Deep Dive: Kubespray (a sig-cluster-lifecycle's project)

张荣  Rong Zhang, Suning.com, (@riverzhang on github)
• Kubespray Overview
• kubespray Community
• Is it too slow?
• Topics
Kubespray Overview
Kubespray is a sig-cluster-lifecycle’s project to create, configure and manage kubernetes clusters. It provides optional, additive functionality on top of core kubernetes.
Mission of kubespray

Easily install and manage Kubernetes clusters
Kubespray at a glance

• Cluster lifecycle manager
• Flexible and composable
• Production ready
• Ansible based
• One package-based component: Docker, Cri-o etc...
• Cross-platform
• Multi-arch
• Community driven since 2015
• Base of kubeadm since 2018
• Just bring your own machine
• Certified Kubernetes Installer (CNCF)
Deployment workflow

- Bootstrap OS
- Preinstall step
- Install Docker
- Install etcd
- Install Kubernetes Master
- Install Kubernetes Minion
- Configure network plugin
- Addons
Lifecycle of cluster operations

- Support full lifecycle of cluster operations
  - New cluster
  - Upgrade cluster
  - Scale a cluster
  - Remove nodes or an entire cluster
- Backup and restore
  - etcd snapshots taken during upgrade
For Kubeadm - automagic

For etcd:

- All nodes should be in inventory
- First, check if a cert was created. If not, then gen_certs is set for the host
- Next, check if cert is present on target host. If not, then sync_certs is set for the host
- Gen certs script runs and generates all certs for nodes in gen_certs
- All masters get a copy of all certs
- All nodes get a unique client cert
- Master nodes get three certs: member, client, and admin
- There are no differences in access. Any cert can read/write anything
High Availability

- **Etcd**
  - Native support for all clients to connect to all ETCD instances

- **Apiserver**
  - External LB (Cloud LB, F5)
  - Local LB (nginx, proxy), static pod in kubernetes cluster
Local LB (default)
User options

**Host Provider**
- Cloud Services
  - GCE
  - AWS
  - OpenStack
  - Azure
  - Digital Ocean
  - Packet
- On-prem
  - Bare metal
  - VMware
  - KVM
  - Vagrant

**OS**
- Operating System
  - Centos
  - SuSe
  - Debian
  - Container Linux
  - REHL
  - Redora
  - Atomic
  - Ubuntu

**Network**
- Plugins
  - Weave
  - Flannel
  - Calico
  - Kube-Router
  - Canal
  - Contiv
  - Multus
  - Cilium

**Certificate management**
- Certs
  - kubeadm
  - Openssl

**Container Engine**
- Engines
  - docker
  - cri-o
  - containerd

**Kubernetes Feature**
- Features
  - Cloud-provider
  - Podsecuritypolicy
  - basic auth
  - OIDC
  - QOS
  - GPU
  - Audit
  - Proxy-mode
  - ...

**Deployment mode**
- Etcd mode
  - Etcd cluster
  - Etcd events cluster
- HA mode
  - Cloud LB
  - Local LB (nginx,proxy)
Cross-platform

- Terraform
- VAGRANT
- TKE
- Other provisioning

Bare metal

- OpenStack
- GCP
- Azure
- AWS
- Other cloud
Multi-arch

options image_arch

• Arm
• Arm64
• Amd64

https://how-to-deploy-multi-arch-kubernetes-cluster-using-kubespray
kubespray Community
Community

Stars: 6400+
Forks: 2600+
Commits: 4400+
Contributors: 450+
Ensure that Kubespray is production-ready

- All components/features are tested
- All changes are safe for upgrades
- All components are HA-ready and scalable
- Minimal comprehensive set of applications

Kubespray is inclusive:

- All components run on all supported OSes
- Include automated support for many choices (container runtime, network plugins)
- Ansible architecture enables users to compose their own deployment
- All options are configurable, but defaults to upstream defaults

Kubespray is opinionated on deployment strategy:

- All components are binary or Docker container
- No system packages
- On-premise is a first-class target for deploy
Continuous Integration

How is Kubespray tested?

How to balance code coverage vs speed?

20+ test cases on Packet+Kubevirt, GCE, OPENSTACK, OVH

We support:

- 6 operating systems
- 7 network plugins
- On prem/cloud deployments
- CI strategies:
  - All-in-one
  - Separate roles (kube-master, etcd, kube-node)
  - HA
  - Graceful upgrade
  - Non-graceful upgrade

Full tests are only started after a maintainer approves PR for testing.
Contribution guidelines

How can we improve contributor experience?

Keep it readable:
- `yamllint`
- `ansible-lint`

All variables have a default defined. But it can get complicated:
- `roles/kubernetes/node/results/main.yml`
- `roles/kubespray-defaults/results/main.yml`
- `inventory/sample/group-vars/k8s-cluster.yml`

For interoperability, try to write tasks in an OS-agnostic way
Larger contributions

All features need a maintainer.
If it breaks and nobody else can maintain it, we might drop it.

OK to add:
- New OS
- New network plugin
- New storage plugin

Not accepted:
- Plugins that only work on a single OS (but maybe okay for limited feature preview?)
- Helm charts (wrap your deploy instead)
Long-term support in Kubespray

Master branch supports N-2 Kubernetes releases

Tagged releases are also an option

Stable branches since Kubernetes 2.11 (Kubespray release 2.7)
Features
• Support for Kubeadm experimental control plane
• Nodelocaldns mode is enabled by default
• Add HAProxy as internal loadbalancer
• ARM support
• ClearLinux OS
• Local-path-provisioner

CI
• Packet+Kubevirt, OVH, terrfrom+openstack
• K8s Conformance
Roadmap 2019

• Improve observability options out of the box
• Adopt (and build) new tools, best practices and features in alignment with the rest of SIG Cluster Lifecycle
  • kubeadm
  • ComponentConfig
  • etcdadm
• Decentralized orchestration
  • Auto-scaling
  • Automatic upgrades
  • Fast provisioning at scale
• Multi-arch
• CI
Is it too slow?
Is it too slow?

Basic tips:

● Use a decent sized machine (8gb memory)
● Locate Ansible node near the target nodes
● Low latency
● Deploy masters first, then kube-nodes
  ○ --limit kube-master:etcd
  ○ --limit kube-node:!kube-master:!etcd
● Redeploy using tags (master, node, etcd, kubernetes-apps, helm)
Kubeadm Experimental control plane

- Speeds up deploy slightly
- Reduces the requirement of config management software to copy certs
- Certificates get uploaded to a secret, encrypted with a key
Topics

- Continuous Integration
- Speeding up deployment
- Working Kubespray into your provisioning + deployment
- Certificate management
- kubeadm in Kubespray
- Release support flexibility
- Kubeadm experimental control plane
- Contributing to Kubespray
- What are kubernetes defaults? Where do I put new ones? Contribute to a role and you need to define a variable
- Where to add new features? New roles. Upgrade considerations
Managing Kubernetes in Air Gap/Offline Environments - Rong Zhang, Suning.com
Tuesday, June 25 • 18:15 - 18:50
Q&A