

# Kubernetes with Terraform, Ansible and OpenShift on OpenStack and Anywhere

(QuadrupleO: OpenShift Origin On OpenStack)

2017, January 26th

Presenter: Arash Kaffamanesh  
@cloudsskyone  
[www.cloudssky.com](http://www.cloudssky.com)

OpenStack Cologne / Cloud Cologne Meetup



# Agenda

- Organizational Topics / Meetups / Events in 2017
- Introduction to Kubernetes, OpenShift Origin, Terraform and Ansible.
- OpenShift Origin On OpenStack (QuadrupleO)
- How to run multiple Kubernetes Clusters with OpenShift (Origin) anywhere
- Get the Terraform code and start open shifting with QuadrupleO
- Demo: Run Rancher and the IoT OCP App on OpenShift

# Organizational Topics & Events

- Most of our Meetups in 2017 will be held through Online Meetings
- The OpenStack DACH e.V. association annual general meeting will take place on March 2nd in Berlin by WeWork (new members are more than welcome to join)
- Openshift Commons Gathering on March 28. March in Berlin
- CloudNativeCon + KubeCon on March 29. + 30. in Berlin

# Speakers + Sponsors Welcome

- For our upcoming meetups we're seeking cloud-native speakers for great talks
- New Sponsorship Program for [OpenStack Cologne](#) and [Cologne Cloud](#) Meetup Groups available soon
- Please get in touch if you'd like to support us with Talks and Sponsorship

