

KUBERNETES

7 Key Considerations for Kubernetes in Production

20 Feb 2019 3:00am, by [Vamsi Chemitiganti](#) and [Peter Fray](#)



Vamsi Chemitiganti

Vamsi Chemitiganti is Chief Strategist at Platform9 Systems. Vamsi works with Platform9's Client CXOs and Architects to help them on key business transformation initiatives. He holds a BS in Computer Science and Engineering as well as an MBA from the University of Maryland, College Park.

Today Enterprise IT does not question the value of containerized applications anymore. Given the move to adopting DevOps and cloud native architectures, it is critical to leverage container capabilities in order to enable digital transformation.

Enterprises are using Kubernetes to create modern architectures composed of microservices and serverless functions which scale seamlessly.

However, two years of working with Kubernetes for enterprise applications, and large-scale production deployments have taught us valuable real-world lessons about the challenges of Kubernetes in the enterprise, and what it REALLY takes in order to make it ready for prime time and enable organizations to safely bet on Kubernetes to power mission-critical enterprise application. Large and complex enterprises that have invested in container-based applications often struggle to realize the value of Kubernetes and container technology, due to operational or Day-two management challenges. In this post, we share seven fundamental capabilities large enterprises need to instrument around their Kubernetes investments in order to be able to effectively implement it and utilize it to drive their business.

Introduction



Peter Fray

Peter Fray is Chief Technologist at Platform9. With a unique blend of technical experience — from IaaS, networking, Kubernetes, databases, and more — he helps large enterprises successfully adopt cloud, containers and modern technologies to accelerate application delivery and improve data center utilization.

Typically, when developers begin to experiment with Kubernetes, they end up deploying Kubernetes on a set of servers. This is only a proof of concept (POC) deployment, and what we see is that this basic deployment is not something you can take into production for long-standing applications, since it is missing critical components to ensure smooth operations of mission-critical Kubernetes-based apps. While deploying a local Kubernetes environment can be a simple procedure that's completed within days, an enterprise-grade deployment is quite another challenge.

A complete Kubernetes infrastructure needs proper DNS, load balancing, Ingress and K8's role-based access control (RBAC), alongside a slew of additional components that then makes the deployment process quite daunting for IT. Once

whenever there's a new version of Kubernetes released by the community, and your production clusters need to be upgraded without risking any application downtime.

Bare-bone Kubernetes is never enough for real-world production applications. Here are seven key services you need around bare-bone Kubernetes to enable mission-critical production use:

#1 Managed Kubernetes Service Ensures SLA and Simplifies Operations

While it may be controversial for some, I rather start with the most critical point and just cut to the chase: for production, mission-critical apps — do not fall down the DIY trap with Kubernetes.

Yes: Kubernetes is awesome. Containers are the future of modern software delivery, and Kubernetes is the best — and the de facto standard — for orchestrating containers. But it is notoriously complex to manage for enterprise workloads, where SLAs are critical.

At the high level, Kubernetes controls how groups of containers (Pods) that make up an application are scheduled, deployed and scaled, and how they leverage the network and the underlying storage. Once you deploy your Kubernetes cluster(s), IT Operations teams must work out how these pods are exposed to consuming applications via request routing, how they ensure the health of these pods, HA, zero-downtime environment upgrades, and more. As clusters grow, IT needs to enable developers to onboard in seconds, enable monitoring, troubleshooting, and ensure smooth operations.

The biggest challenge we find customers dealing with is Day 2 operations with Kubernetes.

Once you go beyond the first install or so, various issues emerge:

1. Configuring persistent storage and networking on a large scale
2. Staying up to date with the rapidly moving community releases of Kubernetes
3. Patching both applications and the underlying Kubernetes version for security as well as regular updates
4. Setting up and maintaining monitoring and logging

The operational pain of managing production-grade Kubernetes is compounded by the industry-wide talent scarcity and skills gap. Most organizations today struggle to hire the much sought-after Kubernetes experts, and they themselves lack advanced Kubernetes experience to ensure smooth operations at scale.

Managed Kubernetes services essentially deliver an enterprise-grade Kubernetes, without the operational burden. These could be services offered exclusively by the public cloud providers like AWS or Google Cloud, or solutions that enable organizations to run Kubernetes on their own data centers or on hybrid/multicloud environments.

And even with managed services, you need to be mindful that different types of solutions would use “Managed” or “Kubernetes-as-a-Service” to describe very different levels of management experience. Some would only allow you an easy, self-service way to deploy a Kubernetes cluster, while others would take over some of the ongoing operations for managing this cluster, while others still offer a fully-managed service that does all of the heavy lifting for you, [with service level agreement guarantee](#), and without requiring any management overhead or Kubernetes expertise from the customer.

