

Submitted By: Annie Jain

Sap Id: 500083967

Roll No: R214220179

Batch: B3 Hons

CLOUD APPLICATION DEVELOPMENT

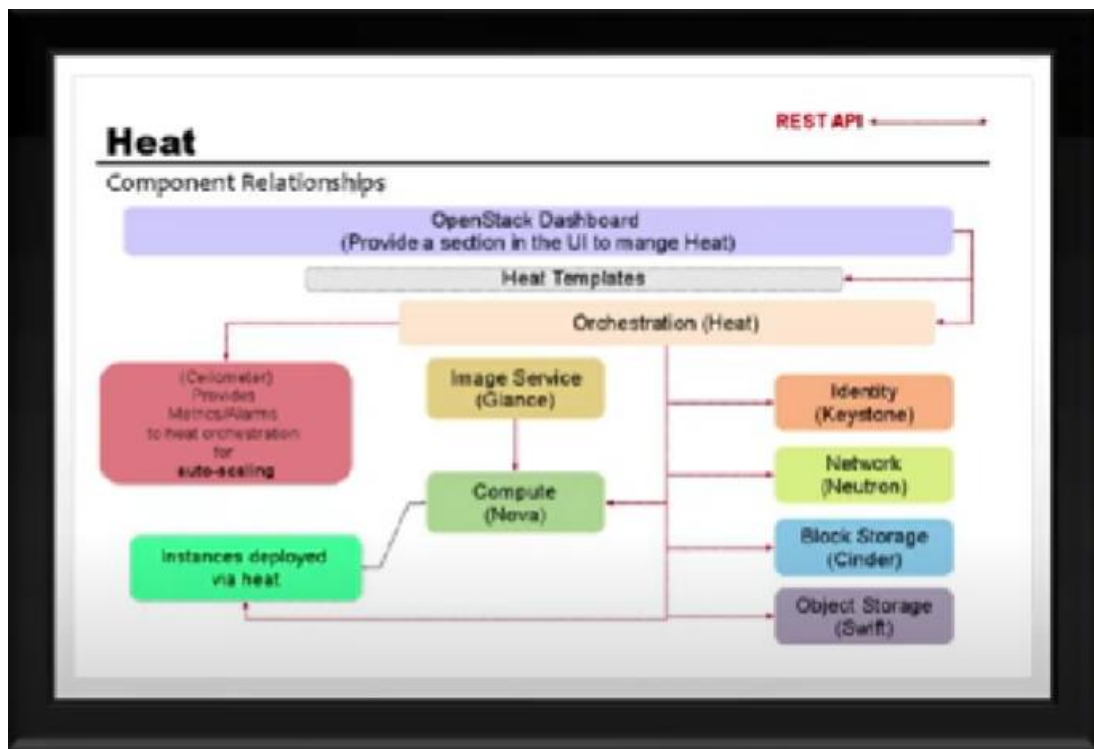
OPENSTACK LAB EXPERIMENT – 10

OBJECTIVE: Deploying an application using OpenStack's Heat Orchestration Service.

Introduction

OpenStack Heat is a service that allows users to automate the deployment of cloud infrastructure and applications. In this report, we will describe the deployment of an application using OpenStack Heat, including the Heat template used, the stack launch process, and the monitoring and updating of the stack.

The application deployed is a simple web application that serves a static HTML page. The application requires a virtual machine instance and a network configuration with appropriate security group settings to allow access to the web server.