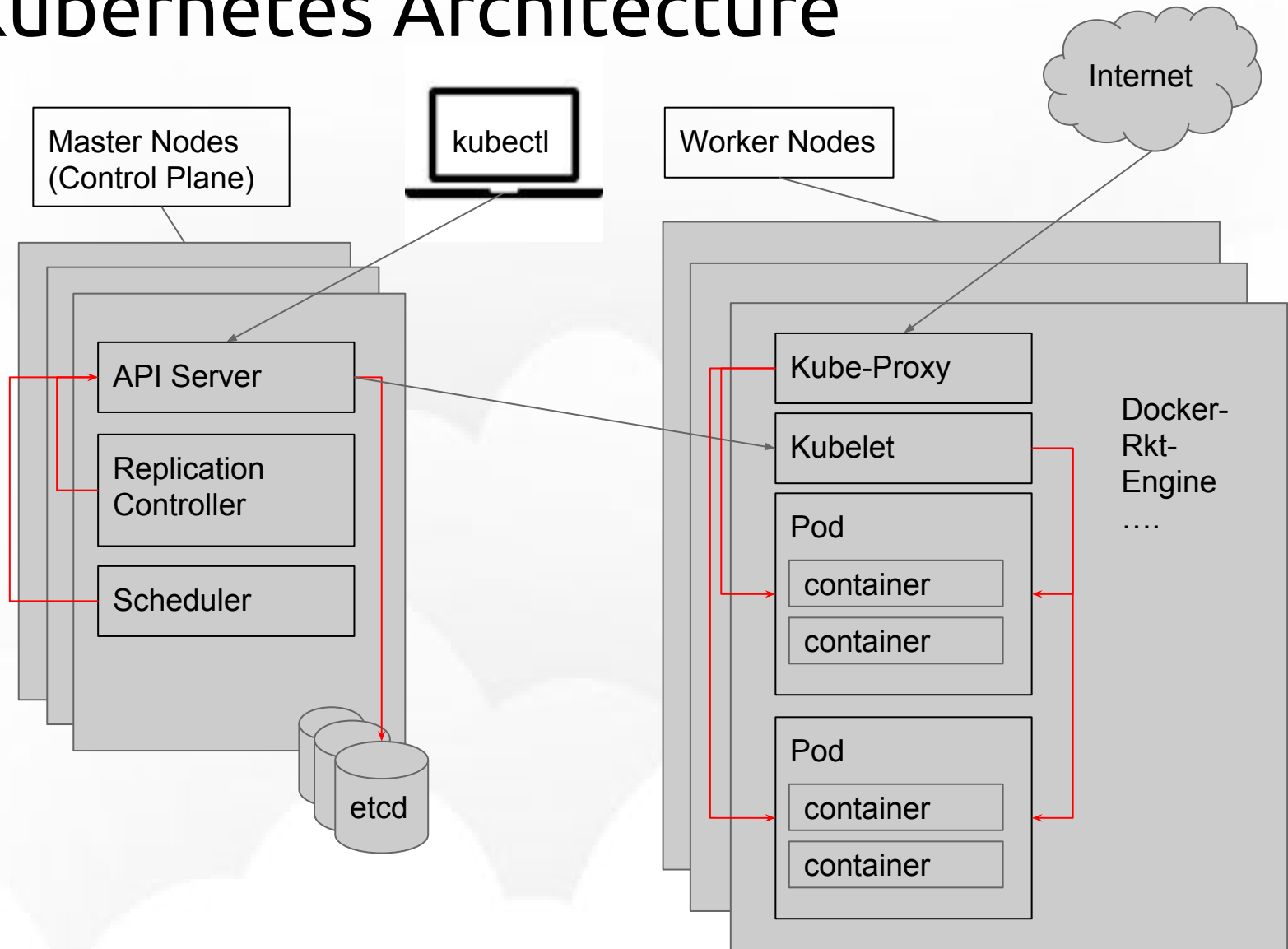
# Desired Topics for our 2017 Sessions

- Topics which might be of our interest:
  - Serverless: Kubeless / Funktion / OpenWhisk
  - Container Run Times
  - Mesos / Mesosphere
  - Docker Data Center
  - Rancher / Cattle
  - CoreOS / Tectonic
  - Cloud-Native OpenStack
  - Kubernetes-X / OpenShift-X / Cloud Foundry

# About Kubernetes (a.k.a K8S or Kube)

- Google's Container Cluster Manager under the Umbrella of CNCF (cloud native computing foundation)
- Provides a REST API to describe the desired **state** of applications, deploys group of Pods, replicates Pods, redeploys Pods if they fail, provides service discovery, auto-scaling, etc..
- Helps to build resilient, reliable, scalable and self-healing applications **ANYWHERE**
- **ANYWHERE == Bare Metal, OpenStack, CloudStack, AWS, GCP / GCE, Azure, ...**

# Kubernetes Architecture



# Kubernetes / OpenShift vocabulary

- Pods
- Replication Controller
- Services
- Secrets
- Deployment Config
- ConfigMap
- Labels
- Annotations
- Router (HAProxy)
- Registry (Docker Image Registry)





