BOSH Documentation

1 Introduction to BOSH

BOSH is a framwork and tool-chain for release engineering, deployment and life cycle management of distributed services. In this manual we describe the architecture, topology, configuration and use of BOSH, as well as the structure and conventions used in packaging and deployment.

BOSH introduces a fairly prescriptive way of managing systems and services. It was originally developed in the context of the Cloud Foundry Application Platform as a Service, but even if this has been the primary consumer, the framework is general purpose and can be used to deploy many different

TODO: what is BOSH not

2 BOSH Component Definitons

2.1 Infrastructure as a Service (IaaS)

The core BOSH engine is abstracted away from any particular Infrastructure as a Service (IaaS), such as VMware vSphere, AWS or OpenStack. The interface to thse is implemented as plugins to BOSH. At the time of writing only the vSphere implementation is fully featured.

2.2 Cloud Provider Interface (CPI)

As a user of BOSH you're not directly exposed to the the BOSH Cloud Provider Interface, but it can be helpful to understand it's primitives when learning how BOSH works.

create_stemcell delete_stemcell create_vm delete_vm configure_networks create_disk delete_disk attach_disk detach_disk

Please refer to the API documentation +director/lib/director/cloud.rb+ for a detailed explanation of the CPI primitives.

The CPI is used primarily to do low level creation and management of resources in an IaaS, once a resouce is up and running command and control is handed over to the higher level BOSH Director-Agent interaction.

2.3 BOSH Director

The Director is the core orchestrating component in BOSH which controls creation of VMs, deployment and other life cycle events of software and services.

2.4 BOSH CLI

TODO

2.5 BOSH Agent

Every VM created by BOSH includes a an Agent, which does initial configuration of a system once the Director has created it through the CPI and further install software and services, once the Director instructs it to apply a Job.

2.6 Stemcells

A BOSH stemcell is typically a simple VM template with an embedded BOSH Agent. These are uploaded using the BOSH CLI and used by the Director when creating VMs through the CPI. When the Director create a VM through the CPI, it will pass along configurations for networking and storage as well as the location and credentials for the BOSH Message Bus and the BOSH Blobstore.

2.7 Releases

A Release in BOSH is a packaged bundle of service descriptors (known as Jobs in BOSH), a collection of software bits and configurations. A release contains all the static bits (source or binary) required to have BOSH manage an application or a distributed service. A Release is typcially not restricted to any particular environment an as such it can be re-used across clusters handling different stages in a service life cycle, such as development, QA, staging or production. The BOSH CLI manages both the creation of releases and the deployments into specific environment.

2.8 Deployments

While BOSH Stemcells and Releases are static compnents, we say that they are bound together into a Deployment by what we call a Deployment Manifest. In the Deployment Manifest you declare pools of VMs, which networks they live on, which Jobs (service componens) from the Release you want to activate. Job configuration specify life cycle parameters, the number instances of a Job, as well as network and storage requirements. In the Deployment Manifest you can also speficy properties at various levels used to paramaterize configuration templates contained in the Release.

Using the BOSH CLI you specify a Deployment Manifest and perform a Deploy operation (+bosh deploy+), which will take this specification and go out to your cluster and either create or update resources in accordance to the specification.

2.9 Blobstore

2.10 BOSH Monitor

2.11 Message bus

3 Installing BOSH

TODO: replace this with gem install of cli gem TODO: remove this section when we don't need chef_deployer anymore

BOSH is a Ruby based toolchain and we suggest that you are set up with the following

- rbenv available at https://github.com/sstephenson/rbenv[1]
- Ruby 1.9.2

4 Configure BOSH Director

[NOTE] The current +chef-solo+ based installer is being re-written as a mini-bosh instance.

To install BOSH into an infrastructure we currently assume that the target VMs have been created.

TODO: check if we can provide vm_builder instructions for creating and //uploading these to IaaS.

```
    ~/projects/deployments/mycloud/cloud
      assets/
        director/
          director.yml.erb          <1>
          chef.rb                   <2>
          config.yml                <3>

cd ~/projects/bosh/chef_deployer
rake install

cd ~/projects/bosh/release
chef_deployer deploy ~/projects/deployments/mycloud/cloud
```

[1] https://github.com/sstephenson/rbenv

5 BOSH CLI

Go Oleg

6 Some section that has stemcells + releases /Director interaction

upload

7 Releases

7.1 Release Repository

A BOSH release is built from a directory tree following a structure described in this section:

7.2 Jobs

TODO: job templates TODO: use of properties TODO: "the job of a vm" TODO: monitrc (gonit) TODO: DNS support

7.3 Packages

TODO: ishisness! TODO: compilation TOOD: dependencies TODO: package specs

7.4 Sources

final release

7.5 Blobs

TODO: configuration options for Blobstore (Atmos vs S3)

7.6 Versioning schemes

7.7 Configuring Releases

7.8 Building Releases

7.9 Final Releases

8 BOSH Deployments

TODO: capture all the steps that the deployment does

8.1 BOSH Property Store

8.2 BOSH Deployment Manifest

TODO: options global/job propertes TODO: cloud_properties for the cli

9 BOSH Troubleshooting

TODO: cloud check TODO: BOSH SSH TODO: logs