

# Why we Picked CF as the Basis for our Public Cloud Multi-Tenant Platform

Mike Root @mikersj

Jeroen van Rotterdam @jvanrotterdam

# THE ANNOYING

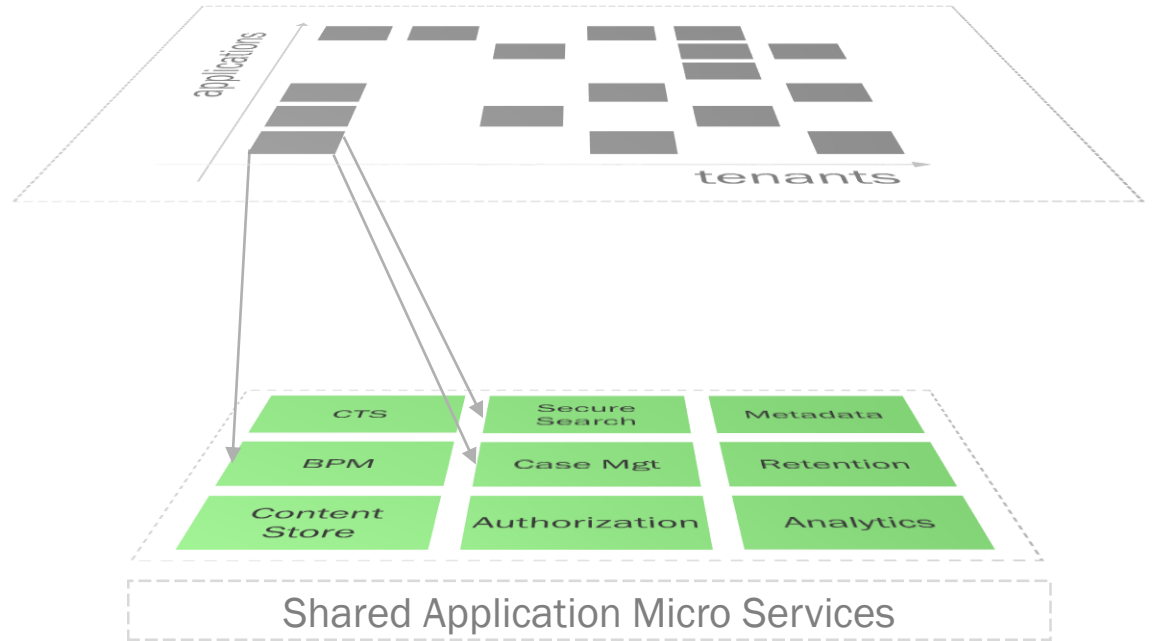
Our Stuff is Complex



WE ONLY WANT TO BUILD APPLICATIONS

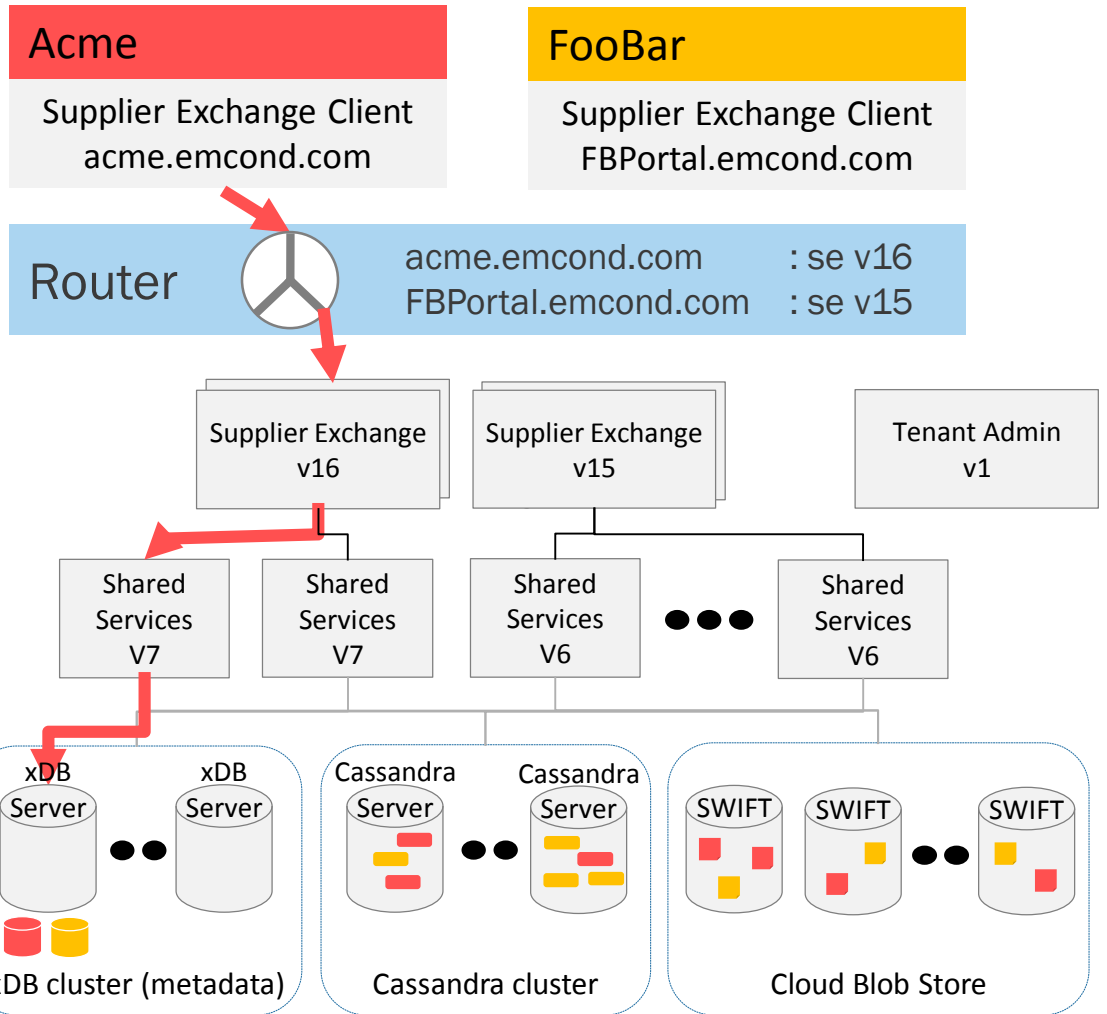
# THE ANNOYING

Our Stuff is Complex

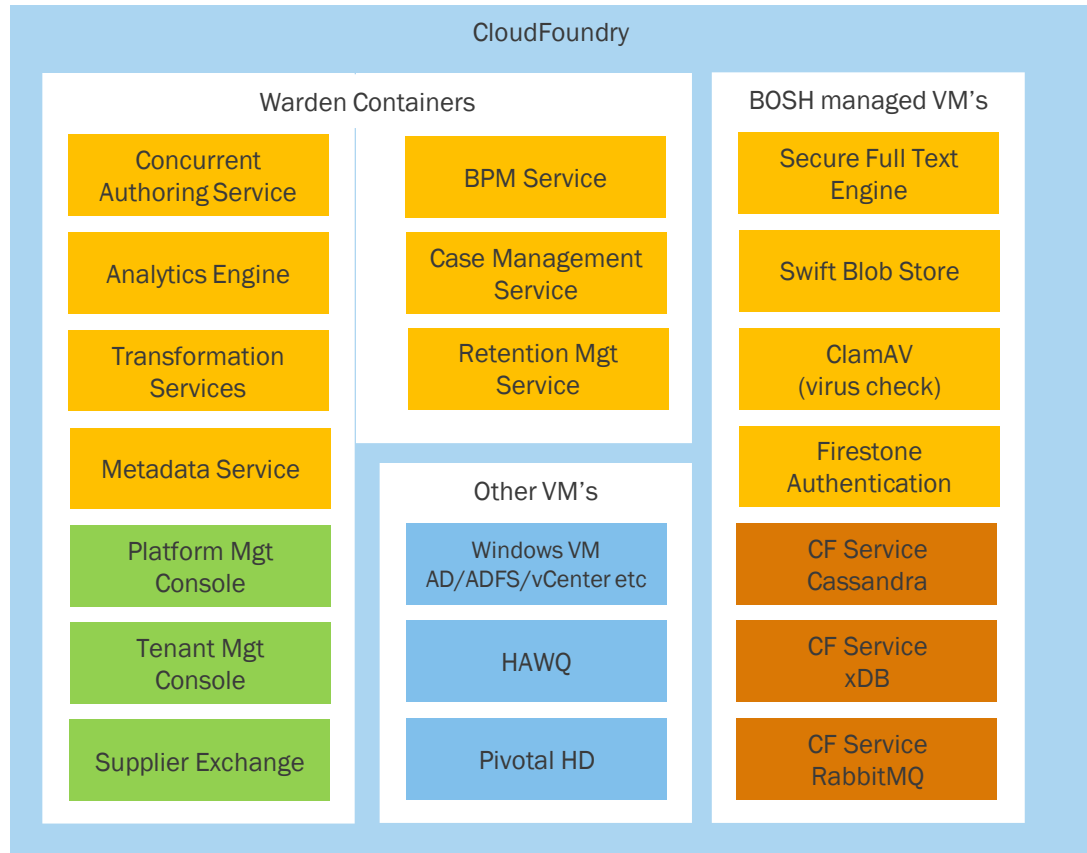


