Building a production grade PostgreSQL Cloud Foundry Service
Production grade?
Production Ready Litmus Test:
Run a public CF.
Design Decisions
Defining service instances.
PostgreSQL Server vs. Cluster
Replication
Sync vs. Async
Async replication is built-in since PostgreSQL 9
PostgreSQL
Replication Limitations
Failure Detection

Automatic Failover
repmgr
Replication Manager for PostgreSQL clusters
- Monitoring replication performance
- Failure detection
- Automated failover
  - Leader election
  - Master promotion
Shared vs Dedicated PostgreSQL VMs
Shared
Shared PostgreSQL Cluster > Bad idea

- Single VM or single cluster of VMs
- Single PostgreSQL server or single PostgreSQL cluster
- Isolation limited to PostgreSQL multi-tenancy capabilities
Shared PostgreSQL = SPOF
Cloud Foundry Runtime

PostgreSQL Cluster

3 VMs

Service Instance | Service Instance | Service Instance
--- | --- | ---
Service Instance | Service Instance | Service Instance
Service Instance | Service Instance | Service Instance
Your shared PostgreSQL cluster goes down, all your PostgreSQL database instances go down.
Dedicated
Dedicated PostgreSQL instances > Good idea

- Service instance = dedicated VM or dedicated cluster of VMs
- Uses infrastructure isolation to enable multi-tenancy support
PostgreSQL failures are contained. Only one service instance affected.
Pre- vs. On-Demand Provisioning
Pre-provisioned service instances

- my-single-postgres-1
  - Postgres
  - VM#1

- my-3node-postgres-cluster-2
  - Postgres
  - VM#1
  - Postgres
  - VM#2
  - Postgres
  - VM#3

Service Broker
Easy Deployment

$> cf create-service postgresql single-small my-single-postgres-1
Pre-provisioned service instances

Service Broker

Service Instance
- my-single-postgres-1
  - Postgresql VM#1

Service Instance
- my-3node-postgres-cluster-2
  - Postgresql VM#1
  - Postgresql VM#2
  - Postgresql VM#3
Easy Deployment

$> cf create-service postgresql
cluster-small my-3node-postgres-cluster-2
Pre-provisioned service instances
On-Demand-Provisioned
On-demand provisioned service instances

- Service Broker
- Some Automation
On-demand provisioned service instances

Service Instance
my-single-postgres-1
  Postgresql VM#1

Service Broker

Some Automation
Pre-provisioned service instances

- Service Broker
- Some Automation

Service Instance
- my-single-postgres-1: Postgresql VM#1

Service Instance
- my-3node-postgres-cluster-2:
  - Postgresql VM#1
  - Postgresql VM#2
  - Postgresql VM#3
Automation Technology
Bosh = ❤
Infrastructure independence
VM and persistent disk management
OS independence
Blueprint vs. construction
Bosh Release vs. Bosh deployment
One Bosh release, several service plans. PostgreSQL single vm vs. PostgreSQL cluster
Monitoring & Self-Healing
my-3node-postgres-cluster-1

Postgresql VM#1

Postgresql VM#2

Postgresql VM#3
my-3node-postgres-cluster-1

Postgresql VM#1

Postgresql VM#2

Postgresql VM#3
my-3node-postgres-cluster-1

Postgresql VM#1

Postgresql VM#3
Scalability
Scalability

$> cf service-update my-single-postgres-1 -p cluster-large
Single node postgresql service instance > turned into a 3-node postgresql cluster.
Scalability

```bash
$> cf service-update \nmy-3node-postgres-cluster-1 \n-p cluster-large
```
Small cluster turned into a large cluster.

my-single-postgres-1

Postgresql VM#1
Postgresql VM#2
Postgresql VM#3

my-3node-postgres-cluster-1

Postgresql VM#1
Postgresql VM#2
Postgresql VM#3
Architectural Overview
a9s Deployer

Templates

Deployments

Bosh

a9s Service Broker

my-3node-postgres-cluster

Postgresql VM#1

Postgresql VM#2

Postgresql VM#3

Cloud Foundry Adapter

Middleware Adapter

Service Instance

my-single-postgres-1

Postgresql VM#1

Service Instance

my-3node-postgres-cluster-2

Postgresql VM#1

Postgresql VM#2

Postgresql VM#3

Service Instance

my-3node-postgres-cluster-3

Postgresql VM#1

Postgresql VM#2

Postgresql VM#3

CF Client

create service

create binding

execute deployments

Cloud Controller

create service

Cloud Foundry Adapter

create deployment from template xy with attributes {...}

a9s Service Broker

deploy release abc & deployment manifest xyz

a9s Deployer

Templates

Deploys

Service Instance

my-3node-postgres-cluster-2

Service Instance

my-3node-postgres-cluster-3

…
a9s
Service Broker
a9s Service Broker

- Implements the CF Service Broker API
- Generic
- Data service specific SPIs
a9s
PostgreSQL SPI
The PostgreSQL SPI encapsulates PostgreSQL specific logic.
a9s PostgreSQL SPI

- Service catalog metadata, e.g. service plans
- PostgreSQL instance credential management (service bindings)
a9s Deployer
The a9s Deployer manages Bosh deployments
Component Interaction
CREATE SERVICE

Cloud Controller ➔ A9S Service Broker ➔ SPI ➔ A9S Deployer ➔ BOSH Director

create service
(Service Type, Plan, User Params)

prepare deployment
(Plan, User Params)

Template Placeholders (service specific)

deploy
(template_name, placeholders)

create deployment task

g et deployment status

get deployment status

{status: deploying, vm_identifiers: [], (progress)

get task status

{status: deploying, vm_identifiers: [], running}

get task status

{status: deploying, vm_identifiers: [node-id-1, node-id-2, ...], running}

get task status

{status: deployed, vm_identifiers: [node-id-1, node-id-2, ...], finished}

finish deployment(vm_identifiers)

Hostnames

done
Lessons Learned
• Dedicated service instances are mandatory
• On-demand provisioning is essential
• Bosh does a great job
The greatest challenge with PostgreSQL was
• Finding a PostgreSQL replication & clustering toolset that allows automation
• Iteratively learn how to configure and automate common and edge-case scenarios.
Questions?

@fischerjulian
@anynines
Thank you.

@fischerjulian
@anynines