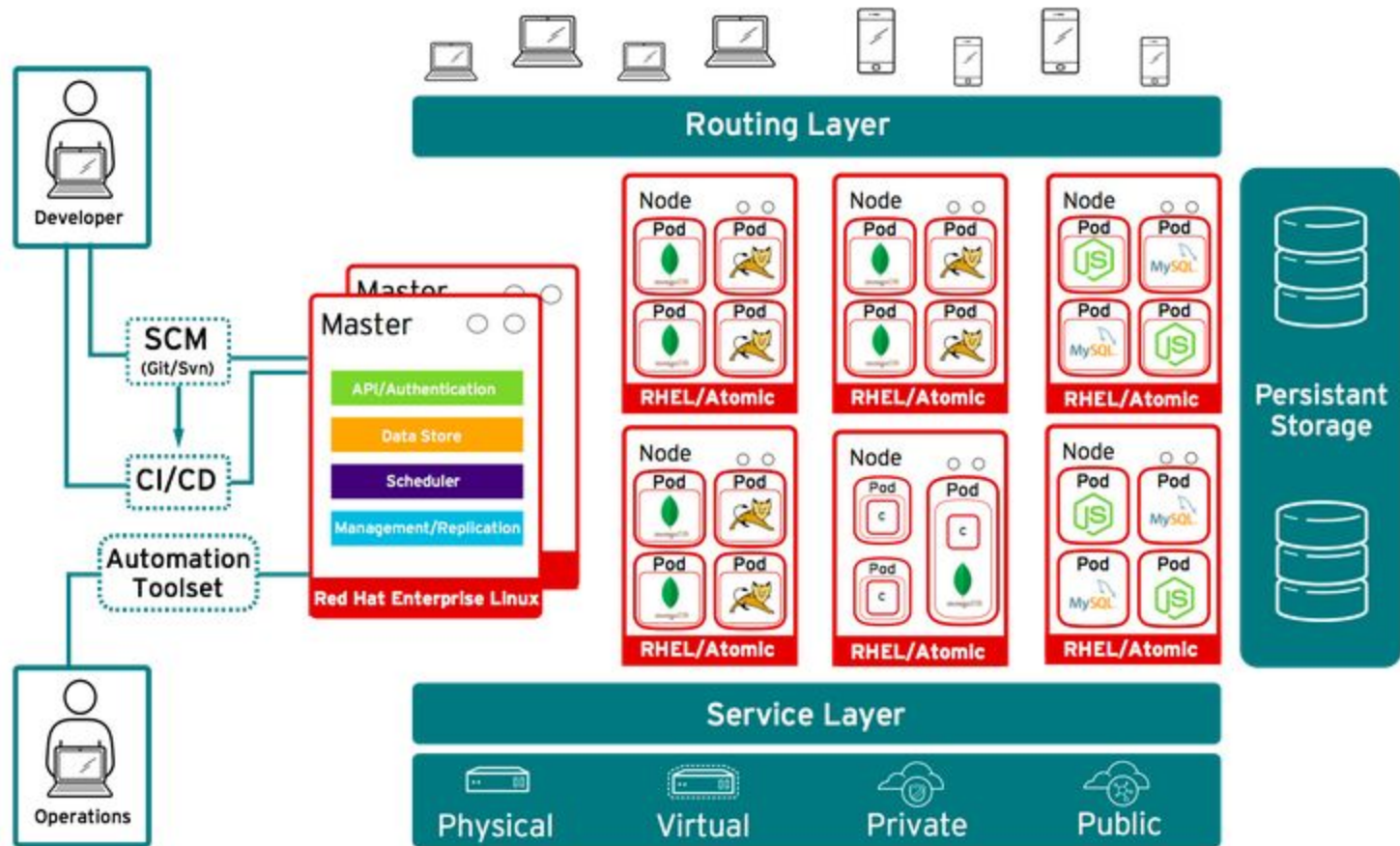
# OpenShift Intro (I)

- OpenShift is Red Hat's Container Platform built on top of K8S
- Turns K8S into a PaaS
- Adds additional tools and services that provide CI / CD and **Continuous Operation** to apply the DevOps Principles and achieve NoOps for developers
- OpenShift is **one of** the Enterprise Grade Kubernetes Implementations
- Hides some K8S complexities and adds some new complexities ;-)

# OpenShift Intro (II)

- OpenShift comes with different flavors:
  - Online
  - Enterprise (3.4)
  - Dedicated
  - Origin (1.4)
- Origin is the upstream project of OpenShift
  - Runs ANYWHERE (on RHEL family)
  - Helps to build your OWN enterprise grade K8S Implementation
- Origin comes with great community support

