

High Availability

Why is High Availability Important?

- Cloud Foundry helps an enterprise prevent application outages and downtime
- The cost of outages can be enormous, and many times is not even quantifiable

Topics

- **Platform Upgrades**
- Component High Availability
- Component Recovery
 - VMs
 - Processes
- Application High Availability
- Application Recovery

Platform Upgrades

- BOSH executes **Canary-style deployments** to minimize downtime
 - Also known as rolling upgrades
- A select number of “canary” VMs from the new release are deployed



Platform Upgrades

- BOSH verifies new VMs have been deployed successfully
- Remaining VMs are deployed only if the upgrade of canary VMs succeeded
 - Otherwise the upgrade is halted



Platform Upgrades

- Success: all VMs upgraded to V1.1
- *No interruption of service*



*Note that Cloud Foundry in general does not patch VMs or app instances, it **recreates** or **replaces** them*

Example: OS Security Upgrade

- Because all VMs use the same OS, Cloud Foundry can use rolling upgrades to replace every OS instance automatically
- This is one of the benefits of having an “opinionated” platform- you are in control of the infrastructure
 - This would be much more challenging on a haphazard collection of machines



Canary-Style Deployment: Implications

- Elastic Runtime can be upgraded to new versions with no application downtime
 - “Apply changes” does not mean “wait for it to finish”
 - You will have limited ability to deploy applications during Elastic Runtime upgrades
- Installing or modifying other tiles does not affect the Elastic Runtime
- For the ops team- upgrade the system during working hours!

Topics

- Platform Upgrades
- **Component High Availability and Recovery**
 - VMs
 - Processes
- Application High Availability
- Application Recovery

Component High Availability

- Some Cloud Foundry components can be scaled to multiple instances for high availability
- Increase number of instances on Resources tab of Elastic Runtime
- Some components are single instance only

Example Scalable Components
HAProxy
NATS
Cloud Controller
Cloud Controller Worker
Router
UAA
Login
Cell/DEA

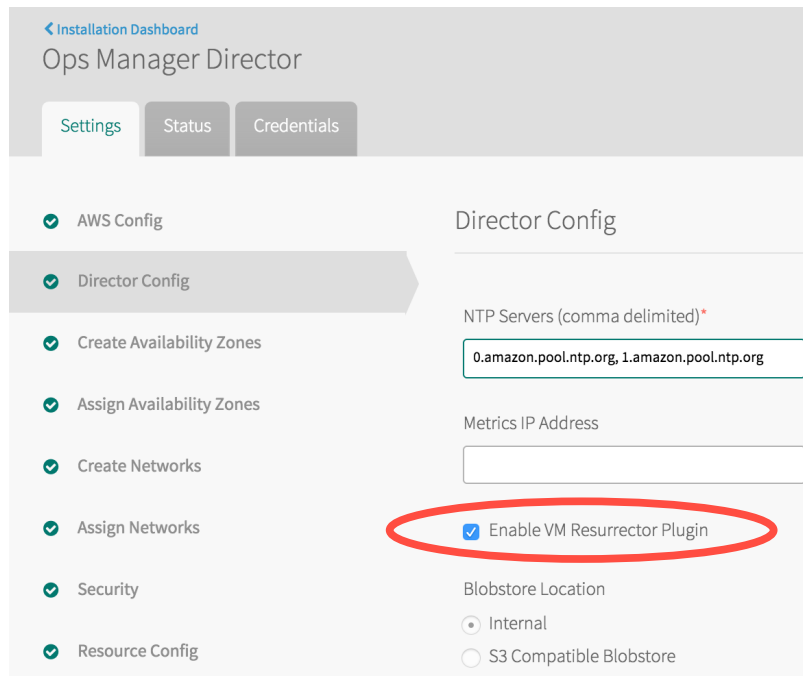
<http://docs.pivotal.io/pivotalcf/opsguide/single-component.html>

Component Recovery – VMs

- Each component of Cloud Foundry is a VM deployed and managed by BOSH
- BOSH health monitor continuously compares current component state to desired state, including:
 - number of instances
 - running processes
- If current state differs from desired state, the health monitor:
 - triggers the VM resurrector, or
 - sends alerts

VM Resurrector Plugin

- The VM resurrector plugin is a health monitor plugin that recreates VMs when a VM's BOSH agent stops sending heartbeats
- It calls the API of the infrastructure to create a VM, then copies necessary files from the BOSH blobstore



