Service Orchestration in an OpenStack Cloud

Rick Ashford
Sr. Technical Specialist
rick.ashford@suse.com

Niel Bornstein
Sr. Technical Specialist
niel.bornstein@suse.com

Deploying Services in the Cloud

How do I deploy it?

There are 3 ways to deploy services in the cloud:

The easy way

The “not quite as easy” way

The “hard up-front, but totally easier in the end” way

How do I deploy it?

The easy way ...

Manual deployment process

- Log into the dashboard
- Go to the images section
- Select your image(s) and launch it
- Configure networking and storage as necessary

Very quick and easy...
If you're doing it once

How do I deploy it?

The “not quite as easy” way ...

Use the API

- Python libraries
- Script out the manipulation of compute, network, and storage

Labor intensive up front, but scales easily to large deployments

Not terribly friendly to all potential cloud users

How do I deploy it?

The “hard up-front, but totally easier in the end” way

OpenStack Heat Project

- Incubated project for Grizzly
- Fully supported as of SUSE OpenStack Cloud 3.0 (Havana)

openstack
CLOUD SOFTWARE

Heat is a service to orchestrate multiple composite cloud applications

How do I deploy it?

The “hard up-front, but totally easier in the end” way

“Heat is a service to orchestrate multiple composite cloud applications using the AWS Cloud Formation template format, through both an OpenStack-native ReST API and a CloudFormation-compatible Query API.”

How do I deploy it?

The “hard up-front, but totally easier in the end” way

“Heat is a service to orchestrate multiple composite cloud applications using the AWS Cloud Formation template format, through both an OpenStack-native ReST API and a CloudFormation-compatible Query API.”

Um ... what?

How do I deploy it?

The “hard up-front, but totally easier in the end” way

Heat allows you to pre-define a set of compute, network, and storage requirements to provide a specific service, and deploy the whole thing *automagically*.

Getting to know Heat

Getting to know Heat

Why “Heat”?

“It makes the clouds rise”

Getting to know Heat

Multiple formats

- AWS Cloud Formations Query API (CFN)
  - JSON
  - YAML
- Heat Orchestration Template (HOT)
  - YAML

Getting to know Heat

YAML (YAML Ain't a Markup Language) Format

```
invoice: 34843
date   : 2001-01-23
bill-to: &id001
    given  : Chris
    family : Dumars
    address:
        lines: |
            458 Walkman Dr.
            Suite #292
        city    : Royal Oak
        state   : MI
        postal  : 48046
ship-to: *id001
product:
    - sku         : BL394D
      quantity    : 4
      description : Basketball
      price       : 450.00
    - sku         : BL4438H
      quantity    : 1
      description : Super Hoop
      price       : 2392.00
tax  : 251.42
total: 4443.52
```

Structured via indentation (one or more spaces). Sequence items denoted by a dash, and key value pairs within a map are separated by a colon.

Getting to know Heat

“Hello world”

```
heat_template_version: 2013-05-23

description: Simple template to deploy a single compute instance

resources:
  blog:
    type: OS::Nova::Server
    properties:
      key_name: rashford
      image: Wordpress-0.0.8-kvm
      flavor: m1.tiny
```

Getting to know Heat

Template Parameters

```
heat_template_version: 2013-05-23

description: Simple template to deploy a single compute instance

parameters:
  key_name:
    type: string
    label: Key Name
    description: Name of key-pair to be used for compute instance
  image_id:
    type: string
    label: Image ID
    description: Image to be used for compute instance
  instance_type:
    type: string
    label: Instance Type
    description: Type of instance (flavor) to be used

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      key_name: { get_param: key_name }
      image: { get_param: image_id }
      flavor: { get_param: instance_type }
```

Getting to know Heat

Template Parameters

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

```
parameters:
  database_password:
    type: string
    label: Database Password
    description: Password to be used for database
    hidden: true
```

Getting to know Heat

Template Parameters: Restricting User Input

```
parameters:
  instance_type:
    type: string
    label: Instance Type
    description: Type of instance (flavor) to be used
    constraints:
      - allow_values: [ m1.medium, m1.large, m1.xlarge ]
        description: Value must be one of m1.medium, m1.large or m1.xlarge.
```

```
parameters:
  database_password:
    type: string
    label: Database Password
    description: Password to be used for database
    hidden: true
    constraints:
      - length: { min: 6, max: 8 }
        description: Password length must be between 6 and 8 characters.
      - allowed_pattern: "[a-zA-Z0-9]+"
        description: Password must consist of characters and numbers only.
      - allowed_pattern: "[A-Z]+[a-zA-Z0-9]*"
        description: Password must start with an uppercase character.
```

Useful Resources

- http://docs.openstack.org/developer/heat/template_guide/openstack.html
- https://github.com/openstack/heat-templates

It's ***hands-on time!***

Thank you.

About the Lab

- SSID: “SUSE Heat Lab”
- Horizon: http://192.168.124.81
- User = geeko$NUM Pass = openstack
- Lab Materials:http://192.168.124.1/heat
- Images: root pwd = linux
- If you want the materials, get scanned and have them note it
- rashford@suse.com
- Floating network id = dbc0cbcf-a59f-4e54-83bc-bfd8b54e9064


SUSE®
We adapt. You succeed.

Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org

Unpublished Work of SUSE. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE.
Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.