ่าน ค่าตัวแปรที่ระบุไว้ใน properties ได้แก่ flavor, image, key_name เป็นต้น ไปใช้ต่อไปใน web_server.yaml สำหรับการสร้าง webserver และเนื่องจากเป็นการกำหนดชนิดเป็น OS::Heat::AutoScalingGroup ทำให้ webserver ที่สร้างจะมีคุณสมบัติของการ autoscale และการกำหนดคุณสมบัติของการทำ autoscale จะระบุไว้ที่บรรทัด ที่ 101-140

```

100
101   web_server_scaleup_policy:
102     type: OS::Heat::ScalingPolicy
103     properties:
104       adjustment_type: change_in_capacity
105       auto_scaling_group_id: {get_resource: web_server_group}
106       cooldown: 300
107       scaling_adjustment: 1
108   web_server_scaledown_policy:
109     type: OS::Heat::ScalingPolicy
110     properties:
111       adjustment_type: change_in_capacity
112       auto_scaling_group_id: {get_resource: web_server_group}
113       cooldown: 300
114       scaling_adjustment: -1
115   web_cpu_alarm_high:
116     type: OS::Ceilometer::Alarm
117     properties:
118       description: Scale-up if the average CPU > 80% for 1 minute
119       meter_name: cpu_util
120       statistic: avg
121       period: 60
122       evaluation_periods: 1
123       threshold: 80
124       alarm_actions:
125         - {get_attr: [web_server_scaleup_policy, alarm_url]}
126       matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
127       comparison_operator: gt
128   web_cpu_alarm_low:
129     type: OS::Ceilometer::Alarm
130     properties:
131       description: Scale-down if the average CPU < 30% for 1 minutes
132       meter_name: cpu_util
133       statistic: avg
134       period: 60
135       evaluation_periods: 1
136       threshold: 30
137       alarm_actions:
138         - {get_attr: [web_server_scaledown_policy, alarm_url]}
139       matching_metadata: {'metadata.user_metadata.stack': {get_param: "OS::stack_id"}}
140       comparison_operator: lt
141

```

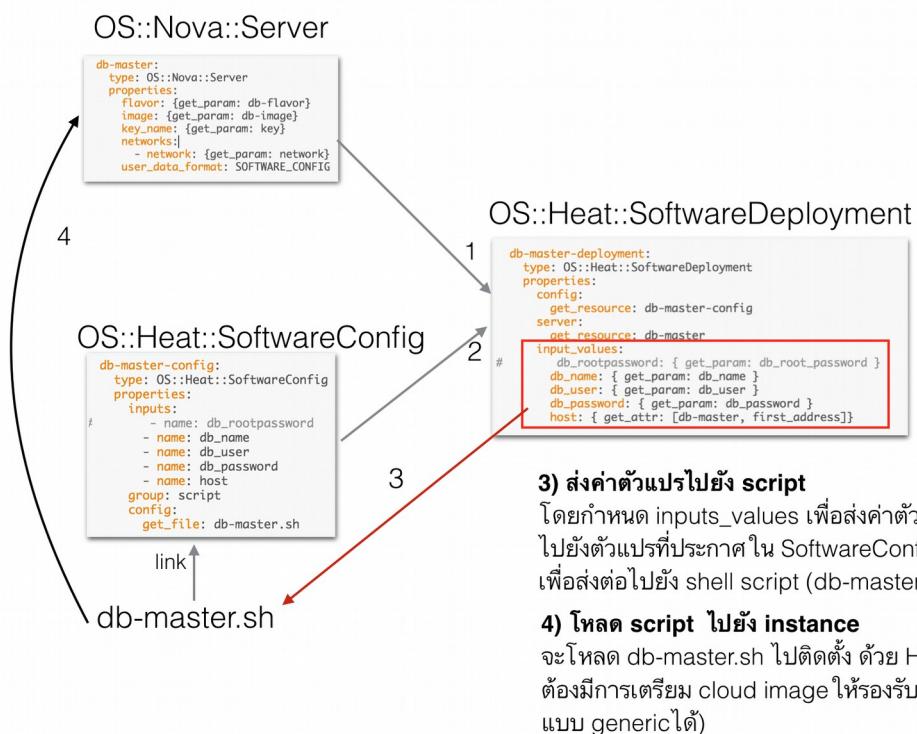
ต่อมาเป็นส่วนที่ใช้สร้าง DB-loadbalancer โดยมีโครงสร้างลักษณะของ Web-loadbalancer แต่ไม่มีการกำหนด floating ip เพราะเป็นการใช้งานภายใน private network ไม่จำเป็นจะต้องเข้าถึงจากภายนอก จาก code ด้านล่าง db-pool ทำหน้าที่เป็น Pool สำหรับ database server และนำเอา pool ที่ได้ไปสร้าง db-lb (Loadbalancer) เพื่อให้สามารถกระจายโหลดกลับมายัง database server ที่เป็นสมาชิกของ db-pool โดยจะใช้งาน port 3306 (Mysql Standard port)

```

143 db-monitor:
144     type: OS::Neutron::HealthMonitor
145     properties:
146         type: TCP
147         delay: 5
148         max_retries: 5
149         timeout: 5
150 db-pool:
151     type: OS::Neutron::Pool
152     properties:
153         protocol: TCP
154         monitors: [{get_resource: db-monitor}]
155         subnet_id: {get_param: subnet_id}
156         lb_method: ROUND_ROBIN
157         vip:
158             protocol_port: 3306
159 db-lb:
160     type: OS::Neutron::LoadBalancer
161     properties:
162         protocol_port: 3306
163         pool_id: {get_resource: db-pool}
164

```

สำหรับการสร้าง โนนดที่ต้องการให้เป็น database นั้นจะทำการติดตั้ง Mariadb Galera cluster เทคนิคการสร้าง จะใช้ resource ที่ชื่อว่า OS::Heat::SoftwareDeployment โดยจะทำหน้าที่รับ instance ที่สร้างมาจาก OS::Nova::Server มาทำการติดตั้งค่า config ที่ประกาศใน OS::Heat::SoftwareConfig ที่ทำหน้าที่ระบุว่าจะใช้งาน shell script ไฟล์ใดในการจะติดตั้ง หากมองในมุมมองของ workflow จะได้ขั้นตอนดังรูปด้านล่าง ขอให้ทำความเข้าใจขั้นตอนก่อนที่จะไปพิจารณาส่วนของ code ต่อไป



จากรูป OS::Heat::SoftwareDeployment มีประกอบด้วย properties หลักคือ config, server, input_values จะ

อ่านค่า db-master-config มาในส่วนของ config: และโหลดค่า db-master มาในส่วนของ server: และจะทำหน้าที่ ส่งผ่านค่าตัวแปร ใน input_values ไปยัง script ที่ระบุไว้ใน OS::Heat::SoftwareConfig ในที่นี่คือ db-master.sh โดยการส่งผ่านค่าตัวแปรดังกล่าวสามารถเรียกใช้งานผ่านทาง shell variable เช่น ถ้าตัวแปรที่ส่งไปคือ db_name เมื่อเรียกใช้งานจะสามารถเรียกได้เป็น \$db_name เป็นต้น ในส่วนนี้จะอยู่ในบรรทัด 165-200

```

165  db-master:
166    type: OS::Nova::Server
167    properties:
168      flavor: {get_param: db-flavor}
169      image: {get_param: db-image}
170      key_name: {get_param: key}
171      networks:
172        - network: {get_param: network}
173      user_data_format: SOFTWARE_CONFIG
174
175  db-master-config:
176    type: OS::Heat::SoftwareConfig
177    properties:
178      inputs:
179      #       - name: db_rootpassword
180      #       - name: db_name
181      #       - name: db_user
182      #       - name: db_password
183      #       - name: host
184      group: script
185      config:
186        get_file: db-master.sh
187
188  db-master-deployment:
189    type: OS::Heat::SoftwareDeployment
190    properties:
191      config:
192        get_resource: db-master-config
193      server:
194        get_resource: db-master
195      input_values:
196      #       db_rootpassword: { get_param: db_root_password }
197      #       db_name: { get_param: db_name }
198      #       db_user: { get_param: db_user }
199      #       db_password: { get_param: db_password }
200      host: { get_attr: [db-master, first_address] }
201

```

ขอให้สังเกตุ เพิ่มเติมดังนี้

ในบรรทัดที่ 172 จะได้ว่า db-master ที่สร้างขึ้นจะอยู่ใน network (จะได้ ip CIDR เดียวกับ private network)

ในบรรทัดที่ 180-183 เป็นการกำหนดค่าตัวแปรภายใต้ค่า key ชื่อว่า name และมี value เป็น ชื่อตัวแปร ที่จะต้องใช้

ในบรรทัดที่ 200 คือการตั้งค่าของ ip ของ instance db-master เพื่อจะส่งต่อไปยัง script db-master.sh

แต่สิ่งที่เป็นหัวใจสำคัญของการทำงานคือการ load นำเอา script ไปทำงานติดตั้งภายใน instance ซึ่งจะ ต้องเรียกใช้งาน heat api จากราบภายใน instance เอง จึงเป็นเหตุผลว่าจะต้องมีการเตรียมสร้าง image ให้สามารถใช้งานกับ OS::Heat::SoftwareDeployment ไม่สามารถใช้งาน image แบบพื้นฐาน โดยได้สร้างเป็น script ชื่อ file:

create-heat-image.sh โดยมีเนื้อหาดังนี้

```
#!/bin/bash
yum install qemu-img
yum install python-pip git
pip install git+git://git.openstack.org/openstack/dib-utils.git

export ELEMENTS_PATH=tripleo-image-elements/elements:heat-templates/hot/software-
config/elements
export BASE_ELEMENTS="centos7 selinux-permissive"
export AGENT_ELEMENTS="os-collect-config os-refresh-config os-apply-config"
export DEPLOYMENT_BASE_ELEMENTS="heat-config heat-config-script"

export IMAGE_NAME=software-deployment-image
diskimage-builder/bin/disk-image-create vm $BASE_ELEMENTS $AGENT_ELEMENTS
$DEPLOYMENT_BASE_ELEMENTS $DEPLOYMENT_TOOL -o $IMAGE_NAME.qcow2

glance image-create --name CentOS-7-x86_64-heat --disk-format qcow2 --container-format
bare --file software-deployment-image.qcow2
```

