Will it run on my OpenStack?

Building an executable validation testsuite for CPIs

Beyhan Veli, SAP SE

Cornelius Schumacher, SUSE Linux

Cloud Foundry Summit, Santa Clara, 2016
Who are we?

Beyhan Veli
SAP SE
beyhan.veli@sap.com
@beyhan

Cornelius Schumacher
SUSE Linux
cschum@suse.com
@cschum
cf push
Will it run on my OpenStack?

• Deploy a BOSH director
• Deploy Concourse
• Configure and run the OpenStack CPI test pipeline
• Manual validation steps described on docs.cloudfoundry.org
Prerequisites

1. An OpenStack environment running one of the following supported releases:
   - Juno
   - Kilo
   - Liberty

   Note: Juno has a bug that prevents BOSH to assign specific IPs to VMs. You have to apply a Nova patch to avoid this problem.

2. The following OpenStack services:
   - Identity: BOSH authenticates credentials and retrieves the endpoint URLs for other OpenStack services.
   - Compute: BOSH boots new VMs, assigns floating IPs to VMs, and creates and attaches volumes to VMs.
   - Image: BOSH stores stemcells using the Image service.
   - (Optional) OpenStack Networking: Provides network scaling and automated management functions that are useful when deploying complex distributed systems.

3. The following OpenStack networks:
   - An external network with a subnet.
   - An private network with a subnet. The subnet must have an IP address allocation pool.

4. Configuration of a new OpenStack Tenant

   1. Automated configuration
      
      You can use a Terraform environment template to configure your OpenStack tenant.

   2. Manual configuration
      
      Note: See the OpenStack documentation for help finding more information.

      Alternatively, you can do the following things manually as described below:
      - Create a Keypair
      - Create and configure Security Groups.
      - Allocate a floating IP address.

Create a Keypair

1. Select Access & Security from the left navigation panel.
2. Select the Keypairs tab.
Step 1: Create a Deployment Manifest

1. Create a deployment directory.

   ```bash
   $ mkdir -p/my-bosh
   ```

2. Create a deployment manifest file named `bosh.yml` in the deployment directory based on the template below.

   In the template, you must replace the `NETWORK-UUID`, `PRIVATE-IP`, `PRIVATE-CIDR`, `PRIVATE-GATEWAY-IP`, `DNS-IP`, `FLOATING-IP`, `OPENSTACK-PASSWORD`, `IDENTITY-API-ENDPOINT`, `OPENSTACK-TENANT`, and `OPENSTACK-USERNAME` properties. We describe replacing these properties in Step 2: Prepare an OpenStack environment.

   ```yaml
   ---
   name: bosh
   releases:
     - name: bosh
       url: https://bosh.io/d/github.com/cloudfounder/sha1: ff2f4e16e827f6b03c8c519cd52a6f8b820cfd
       name: bosh-openstack-cpi
       url: https://bosh.io/d/github.com/cloudfounder/sha1: 6621e6132673f8d9d6ba9c6158d18161b8d
   resource_pools:
     - name: vms
       network: private
       stencils:
         url: https://bosh.io/d/stencils/bosh-openstack
         sha1: e0f33d58013816520e65d3d11df54f5
     cloud_properties:
       instance_type: m3.xlarge
   disk_pools:
     - name: disks
       disk_size: 20_000
   networks:
     - name: private
       type: manual
       subnets:
         - range: PRIVATE-CIDR # ---- Replace with a gateway: PRIVATE-GATEWAY-IP # ---- Replace:
           dns: [DNS-IP] # ---- Replace with your DNS
           cloud_properties: {net_id: NETWORK-UUID}
   ```

   Note: The example below uses several predefined passwords. We recommend replacing them with passwords of your choice.
Clusters with BOSH

Deploying Concourse with BOSH provides a scalable cluster with health management and rolling upgrades.

If you’re not yet familiar with BOSH, learning it will be a bit of an investment, but it should pay off in spades. There are a lot of parallels between the philosophy of BOSH and Concourse.

Prepping the Environment

To go from nothing to a BOSH managed Concourse, you’ll need to do the following:

- **Initialize a BOSH director**, which is the server that orchestrates BOSH deployments.
- **Set up your Cloud Config**, which describes the infrastructure to which Concourse will be deployed.
- **Upload the stemcell**, which contains the base image used for VMs managed by BOSH.

You can skip some of this if you already have a working BOSH director running BOSH v2.55.4 and up.

Deploying Concourse

Once you’ve got all that set up, download the releases listed for your version of Concourse from the Downloads section, and upload them to the BOSH director.

Next you’ll need a Concourse BOSH deployment manifest. An example manifest is below; you’ll want to replace the REPLACE_ME bits with whatever values are appropriate.

Note that the VM types and network names used in this manifest must be present in your Cloud Config. Consult Prepping the Environment if you haven’t set it up yet.

```yaml
---
name: conourse
```
OpenStack Prerequisites and Variables

The concourse pipeline requires some variables to be set. The easiest way is to create a file called `secrets.yml` using the template and fill in the values as you go through the setup steps.

If you are setting this up for your own environment, you are probably not publishing your own releases. Some of the properties defined there are for promoting a real release only; you can fill those with anything you’d like or remove the corresponding tasks entirely from the pipeline.

All instructions include the variables to fill in `monospace`.

