Infrastructure Automation with Deployment Manager

Architecting with GCP Fundamentals: Infrastructure

DEPLOYMENT MANAGER, CLOUD LAUNCHER

DEPLOYMENT MANAGER

Last modified 2017-11-27

Agenda

- **Deployment Manager**
- Configuration
- Cloud Launcher
- Lab
- Quiz

Deployment Manager

- An infrastructure automation tool
  - Creates GCP resources
  - Not limited to 1 VM like an Instance Template
- Create the Deployment Template in a Cloud API-enabled environment such as Cloud Shell
  - View results and manage deployment in console

Comparing orchestration tools

	Deployment Manager	Puppet	Chef	Terraform	Cloud Formation
Imperative vs Declarative	Declarative	Declarative	Imperative	Declarative	Declarative
Hosted	Yes	No	No	No	Yes
Driven by Discovery/Swagger	Yes	No	No	No	No
Multi-Platform	No	Yes	Yes	Yes	No
Integrated with a Platform (IAM, UI, ...)	Yes	No	No	No	Yes

Google Cloud

https://cloud.google.com/solutions/google-compute-engine-management-puppet-chef-salt-ansible

Agenda

- Deployment Manager
- **Configuration**
- Cloud Launcher
- Lab
- Quiz

Creating a Deployment Configuration

- Creating a configuration
  - `*.yaml` file defines the basic configuration
  - Include `import` at the top of the yaml file to expand to full-featured templates written in python or jinja2
  - Program configuration is bidirectional and interactive: receives data like `machine-type` and returns data like `ip-address`
- Use "preview" to validate configuration before using it:
- Example
  - `gcloud deployment-manager deployments update accel --config *.yaml --preview`

Google Cloud

Configuration limits: All configurations are expanded on the server side within a controlled environment that Deployment Manager maintains. In order to prevent abuse, this environment is closely managed by the Deployment Manager team and has some limitations:

Neither your original configuration nor your expanded configuration can exceed 10 MB.
Any configurations uploaded to Deployment Manager are limited in the amount of time the configuration can take to run and the amount of processing power the configuration consumes during expansion. If you run into this limitation, contact Deployment Manager for more information.
Any Python templates you use cannot make any system or network calls. These templates will automatically be rejected.

Templates

- Templates can be nested
  - Isolate specific functions into meaningful files
  - Create reusable assets
  - Example: a separate template for firewall rules
- Templates have properties
- Templates can use environment variables
- Supports the startup script and metadata capabilities
- Deployments can be updated - uses GCP API
  - Add resources: default policy is `acquire` or `create` as needed
  - Remove resources: default policy is to `delete` the resource

Agenda

- Deployment Manager
- Configuration
- **Cloud Launcher**
- Lab
- Quiz

Cloud Launcher

- Pre-packaged solutions by 3rd party vendors
- A "solution marketplace"
- Separate fees
  - license fees for software
  - image usage fees
- Image usage fee vs separate license -up to vendor
- Google updates images, but not running instances
- Cloud technology partners

Agenda

- Deployment Manager
- Configuration
- Cloud Launcher
- **Lab**
- Quiz

Lab: Deployment Manager

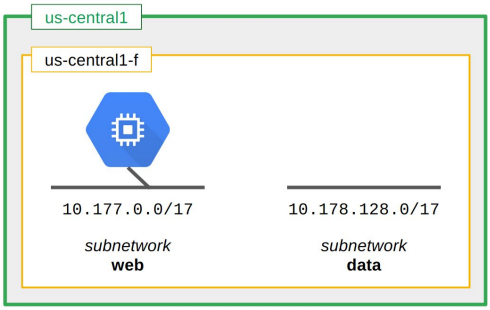
Objectives

In this lab, you learn how to perform the following tasks:

- Download and review a set of Deployment Manager templates
- Use the templates to deploy two networks and a VM instance

Completion: 40 minutes

Access: 80 minutes



Lab Review

In this lab you:

- Customized Deployment Manager templates
- Deployed a network with two subnetworks and a VM

Agenda

- Deployment Manager
- Configuration
- Cloud Launcher
- Lab
- **Quiz**

Quiz

What kinds of files form the Deployment Manager templates?

1. Templates are composed of *.yaml, java, and Node.js files.
2. Templates are composed of *.yaml, python, and jinja2 files.
3. Templates are composed of Powershell, python, and text files.
4. Templates are composed of bash, *.yaml, and Angular.js files.

Quiz

What kinds of files form the Deployment Manager templates?

1. Templates are composed of *.yaml, java, and Node.js files.
2. Templates are composed of *.yaml, python, and jinja2 files. *
3. Templates are composed of Powershell, python, and text files.
4. Templates are composed of bash, *.yaml, and Angular.js files.

Explanation:
*.yaml, python, and jinja2

Quiz

What service does Cloud Launcher provide?

1. Provides pre-packaged 3rd party solutions using Deployment Manager templates.
2. It is an Android app that sends notifications when new Zones are launched.
3. Provides 3rd party solutions using the Google Cloud API and bash scripts.
4. Provides an open source standard alternative to Deployment Manager.

Quiz

What service does Cloud Launcher provide?

1. Provides pre-packaged 3rd party solutions using Deployment Manager templates. *
2. It is an Android app that sends notifications when new Zones are launched.
3. Provides 3rd party solutions using the Google Cloud API and bash scripts.
4. Provides an open source standard alternative to Deployment Manager.

Explanation:
Pre-packaged solutions by 3rd party vendors in a "solution marketplace".

Quiz

Which of the following is true of Deployment Manager Templates?

1. They cannot be nested, can use environment variables, but do not have their own properties.
2. They can be nested, but cannot share data except as passed in an external file.
3. There can be multiple templates, but they cannot be nested and don't share properties or variables.
4. They can be nested, have properties, and can use environment variables.

Quiz

Which of the following is true of Deployment Manager Templates?

1. They cannot be nested, can use environment variables, but do not have their own properties.
2. They can be nested, but cannot share data except as passed in an external file.
3. There can be multiple templates, but they cannot be nested and don't share properties or variables.
4. They can be nested, have properties, and can use environment variables. *

Explanation:
Templates can be nested, have properties, and can use environment variables.

More...

Startup scripts

- https://cloud.google.com/compute/docs/startupscript

Shutdown scripts

- https://cloud.google.com/compute/docs/shutdownscript

Storing and retrieving metadata

- https://cloud.google.com/compute/docs/storing-retrieving-metadata