ส่วนต่อมาถือได้ว่าเป็นส่วนสำคัญของการสร้าง cluster ของ database ในบรรทัดที่ 202-216 และคุณสมบัติของ db-server ที่สร้างเพื่อให้เป็น cluster และจะ sync data กับ db-master ที่สร้างก่อนหน้านี้ และการสร้างตั้งกล่าว เป็นการสร้าง ชนิด OS::Heat::AutoScalingGroup โดยกำหนดให้สร้าง ภายใต้ db-pool

```
202   db_slave_group:
203     type: OS::Heat::AutoScalingGroup
204     properties:
205       min_size: 1
206       max_size: 5
207       resource:
208         type: db_server.yaml
209         properties:
210           flavor: {get_param: db-flavor}
211           image: {get_param: db-image}
212           key_name: {get_param: key}
213           network: {get_param: network}
214           pool_id: {get_resource: db-pool}
215           metadata: {"metering.stack": '44e01704-bb00-41e0-bdf0-84456cc4eb27'}
216           master: { get_attr: [ db-master, first_address ]}
217           #           db_rootpassword: { get_param: db_root_password }
```

ส่วนคุณสมบัติของ Autoscale จะอยู่ใน code ระหว่าง บรรทัดที่ 219-259

```
218 |
219   db_server_scaleup_policy:
220     type: OS::Heat::ScalingPolicy
221     properties:
222       adjustment_type: change_in_capacity
223       auto_scaling_group_id: {get_resource: db_slave_group}
224       cooldown: 300
225       scaling_adjustment: 1
226   db_server_scaledown_policy:
227     type: OS::Heat::ScalingPolicy
228     properties:
229       adjustment_type: change_in_capacity
230       auto_scaling_group_id: {get_resource: db_slave_group}
231       cooldown: 300
232       scaling_adjustment: -1
233
```

```

233
234 db_cpu_alarm_high:
235   type: OS::Ceilometer::Alarm
236   properties:
237     description: Scale-up if the average CPU > 80% for 1 minute
238     meter_name: cpu_util
239     statistic: avg
240     period: 60
241     evaluation_periods: 1
242     threshold: 80
243     alarm_actions:
244       - {get_attr: [db_server_scaleup_policy, alarm_url]}
245     matching_metadata: {'metadata.user_metadata.stack': '44e01704-bb00-41e0-bdf0-84456cc4eb27'}
246     comparison_operator: gt
247 db_cpu_alarm_low:
248   type: OS::Ceilometer::Alarm
249   properties:
250     description: Scale-down if the average CPU < 30% for 1 minutes
251     meter_name: cpu_util
252     statistic: avg
253     period: 60
254     evaluation_periods: 1
255     threshold: 30
256     alarm_actions:
257       - {get_attr: [db_server_scaledown_policy, alarm_url]}
258     matching_metadata: {'metadata.user_metadata.stack': '44e01704-bb00-41e0-bdf0-84456cc4eb27'}
259     comparison_operator: lt
260

```

ส่วนสุดท้ายคือ output เพื่อแสดงผลค่าของ floating ip ที่จะใช้งาน wordpress และ db-master ip

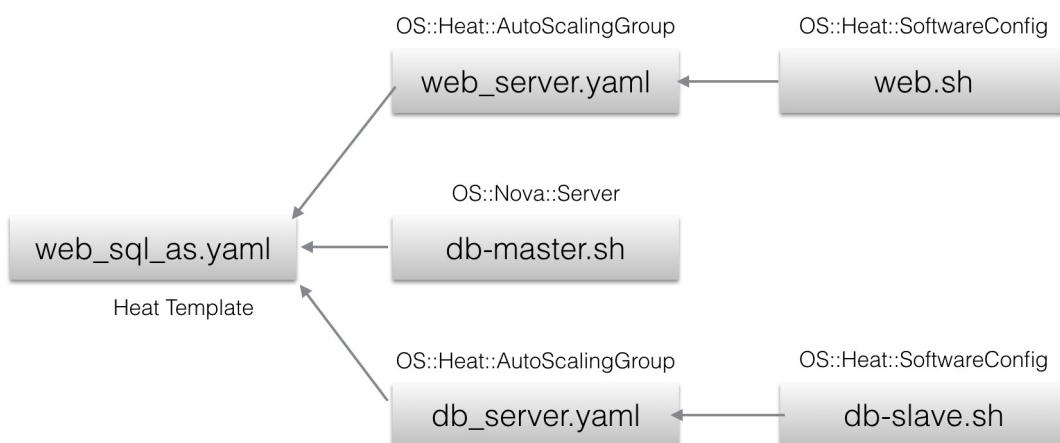
```

261
262 outputs:
263   wordpress_url:
264     value:
265       str_replace:
266         template: http://url/
267         params:
268           url: {get_attr: [wb-floating, floating_ip_address]}
269     description: The url of wordpress blog
270
271 master:
272   value: { get_attr: [ db-master, first_address ]}
273   description: The ip address of master instance
274

```

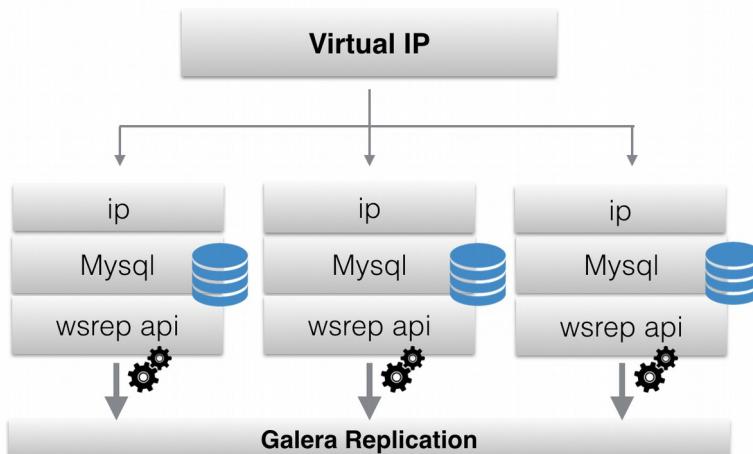
โครงสร้างความสัมพันธ์ระหว่างไฟล์

จาก master file นี้จะมีการเรียกใช้งาน file อื่นประกอบ ดังนั้นเพื่อให้สามารถที่จะทำการเรียนรู้ หรือ debug ระบบได้นั้นขอให้พิจารณาโครงสร้าง file อีกครั้ง ก่อนที่จะดูในรายละเอียด file ย่อยๆของแต่ละส่วน



การติดตั้งการใช้งาน MariDB Galera Cluster

การติดตั้งใช้งานส่วนของ Autoscale สำหรับ Database นั้นจะติดตั้ง MariDB Galera Cluster เพื่อให้มีความสามารถในการรองรับการ sync ฐานข้อมูล โดย MariDB Galera Cluster เป็นการเพิ่มความสามารถให้กับฐานข้อมูลชนิด RDBMS ให้ทำงานเป็น cluster ด้วยการ patch code เพิ่มเติมในส่วนของ Mysql-wsrep จาก Codership โปรเจค wsrep (ย่อมาจาก Write Set REPLICATION) เป็นโปรเจคโอเพนซอร์ส มีวัตถุประสงค์เพื่อสร้าง generic pluggable replication API ให้แก่ฐานข้อมูลที่เป็นลักษณะแบบฐานข้อมูลเชิงสัมพันธ์ สามารถติดตามได้ที่ (<https://launchpad.net/wsrep>) ตามรูปภาพ ด้านล่าง



การติดตั้งการใช้งาน MariDB Galera Cluster สำหรับ database Master

ติดตั้ง repository เพื่อใช้งาน mariadb เวอร์ชัน 10.0 ในบรรทัดที่ 6-12 พร้อมกับติดตั้ง package ในบรรทัดที่ 14

```
1 #!/bin/bash -v
2 echo root:centos | chpasswd
3
4
5 setenforce 0
6 cat << MARIA > /etc/yum.repos.d/mariadb.repo
7 [mariadb]
8 name = MariaDB
9 baseurl = http://yum.mariadb.org/10.0/centos7-amd64
10 gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
11 gpgcheck=1
12 MARIA
13
14 yum install MariaDB-Galera-server MariaDB-client galera -y
15 yum install socat vim tmux -y
16
17 systemctl start mysql
18 systemctl enable mysql
19
```

ติดตั้ง firewall เพื่อเปิด port 3306, 4444, 4567, 4568, 22

```
20 yum install firewalld -y
21 systemctl start firewalld
22 systemctl enable firewalld
23 for i in 3306 4444 4567 4568 22; do sudo firewall-cmd --permanent --add-port=$i/tcp; done;
24 firewall-cmd --reload
```

สร้าง user สำหรับ wsrep_sst และมี password คือ wpass ในทุก node ของ cluster และสำหรับ master node ก็ให้สร้าง wordpress database ในบรรทัดที่ 34-35 โดยรับค่าจาก \$db_name, \$db_password , \$db_user มาจาก heat (wordpress database จะถูก replicate ไปยัง โนนดอื่นอัตโนมัต)

```

25 mysql -uroot -e "SET wsrep_on=OFF; GRANT ALL ON *.* TO wsrep_sst@'%' IDENTIFIED BY 'wpass';"
26 mysql -uroot -e "SET wsrep_on=OFF; DELETE FROM mysql.user WHERE user=''";
27 mysql -uroot -e "FLUSH PRIVILEGES"
28
29 #set wordpress database
30 echo "$db_name:$db_name:$db_user:$db_user/$db_password:$db_password" > /root/db-master-vardb.txt
31 cat << EOF | mysql -u root
32 #GRANT ALL PRIVILEGES ON *.* TO "sst"@"%" IDENTIFIED BY "Passw0rd";
33 CREATE DATABASE $db_name;
34 GRANT ALL PRIVILEGES ON $db_name.* TO "$db_user"@"%" IDENTIFIED BY "$db_password";
35 FLUSH PRIVILEGES;
36 EXIT
37 EOF

