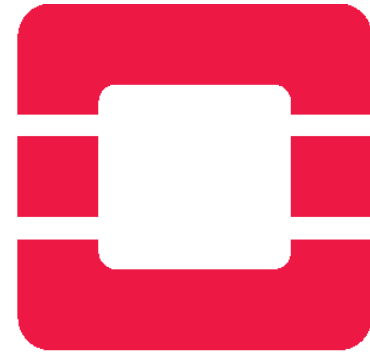




# Heat

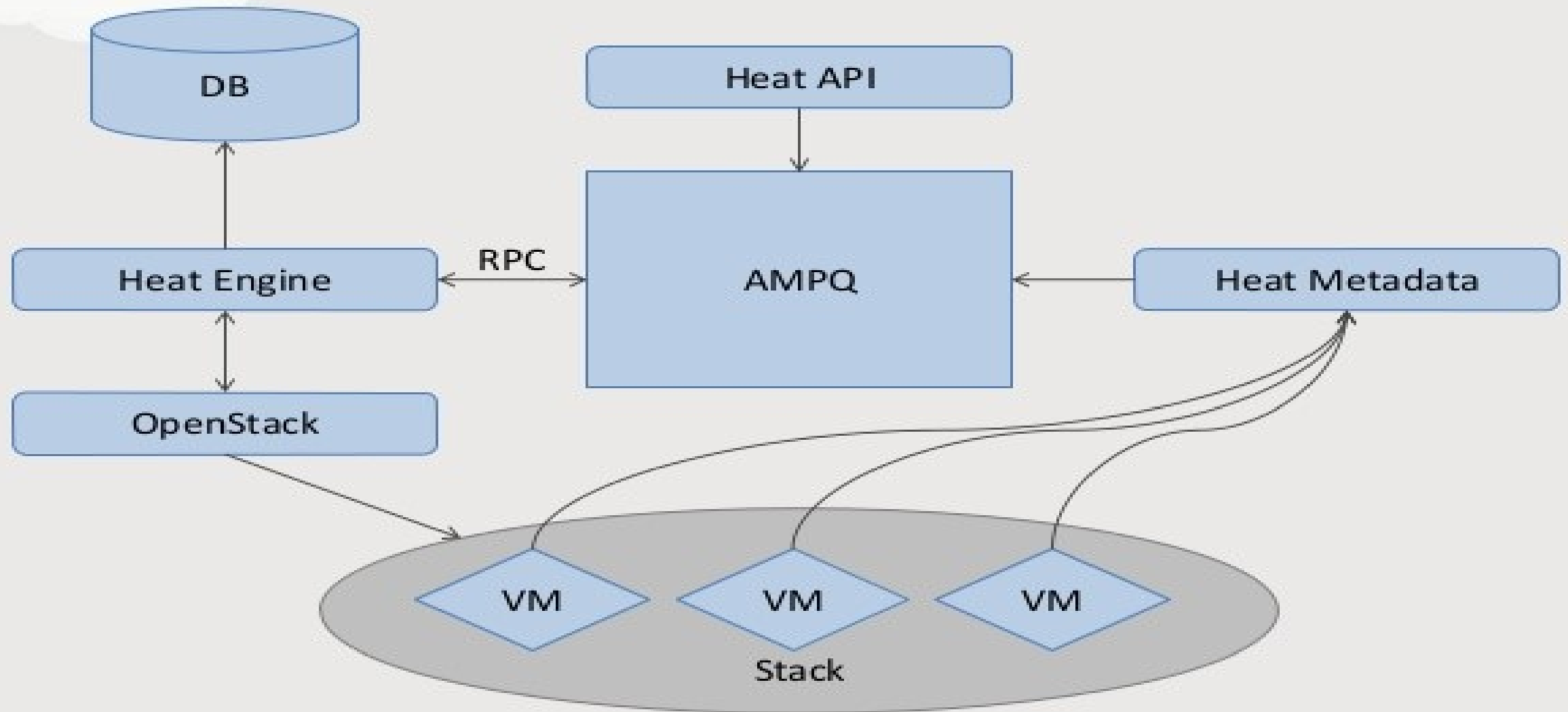


# openstack®

# Heat

- Heat เป็น Orchestration และ Automation Tools ที่อยู่ใน OpenStack ซึ่ง Heat จะเป็น Service ที่อาศัยข้อมูลที่ส่งมาจาก Ceilometer มาทำงานต่อ
- Heat จะอ่าน HOT (Heat Orchestration Template) ที่เป็น YAML file ในการ Deploy App ตามที่กำหนดไว้ใน template
- จุดเด่นของ Heat คือ การทำ Autoscaling จะขยาย หรือ ลดขนาด จำนวนของ instance ตามโหลดการทำงานของระบบ