The screenshot shows the 'Installation Dashboard' for 'Ops Manager Director'. The 'Settings' tab is active, displaying a list of configuration items on the left and a 'Director Config' section on the right. The 'Director Config' section includes fields for 'NTP Servers (comma delimited)*' (filled with '0.amazon.pool.ntp.org, 1.amazon.pool.ntp.org'), 'Metrics IP Address' (empty), and 'Blobstore Location' (radio buttons for 'Internal' and 'S3 Compatible Blobstore'). The 'Enable VM Resurrector Plugin' checkbox is checked and highlighted with a red oval.

Installation Dashboard
Ops Manager Director

Settings Status Credentials

✓ AWS Config

✓ Director Config

✓ Create Availability Zones

✓ Assign Availability Zones

✓ Create Networks

✓ Assign Networks

✓ Security

✓ Resource Config

Director Config

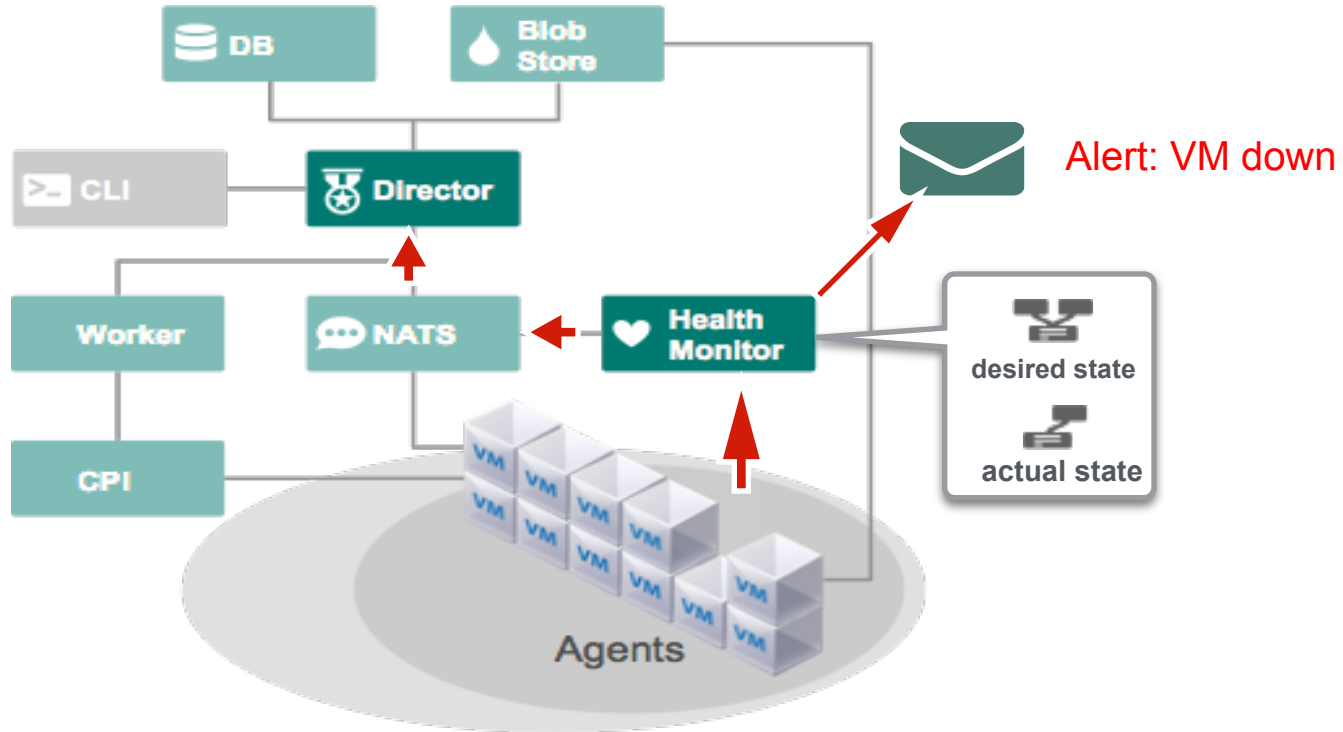
NTP Servers (comma delimited)*
0.amazon.pool.ntp.org, 1.amazon.pool.ntp.org