WE ONLY WANT TO BUILD APPLICATIONS

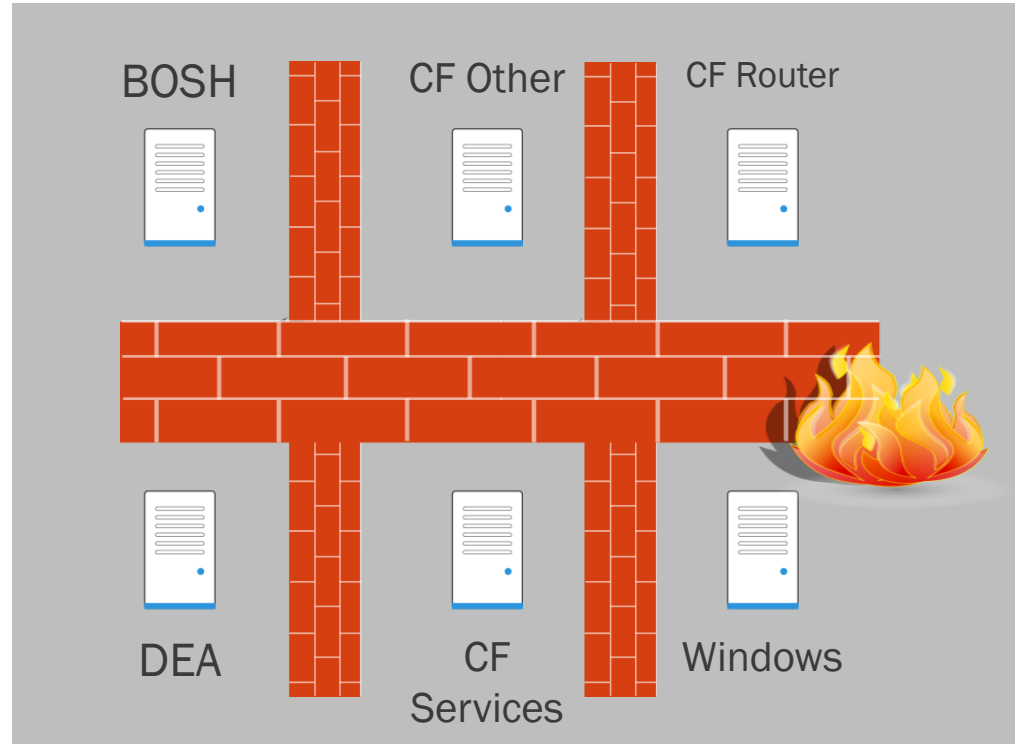
# TENANT PARTITIONING



# PRODUCTION SETUP



# NETWORK SEGREGATION



# CLOUDFOUNDRY BENEFITS

BOSH provisions the entire datacenter

- Repeatable at any time
- No downtime

Cloud Foundry

- Handles application scalability
- Tenant (Customer) modifiable URL

Upgrade tool

- Uses the CF API
- Blue Green upgrade (0 downtime)
- Dynamically move/configure tenants

Promotion process is fully automated

## Example

Shellshock, OS Security vulnerability

- 1 We updated the stem-cell (OS)
- 2 Ran “bosh deploy”
  - 16 CF environments updated (dev/test/pre-prod/prod)
  - 16x30 VM’s updated
  - No down time
- 3 Drank beer (optional)