# Heat basic architecture



# องค์ประกอบภายใน Heat

- Heat-API เป็นส่วนประกอบที่ทำหน้าที่เป็น REST API process สำหรับในการติดต่อ service อื่นๆ
- Heat-API-CFN เป็นส่วนประกอบที่ทำหน้าที่รองรับ AWS API โดยจะส่งไปยังส่วนประกอบอื่นๆ ใน Heat โดยผ่าน RPC (Remote Procedure Call)
- Heat-Engine เป็น engine หลักสำหรับประมวลผล HOT
- Heat-cli เป็นเครื่องมือ client สำหรับการเข้าใช้งาน Heat

# การทดสอบการทำงานระหว่าง Heat Template และ Heat Engine

การสร้าง Instance ด้วย HOT  
(แบบ single instance)

# การสร้าง Instance ด้วย HOT (แบบ single instance)

## การเตรียม resource

- Image
- Flavor
- Key-pair
- Private network และ Private Subnet
- Router
- Public network ที่ได้สร้างไว้ก่อนหน้านี้แล้ว

## การสร้าง Instance ด้วย HOT(แบบ single instance)

- การทดสอบนี้เป็นการทดสอบการสร้าง และการใช้งาน HOT แบบง่าย ไม่มีความซับซ้อน ซึ่งจะอ่านค่า parameters ไปใช้ในส่วนของการสร้าง resource
- การทำงานของ Heat engine จะทำการอ่านและตรวจสอบ template ก่อนจะสั่งให้ Heat template ทำงาน จึงต้องเตรียมค่าตัวแปรต่างๆให้พร้อมเสียก่อน หากตัวแปรไม่ครบถ้วนตามที่ระบุไว้ใน template เช่น image, network, flavor, key-pair เป็นต้น Heat template ก็จะไม่สามารถใช้งานได้

## อธิบายรายละเอียดแต่ละส่วนของ template single-instance.yaml

- โครงสร้างจะประกอบด้วยส่วนของ parameter ทำหน้าที่ รองรับค่าที่ผู้ใช้ป้อนมา และจะกำหนดให้เป็นค่า default หากไม่มีการป้อนค่า และส่วนของ resource ที่จะส่งต่อไปยัง heat engine เพื่อไปสร้างสิ่งต่างๆใน OpenStack

```
1  heat_template_version: 2015-04-30
2
3  description: Simple template to deploy a single compute instance
4
5  > parameters: ...
26
27 > resources: ...
```



# Parameters

```
5  parameters:
6    image:
7      type: string
8      label: Image name or ID
9      description: Image to be used for compute instance
10     default: Cirros
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: keypair
21  private_network:
22    type: string
23    label: Private network name or ID
24    description: Network to attach instance to.
25    default: Private
```

## Parameters 4 ตัวแปร คือ

- Image คือ ชื่อของ image ที่จะนำไปสร้าง instance มีค่า default เป็น Cirros
- Flavor คือ ค่าของ spec ของ instance มีค่า default เป็น m1.small
- Key คือ ค่าของ keypair ที่จะนำไปใช้เพื่อการ ssh ไปใช้งาน มีค่า default เป็น keypair
- Private network คือ ชื่อของ network สำหรับสร้าง instance มีค่า default เป็น Private

# Resources

```
27  resources:
28      my_instance:
29          type: OS::Nova::Server
30          properties:
31              name: my_instance
32              image: { get_param: image }
33              flavor: { get_param: flavor }
34              key_name: { get_param: key }
35              networks:
36                  - network: { get_param: private_network }
```