Metrics IP Address

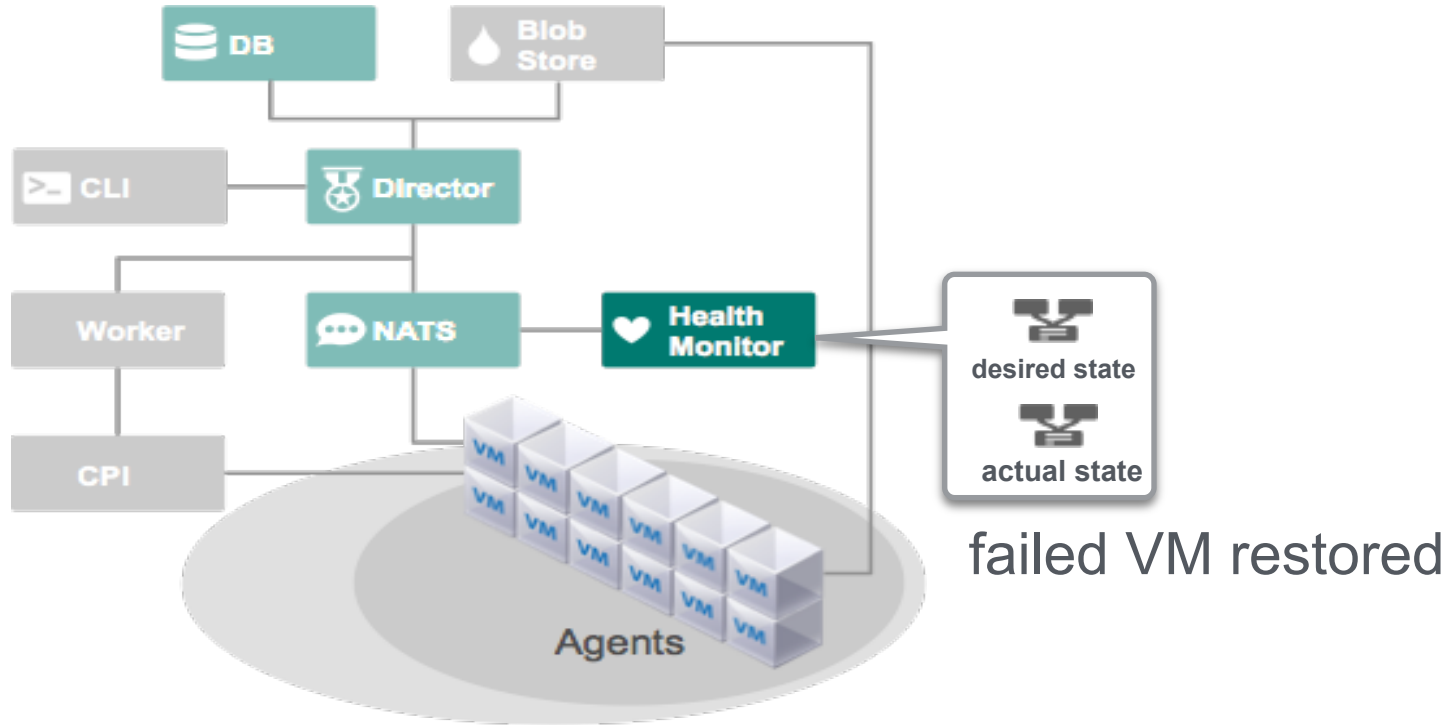
☒ Enable VM Resurrector Plugin

Blobstore Location
☐ Internal
☐ S3 Compatible Blobstore

Scenario 1: Component VM Failure



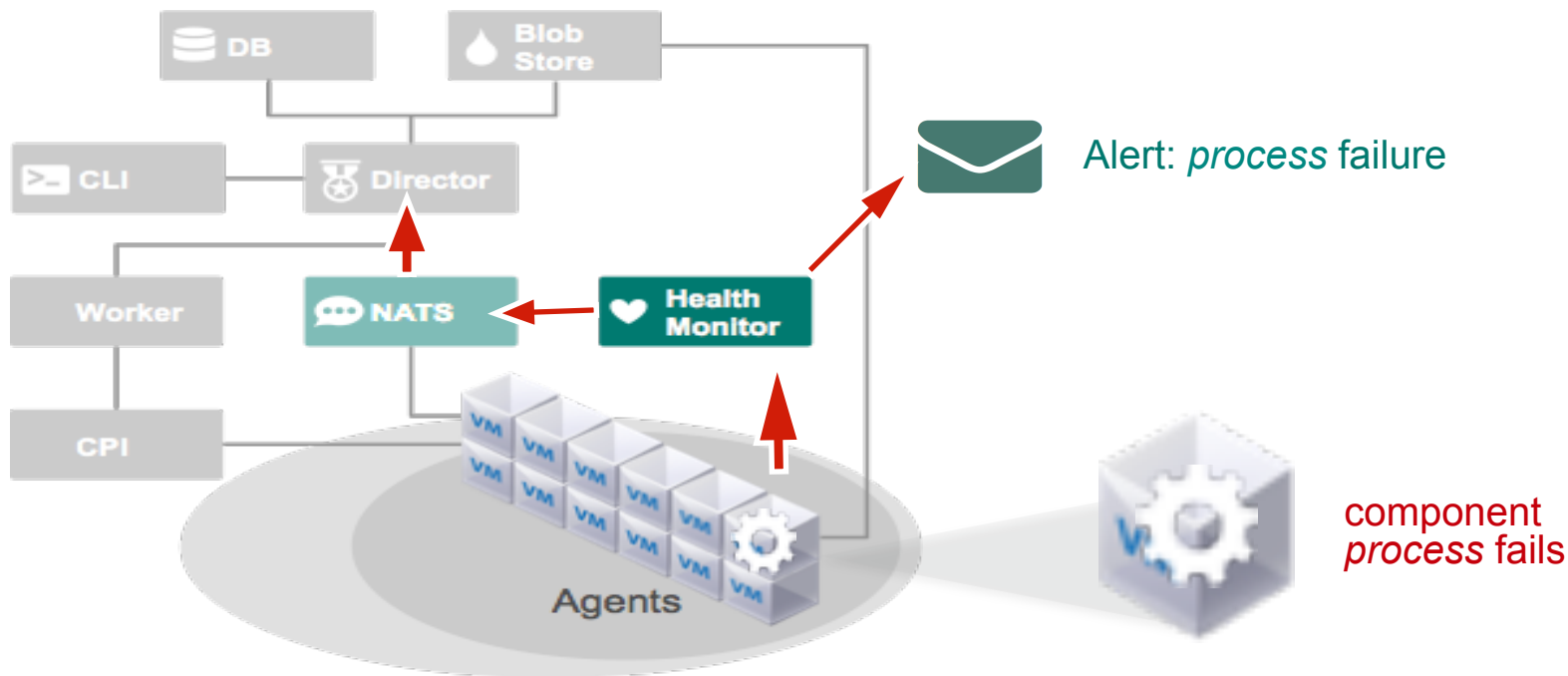
Scenario 1: Component VM Failure



