A Story of GitOps or How Kohl's Manages OpenShift at-scale

Heiko Zuerker / Senior Staff Architect @ Kohl's
Johnathan Kupferer / Solutions Architect @ Red Hat
05/08/2019

KOHL'S

1,100+

STORES IN 49 STATES

$20B

ANNUAL REVENUE

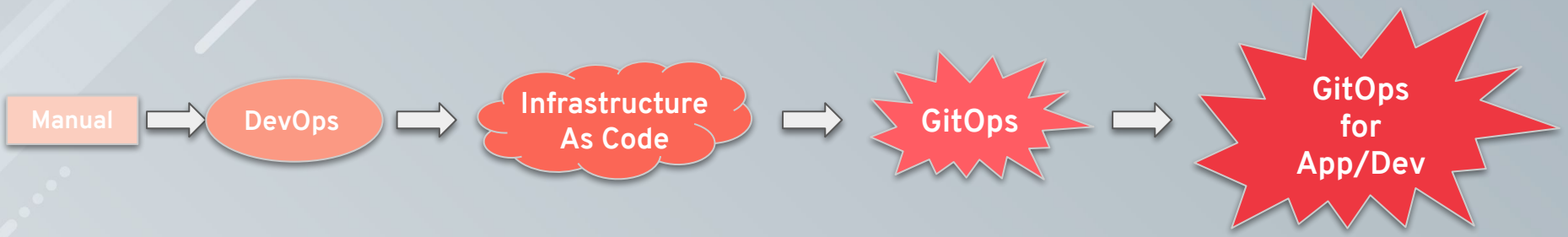
140K

EMPLOYEES

$700M

TO COMMUNITIES

OUR OPENSHIFT JOURNEY



2016	2017	2018	2019
OpenShift 3.4	**OpenShift 3.6**	**OpenShift 3.9**	**OpenShift 3.11**
Prototype	*"Sorta" stable*	*Zero issues during "peak"*	*Rock-solid, cheap, & stateful*
3 Clusters	**10 Clusters** (Planned and staffed for 4)	**19 Clusters**	**18 Clusters**
	<200 Nodes <200 Containers	1000+ Nodes 8000+ Containers	1200+ Nodes 25000+ Containers
	Dedicated L2 Support	**Dedicated L2 Support**	**Shared L2 Support**

OPENSHIFT @ KOHL'S

MAIN DESIGN FEATURES

- ❑ Shared & dedicated clusters
- ❑ Agility & Flexibility
- ❑ Organic Patching
- ❑ Drift Management
- ❑ Support snowflakes
- ❑ Managed via GitOps
- ❑ Automate Everything
- ❑ Everything-as-Code
- ❑ Idempotent Automation
- ❑ Immutable Infrastructure
- ❑ Ephemeral Nodes*
- ❑ Platform Autoscaling

Applying container principles and best practices to the platform itself!

* Everything except master and gluster nodes (for now)

INNER SOURCE

TAKING OPEN SOURCE PRINCIPLES AND APPLYING THEM INTERNALLY!

Everything that makes Open Source great, for example...

- ❑ Open and inviting
- ❑ All repos can be viewed
- ❑ External changes are encouraged
- ❑ Collaboration across teams
- ❑ “Outsourcing” of work
- ❑ Shared ownership
- ❑ Fast turnaround
- ❑ Quality code

GitOps = Turning Day 2 Operations into Code!