# Resource

- ในส่วนของ resource เป็นการกำหนดให้มีการสร้าง instance ชื่อ my\_instance เนื่องจากเป็นการสร้าง instance จึงกำหนดให้มีการใช้ type OS::Nova::Server และมีการกำหนดค่า property ให้แก่ OS::Nova::Server คือ

image ใช้คำสั่ง get\_param เพื่อดึงค่า ตัวแปรชื่อ image ในส่วนของ parameters

flavor ใช้คำสั่ง get\_param เพื่อดึงค่า ตัวแปรชื่อ flavor ในส่วนของ parameters

key ใช้คำสั่ง get\_param เพื่อดึงค่า ตัวแปรชื่อ key ในส่วนของ parameters

network ใช้คำสั่ง get\_param เพื่อดึงค่า ตัวแปรชื่อ private\_network ในส่วนของ parameters

# วิธีการทดสอบ

- เลือก Launch Stack ผ่านหน้า dashboard

The screenshot shows the AWS CloudFormation console dashboard. On the left sidebar, the 'Orchestration' menu item is selected (1), and the 'Stacks' section is highlighted (2). The main content area is titled 'Stacks'. It includes a search filter box, a '+ Launch Stack' button (3), and a 'Preview Stack' button. Below these elements is a table with columns: Stack Name, Created, Updated, Status, and Actions. The table is currently empty, displaying the message 'No items to display.' and 'Displaying 0 items'.

# วิธีการทดสอบ

- เลือก heat template file เพื่อ upload file แล้วกด next

## Select Template ×

---

Template Source \*

File ▼

Template File ?

Choose File No file chosen **4**

Environment Source

File ▼

Environment File ?

Choose File No file chosen

Description:

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

Cancel

Next

# วิธีการทดสอบ

- หากทำการ upload สำเร็จ dashboard จะแสดงแบบ form ที่ได้กรอกข้อมูลแล้ว

The screenshot shows a 'Launch Stack' dialog box with the following fields and annotations:

- Stack Name \***: An empty text input field, annotated with a red circle containing the number 1.
- Description:**: A text area containing the text 'Create a new stack with the provided values.', annotated with a red circle containing the number 2.
- Creation Timeout (minutes) \***: A text input field containing the value '60', annotated with a red circle containing the number 3.
- Rollback On Failure**: A checkbox that is currently unchecked, annotated with a red circle containing the number 4.
- Password for user "demo" \***: A password input field with masked characters and a toggle icon, annotated with a red circle containing the number 5.
- Flavor**: A text input field containing the value 'm1.small', annotated with a red circle containing the number 6.
- Image name or ID**: An empty text input field, annotated with a red circle containing the number 7.
- Key name**: An empty text input field, annotated with a red circle containing the number 8.
- Private network name or ID**: A text input field containing the value 'private\_network', annotated with a red circle containing the number 9.

At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Launch', with the 'Launch' button annotated with a red circle containing the number 10.

# รายละเอียดของแต่ละช่องของ form

- Stack name ชื่อของ stack
- Creation Timeout (minutes) เนื่องจาก heat จะทำงานติดต่อกับ service อื่นๆ ใน OpenStack ค่าดังกล่าวจะหมายถึง เวลาสูงสุดที่จะรอได้ก่อนที่จะเปลี่ยนสถานะไปเป็น fail
- Password for user “demo” เป็น password ที่เข้ารับยืนยันตัวตนสำหรับการเริ่มทำงานกับ heat สำหรับการทดสอบนี้ใช้ user : demo โดยถูกสร้างจาก admin
- Flavor ขนาดของ flavor ที่ใช้ใน stack ซึ่งถูกสร้างจาก parameter ใน HOT

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



# รายละเอียดของแต่ละช่องของ form

- Image name or ID ชื่อหรือหมายเลข ID ของ image ที่ใช้ใน stack ซึ่งถูกสร้างจาก parameter ใน HOT

```
6     image:
7     type: string
8     label: Image name or ID
9     description: Image to be used for compute instance
10    default: Cirros
```

- Key name ชื่อ key ที่ใช้สำหรับ instance ซึ่งถูกสร้างจาก parameter ใน HOT

```
16    key:
17    type: string
18    label: Key name
19    description: Name of key-pair to be used for compute instance
20    default: keypair
```