# CLOUDFOUNDRY BENEFITS

WE DIDN'T BUILD  
ANY OF THIS

## Monitoring

- Monitoring VMs
- Monitoring applications

## Resource scaling

## High Availability

- VM HA
- Application HA

## Log collection

## Health metrics

WE ONLY WANT TO BUILD APPLICATIONS



# SPIFF GENERATION OF BOSH MANIFESTS

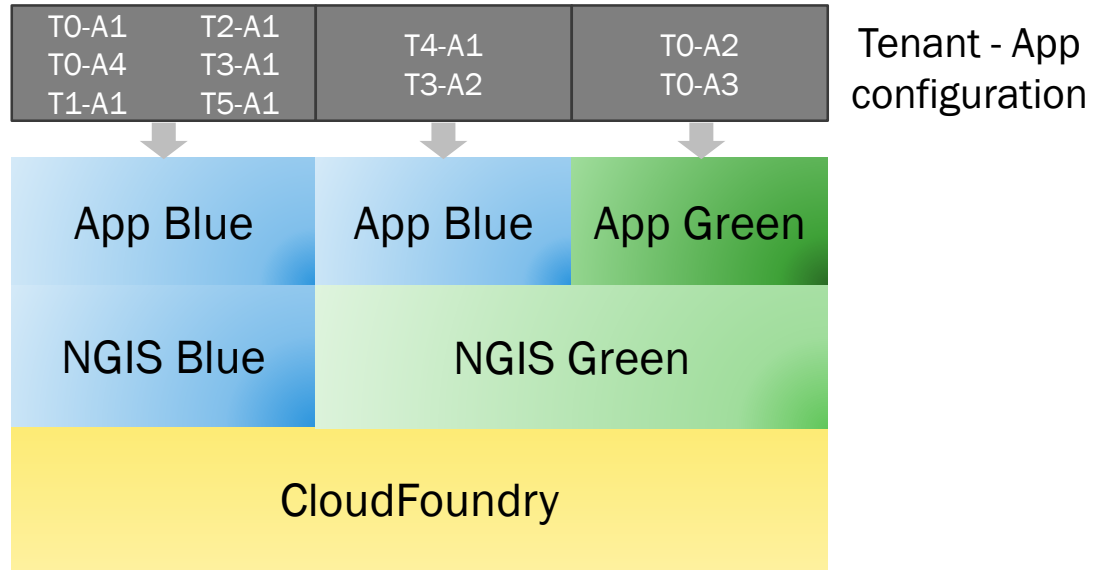
- 16 Deployments managed with auto generated manifests
- Core templates
- 16 Instance specific deployments

Support for multiple networks  
→ Contribute to the community

```
$> spiff merge cf-jobs.yml cf-network.yml vcenter.yml prod.yml  
ci_merged.yml  
dev_merged.yml  
preprod_merged.yml  
prod_merged.yml
```

# CLOUDFOUNDRY UPGRADE TOOL

## BLUE GREEN DEPLOYMENT



- CloudFoundry API to deploy applications
- SaaS REST to move/configure tenants  
→ e.g. new security rules, enable features etc
- Old and new versions running in parallel

# GAPS WE CONQUERED



- OS hardening; Ubuntu patch version, ssh permissions, file access etc etc
- Deploying in multiple networks with firewalls
  - CF/BOSH/DEA/Router/Services
  - Spiff not setup for multiple networks
- Keeping up to date with latest CF release
- Converting v1 services to v2 services
- DR setup
- CF cli is constantly changing, but CF api is more stable
  - We use the API for the upgrade tool
- Debugging applications in CloudFoundry (step through code)
- Filesystem options/type for persistent storage (swift)
- Filesystem size of /tmp is not configurable
- BOSH builds are more difficult because BOSH doesn't support artifactory as a BLOB store
- Managing multiple BOSH deployments
- Log files other than stderr/stdout
  - Collecting them
  - Limited disk space
- SPIFF manifests are harder to read by a human
  - Spiff diff helps
- Application needs to report ready before ready for large application
- Router networking resources exhausted
  - keep-alive disabled
- CF staging blobs not cleaned up properly
- CF routes not cleaned up properly

WE ONLY WANT TO BUILD APPLICATIONS

# CONTRIBUTIONS TO CLOUDFOUNDRY



## Bosh releases

- ClamAV → First EMC Contribution to CF.org
- SWIFT
- HA Proxy
- Deployment VM
  - BOSH CLI
  - CF CLI
  - User management ability
- Cassandra as a CF Service
- Zabbix Agent (open source monitoring tool)
- Network yml