Thinking about Day 2 on Day 1

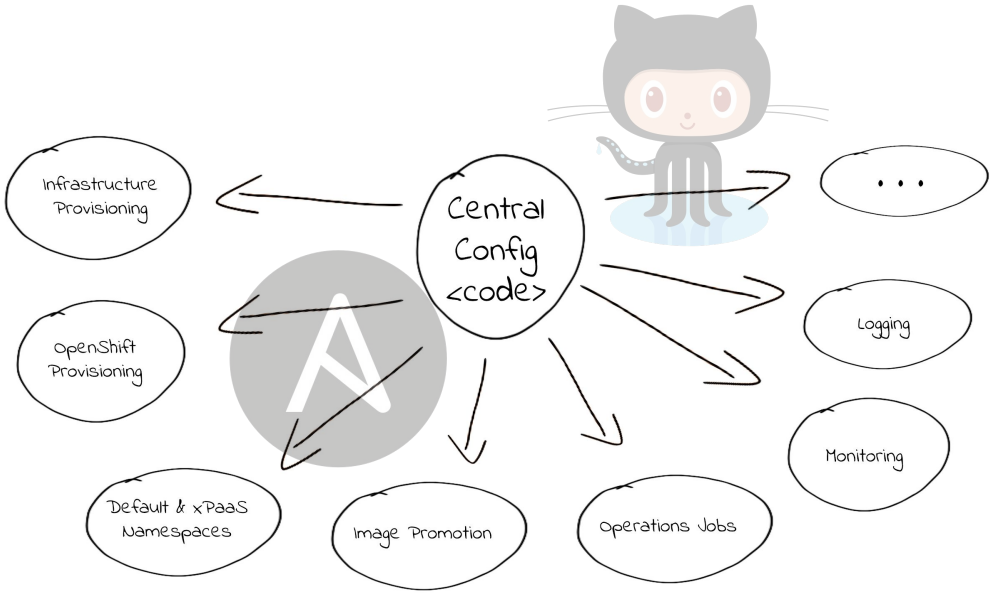
Leveraging best practices learned from DevOps and using them to manage "**everything-as-code**"

Next Generation Change Management

- ❏ Version Control
- ❏ Peer Reviews
- ❏ Audit Trail
- ❏ Reproducibility, Consistency, & Reliability

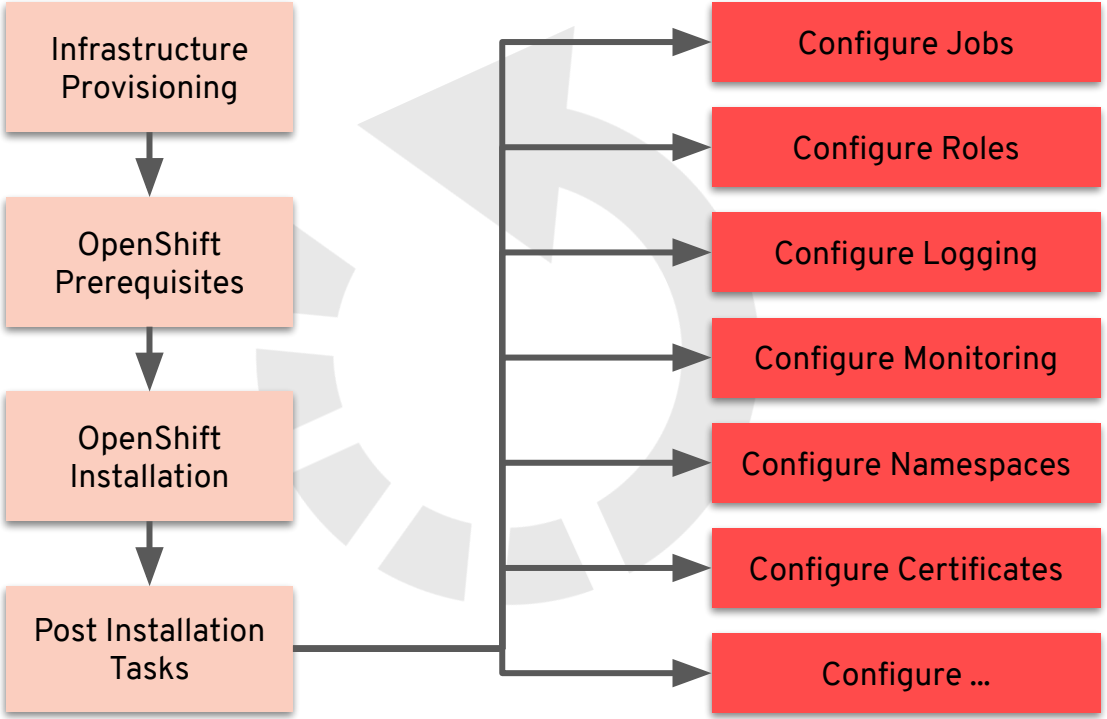
Efficiency

- ❏ Disposable Infrastructure
- ❏ Scalability of Team
- ❏ "Outsource" via Inner Source