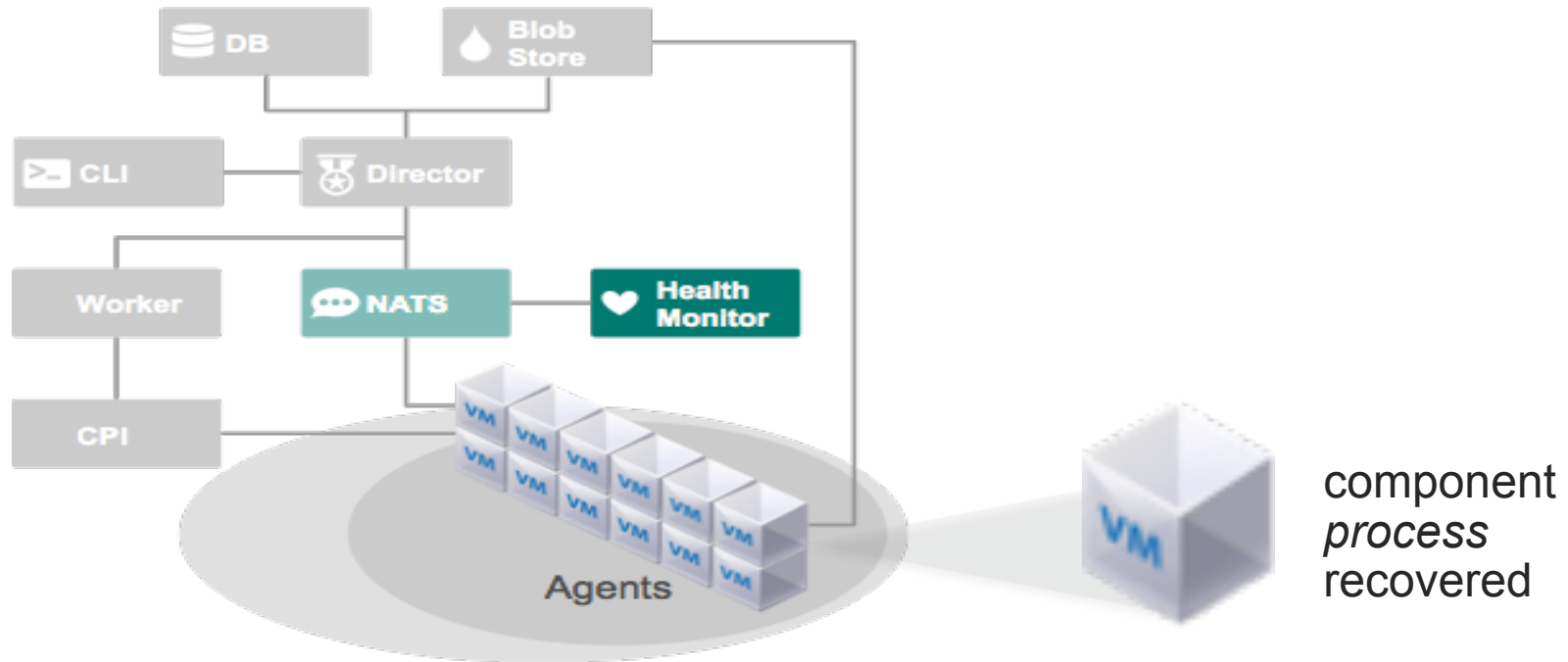
Component Recovery - Processes

- Monit utility running on the component VMs automatically restarts failed processes
- BOSH release defines which processes to monitor

Scenario 2: Component Process Failure