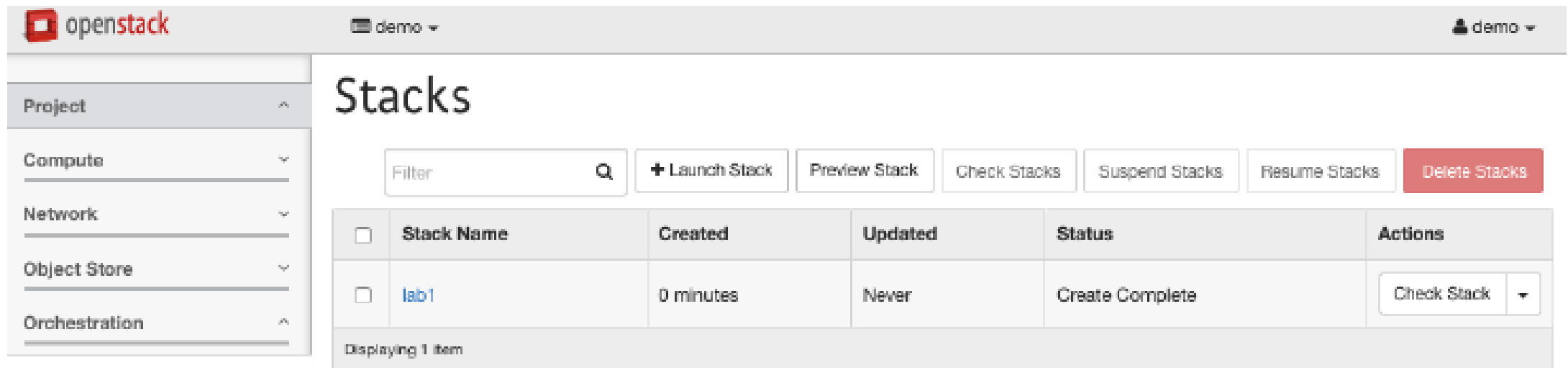
## รายละเอียดของแต่ละช่องของ form

- Private network name or ID ชื่อหรือหมายเลข ID ของ network ที่ใช้ใน stack ซึ่งถูกสร้างจาก parameter ใน HOT

```
21  ▾  private_network:  
22      type: string  
23      label: Private network name or ID  
24      description: Network to attach instance to.  
25      default: Private
```

# วิธีการทดสอบ

- กด Launch เมื่อสำเร็จจะแสดงผลใน Stacks menu



The screenshot shows the OpenStack dashboard interface. On the left is a sidebar with a 'Project' menu containing 'Compute', 'Network', 'Object Store', and 'Orchestration'. The main area is titled 'Stacks'. At the top of this area is a search bar labeled 'Filter' and a row of buttons: '+ Launch Stack', 'Preview Stack', 'Check Stacks', 'Suspend Stacks', 'Resume Stacks', and 'Delete Stacks'. Below these is a table with the following columns: 'Stack Name', 'Created', 'Updated', 'Status', and 'Actions'. There is one row in the table with the stack name 'lab1', created '0 minutes' ago, updated 'Never', and status 'Create Complete'. The 'Actions' column for this row contains a 'Check Stack' button and a dropdown arrow. At the bottom of the table, it says 'Displaying 1 item'.

	Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/>	lab1	0 minutes	Never	Create Complete	Check Stack ▾

Displaying 1 item

# วิธีการสร้าง Stack ด้วย command line

1. Source credential ของ user ที่ต้องการจะสร้าง Stack

```
$ source keystone-demo
```

2. สร้าง instance จากค่า parameter และ resource จาก HOT ไฟล์ (.yaml)

```
$ openstack stack create -t <HOT.yaml> <Name_Stack>
```

3. หลังจากผ่านไประยะเวลาสั้นๆ ตรวจสอบการสร้าง stack โดยใช้คำสั่ง

```
$ openstack stack list
```

4. แสดงชื่อและที่อยู่ IP ของ instance

```
$ openstack stack output show -all <Name_Stack>
```

```
$ openstack server list
```

## การลบ Stack ด้วย command line

```
$ openstack stack delete --yes <Name_Stack>
```

สามารถศึกษาวิธีการเขียน HOT ได้ที่

[https://docs.openstack.org/developer/heat/template\\_guide/index.html](https://docs.openstack.org/developer/heat/template_guide/index.html)