# OpenShift (Kubernetes Platform :-)



Source: <https://docs.openshift.org/latest/architecture/index.html>

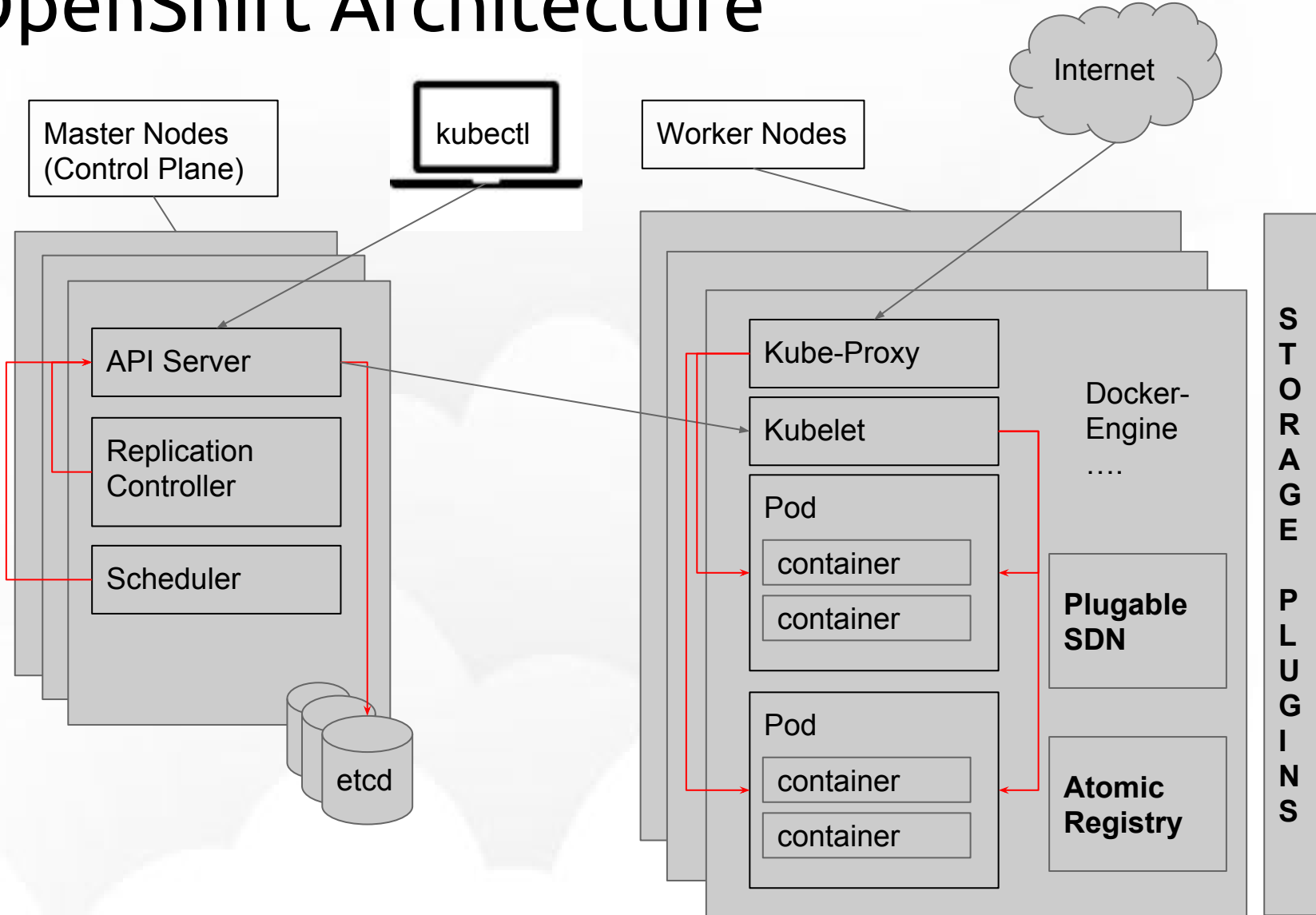
# What does OpenShift add to K8S?

- Source code management, source to image builds (S2I), and a large set of builder images and quick starts
- Managing and promoting images at scale as they flow through your system
- Flexible deployment models: rolling, blue/green and [load balancing for A/B testing](#)
- Team and user tracking for organizing a large developer organization

# What does OpenShift Enterprise add to Origin

- Enterprise Grade Support
- Enterprise version lags behind Origin in favor of stability
- Some Application templates and utilities come as part of packages for RHEL users
- In short: stability over early adoption
- From our experience Origin is stable enough for DEV / Test and even for Production if you can afford to support it on your own or through community support!