Scenario 2: Component Process Failure

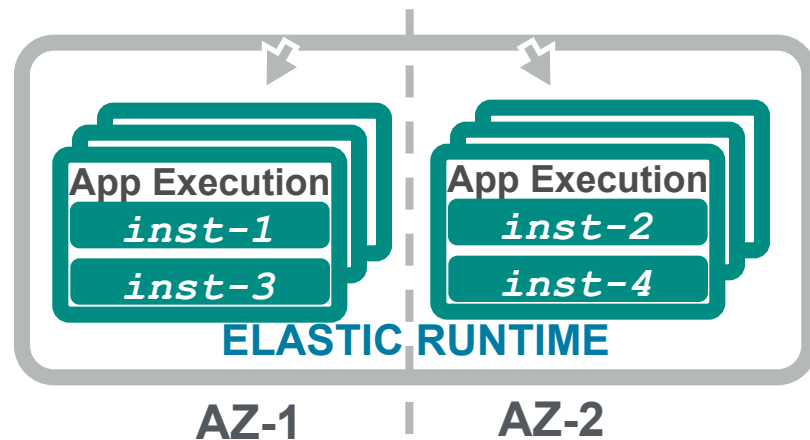


Topics

- Platform Upgrades
- Component High Availability
- Component Recovery
 - VMs
 - Processes
- **Application High Availability**
- Application Recovery