```

สร้าง template script ของ wsrep api ชื่อ wsrep.conf ไว้ใน /etc/my.cnf.d/ มีการระบุ wsrep_provider library เป็น /usr/lib64/galera/libgalera_smm.so ในบรรทัด 52

```

40 #stop mysql before start cluster
41 systemctl stop mysql
42 cat << WSREP > /etc/my.cnf.d/wsrep.cnf
43 [mariadb-10.0]
44 #mysqlconfig
45 bind-address=0.0.0.0
46 binlog_format=row
47 default_storage_engine=InnoDB
48 innodb_autoinc_lock_mode=2
49 innodb_doublewrite=1
50
51 #wsrep config
52 wsrep_provider=/usr/lib64/galera/libgalera_smm.so
53 wsrep_cluster_name="OpenStack"
54 wsrep_sst_auth=wsrep_sst:wpass
55 wsrep_cluster_address="gcomm://"
56 wsrep_sst_method=rsync
57 wsrep_node_address="{NODE_ADDRESS}"
58 wsrep_node_name="{NODE_NAME}"
59 log-error = error.log
60 WSREP

```

แทนค่า ตัวแปร '{NODE_ADDRESS}' คือค่าของ ipaddress ที่อยู่ที่ interface eth0 การหาค่าของ ip สามารถใช้ บรรทัดที่ 63 หรือ 64 อย่างได้อย่างหนึ่งได้, '{NODE_NAME}' คือค่าของ hostname ของเครื่อง

```

61
62 export NODE_NAME=$HOSTNAME
63 export NODE_ADDRESS=$(ip addr show eth0 | grep "inet\b" | awk '{print $2}' | cut -d/ -f1)
64 #export NODE_ADDRESS=$host
65
66 sed -i "s/{NODE_ADDRESS}/$NODE_ADDRESS/" /etc/my.cnf.d/wsrep.cnf
67 sed -i "s/{NODE_NAME}/$NODE_NAME/" /etc/my.cnf.d/wsrep.cnf
68
69
70 /etc/init.d/mysql bootstrap
71 /etc/init.d/mysql start
72 sleep 5
73 sleep 5
74

```

การติดตั้งการใช้งาน MariaDB Galera Cluster สำหรับ database server ลูก

คล้ายกับการสร้างของ database master และไม่สร้าง wordpress database เพราะจะใช้วิธีการ sync database แทน

```
1 #!/bin/bash -v
2 echo root:centos | chpasswd
3
4
5 setenforce 0
6 cat << MARIA > /etc/yum.repos.d/mariadb.repo
7 [mariadb]
8 name = MariaDB
9 baseurl = http://yum.mariadb.org/10.0/centos7-amd64
10 gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
11 gpgcheck=1
12 MARIA
13
14 yum install MariaDB-Galera-server MariaDB-client galera -y
15 yum install socat vim -y
16
17 yum install firewalld -y
18 systemctl start firewalld
19 systemctl enable firewalld
20 for i in 3306 4444 4567 4568 22; do sudo firewall-cmd --permanent --add-port=$i/tcp; done;
21 firewall-cmd --reload
22
23 systemctl start mysql
24 systemctl enable mysql
25
26 mysql -uroot -e "SET wsrep_on=OFF; GRANT ALL ON *.* TO wsrep_sst@'%' IDENTIFIED BY 'wspass';"
27 mysql -uroot -e "SET wsrep_on=OFF; DELETE FROM mysql.user WHERE user='';"
28 mysql -uroot -e "FLUSH PRIVILEGES"
29
```

และในส่วนของ template นี้จะต้องใส่ค่าของ "gcomm://{{PRIMARY_NODE_IP}}" ซึ่งจะเป็น ip ของ database master node มีผลทำให้ database slave ทุกเครื่องที่สร้างใน autoscale group จะซื้อไปยัง master ทุกครั้ง

```
30 #systemctl stop mysql
31
32 cat << WSREP > /etc/my.cnf.d/wsrep.cnf
33 [mariadb-10.0]
34 #mysqlconfig
35 bind-address=0.0.0.0
36 binlog_format=row
37 default_storage_engine=InnoDB
38 innodb_autoinc_lock_mode=2
39 innodb_doublewrite=1
40
41 #wsrep config
42 wsrep_provider=/usr/lib64/galera/libgalera_smm.so
43 wsrep_cluster_name="OpenStack"
44 wsrep_sst_auth=wsrep_sst:wspass
45 wsrep_cluster_address="gcomm://{{PRIMARY_NODE_IP}}"
46 wsrep_sst_method=rsync
47 wsrep_node_address="{{NODE_ADDRESS}}"
48 wsrep_node_name="{{NODE_NAME}}"
49 log-error = error.log
50 WSREP
```

แทนค่าตัวแปร ในบรรทัดที่ 53-60

```
51
52 echo "master is: $master/host is: $host" > /root/variable.txt
53 export PRIMARY_NODE_IP=$master
54 export NODE_NAME=$HOSTNAME
55 export NODE_ADDRESS=$(ip addr show eth0 | grep "inet\b" | awk '{print $2}' | cut -d/ -f1)
56 #export NONE_ADDRESS=$host
57
58 sed -i "s/{PRIMARY_NODE_IP}/$PRIMARY_NODE_IP/" /etc/my.cnf.d/wsrep.cnf
59 sed -i "s/{NODE_ADDRESS}/$NODE_ADDRESS/" /etc/my.cnf.d/wsrep.cnf
60 sed -i "s/{NODE_NAME}/$NODE_NAME/" /etc/my.cnf.d/wsrep.cnf
61
62
63 /etc/init.d/mysql restart
64 sleep 5
65
```

การติดตั้ง webserver ด้วย script ชื่อ web.sh

ใน script จะแบ่งออกเป็น ส่วนสำคัญ ได้แก่ การติดตั้ง package ในบรรทัดที่ 3-8 ติดตั้ง apache webserver และ php php-mysql เป็นต้น พร้อมกับการ start ให้ web server ทำงาน

```
1 #!/bin/bash -v
2 echo root:centos | chpasswd
3 yum -y -q install httpd
4 yum -y install php php-gd php-mysql httpd wget tmux vim unzip
5
6 systemctl enable httpd.service
7 systemctl restart httpd.service
8 setsebool -P httpd_can_network_connect_db=1
9
```

ต่อมาจะเป็น Download wordpress ในเวอร์ชันล่าสุด จาก wordpress.org และแก้ค่า config ของ wp-config.php ด้วยการแทนค่าด้วย ตัวแปร ในบรรทัดที่ 16-21

```
10 # download wordpress
11 cd /root
12 wget http://wordpress.org/latest.tar.gz
13 rm /var/www/html/index.html
14 tar -xzf latest.tar.gz
15
16 echo "db_user:$db_user/db_name:$db_name/db_password:$db_password" > /root/web.sh.var
17 cp wordpress/wp-config-sample.php wordpress/wp-config.php
18 sed -i "s/database_name_here/$db_name/" wordpress/wp-config.php
19 sed -i "s/username_here/$db_user/" wordpress/wp-config.php
20 sed -i "s/password_here/$db_password/" wordpress/wp-config.php
21 sed -i "s/localhost/$db_host/" wordpress/wp-config.php
22
23 cp -avR wordpress/* /var/www/html/
24
25 mkdir /var/www/html/wp-content/uploads
```

ตั้งค่า firewall ให้เปิด port 80/3306

```
26
27 yum install firewalld -y
28 systemctl start firewalld
29 systemctl enable firewalld
30
31 firewall-cmd --permanent --add-port=80/tcp
32 firewall-cmd --permanent --add-port=3306/tcp
33 firewall-cmd --reload
```

การติดตั้งการใช้งาน Golang Script ชื่อ rclone (โปรเจคโอเพนซอร์ส <http://rclone.org/>) เพื่อทำหน้าที่สำหรับการ cloud sync กับ swift object storage ใน openstack ในระดับของ file system ในบรรทัดที่ 35-40 การทำงานของ rclone จะทำงานร่วมกันกับ node.js ที่จะทำหน้าที่ monitor folder ภายใน wordpress

```

34
35 yum -y install epel-release
36 yum -y install nodejs npm vim tmux git
37 yum -y install golang
38 mkdir /root/gopath
39 export GOPATH="/root/gopath/"
40 go get github.com/ncw/rclone
41 sleep 5
42 cp /root/gopath/bin/rclone /usr/bin
43 npm install fsmonitor
44 npm install -g fsmonitor
45 npm install forever
46 npm install -g forever
47
48 cat << RCLONE > /root/.rclone.conf
49 [remote]
50 type = swift
51 user = demo
52 key = isyl]y[openstack
53 auth = http://103.27.200.234:5000/v2.0
54 tenant = demo
55 region = RegionOne
56
57 RCLONE

```

ทำการ sync folder wp-content/uploads และ wp-content/themes บน objectstorage ในบรรทัดที่ 59-62 ด้วยคำสั่ง 'rclone sync <cloudfolder> <serverfolder>'

```

58
59 rclone mkdir remote:uploads
60 rclone mkdir remote:themes
61 rclone sync remote:uploads /var/www/html/wp-content/uploads
62 rclone sync remote:themes /var/www/html/wp-content/themes
63
64 cat << BACKUP > backup.js
65 var sys = require('sys');
66 var exec = require('child_process').exec;
67 function reports(error, stdout, stderr) {sys.puts(stdout)};
68 fsmonitor = require('fsmonitor');
69 ▼ fsmonitor.watch('/var/www/html/wp-content/uploads', null, function(change) {
70 console.log("Change detected:\n" + change);
71
72 console.log("Added files: %j", change.addedFiles);
73 console.log("Modified files: %j", change.modifiedFiles);
74 console.log("Removed files: %j", change.removedFiles);
75
76 console.log("Added folders: %j", change.addedFolders);
77 console.log("Modified folders: %j", change.modifiedFolders);
78 console.log("Removed folders: %j", change.removedFolders);
79 exec('rclone sync /var/www/html/wp-content/uploads remote:uploads',reports)
80 });
81
82 fsmonitor2 = require('fsmonitor');
83 ▼ fsmonitor2.watch('/var/www/html/wp-content/themes', null, function(change) {
84 console.log("Change detected:\n" + change);
85
86 console.log("Added files: %j", change.addedFiles);
87 console.log("Modified files: %j", change.modifiedFiles);
88 console.log("Removed files: %j", change.removedFiles);
89
90 console.log("Added folders: %j", change.addedFolders);
91 console.log("Modified folders: %j", change.modifiedFolders);
92 console.log("Removed folders: %j", change.removedFolders);
93 exec('rclone sync /var/www/html/wp-content/themes remote:themes', reports)
94 });
95 BACKUP
96 forever start backup.js
97
98 systemctl restart httpd.service
--
```

