Exploring helper scripts

Helper scripts, or *template modules*, are helper files that can make your templates more efficient by performing specific functions. For example, you can use helper scripts to interpret resource metadata, create files, and launch services.

You will now explore a Python helper script that names a virtual machine, given a prefix and a suffix.

Before you begin

- If you want to use the command-line examples in this guide, install the `gcloud` command-line tool (/sdk).
- If you want to use the API examples in this guide, set up API access (/deployment-manager/docs/reference/latest).

Open the helper script

The basic helper script in this example generates the name for a virtual machine. To view the script, run these commands:

```
ploymentmanager-samples/examples/v2/step_by_step_guide/create_a_helper_script

common.py  # use your preferred text editor
```

The `GenerateMachineName()` function takes a prefix and suffix, and generates a name in the format `prefix-suffix`:

examples/v2/step_by_step_guide/create_a_helper_script/common.py (https://github.com/GoogleCloudPlatform/deployment-manager-samples/blob/master/examples/v2/step_by_step_guide/create_a_helper_script/common.py)

ntmanager-samples/blob/master/examples/v2/step_by_step_guide/create_a_helper_script/common.py)

```
"""Generates name of a VM."""


def GenerateMachineName(prefix, suffix):
  return prefix + "-" + suffix
```

Using the helper script in the template

To use `common.py` in `vm-template.py`, several changes must be made to the template.

To view the changes, open `vm-template.py`:

```
vm-template.py
```

The template contains code comments that highlight the changes.

Note that the template imports `common.py` at the top of the file. In the `resources` section, the `name` fields for the VMs now call `GenerateMachineName()`.

examples/v2/step_by_step_guide/create_a_helper_script/vm-template.py (https://github.com/GoogleCloudPlatform/deploymentmanager-samples/blob/master/examples/v2/step_by_step_guide/create_a_helper_script/vm-template.py)

tmanager-samples/blob/master/examples/v2/step_by_step_guide/create_a_helper_script/vm-template.py)

```
"""Creates the virtual machine."""

# `common.py` is imported below.
import common

COMPUTE_URL_BASE = 'https://www.googleapis.com/compute/v1/'


def GenerateConfig(context):
  """Generates configuration of a VM."""
```

```
resources = [{
    'name': common.GenerateMachineName('myfrontend', 'prod'),
    'type': 'compute.v1.instance',
    'properties': {
        'zone': 'us-central1-f',
        'machineType': COMPUTE_URL_BASE + 'projects/' + context.env['project']
                       + '/zones/us-central1-f/machineTypes/f1-micro',
        'disks': [{
            'deviceName': 'boot',
            'type': 'PERSISTENT',
            'boot': True,
            'autoDelete': True,
            'initializeParams': {
                'sourceImage': COMPUTE_URL_BASE + 'projects/'
                               'debian-cloud/global/images/family/debian-9'}
        }],
        'networkInterfaces': [{
            'network': COMPUTE_URL_BASE + 'projects/' + context.env['project']
                       + '/global/networks/default',
            'accessConfigs': [{
                'name': 'External NAT',
                'type': 'ONE_TO_ONE_NAT'
            }]
        }]
    }
}]
return {'resources': resources}
```

View the changes to the configuration

To view the updated configuration, open `two-vms.yaml`:

```
two-vms.yaml
```

Note that the helper script `common.py` must be imported in the configuration as well.

Deploy your configuration:

```
d deployment-manager deployments create deployment-with-helper-script --config two-v
```

To view the deployment, including the resources with the generated names, run:

```
ud deployment-manager deployments describe deployment-with-helper-script
```

Delete Your Deployment

We recommend that you delete the deployment to avoid charges. You don't need this deployment for the next step. Run the following command to delete the deployment:

```
d deployment-manager deployments delete deployment-with-helper-script
```

Looking ahead: updating deployments

Next, learn to add, delete, and change the properties of resources in a deployment as your application evolves.

Previous
← Understanding template properties and using environment variables
(/deployment-manager/docs/step-by-step-guide/using-template-and-environment-variables)

Next
Updating deployments →
(/deployment-manager/docs/step-by-step-guide/updating-a-deployment)

Last updated 2020-07-27 UTC.