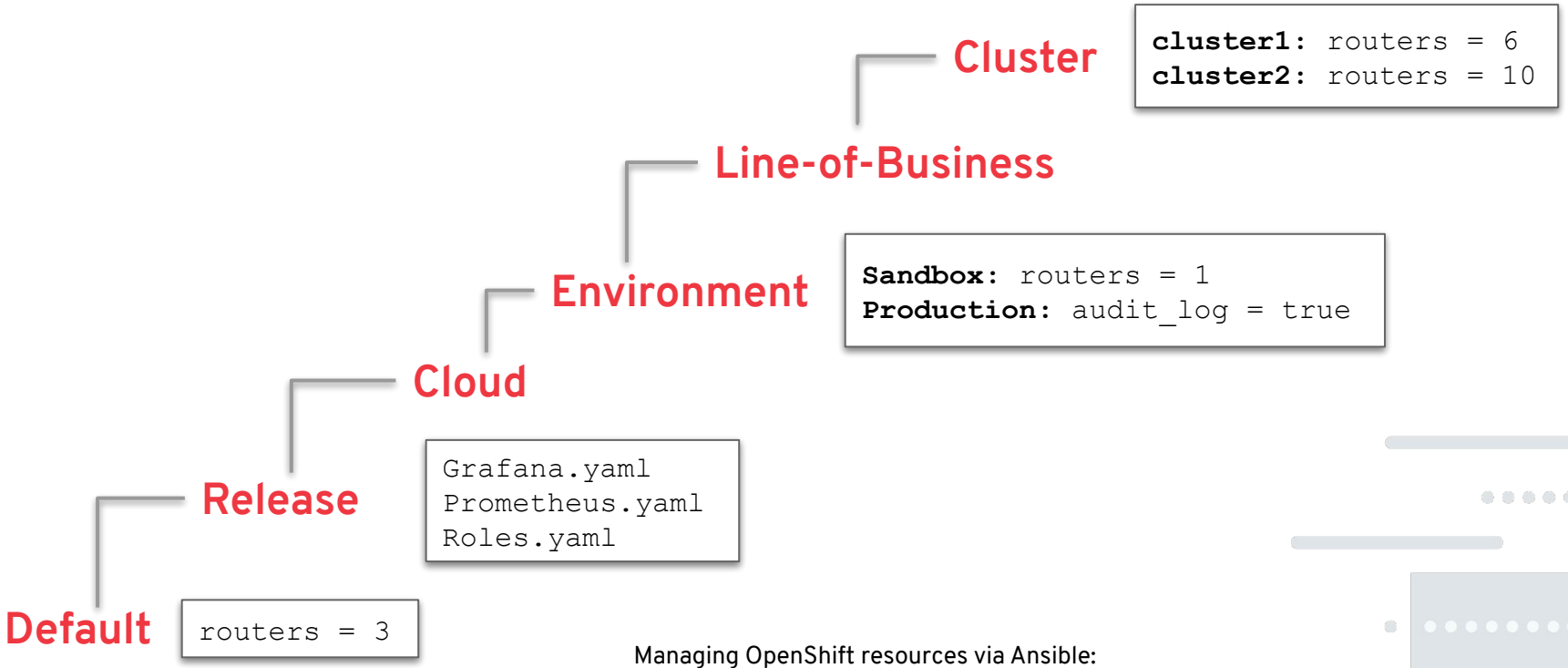
Infrastructure Pipeline

SBX → DEV → QA → PROD



EXAMPLE: CONFIGURATION

(SIMPLIFIED)



Managing OpenShift resources via Ansible:
https://github.com/gnuthought/ansible-role-openshift-provision