การสร้าง Heat template

การใช้งาน heat template จะใช้งานผ่านทาง command line ด้วยคำสั่ง heat stack-create และก่อนใช้งานคำสั่ง จะต้องทำการ source ไฟล์ demo-openrc.sh ซึ่งเป็น file credential ของ userdemo สามารถ Download file นี้ได้โดยการ log in ด้วย user demo ใน openstack และไปที่ เมนู Compute > Access & Security > Download Openstack RC file

The screenshot shows the OpenStack dashboard with the 'Access & Security' tab selected. On the right, under 'API Access', there is a section titled 'Service Endpoint' listing various services and their endpoints. At the bottom of this section is a button labeled 'Download OpenStack RC File' which is highlighted with a red box and a red arrow pointing to it.

```
source /root/demo-openrc.sh
Please enter your OpenStack Password:
<กรอกรหัสของ Demo>
git clone git@github.com:thaiopen/privatecloud.git
cd privatecloud/workshop6
```

```
heat stack-create websql -f web_sql_as.yaml
```

```
heat stack-list
```

id	stack_name	stack_status	creation_time	updated_time
409faba1-dfd2-42be-8274-32384c5604be	websql	CREATE_COMPLETE	2016-01-03T17:36:08	None

ตรวจสอบ resource-list

```
# heat resource-list 409faba1-dfd2-42be-8274-32384c5604be
```

resource_name	physical_resource_id	resource_type	resource_status
db-lb		OS::Neutron::LoadBalancer	CREATE_COMPLETE
db-master	64e6f41a-0095-450c-89a9-10591dd74f9d	OS::Nova::Server	CREATE_COMPLETE
db-master-config	5b5d8dbe-a595-4f19-9f29-6c04bcf726ca	OS::Heat::SoftwareConfig	CREATE_COMPLETE
db-master-deployment	8da88360-317a-4194-8d23-c8a60d53cc59	OS::Heat::SoftwareDeployment	CREATE_COMPLETE
db-monitor	35d1e359-eb66-4685-b05a-fcde2737bf81	OS::Neutron::HealthMonitor	CREATE_COMPLETE
db-pool	df5ef033-59b9-4ceb-af71-c028700ba0e6	OS::Neutron::Pool	CREATE_COMPLETE
db_cpu_alarm_high	74c16c79-4fc6-4eb6-aiae-c3b25857b638	OS::Ceilometer::Alarm	CREATE_COMPLETE
db_cpu_alarm_low	fbb7f3df-78f3-4ccc-95dd-fa2bd4b7b9a3	OS::Ceilometer::Alarm	CREATE_COMPLETE
db_server_scaledown_policy	6e799ac92cfb4db080d552bedbd4de28	OS::Heat::ScalingPolicy	CREATE_COMPLETE
db_server_scaleup_policy	fb9659ebcb7b44a0836d8e0befab70da	OS::Heat::ScalingPolicy	CREATE_COMPLETE
db_slave_group	f1fe3f08-56c6-425a-a4cb-fe88c9dbb69b	OS::Heat::AutoScalingGroup	CREATE_COMPLETE

03T17:36:08	aa9ff8ad-b6e4-48e3-886b-5335e80586c4 OS::Neutron::FloatingIP	CREATE_COMPLETE 2016-01-
web-floating	OS::Neutron::LoadBalancer	CREATE_COMPLETE 2016-01-
03T17:36:08	66df3f3c-91c5-4369-bacf-fa340807e085 OS::Neutron::HealthMonitor	CREATE_COMPLETE 2016-01-
web-lb	2183e72b-90f4-4d34-a225-b6dd8a636124 OS::Neutron::Pool	CREATE_COMPLETE 2016-01-
03T17:36:08	fc558f22-1048-4234-bcd3-fb1146d7aec OS::Ceilometer::Alarm	CREATE_COMPLETE 2016-01-
web-monitor	2bcc8bba-cfed-4b5f-9f7f-7b68081edc29 OS::Ceilometer::Alarm	CREATE_COMPLETE 2016-01-
03T17:36:08	4aa9c01f-4200-4057-9c68-f7960493f016 OS::Heat::AutoScalingGroup	CREATE_COMPLETE 2016-01-
web-pool	796368e1ab9b49838c919a99f83497b5 OS::Heat::ScalingPolicy	CREATE_COMPLETE 2016-01-
03T17:36:08	0672cb19327b442bb223699f81803313 OS::Heat::ScalingPolicy	CREATE_COMPLETE 2016-01-
web_cpu_alarm_high		
web_cpu_alarm_low		
web_server_group		
web_server_scaledown_policy		
web_server_scaleup_policy		
03T17:36:08		
+-----+		
+-----+		

ผลที่ได้ดังนี้เมื่อตรวจสอบฝ่าน horizon

The screenshot shows the OpenStack Horizon interface with the 'demo' project selected. In the left sidebar, under 'Orchestration', the 'Stacks' tab is active. A search bar at the top right contains the text 'Check Stack'. Below the search bar, there are tabs for 'Topology', 'Overview', 'Resources', 'Events', and 'Template'. The 'Topology' tab is selected, showing a network diagram for the 'websql' stack. The diagram consists of five main nodes: one central 'websql' node and four database nodes branching off from it. Each database node is also connected to a monitoring node (represented by a bell icon).

ตรวจสอบ web-pool (ip 192.168.10.168, floatip 103.27.200.240) และ db-pull (ip 192.168.10.167)

The screenshot shows the OpenStack Horizon interface with the 'demo' project selected. In the left sidebar, under 'Network', the 'Load Balancers' tab is active. At the top, there are tabs for 'Pools', 'Members', and 'Monitors'. Below these are buttons for '+ Add Pool' and 'Delete Pools'. A table lists the current load balancer pools:

	Name	Description	Provider	Subnet	Protocol	LB Method	Status	VIP	Admin State	Actions
<input type="checkbox"/>	websql-web-pool-t2lesabzvbyy		haproxy	private_subnet 192.168.10.0/24	HTTP	Round Robin	Active	web-pool.vip Address: 192.168.10.168 Floating IP: 103.27.200.240	UP	<button>Edit Pool</button>
<input type="checkbox"/>	websql-db-pool-eiu7liddf45q		haproxy	private_subnet 192.168.10.0/24	TCP	Round Robin	Active	db-pool.vip Address: 192.168.10.167	UP	<button>Edit Pool</button>

Displaying 2 items

ตรวจสอบ members pool จะพบว่า มี instance ของ web อุปกรณ์ที่ 192.168.10.171 และ instance ของ database ลูกอุปกรณ์ที่ 192.168.10.171

The screenshot shows the OpenStack dashboard with the 'Load Balancer' tab selected. The 'Members' tab is active. There are two entries in the table:

	IP Address	Protocol Port	Weight	Pool	Status	Admin State	Actions
<input type="checkbox"/>	192.168.10.171	80	1	websql-web-pool-t2tesabzvbyy	Active	UP	<button>Edit Member</button>
<input type="checkbox"/>	192.168.10.172	3306	1	websql-db-pool-eiu7liddf45q	Active	UP	<button>Edit Member</button>

Displaying 2 items

ตรวจสอบ instance ทั้งหมด จะพบว่า websql-db-master มี ip 192.168.10.166 (ไม่มีอยู่ใน db-pool)

The screenshot shows the OpenStack dashboard with the 'Instances' tab selected. There are three entries in the table:

	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	we-t4cjctz7-tnpruwoozsy6-db-slave-server-gizwmamwappy	CentOS-7-x86_64-heat	192.168.10.172	m1.small	cloudkey1	Active	nova	None	Running	1 day, 12 hours	<button>Create Snapshot</button>
<input type="checkbox"/>	we--awg4z3aoxxcs-wrqwtgxibypn-web-server-nzdtsf2a2qlg	CentOS-7-x86_64-heat	192.168.10.171	m1.small	cloudkey1	Active	nova	None	Running	1 day, 12 hours	<button>Create Snapshot</button>
<input type="checkbox"/>	websql-db-master-h3lwbcbmjfq	CentOS-7-x86_64-heat	192.168.10.166	m1.small	cloudkey1	Active	nova	None	Running	1 day, 12 hours	<button>Create Snapshot</button>

Displaying 3 items

ตรวจสอบ folder ใน object storage

The screenshot shows the OpenStack dashboard with the 'Object Store' tab selected. The 'Containers' tab is active. There are two entries in the table:

	Object Count: 0 Size: 0 bytes Access: Private	View Details
themes	Object Count: 0 Size: 0 bytes Access: Private	<button>View Details</button>
uploads	Object Count: 0 Size: 0 bytes Access: Private	<button>View Details</button>