## Service Brokers

- Cassandra
- xDB

WE ONLY WANT TO BUILD APPLICATIONS

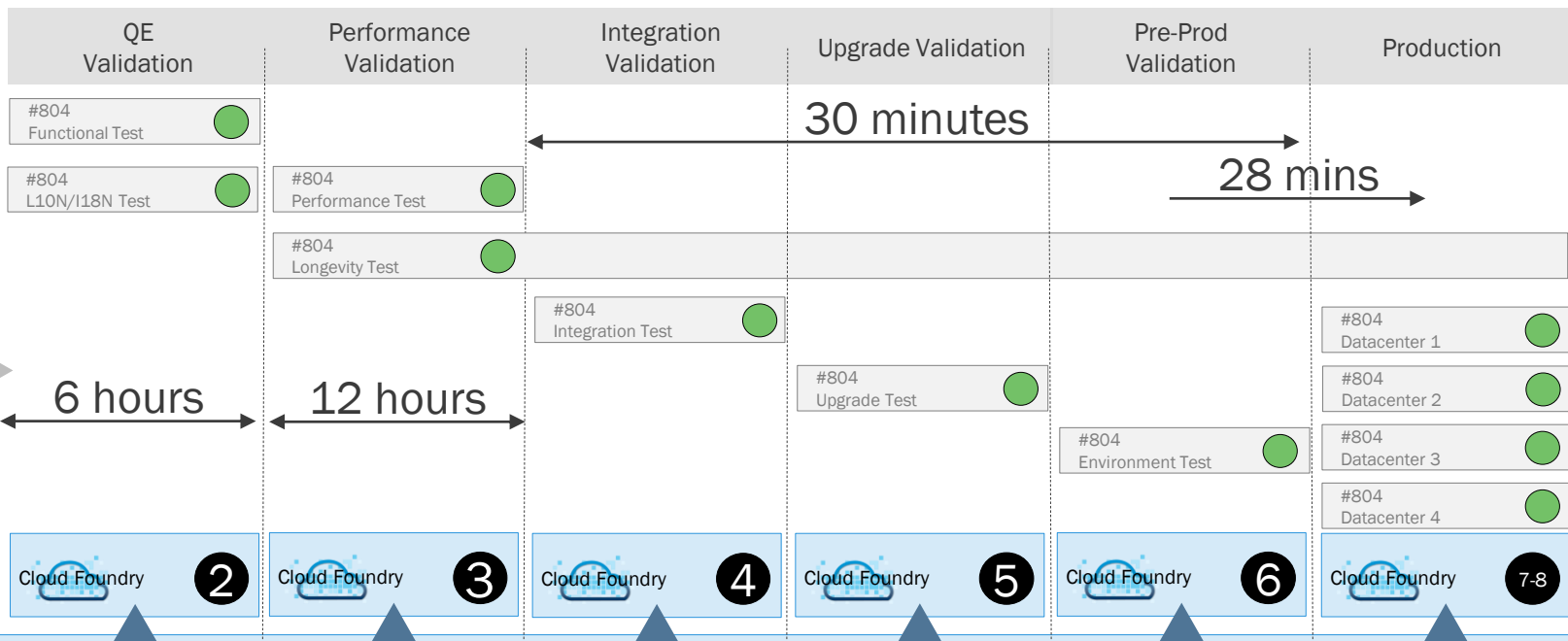
# PROMOTION PROCESS

CI

DEV  
Validation

- Build #806 ●
- Build #805 ●
- Build #804 ●
- Build #803 ●
- Build #802 ●

## SaaS Continuous Deployment Process



Cloud Foundry **1**

Cloud Foundry **2**

Cloud Foundry **3**