PLATFORM AUTO SCALING

SCALING THE UNDERLYING NODES BASED ON THE CURRENT RESOURCE DEMANDS (CPU & MEMORY REQUESTS)

- ❏ Implemented in GCP for OpenShift 3.6 leveraging kube-autoscaler
- ❏ ~3 minutes for new nodes to become available
- ❏ Maximum node lifetime 7 days
- ❏ No guarantees for how long pods will live

} Chaos-Engineering Lite

3.6 & 3.9	3.11
custom cluster-join logic	TLS bootstrapping
custom scale-down logic	kube-autoscaler
custom capacity logic	pre-scaling with preemptible pods
scale-down oldest node	kube-autoscaler

ORGANIC PATCHING

RELY ON PLATFORM AUTOSCALING TO SLOWLY REPLACE NODES, AS THE CLUSTER FLEXES UP AND DOWN

- ❏ Cordon “outdated” nodes to force new pods onto new nodes
- ❏ “Nudge” the cluster every couple of hours (scale’n’drain)
- ❏ Used for configuration changes
- ❏ Used for platform patching

“THE LAW” / “TOUGH LOVE”

- ❏ Containers are not VMs and will not be treated as such
- ❏ Pods must be immutable
- ❏ Pods must have health checks and liveliness probes
- ❏ Pod lifetime not guaranteed & we won’t warn you about shutdowns
- ❏ No manual config beyond development environments
- ❏ No privileged containers
- ❏ You’re responsible for your own HA & DR
- ❏ You must manage your own state and backup your own data

EMBRACING FAILURE

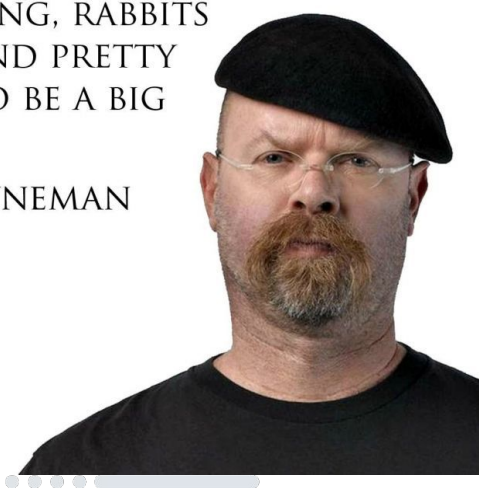
"Failure is always an option."

~ Adam Savage

"It's a beautiful day at the ~~bomb~~ ~~range~~. Birds are singing, rabbits are hopping about...and pretty soon there's going to be a big explosion."

~ Jamie Hyneman

datacenter / cloud



IT'S EARLY DECEMBER...

WE ARE IN PEAK SEASON...

and here comes... CRITICAL CVE-2018-1002105

"Kubernetes privilege escalation and access to sensitive information in OpenShift products and services."

Patched 19 OpenShift clusters in 2 days
400+ application nodes (VMs) replaced
1000+ business application pods moved
0 (zero) business impact!

GitOps for App/Dev

- ❏ Stop funneling requests through a central team
- ❏ We don't care (much) what teams do in "their namespaces"
  - ☆ Chargeback
  - ☆ Cluster stability & security
- ❏ Guardrails via policy automation
- ❏ Remove dependencies to external teams
  - ☆ Automate DNS entries for custom routes
- ❏ Ephemeral namespaces

Kohl's k8s GitOps Operator soon to be released under: https://github.com/KohlsTechnology

2019 & BEYOND

Decentralizing → “OpenShift-in-a-box” / self-managing clusters

Nightly cluster build, validation, & full GitOps test suite

Consistent automation for all k8s platforms

Open Sourcing as much as we can

Less custom engineering

OpenShift 4.x

RED HAT SUMMIT

THANK YOU


plus.google.com/+RedHat


facebook.com/redhatinc


linkedin.com/company/red-hat


twitter.com/redhat


youtube.com/user/RedHatVideos