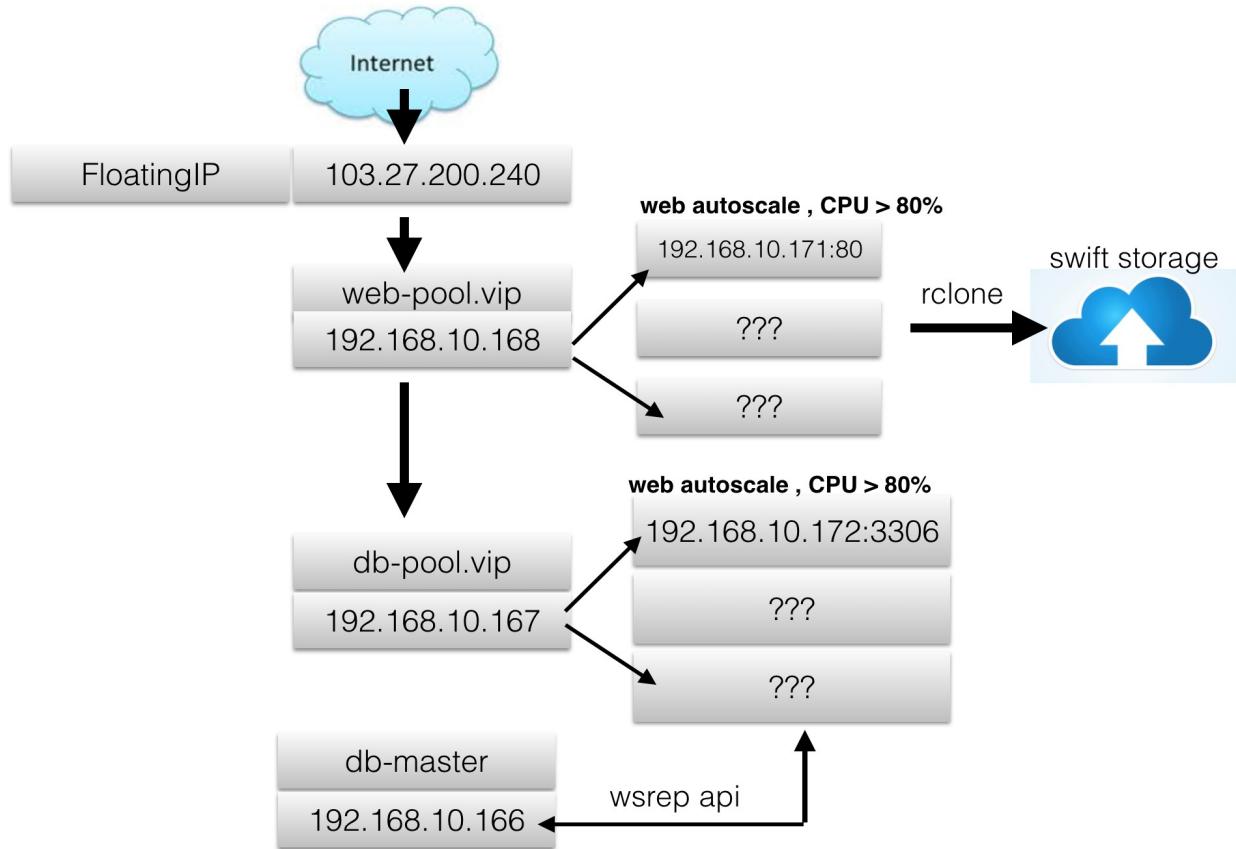
Displaying 2 items

Select a container to browse.

Displaying 0 items

โครงสร้าง Heat Template ตาม architecture

เมื่อนำเอา ip ของ instance ที่สร้าง และ loadbalance มาสร้าง เป็นภาพการเขียนโดยตาก architecture จะได้ภาพรวมดังนี้



อ้างอิง: ข้อกำหนด Heat Orchestration Template (HOT) Specification

HOT คือ Template format ที่ตั้งใจใช้สำหรับการแทนที่ CloudFormation-compatible format (CFN) ที่ใช้มาก่อน โดยเริ่มต้นการใช้งานเป็นมาตรฐานมาตั้งแต่ version icehouse (April 2014) มีโครงสร้างต่อไปนี้

```
heat_template_version: 2015-04-30

description:
  # a description of the template

parameter_groups:
  # a declaration of input parameter groups and order

parameters:
  # declaration of input parameters

resources:
  # declaration of template resources

outputs:
  # declaration of output parameters
```

heat_template_version คือ ค่า key ที่ใช้เพื่อเป็นการกำหนด version ของ YAML document
description คือ ค่า key ใช้สำหรับรายละเอียดที่เกี่ยวข้องกับ template , workload ที่จะใช้ภายใน template

parameter_groups คือ ค่า key ที่ใช้สำหรับ กำหนดลักษณะวิธีการกำหนด ค่า input parameter ค่านี้เป็นค่า option จะใช้หรือไม่ก็ได้

parameters คือ ส่วนที่ใช้เพื่อกำหนดค่าของ input parameters ที่จะต้องใช้ใน template โดยสามารถจะรับค่าจากผู้ใช้งานหรือสามารถกำหนดเป็นค่า default

resources ในส่วนนี้ จะใช้เป็นส่วนของการประกาศ ว่าจะสร้าง (instantiated) virtual machine อย่างไร outputs ในส่วนนี้จะใช้สำหรับ output parameters ที่สามารถให้ user สามารถใช้งานได้หลังจากการ instantiation

ความสำคัญของ Heat Template Version

ค่าที่ระบุใน heat template version จะเป็นค่าที่บอกให้ heat engine ทราบว่า นอกจากรูปแบบ format ที่สามารถใช้ได้แล้วยังบอกถึง function หรือ feature อื่นๆ ที่ต้อง implement ในแต่ละ version และผู้ใช้งานที่สามารถเรียกใช้งานได้ภายใน template อีกด้วย

version 2013-05-23

version นี้เริ่มต้นใช้งานมากตั้งแต่แรก ในเวอร์ชัน icehouse มีความสามารถสนับสนุน function ดังต่อไปนี้

```
get_attr
```

```
get_file
get_param
get_resource
list_join
resource_facade
str_replace
Fn::Base64
Fn::GetAZs
Fn::Join
Fn::MemberListToMap
Fn::Replace
Fn::ResourceFacade
Fn::Select
Fn::Split
Ref
```

version 2014-10-16

เริ่มต้นการใช้งาน ในเวอร์ชั่น juno โดยได้ดำเนินการปรับปรุง ลด function ส่วนมากที่ให้การสนับสนุน CFN พังก์ชั่น ที่เคยให้การสนับสนุนในเวอร์ชั่นของ icehouse และ สามารถสนับสนุน function ดังต่อไปนี้

```
get_attr
get_file
get_param
get_resource
list_join
resource_facade
str_replace
Fn::Select
```

version 2015-04-30

เริ่มต้นการใช้งานในเวอร์ชั่น kilo โดยในเวอร์ชั่นนี้ได้ดำเนินการ เพิ่ม function ชื่อว่า repeat, digest ลงไป จากเวอร์ชั่นก่อนหน้า และสามารถสนับสนุน function ดังต่อไปนี้

```
get_file
get_file
get_param
get_resource
list_join
repeat
digest
resource_facade
str_replace
Fn::Select
```

version 2015-10-15

ได้มีเป้าหมายเพื่อที่จะใช้เพื่อรับในเวอร์ชั่นของ Liberty มีพังก์ชั่นดังต่อไปนี้

```
get_attr  
get_file  
get_param  
get_resource  
list_join  
repeat  
digest  
resource_facade  
str_replace  
str_split
```

รูปแบบการกำหนดค่าใน ส่วน Parameter Groups Section

ในส่วน ของ parameter_groups เป็นการบ่งบอกว่าแต่ละ parameter นั้นถูกจัดกลุ่มอย่างไร วัตถุประสงค์ของการใช้ การจัดกลุ่มตัวแปร ก็เพื่อ จะมีผลต่อการกำหนด userinterface ให้เหมาะสมเมื่อมีการนำข้อมูลเข้า สามารถกำหนดรูปแบบได้ดังต่อไปนี้

```
parameter_groups:  
- label: <human-readable label of parameter group>  
  description: <description of the parameter group>  
  parameters:  
    - <param name>  
    - <param name>
```

รูปแบบการกำหนดค่าใน ส่วน Parameter Section

ใช้กำหนดค่าของตัวแปร ที่เป็น input parameters จาก user และนำไปเป็น input ต่อไปยัง template โดยแต่ละค่า ตัวแปรนั้น จะแบ่งออกเป็น Block level ช้อนๆ กัน มีรูปแบบการประกาศดังต่อไปนี้

```
parameters:  
  <param name>:  
    type: <string | number | json | comma_delimited_list | boolean>  
    label: <human-readable name of the parameter>  
    description: <description of the parameter>  
    default: <default value for parameter>  
    hidden: <true | false>  
    constraints:  
      <parameter constraints>
```

ความหมายตัวแปร

parameter name คือ ชื่อตัวแปรที่ต้องการใช้ งานใน template โดยที่รับค่าจาก ผู้ใช้งานมาเก็บไว้ที่ตัวแปร ก่อนการนำไปใช้งาน

type หรือ ชนิดตัวแปรที่รับ จะต้องกำหนดด้วยชนิดดังต่อไปนี้

Type	Description	Examples
string	A literal string.	"String param"
number	An integer or float.	"2"; "0.2"
comma_delimited_list	An array of literal strings that are separated by commas. The total number of strings should be one more than the total number of commas.	["one", "two"]; "one, two"; Note: "one, two" returns ["one", "two"]
json	A JSON-formatted map or list.	{"key": "value"}
boolean	Boolean type value, which can be equal "t", "true", "on", "y", "yes", or "1" for true value and "f", "false", "off", "n", "no", or "0" for false value.	"on"; "n"

label คือชื่อตัวแปร ในแสดงผล หน้า เวป เพื่อที่ช่วยให้เป็นการนำเข้าข้อมูลให้ถูกต้องใน form input

description คำอธิบายตัวแปร

default ใช้กำหนดค่าให้เป็นค่า default สำหรับตัวแปรนี้

hidden ใช้กำหนดให้เป็นการซ่อนการพิม ใช้สำหรับการกรอกข้อมูล ในช่อง password

constraints เป็นการกำหนดค่าข้อจำกัดของการกรอกข้อมูลที่สามารถจะรับได้ เพื่อใช้สำหรับเป็นการ validation และใช้เพื่อป้องการผิดพลาดในขณะที่ deploy hot template การกำหนดค่าของ constrain ใช้เป็น block parameter มี format ดังต่อไปนี้

```
constraints:
  - <constraint type>: <constraint definition>
    description: <constraint description>
```