External Prerequisites

- Install `concourse`
- Create S3 buckets for the artifacts created during the build
  - s3_openshift_cpi_pipeline_bucket_name
  - s3_openshift_cpi_state_files_bucket_name
- Credentials for these S3 buckets
  - s3_openshift_cpi_pipeline_access_key
  - s3_openshift_cpi_pipeline_secret_key
  - s3_openshift_cpi_state_files_access_key
  - s3_openshift_cpi_state_files_secret_key
- Create a github repository used for locus in the pipeline
  - pipeline_lock_repository_url
- Choose a name for the pipeline lock in this repository
  - pipeline_lock_pool_name

OpenStack Prerequisites

- Create a project where the tests can run. The `cleanup` task at the end will clean everything in that project.
  - openshift_tenant
- Fill in some infrastructure details
  - Keystone v2 auth url: openshift_auth_url_v2
  - Keystone v3 auth url (optional): openshift_auth_url_v3
  - User for the tenant: openshift_username
  - Password for the tenant: openshift_api_key
  - openshift_flavor_with_ephemeral_disk
  - openshift_flavor_with_no_ephemeral_disk
- Create SSH key pairs
  - bosh director and bats: openshift_default_key_name
  - Reverse tests
    - lifecycle_openshift_default_key_name
    - v3_e2e_default_key_name
- Create networks with DHCP enabled. Each test uses a different network so they can be run in parallel.
  - Reverse tests
    - lifecycle_openshift_net_id
    - bosh ubuntu
    - bats_dynamicsubuntu_primary_net_id
Validate your OpenStack Instance

On this page:

1. Can you access the OpenStack APIs for your instance of OpenStack?
2. Can you access OpenStack metadata service from a virtual machine?
3. Can you ping one virtual machine from another?
4. Can you invoke large numbers of API calls?
5. Can you create and mount a large volume?
6. Can you upload and deploy an Ubuntu 14.04 64-bit Server Cloud Image?
7. Can networking be configured for both external and internal IPs?
8. Can you access the Internet from within instances?

Page last updated: March 22, 2016

This topic helps you validate your target OpenStack in preparation for installing BOSH and deploying Cloud Foundry.

You will need a running OpenStack environment. Note that only OpenStack releases newer than Folsom are supported.

The requirements listed here are considered necessary but not sufficient for BOSH to be able to use your OpenStack deployment. If you cannot perform any one of these tasks successfully, BOSH will not work; however, satisfying all these requirements does not ensure that BOSH will work.

See Troubleshooting Cloud Foundry on OpenStack for additional troubleshooting information.

Can you access the OpenStack APIs for your instance of OpenStack?

You should verify that you have your OpenStack API credentials and can make API calls. Credentials are a combination of your user name, password, and the tenant or project your cloud is running under. Some providers also require you to set the region.
We can do better, can't we?
Goal

Is my OpenStack ready?
For BOSH?
For Cloud Foundry?
For my product?

Put expert knowledge to answer these questions into a simple tool,
the Cloud Foundry OpenStack validator

Any of you should be able to use this validator
Overview

My Product

Service A

Service B

My App

BOSH

CPI

Deploy

API

API

openstack

Cloud Foundry
Demo
Your OpenStack
  can create a VM
  can create a disk
  can attach the disk to the vm
  can detach the disk from the VM
  can delete the disk
  can delete the VM

Finished in 2 minutes 8.2 seconds (files took 0.10211 seconds to load)
6 examples, 0 failures
$ src/run_validator /var/folders/3_/tksbx/fdd0lvglv0nkyr76vjr0000gp/T/tmp.xmn81f44/
CPI already installed at /var/folders/3_/tksbx/fdd0lvglv0nkyr76vjr0000gp/T/tmp.xmn81f44/

Your OpenStack
  can create a VM
  can create a disk
      Disk could not be created. OpenStack error: Unable to connect to the OpenStack Keystone API

Finished in 0 minutes 45.3 seconds (files took 0.08447 seconds to load)
2 examples, 1 failures

$
Test Case

```ruby
it 'can create a disk' do
  @disk_cid = with_cpi('Disk could not be created.') {
    cpi.create_disk(2048, {}, @vm_cid)
  }
  expect(@disk_cid).to be
end
```
Test Scenarios

OpenStack Validator

RSpec Tests

API

SSH

CPI

API

API
OpenStack Validator

Binary + Config + CPI Release + Stemcell
• Unpack CPI and stemcell
• Install CPI
• Read config
• Upload stemcell
• Execute tests
• Report results
This is better, isn't it?
What we’ve learned so far

• BOSH OpenStack CPI Project
• SAP and SUSE cooperating on the project

• Emerging topic: How to validate OpenStack instances?
Experience from setting up CI system

- Manual validation steps
- Performance issues
- Access (SSL, external and internal connectivity)
- Network configuration (MTU settings)

- In hindsight: Having a validator would have been great
Outlook

• Have tests for entire stack (OpenStack, BOSH, CF)
• Continuous improvement
• Extend validator with your own tests
• Extend to other infrastructures?
Find us on GitHub:

https://github.com/cloudfoundry-incubator/cf-openstack-validator

Talk to us:

Beyhan Veli <beyhan.veli@sap.com>
Cornelius Schumacher <cschum@suse.com>