On-Demand Service Provisioning with BOSH 2.0

Craig Furman & Alex Ley
Pivotal
Hey Ops,

Please can you create a new database in the mySQL cluster, and then create a user.

Once we have this you will need to redeploy our app with the updated config.

Thanks,

Alex
Hi Alex,

Please can you let us know your cost center and business approver?

Then we will be able to provision this in 3-5 days.

Regards,

Ops
> cf create-service redis shared-vm my-redis
But behind the scenes...

deploying and managing stateful services is HARD
Virtual Machine
Virtual Machine

Redis Process

Port: 6379

Redis Config
Virtual Machine

Redis Process

Redis Config

Port: 6379

Persistent Disk

Redis State
> cf create-service redis shared-vm another-redis
Virtual Machine

Redis Process 1
Port: 6379

Redis Process 2
Port: 6400

Redis Config One
Redis Config Two

Persistent Disk
Redis 1 State
Redis 2 State
Virtual Machine

- Redis Process 1
  - Port: 6379
- Redis Process 2
  - Port: 6400
- Redis Config One
- Redis Config Two

Orchestration Agent

Monitoring Agent

Persistent Disk
- Redis 1 State
- Redis 2 State
> cf create-service redis dedicated-vm prod-redis
Virtual Machine

Redis Process

Redis Config

Port: 6379

Persistent Disk

Redis State
BOSH

Not this BOSH
We

BOSH
> cf create-service redis dedicated-vm 4th-redis
> cf create-service redis dedicated-vm 4th-redis

FAILED: Service instance limit reached
We're fresh out of Redis

it@company.io

We're fresh out of Redis

Hey Ops,

Please can you re-deploy the Cloud Foundry Redis service with more dedicated Redis instances in the pool.

Thanks,

Alex
What if we could provision new resources on-demand using BOSH?
cf create service

= bosh deploy
BOSH 1.0 Manifest

- name: dedicated-node
  templates:
  - name: dedicated-node
    release: redis
  instances: 3
  resource_pool: service-vms
  persistent_disk: 4096
  networks:
  - name: services-network
    static_ips:
    - 10.0.26.6
    - 10.0.26.7
    - 10.0.26.8

networks:
- name: services-network
  type: manual
  subnets:
  - range: 10.0.26.0/24
    gateway: 10.0.26.1
    reserved:
    - 10.0.26.2 - 10.0.26.4
    static:
    - 10.0.26.5 - 10.0.26.25
    dns: [10.0.0.2]
  cloud_properties:
    subnet: subnet-abcd

resource_pools:
- name: service-vms
  network: services1
  stemcell:
    name: bosh-aws-xen-hvm-ub
    version: 3191
  cloud_properties:
    instance_type: m3.medium
BOSH 1.0 manifest generation challenges

- CPI-specific resource definitions
- Static IP management
- Manually create multi AZ’s deployments via multiple BOSH jobs
BOSH 2.0
BOSH 2.0

- Dynamic IP management
- Global Cloud Config
- 1st class support for multi-AZ job striping
- Manifest enhancements
name: on-demand-broker

releases:
  - name: &broker-release on-demand-service-broker
    version: latest

stemcells:
  - alias: trusty
    os: ubuntu-trusty
    version: latest

instance_groups:
  - name: broker
    instances: 1
    jobs:
      - name: broker
        release: &broker-release
        vm_type: medium
        persistent_disk_type: ten
        stemcell: trusty
        azs: [az1, az2]
        networks:
          - name: odb
Only using Platform (CF + BOSH) capabilities
Great for Apps Devs
> cf update-service myredis -c '{"upgrade": "v10"}'
> cf update-service mycassandra -c '{"seed_nodes": "5"}'}
Time for a richer Cloud Foundry Services experience?
> cf upgrade-service myredis v10
> cf scale-service cassandra seed_nodes 10
Operational Challenges

● Potential for lots of BOSH deployments
  ○ Resource usage monitoring
  ○ Quota management & Chargebacks
  ○ Will the BOSH director become the bottleneck?
  ○ IaaS limits?

● How much control to give app developers
  ○ Level of customisation
  ○ Upgrade lifecycle
Questions?

@alexevade
cfurman@pivotal.io