ตัวอย่างการกำหนดค่าในส่วนของ parameter

```
parameters:
  user_name:
    type: string
    label: User Name
    description: User name to be configured for the application
    constraints:
      - length: { min: 6, max: 8 }
        description: User name must be between 6 and 8 characters
      - allowed_pattern: "[A-Z]+[a-zA-Z0-9]*"
        description: User name must start with an uppercase character
```

length เป็นการกำหนด ขนาดยาวของ input ที่รับเข้า ใช้กับตัวแปรชนิดของ string มี รูปแบบ syntax ดังนี้

```
length: { min: <lower limit>, max: <upper limit> }
```

range เป็นการกำหนดช่วงของ input ใช้สำหรับตัวแปร ชนิดตัวเลข มีรูปแบบ syntax ดังนี้

```
range: { min: <lower limit>, max: <upper limit> }
```

allowed_values เป็นการกำหนดค่า constrain ที่สามารถยอมรับได้ ใน input สามารถกำหนดดังนี้

```
allowed_values: [ <value>, <value>, ... ]
```

หรือสามารถกำหนดใน format YAML ได้ดังนี้

```
allowed_values:  
- <value>  
- <value>  
- ...
```

ตัวอย่างเช่น การกำหนดค่าใน allowed_values

```
parameters:  
instance_type:  
  type: string  
  label: Instance Type  
  description: Instance type for compute instances  
constraints:  
  - allowed_values:  
    - m1.small  
    - m1.medium  
    - m1.large
```

สามารถกำหนดค่า allowed ได้ด้วยการกำหนดรูปแบบ regular expression

```
allowed_values: [ <value>, <value>, ... ]
```

ตัวอย่าง

```
parameters:  
user_name:  
  type: string  
  label: User Name  
  description: User name to be configured for the application  
constraints:  
  - allowed_pattern: "[A-Z]+[a-zA-Z0-9]*"  
  description: User name must start with an uppercase character
```

custom_constraint

เป็นการกำหนดความสามารถของการ validation ให้มีประสิทธิภาพสูงมากขึ้น โดยผู้ใช้งานสามารถกำหนดได้เอง มีรูปแบบดังนี้

```
custom_constraint: <name>
```

ตัวอย่าง

```
parameters:  
key_name  
  type: string  
  description: SSH key pair  
constraints:  
  - custom_constraint: nova.keypair
```

โดยค่าตัวแปร <name> จะต้องสอดคล้องกับ plugin ที่สนับสนุนค่า นี้

Name	Plug-in
cinder.backup	heat.engine.clients.os.cinder:VolumeBackupConstraint
cinder.snapshot	heat.engine.clients.os.cinder:VolumeSnapshotConstraint
cinder.volume	heat.engine.clients.os.cinder:VolumeConstraint
cinder.vtype	heat.engine.clients.os.cinder:VolumeTypeConstraint
cron_expression	heat.engine.constraint.common_constraints:CRONExpressionConstraint
designate.domain	heat.engine.clients.os.designate:DesignateDomainConstraint
glance.image	heat.engine.clients.os.glance:ImageConstraint
ip_addr	heat.engine.constraint.common_constraints:IPConstraint
iso_8601	heat.engine.constraint.common_constraints:ISO8601Constraint
keystone.domain	heat.engine.clients.os.keystone:KeystoneDomainConstraint
keystone.group	heat.engine.clients.os.keystone:KeystoneGroupConstraint
keystone.project	heat.engine.clients.os.keystone:KeystoneProjectConstraint
keystone.role	heat.engine.clients.os.keystone:KeystoneRoleConstraint
keystone.service	heat.engine.clients.os.keystone:KeystoneServiceConstraint
keystone.user	heat.engine.clients.os.keystone:KeystoneUserConstraint
mac_addr	heat.engine.constraint.common_constraints:MACConstraint
manila.share_network	heat.engine.clients.os.manila:ManilaShareNetworkConstraint
manila.share_snapshot	heat.engine.clients.os.manila:ManilaShareSnapshotConstraint
manila.share_type	heat.engine.clients.os.manila:ManilaShareTypeConstraint
monasca.notification	heat.engine.clients.os.monasca:MonascaNotificationConstraint
net_cidr	heat.engine.constraint.common_constraints:CIDRConstraint
neutron.network	heat.engine.clients.os.neutron:NetworkConstraint
neutron.port	heat.engine.clients.os.neutron:PortConstraint
neutron.router	heat.engine.clients.os.neutron:RouterConstraint
neutron.subnet	heat.engine.clients.os.neutron:SubnetConstraint
nova.flavor	heat.engine.clients.os.nova:FlavorConstraint
nova.keypair	heat.engine.clients.os.nova:KeypairConstraint
nova.network	heat.engine.clients.os.nova:NetworkConstraint
nova.server	heat.engine.clients.os.nova:ServerConstraint
sahara.image	heat.engine.clients.os.sahara:ImageConstraint
sahara.plugin	heat.engine.clients.os.sahara:PluginConstraint
timezone	heat.engine.constraint.common_constraints:TimezoneConstraint
trove.flavor	heat.engine.clients.os.trove:FlavorConstraint

Pseudo parameter

ใน heat template มีการกำหนดค่าตัวแปรเพื่อที่สามารถเข้าถึงข้อมูลของ Template เอง ได้แก่

```
OS::stack_name
OS::stack_id
OS::project_id
```

หมายเหตุ OS::project_id เริ่มต้น ใช้งานตั้งแต่ เวอร์ชัน 2015.1 (kilo)

Resource section

ในส่วนนี้ เป็นการกำหนด ให้ heat template สร้าง instance, network, storage volumes ตามต้องการ โดยในแต่ละ resource จะสามารถแบ่งได้ด้วย block มีรูปแบบดังนี้

```
resources:
```

```

<resource ID>:
  type: <resource type>
  properties:
    <property name>: <property value>
  metadata:
    <resource specific metadata>
  depends_on: <resource ID or list of ID>
  update_policy: <update policy>
  deletion_policy: <deletion policy>

```

ความหมายตัวแปร

resource ID ชื่อของ resource ที่ต้องการสามารถเลือกได้ตามที่ต้องการ แต่จะต้องไม่ซ้ำกัน (กำหนดโดยผู้ใช้)
type คือชนิดของ resource ได้แก่ OS::Nova::Server หรือ OS::Neutron::Port ค่านี้ขาดไม่ได้ จำเป็นต้องมี
properties รายการของ ค่า properties ของ resource ที่ใช้ (option)

metadata ค่า metadata ของ resource (option)

depends_on ระบุ dependency ไปยัง resource อื่น

update_policy คือ update_policy สำหรับ resource กำหนดในรูปแบบของ nested dictionary

deletion_policy คือ deletion_policy สำหรับ resource

ตัวอย่าง

```

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      flavor: m1.small
      image: F18-x86_64-cfntools

```

ตัวอย่างของ resource dependency

```

resources:
  server1:
    type: OS::Nova::Server
    depends_on: server2

  server2:
    type: OS::Nova::Server

```

ตัวอย่างของ resource dependency สามารถระบุมากกว่า 1 โดยกำหนดไว้ใน [value1, value2]

```

resources:
  server1:
    type: OS::Nova::Server
    depends_on: [ server2, server3 ]

  server2:
    type: OS::Nova::Server

```

```
server3:  
  type: OS::Nova::Server
```

Outputs Sections

เป็นการกำหนดค่าของ output parameter เพื่อที่จะสามารถให้เป็นตัวแปรให้แก่ user สามารถใช้งานได้ หลังจากที่ stack ได้สร้างขึ้น

```
outputs:  
  <parameter name>:  
    description: <description>  
    value: <parameter value>
```

ความหมายตัวแปร

parameter name กำหนดชื่อตัวแปร ที่จะใช้

description คำอธิบาย ตัวแปร

parameter value เป็นค่าของ output parameter และจะเอาไปใช้งาน

ตัวอย่างใช้สำหรับการกำหนดค่าของ ipaddress ให้แก่ compute resource (vm)

```
outputs:  
  instance_ip:  
    description: IP address of the deployed compute instance  
    value: { get_attr: [my_instance, first_address] }
```

พังก์ชันภายใน hot (Intrinsic function)

HOT ได้ออกแบบให้มีการเรียกงานพังก์ชัน ที่อยู่ใน ระบบ เพื่อให้สามารถใช้งานตามที่ต้องการได้ เช่นการ get ค่าจากตัวแปรจากการอักข้อมูล

พังก์ชัน get_attr

ใช้สำหรับการคืนค่า โดยการใช้ keys โดยมีรูปแบบดังนี้

```
get_attr:  
  - <resource name>  
  - <attribute name>  
  - <key/index 1> (optional)  
  - <key/index 2> (optional)  
  - ...
```

ความหมายตัวแปร

resource name ชื่อ resources ที่ต้องการอ่านค่า

attribute ใช้กำหนดค่า สำหรับ attribute data structure ที่มีความซับซ้อน เช่น list หรือ map การอ่านค่าตัวแปรจะอ้างอิงตามลำดับของตัวแปร ใน data structure

ตัวอย่าง

```
resources:
  my_instance:
    type: OS::Nova::Server
    # ...

outputs:
  instance_ip:
    description: IP address of the deployed compute instance
    value: { get_attr: [my_instance, first_address] }
  instance_private_ip:
    description: Private IP address of the deployed compute instance
    value: { get_attr: [my_instance, networks, private, 0] }
```

ในตัวอย่างนี้ค่าของ networks attribute ประดับวย ข้อมูลได้แก่

```
{"public": ["2001:0db8:0000:0000:0000:ff00:0042:8329", "1.2.3.4"],
 "private": ["10.0.0.1"]}
```

ฟังก์ชัน get_file

ฟังก์ชันจะคืนค่าของ content ที่อยู่ใน file ใน template มีรูปแบบดังนี้

```
get_file: <content key>
```