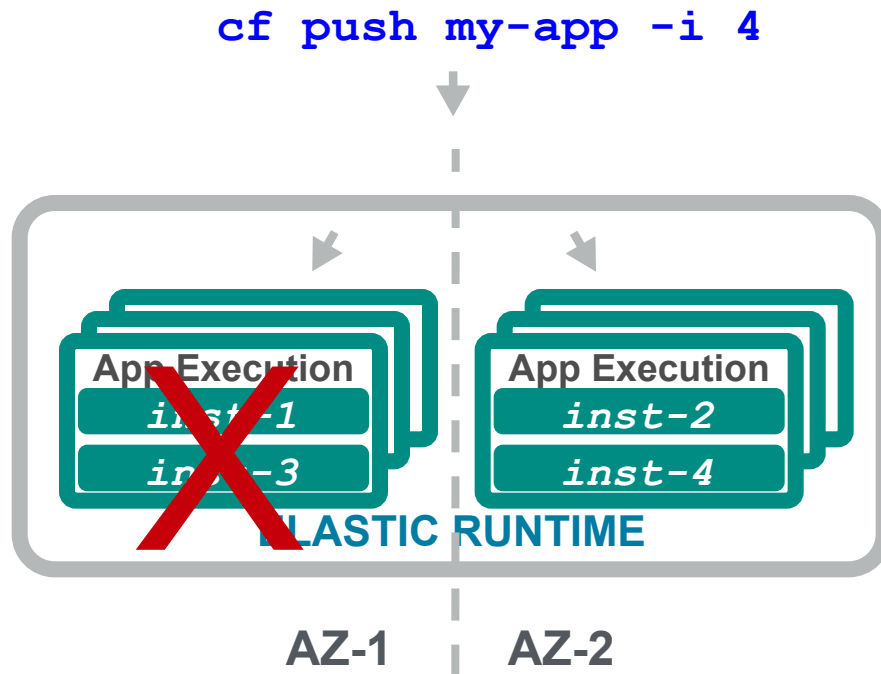
Application High Availability

- Cloud Foundry will scale application instances across *availability zones*
- Availability zones correspond to independent infrastructure segments
 - Different racks, or even different data centers
 - Provide physical isolation, redundancy
 - Feature of the IaaS layer
 - Be aware of *latency* between zones



Scenario: Availability Zone Failure

- If one zone fails, the application instances in the other zone pick up the load
- *No outage*

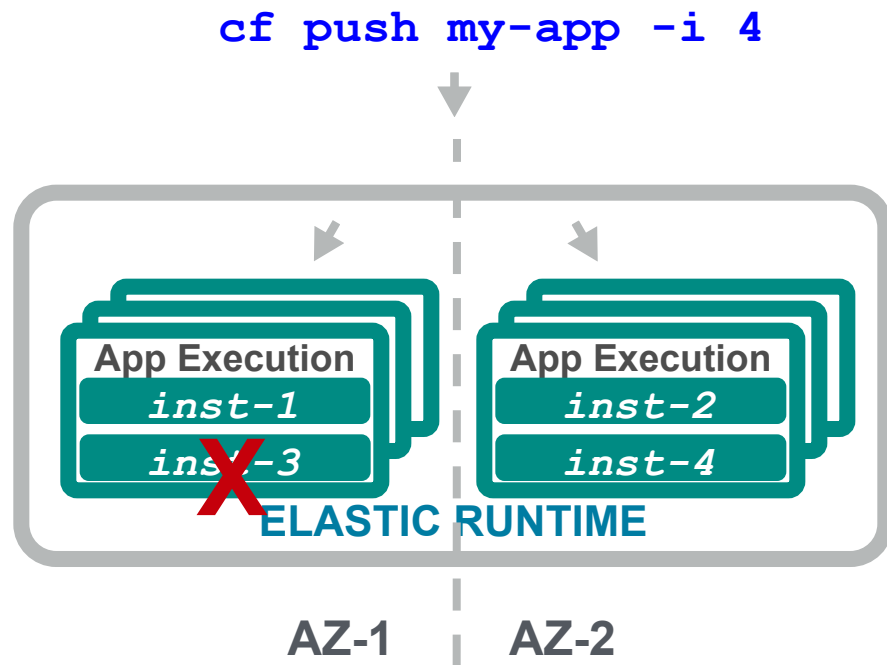


Topics

- Platform Upgrades
- Component High Availability
- Component Recovery
 - VMs
 - Processes
- Application High Availability
- **Application Recovery**

Application Recovery

- The Elastic Runtime monitors application instance processes, and detects failures
- On failed instances, the container is destroyed and a new container is created
 - Droplet is copied from the Cloud Controller blobstore to the container and started



Summary

- Canary style deployments minimize impact of upgrades
- Components should be scaled when possible
- There are four levels of high availability:
 - Enable the VM Resurrector to recover failed VMs
 - Component processes will be automatically restarted
 - Availability zones protect against infrastructure failures
 - Application instances will be automatically recreated and restarted

Note: The PWS Ops team uses a 2 minute delay before being notified of system problems- this is because the system usually repairs itself within that time

Lab

Explore some of the high availability behavior of Pivotal Cloud Foundry