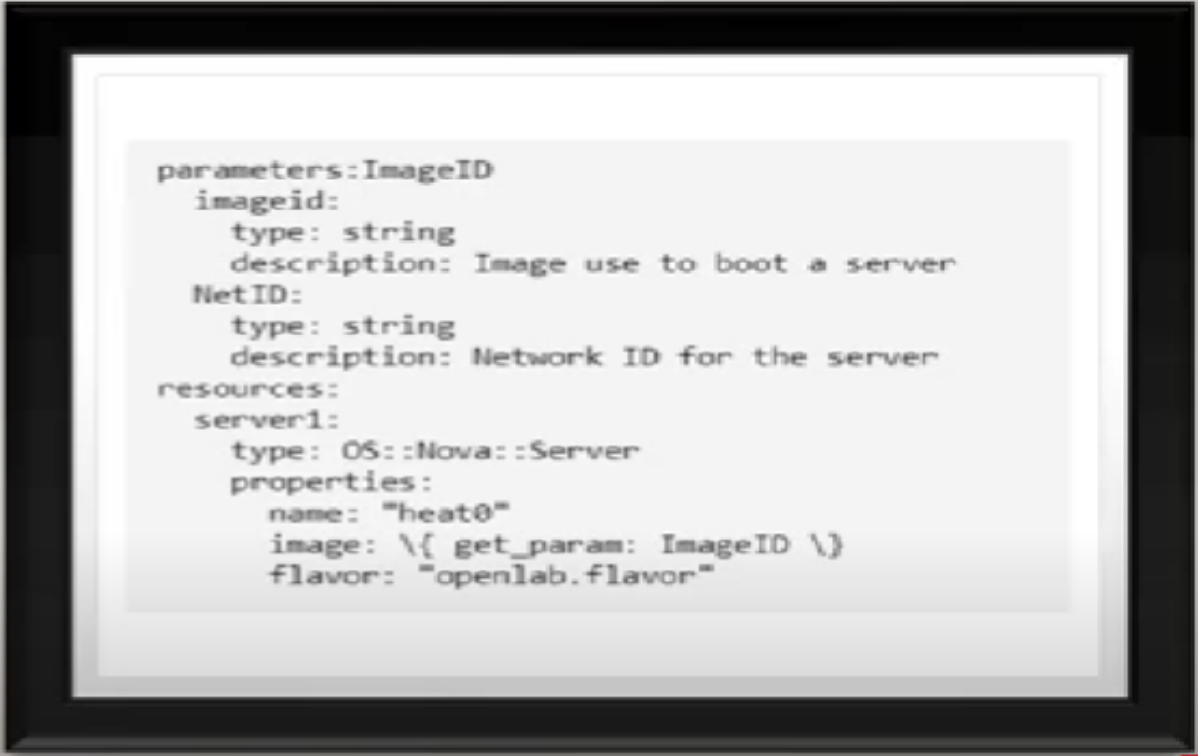


Heat Template

The Heat template used to deploy the application is a YAML-formatted file that defines the necessary resources and their configurations. The template includes the following resources:

- A virtual machine instance with the required image and flavor configurations.
- A security group allowing incoming traffic on port 80.
- A network configuration with the required subnet and router settings.

The template also includes several parameters, such as the image ID, flavor name, and network CIDR.

A framed image showing a Heat template snippet. The snippet is a YAML configuration for a server resource. It includes parameters for ImageID and NetID, and a resource named server1 of type OS::Nova::Server. The server1 resource has properties for name, image, and flavor. The image property is a Jinja2 template that uses the ImageID parameter. The flavor property is a string.

```
parameters:ImageID
  imageid:
    type: string
    description: Image use to boot a server
NetID:
  type: string
  description: Network ID for the server
resources:
  server1:
    type: OS::Nova::Server
    properties:
      name: "heat0"
      image: \{ get_param: ImageID \}
      flavor: "openlab.flavor"
```

Launching the Stack

To launch the stack, we used the OpenStack CLI and specified the name of the stack, the path to the template, and the required parameters. The command used was:

```
heat stack-create myapp -f /path/to/template.yaml -P "image_id=Ubuntu 20.04 LTS" -P "flavor=m1.small" -P "cidr=10.0.0.0/24"
```

The stack launch process took approximately 10 minutes, during which time Heat created the necessary resources and configured them according to the specifications in the Heat template. No issues were encountered during the deployment process.

Monitoring and Updating the Stack

After launching the stack, we monitored the deployment process using the OpenStack dashboard and the CLI. We were able to view the status of the stack and the individual resources, and we verified that the application was accessible via a web browser.

We also tested the updating process by modifying the Heat template to add an additional security group, and then using the **heat stack-update** command to apply the changes. The updating process took approximately 5 minutes and was successful, with the new security group being added to the stack.

Results

The deployment of the application using OpenStack Heat was successful, with all resources being created and configured according to the specifications in the Heat template. The application was accessible via a web browser, and the updating process worked as expected.

The use of OpenStack Heat allowed us to automate the deployment of the application, ensuring consistency and simplifying management. The Heat template provided a reusable definition of the necessary resources and configurations, which can be modified and used for future deployments.

Conclusion

In conclusion, OpenStack Heat provides a powerful tool for automating the deployment of cloud infrastructure and applications. The deployment of our simple web application using Heat was successful, and the use of Heat templates allowed for consistent and efficient deployment. We recommend using OpenStack Heat for future cloud deployments