ในตัวอย่างนี้ใช้อ่านค่าจาก file url

```
resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      # general properties ...
      user_data:
        get_file: my_instance_user_data.sh
  my_other_instance:
    type: OS::Nova::Server
    properties:
      # general properties ...
      user_data:
        get_file: http://example.com/my_other_instance_user_data.sh
```

สามารถกำหนดรูปแบบ

- file:///path/to/my_instance_user_data.sh
- http://example.com/my_other_instance_user_data.sh

ฟังก์ชัน get_param

ฟังก์ชัน get_param จะเป็นการอ้างอิงไปยัง input parameter และเป็นการเรียกคืนค่าจาก ตัวแปร

```
get_param:
```

```
- <parameter name>
- <key/index 1> (optional)
- <key/index 2> (optional)
- ...
```

ตัวอย่างการใช้งาน พัฟ์ชัน get_param

```
parameters:
  instance_type:
    type: string
    label: Instance Type
    description: Instance type to be used.
  server_data:
    type: json

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      flavor: { get_param: instance_type }
      metadata: { get_param: [ server_data, metadata ] }
      key_name: { get_param: [ server_data, keys, 0 ] }
```

ในตัวอย่างนี้ ถ้าหากตัวแปร instance_type และ server_data มีตัวแปรดังนี้

```
{"instance_type": "m1.tiny",
{"server_data": {"metadata": {"foo": "bar"}, "keys": ["a_key", "other_key"]}}}
```

พัฟ์ชัน get_resource

พัฟ์ชัน จะคืนค่า resource อื่น มาใช้งานภายใต้ template ปัจจุบัน ในขณะ runtime โดยอ้างอิงโดยใช้ Resource

```
get_resource: <resource ID>
```

ตัวอย่าง

```
resources:
  instance_port:
    type: OS::Neutron::Port
    properties: ...

  instance:
    type: OS::Nova::Server
    properties:
      ...
      networks:
        port: { get_resource: instance_port }
```

พัฟ์ชัน list_join

ทำหน้าที่เชื่อม list คำเข้าด้วยกัน ด้วย delimiter มีรูปแบบการใช้งานดังนี้

```
list_join:  
- <delimiter>  
- <list to join>
```

เช่น

```
list_join: [', ', ['one', 'two', 'and three']]
```

ฟังก์ชัน digest

function digest ใช้สำหรับ digest operation ด้วยค่าที่กำหนดให้ มีรูปแบบการใช้งาน

```
digest:  
- <algorithm>  
- <value>
```

ความหมายตัวแปร

algorithm อัลกอริทึม ที่สามารถใช้งานได้ md5, sha1, sha224, sha256, sha384, sha512 หรือ รองรับการใช้งาน OpenSSL

```
# from a user supplied parameter  
pwd_hash: { digest: ['sha512', { get_param: raw_password }] }
```

ฟังก์ชัน repeat

ฟังก์ชัน รองรับกันทำงานในการวนลูปค่า content ได้ใน list ในสำหรับความต้องการที่จะทำซ้ำใน template รูปแบบคำสั่ง

```
repeat:  
  template:  
    <template>  
  for_each:  
    <var>: <list>
```

ตัวอย่างการใช้งาน security group resource เพื่อ อ่านค่า ports parameters

```
parameters:  
  ports:  
    type: comma_delimited_list  
    label: ports  
    default: "80,443,8080"  
  
resources:  
  security_group:  
    type: OS::Neutron::SecurityGroup  
    properties:  
      name: web_server_security_group
```

```

rules:
repeat:
for_each:
%port%: { get_param: ports }
template:
protocol: tcp
port_range_min: %port%
port_range_max: %port%

```

อีกตัวอย่าง ใช้งานการ วนลูป เพื่อ enable ค่า security group ร่วมกับค่าของ protocols

```

parameters:
ports:
type: comma_delimited_list
label: ports
default: "80,443,8080"
protocols:
type: comma_delimited_list
label: protocols
default: "tcp,udp"

resources:
security_group:
type: OS::Neutron::SecurityGroup
properties:
name: web_server_security_group
rules:
repeat:
for_each:
%port%: { get_param: ports }
%protocol%: { get_param: protocols }
template:
protocol: %protocol%
port_range_min: %port%

```

พังก์ชัน resource_facade

พังก์ชัน resource_facade ใช้สำหรับการดึงค่าจาก template แม่ (parent template)

```
resource_facade: <data type>
```

ค่า data type สามารถเป็นไปได้ดังนี้ metadata, deletion_policy, หรือ update_policy

พังก์ชัน str_replace

พังก์ชัน str_replace ใช้สำหรับการสร้าง string ขึ้นมาใหม่ โดยการกำหนดรูปแบบของ string template ที่กำหนดเอาไว้ก่อน มีทำหน้าเป็น placeholder ให้กับตัวแปลง รูปแบบการใช้งานดังนี้

```
str_replace:
template: <template string>
```

```
params: <parameter mappings>
```

ตัวอย่างการใช้งาน str_replace function ในส่วนของ output section

```
resources:
  my_instance:
    type: OS::Nova::Server
    # general metadata and properties ...

outputs:
  Login_URL:
    description: The URL to log into the deployed application
    value:
      str_replace:
        template: http://host/MyApplication
        params:
          host: { get_attr: [ my_instance, first_address ] }
```

```
parameters:
  DBRootPassword:
    type: string
    label: Database Password
    description: Root password for MySQL
    hidden: true

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      # general properties ...
      user_data:
        str_replace:
          template: |
            #!/bin/bash
            echo "Hello world"
            echo "Setting MySQL root password"
            mysqladmin -u root password $db_rootpassword
            # do more things ...
        params:
          $db_rootpassword: { get_param: DBRootPassword }
```

ฟังก์ชัน str_split

ฟังก์ชัน str_split ใช้สำหรับการแยกคำต่างๆ ใน string ด้วยการระบุ delimiter

```
str_split:
  - ','
  - string,to,split
```

หรือ

```
str_split: [',', 'string,to,split']
```

ผลที่ได้

```
['string', 'to', 'split']
```

ฟังก์ชัน get_param, get_attr, get_resource

ตาม มาตรฐานของ HOT (http://docs.openstack.org/developer/heat/template_guide/hot_spec.html) ได้อธิบาย เกี่ยวกับฟังก์ชัน ที่สามารถเรียกใช้งานภายใน template ซึ่งจะเรียกว่า Intrinsic Function ในส่วนต่อไป จะเป็นการอธิบายเพิ่มรายของการใช้ฟังก์ชัน และรูปแบบการใช้งานหรือ syntax ของ Intrinsic Function. ฟังก์ชันเหล่านี้สามารถเรียกใช้งานในส่วนของคุณสมบัติ (properties section) ในขอบเขตของ Resources หรือในส่วนของขอบเขต Output

ฟังก์ชัน get_attr

จะเป็นการอ้างอิงตัวแปรของ Resource ค่าที่ได้จะได้ในขณะ runtime เมื่อมีการสร้าง resource นั้นๆ มี syntax ดังนี้

```
get_attr:  
- <resource name>  
- <attribute name>  
- <key/index 1> (optional)  
- <key/index 2> (optional)  
- ...
```

ความหมาย

ค่า<resource name> คือชื่อของ resource ซึ่งจะถูกใช้สำหรับการหาค่า(resolve) และจะต้องมีชื่อปรากฏอยู่ก่อนในส่วนของ resource หากไม่มีจะเกิด error

ค่า<attribute name> ชื่อของ attribute ที่เราต้องการหาค่า หากค่าที่คืนมาเป็นตัวแปรที่มีความซับซ้อนของโครงสร้างของข้อมูล เช่นคือมาเป็นชนิดตัวแปร List หรือ Dict จะต้องมีตัวแปร Key/index เพื่อมาเลือกรับค่าที่ต้องการนำไปใช้

ตัวอย่างการใช้งานฟังก์ชัน get_attr

```
resources:  
  my_instance:  
    type: OS::Nova::Server  
    # ...  
  
outputs:  
  instance_ip:  
    description: IP address of the deployed compute instance  
    value: { get_attr: [my_instance, first_address] }  
  instance_private_ip:  
    description: Private IP address of the deployed compute instance  
    value: { get_attr: [my_instance, networks, private, 0] }
```

จากรูปด้านบนเป็นการเรียกใช้งาน get_attr ในส่วนของ outputs เพื่อเป็นการดึงค่าจาก resource ชื่อ my_instance ที่มีชนิดเป็น OS::Nova::Server จากตัวอย่างนี้ หากค่าของ networks attribute มีค่าดังนี้

```
{"public": ["2001:0db8:0000:0000:0000:ff00:0042:8329", "1.2.3.4"],  
 "private": ["10.0.0.1"]}
```

ดังนั้นค่าที่ได้จากการ resolve ฟังก์ชัน get_attr ของ first_address จะมีค่าเป็น private คือ 10.0.0.1 หากกลับไปดูข้อมูลใน Horizon ในเมนู orchestration > Resource Type และทำการเลือก OS::Nova::Server จะอธิบาย DataStructure ของ Attributes ที่สามารถใช้ได้

Resource Type Details

Resource Type

OS::Nova::Server

Attributes

```

accessIPv4: {description: The manually assigned alternative public IPv4 address of
the server.}
accessIPv6: {description: The manually assigned alternative public IPv6 address of
the server.}
addresses: {description: 'A dict of all network addresses with corresponding port_id.
The port ID may be obtained through the following expression: "{get_attr: [<server>,
addresses, <network name>, 0, port]}".'}
console_urls: {description: 'URLs of server's consoles. To get a specific console
type, the requested type can be specified as parameter to the get_attr function,
e.g. get_attr: [<server>, console_urls, novnc ]. Currently supported types are
novnc, xvnc, spice-html5, rdp-html5, serial.'}
first_address: {description: 'Convenience attribute to fetch the first assigned network
address, or an empty string if nothing has been assigned at this time. Result
may not be predictable if the server has addresses from more than one network.'}
instance_name: {description: AWS compatible instance name.}
name: {description: Name of the server.}
networks: {description: 'A dict of assigned network addresses of the form: {"public":'
[ip1, ip2...], "private": [ip3, ip4]}.'
show: {description: A dict of all server details as returned by the API.}