# OpenShift Architecture



# Customized CI / CD, extend OpenShift

- With [OpenShift Pipeline Jenkins Plugin](#)
  - Create Pipeline BuildConfig
  - Jenkins auto-provisioning
  - Web Console Visualization
- Have a look on [fabric8.io](#), the Integrated Development Platform for K8S / OpenShift
- Don't miss: [Fabric8 OpenShift Manifests](#)

# About Ansible

- Ansible is Red Hat's Automation Engine that automates Cloud Provisioning through Configuration Management, Application Deployment and Infra Orchestration
- Uses YAML to describe automation jobs through easy to write Inventory files and PlayBooks
- Supports continuous deployments with zero downtime rolling updates



# About Terraform

- Terraform is HashiCorp's Infrastructure Orchestration Tool that helps to describe the **desired state** of the infrastructure in declarative way!
- Deploys groups of resources (e.g. Bare Metal, VMs, Containers, Networks, Nat Gateways, Loadbalancers, etc..)
- Uses HCL (HashiCorp Configuration Language) to describe the desired state of your infrastructure
- Terraform is Cloud Agnostic, helps to avoid Heat- or CloudFormation- Lock-In → DEMO

# Terraform DEMO on OpenStack & AWS

# DEMO



TERRAFORM

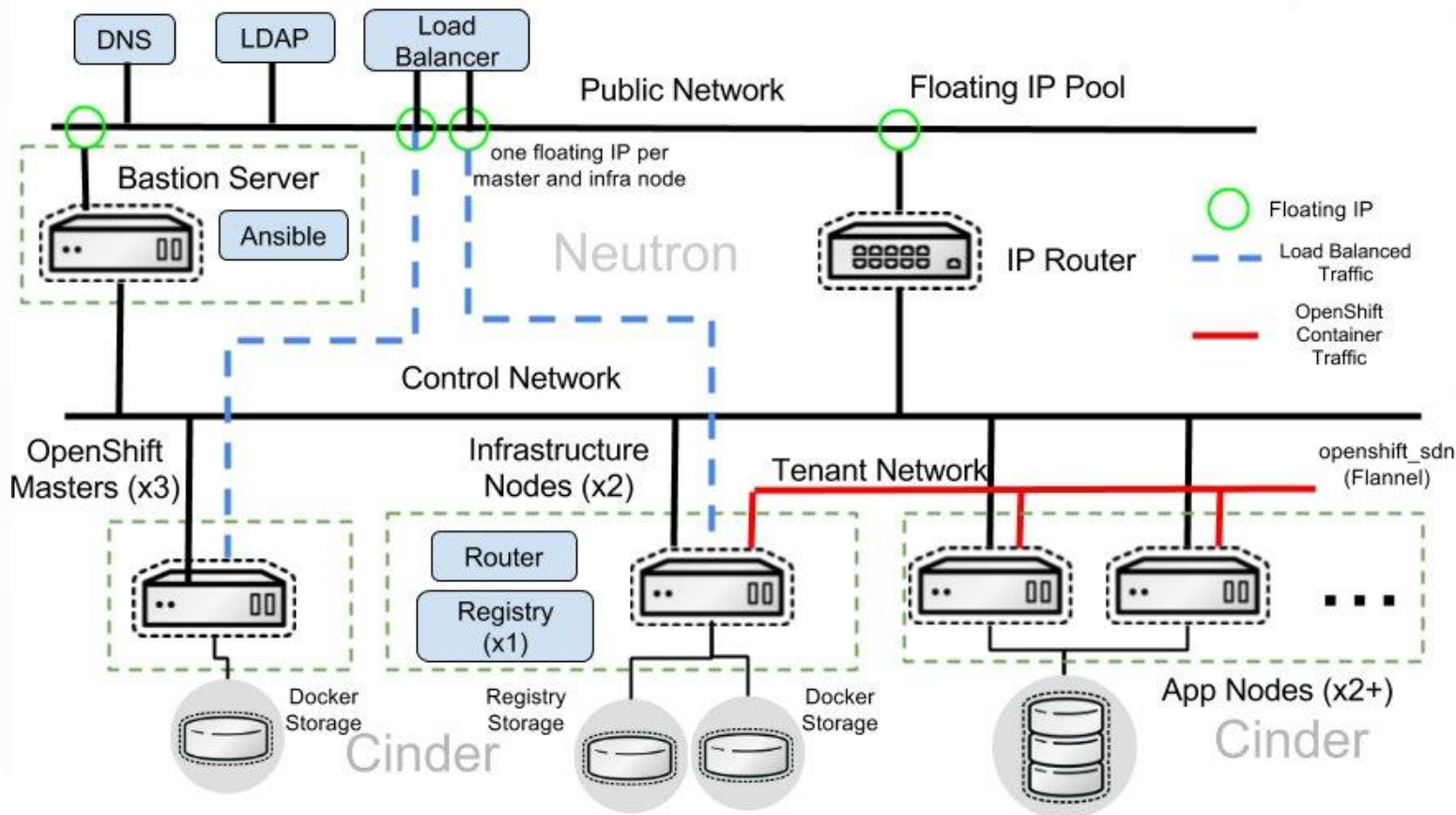
# QuadrupleO

- **OpenShift Origin On OpenStack**
- Uses OpenStack RDO with CentOS 7.3 Golden Images
- DEMO on OpenStack
- Read the full story and get the guide:
  - <https://goo.gl/VQT9T4>
  - <https://goo.gl/gbnMDD>
- **Known Issues with Origin:**
  - Exited containers and dangling images do not get purged automatically
  - Persistent Volumes with NFS is not stable
  - Cinder Integration didn't worked out of the box for us

# OCP On RHOSP

## OpenShift in the Cloud: Architecture Template

Version 0.4  
6 Oct 2016  
Mark Lamourine  
<mlamouri@redhat.com>



Source: <https://blog.openshift.com/openshift-container-platform-reference-architecture-implementation-guides/>

# Run OpenShift ANYWHERE

- OpenShift Container Platform Reference Architecture Implementation Guides
  - Deploying on Red Hat OSP
  - Deploying on Amazon Web Services
  - Deploying on Google Cloud Engine
  - Deploying on VMware vCenter
  - Deploying on Microsoft Azure is targeted to release in February 2017

Source:

<https://blog.openshift.com/openshift-container-platform-reference-architecture-implementation-guides/>

# Questions to ask

- Why OpenShift and not pure Kubernetes?
- Why OpenShift On OpenStack?
- Why not running OpenShift on Bare Metal?
- Why not just using OpenShift dedicated on GCE or use GKE?
- Why not AppScale / Cloud Foundry / Magnum / Rancher / Tectonic, etc. as alternatives?

# References / Books

- [OpenShift Origin Docs](#)
- [Free OpenShift eBook](#) (for developers)
- The [TerraformBook.com](#)
- [QuadrupleO](#)
- [OpenShift 3.3 Pipelines – Deep Dive](#)
- Blog post: [OpenShift makes Kubernetes more Exciting](#)

# Last but not least

- Thinking about Cloud-Native OpenStack:
  - to run OpenStack On OpenShift with Kolla Containers
  - or go with [OpenStack-Helm](#) project (initiated by AT&T)

```
Project: kubernauts
Pod: horizon-2-2pk8o
horizon-2-2pk8o
app horizon deployment
Details Environment
Status
State:
Last State
Ready:
Restart Count:
Debug in terminal
Show annotations

Debug Container horizon
horizon-2-2pk8o-debug -- Running
This temporary pod has a modified entrypoint command to debug a failing container. The pod will be available for one hour and will be deleted when the terminal window is closed.
Original Command: /bin/sh -c kolla_start

/tutorials/stateful-application/zokeeper.yaml^C -f http://k8s.io/docs
()root@horizon-2-2pk8o-debug /# /bin/sh -c kolla_start
INFO: main:Kolla config strategy set to: None
INFO: main:loading config file at /var/lib/kolla/config_files/config.json
ERROR: main:unexpected error:
Traceback (most recent call last):
  File "/usr/local/bin/kolla_set_configs", line 304, in <module>
    exit_code = main()
  File "/usr/local/bin/kolla_set_configs", line 298, in main
    execute_config_strategy()
  File "/usr/local/bin/kolla_set_configs", line 231, in execute_config_strategy
    config = load_config()
  File "/usr/local/bin/kolla_set_configs", line 174, in load_config
    config = load_from_file()
  File "/usr/local/bin/kolla_set_configs", line 162, in load_from_file
    with open(config_file) as f:
IOError: [Errno 2] No such file or directory: '/var/lib/kolla/config_files/config.json'
()root@horizon-2-2pk8o-debug /#
```



**Q&A**

**Thank You for  
Your attention!**