Cloud Foundry **4**

Cloud Foundry **5**

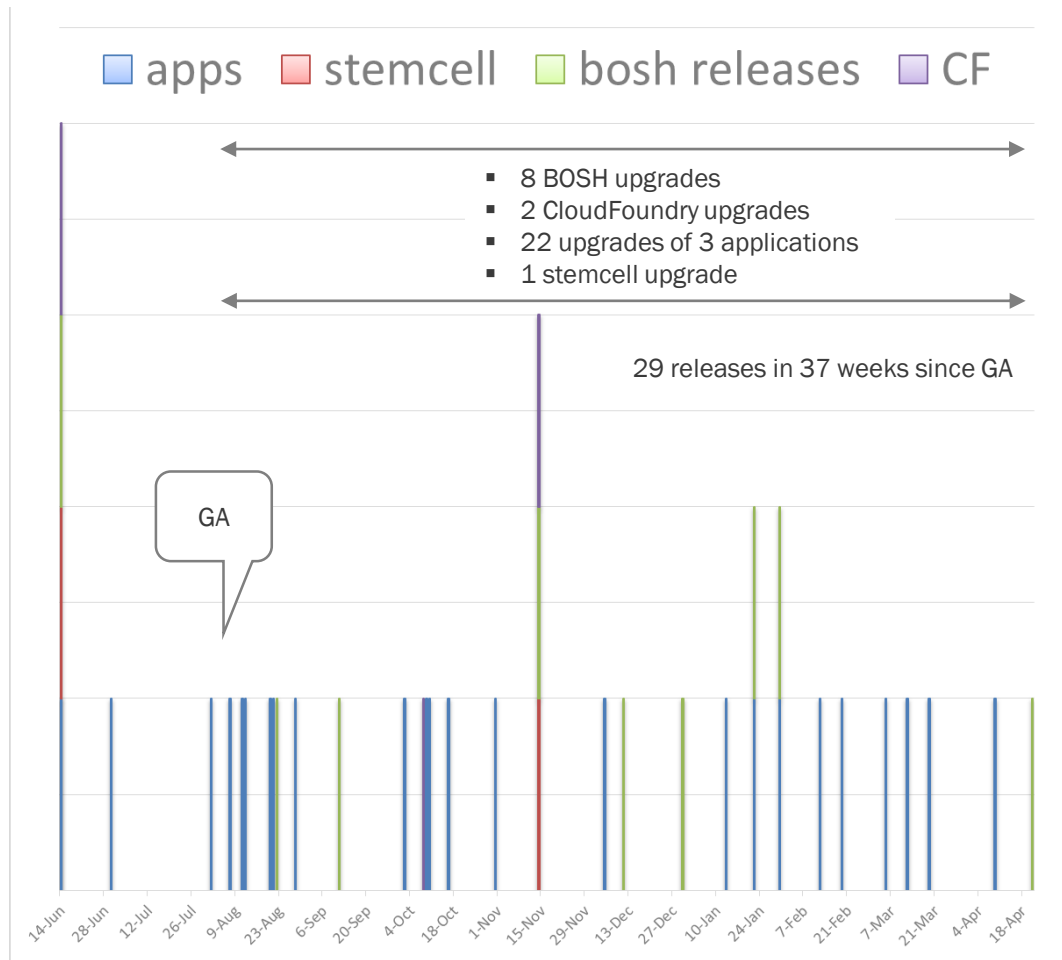
Cloud Foundry **6**

Cloud Foundry **7-8**

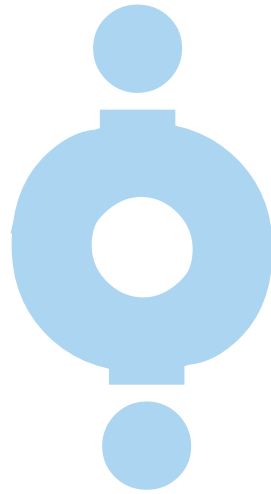
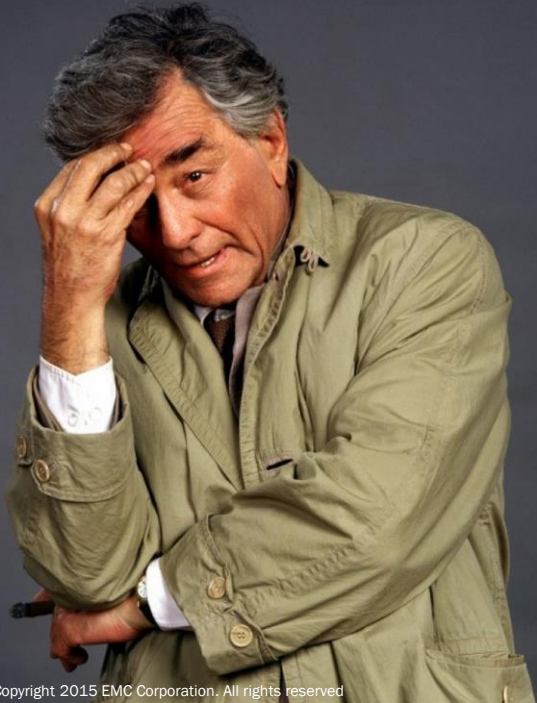


Upgrade Tool / BOSH: Configuration Management, Orchestration

# DEPLOYMENT HISTORY



# Q & A



NOT THAT WE HAVE TIME FOR THIS

EMC<sup>2</sup>®