```

ใน HOT version “2015-10-15” หากกำหนดค่า <resource name> แต่ไม่มีการกำหนด <attribute name> จะหมายถึงเป็นการคืนค่า dict ของทุกๆ Attribute ดังรูป ด้านล่าง

```
get_attr:
- <resource_name>
```

ฟังก์ชัน get_param

จะเป็นการอ้างอิงตัวแปรสำหรับ input parameter ใน template จะใช้ get_param เพื่อเป็นการ insert ค่าจาก input ที่ user กำหนดมาจากส่วน parameters หรือเป็นการกำหนดให้เป็นค่า default many ส่วนของ Resource เพื่อจะใช้สร้าง resource มี syntax ดังนี้

```
get_param:
- <parameter name>
- <key/index 1> (optional)
- <key/index 2> (optional)
- ...
```

ความหมาย

ค่า <parameter name> ค่าตัวแปร ที่กำหนดในส่วนของ parameter หากค่าที่ให้เป็นตัวแปรที่มีโครงสร้างซับช้อนจะต้องมีค่า <key> หรือ <index> มารับ

ตัวอย่างการใช้งาน

```
parameters:
  instance_type:
    type: string
    label: Instance Type
    description: Instance type to be used.
  server_data:
    type: json

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      flavor: { get_param: instance_type }
      metadata: { get_param: [ server_data, metadata ] }
      key_name: { get_param: [ server_data, keys, 0 ] }
```

จากตัวอย่างนี้ ค่าตัวแปรที่ระบุในส่วนของ parameter จะมีอยู่ 2 ค่า คือ instance_type และ server_data หากค่าของตัวแปร มีค่าดังต่อไปนี้

```
{"instance_type": "m1.tiny",
"server_data": {"metadata": {"foo": "bar"},
"keys": ["a_key", "other_key"]}}
```

เมื่อมีการใช้งาน get_param ในส่วนของ resource จะทำให้ properties ที่ชื่อ flavor ได้รับกำหนดค่าเป็น m1.tiny ส่วน metadata จะได้ค่าตัวแปรชนิด Dict ที่มีค่า {"foo": "bar"} เนื่องจากมีค่า key ที่ชื่อ metadata มารับ และ เช่นเดียวกัน ค่าตัวแปร key_name จะได้ค่าค่าที่เป็นชนิดตัวแปรชนิด List คือ ["a_key", "other_key"]

ฟังก์ชัน get_resource

เป็นฟังก์ชันที่ในสำหรับการอ้างอิงไปยัง Resource อื่นที่อยู่ template เดียวกัน ในขณะ runtime นั้น ค่าที่คืนมาจากการ调用 get_resource คือ resource id มี syntax ดังต่อไปนี้

```
get_resource: <resource ID>
```

ยกตัวอย่างการใช้งาน

```
resources:
  instance_port:
    type: OS::Neutron::Port
    properties: ...

  instance:
    type: OS::Nova::Server
    properties:
      ...
    networks:
      port: { get_resource: instance_port }
```

นอกจากนั้นยังมีอีก intrinsic พิมพ์ชื่อ str_replace ที่พบบ่อย ทำหน้าที่นำค่าของ parameter มาแทนใน user_data ซึ่งเป็นส่วนที่สร้างการ provision โดย Cloud-init จะนำไปใช้งานขณะที่สร้าง instance หากไม่มีการแทนค่า จาก parameter ดังตัวอย่างเช่น

```
1 resources:
2
3   the_server:
4     type: OS::Nova::Server
5     properties:
6       # flavor, image etc
7     user_data: |
8       #!/bin/bash
9       echo "Running boot script"
10      # ...
```

แต่เมื่อจะต้องมีการดึงค่าจาก ส่วน parameter จะต้องใช้ str_replace ตัวอย่างเช่น

```
1 parameters:
2   foo:
3     default: bar
4
5 resources:
6
7   the_server: 1
8     type: OS::Nova::Server
9     properties:
10    # flavor, image etc
11    user_data:
12      str_replace:
13        template: |
14          #!/bin/bash
15          echo "Running boot script with $FOO"
16          # ...
17
18 params:
19   $FOO {get_param: foo}
```

สรุปการใช้งานคำสั่ง เพื่อการบริหารจัดการ Heat Template

1 แสดง stack ทั้งหมดที่ใช้งานอยู่

```
# heat stack-list
```

```
[root@allinone ~(keystone_admin)]# heat stack-list
+-----+-----+-----+-----+
| id | stack_name | stack_status | creation_time |
+-----+-----+-----+-----+
| e978a8f4-54fc-42d9-b92b-b11d569f5d07 | webfarm | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
+-----+-----+-----+-----+
```

2 แสดงรายละเอียดของ stack

```
#heat stack-show webfarm
```

3 แสดงรายละเอียดของ event ที่เกิดขึ้นภายใน stack

```
# heat event-list webfarm
```

```
[root@allinone ~(keystone_admin)]# heat event-list webfarm | grep cpu
| cpu_alarm_low | f5840c4a-7ce2-4c77-a644-53a0f5629ea1 | state changed
CREATE_IN_PROGRESS | 2015-11-20T10:34:24Z |
| cpu_alarm_high | 29067138-8385-4060-89e9-5a435f83ed99 | state changed
CREATE_IN_PROGRESS | 2015-11-20T10:34:24Z |
| cpu_alarm_low | 8d4ac263-bd70-4cc4-8b40-0bae6feebcb1 | state changed
CREATE_COMPLETE | 2015-11-20T10:34:25Z |
| cpu_alarm_high | 74e6bc8b-8e97-4b0a-8745-29b4f7fd734b | state changed
CREATE_COMPLETE | 2015-11-20T10:34:25Z |
```

3 แสดงผลย่อของ event

```
# heat event-show webfarm cpu_alarm_low f5840c4a-7ce2-4c77-a644-53a0f5629ea1
```

4 แสดงผล resource ใน stack

```
# heat resource-list webfarm
```

```
[root@allinone ~(keystone_admin)]# heat resource-list webfarm
+-----+-----+-----+-----+-----+
| resource_name | physical_resource_id | resource_type | resource_status | updated_time |
+-----+-----+-----+-----+-----+
| cpu_alarm_high | 2f97bafa-4304-4eaa-b8c6-fd31f68ba6c3 | OS::Ceilometer::Alarm | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| cpu_alarm_low | f9c6b7f0-e1af-4c34-95ec-535fa9ff131b | OS::Ceilometer::Alarm | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| lb |  | OS::Neutron::LoadBalancer | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| lb_floating | 40ae8ab3-2376-4656-ad94-050163af268f | OS::Neutron::FloatingIP | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| monitor | 44238922-7411-4413-bfa6-640bf6db51f1 | OS::Neutron::HealthMonitor | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| pool | e1e68590-6b5b-4926-a360-63ca0189b785 | OS::Neutron::Pool | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| web_server_scaledown_policy | 49703eb94a483795bbac3c8d66fb64 | OS::Heat::ScalingPolicy | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| web_server_scaleup_policy | 885792bb43b540b1bf3c74bded2ec2d3 | OS::Heat::ScalingPolicy | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
| webserver | 5767e810-4c50-40e1-a0d3-1add1c2b3f67 | OS::Heat::AutoScalingGroup | CREATE_COMPLETE | 2015-11-20T10:34:02Z |
+-----+-----+-----+-----+-----+
```

```
# heat resource-show webfarm cpu_alarm_high
```

```
[root@allinone ~(keystone_admin)]# heat resource-show webfarm cpu_alarm_high
+-----+-----+
| Property | Value
+-----+-----+
| attributes | {}
| description |
| links | http://192.168.1.2:8004/v1/b2bf8657138a4a7f98023a7c88187159/stacks/webfarm/e
| logical_resource_id | cpu_alarm_high
| physical_resource_id | 2f97bafa-4304-4eaa-b8c6-fd31f68ba6c3
| required_by |
| resource_name | cpu_alarm_high
| resource_status | CREATE_COMPLETE
| resource_status_reason | state changed
| resource_type | OS::Ceilometer::Alarm
| updated_time | 2015-11-20T10:34:02Z
+-----+-----+
```

5 แสดงผล resource metadata ของ resource ใน stack

```
# heat resource-metadata webfarm pool
```

```
[root@allinone ~(keystone_admin)]# heat resource-metadata webfarm pool
{
    "vip": "41f643af-e779-45e3-b4b6-8ca480856db6"
}
```

6 แสดงผล template ของ resource ใน stack

```
# heat template-show webfarm
```

```
[root@allinone ~(keystone_admin)]# heat template-show webfarm
description: AutoScaling CentOS 7.1 Web Application
heat_template_version: '2014-10-16'
outputs:
  ceilometer_query:
    description: 'This is a Ceilometer query for statistics on the cpu_util meter
      Samples about OS::Nova::Server instances in this stack. The -q parameter selects
      Samples according to the subject's metadata. When a VM's metadata includes
      an item of the form metering.X=Y, the corresponding Ceilometer resource has
      a metadata item of the form user_metadata.X=Y and samples about resources so
      tagged can be queried with a Ceilometer query term of the form metadata.user_metadata.X=Y. In
      this case the nested stacks give their VMs metadata that is passed as a nested
      stack parameter, and this stack passes a metadata of the form metering.stack=Y,
      where Y is this stack's ID.

    '
    value:
      str_replace:
        params:
          stackval: {get_param: 'OS::stack_id'}
        template: 'ceilometer statistics -m cpu_util -q metadata.user_metadata.stack=stackval
          -p 600 -a avg
          '

  pool_ip_address:
    description: The IP address of the load balancing pool
    value:
      get_attr: [pool, vip, address]
scale_dn_url:
  description: 'This URL is the webhook to scale down the autoscaling group. You
    can invoke the scale-down operation by doing an HTTP POST to this URL; no body
    nor extra headers are needed.'
```

6 แสดงผล template ของ resource ใน stack

```
# heat stack-delete webfarm
```