For example, with [Platform9 managed service](#), you can be up and running in less than an hour with enterprise-grade Kubernetes that just works on any environment. The service eliminates the operational complexity of Kubernetes at scale by delivering it as a fully managed service, with all enterprise-grade capabilities included out of the box: zero-touch upgrades, multicluster operations, high availability, monitoring, and more — all handled automatically and backed by a 24x7x365 SLA.

Kubernetes would be the backbone of your software innovation. Don’t scramble or jeopardize your business with a DIY approach, but choose the right managed service from the get-go, to set you up for success.

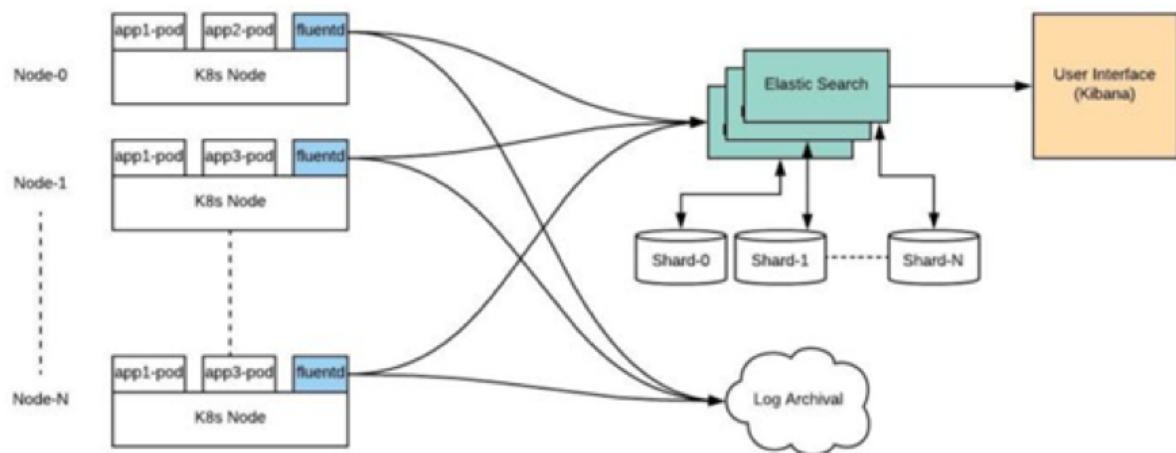
#2 Cluster Monitoring and Logging

Production deployments of Kubernetes often scale up to hundreds of pods and the lack of effective monitoring and logging can result in an inability to diagnose severe

Monitoring provides visibility and detailed metrics of your Kubernetes infrastructure. This includes granular metrics on usage and performance across all cloud providers or private data centers, regions, servers, networks, storage, and individual VMs or containers. A key use of these metrics should be around improving data center efficiency and utilization (which, obviously translates to costs), on both on-prem and public cloud resources.

Logging is a complementary and required capability for effective monitoring. Logging ensures that logs at every layer of the architecture — the Kubernetes infrastructure and its components as well as the application are all captured for analysis, troubleshooting and diagnosis. Centralized, distributed, log management and visualization is a key capability that is either met using proprietary tools or open source tools such as FluentBit, Fluentd, Elasticsearch and Kibana (also known as the [EFK stack](#)).

Monitoring should not only be 24X7, but also provide proactive alerts for possible bottlenecks or issues in Production, as well as role-based dashboards with KPIs around performance, capacity management, and more.



Kubernetes Logging and Monitoring Stack — Architecture

#3 Registry and Package Management — Helm/Terraform

A private registry server is an important function in that it stores Docker images securely. The registry enables image management workflow, with image signing, security, LDAP integration, and more. Package managers, such as Helm, provide a template (called a “chart” in Helm) to define, install, and upgrade Kubernetes-based application.

This streamlines the CI/CD pipeline and release processes of Kubernetes-based applications. Developers can more easily collaborate on their applications, version code changes, ensure deployment and configuration consistency, ensure compliance and security, and roll back to a previous version, if needed. Private Registry along with package management ensure that the right images are deployed into the right containers, and that security is integrated into the process as well.

#4 CI/CD Toolchain for DevOps

Enabling a CI/CD pipeline is critical to improve the quality, security, and to accelerate the speed of releases for Kubernetes-based applications. Today, continuous integration pipelines (such as unit testing, integration testing, etc.) and continuous delivery pipelines (such as deployment processes from dev environments all the way through to production) — are configured using GitOps methodologies and tools. The developer workflows usually start with a “git push” — with every code check-in usually triggering a build, tests and deployments processes. This includes automated Blue/Green or Canary deployments with tools like Spinnaker or others. It’s important that your Kubernetes infrastructure is easily “pluggable” to these CI/CD tools to enable your developers and improve their productivity and the quality of the release.

#5 Cluster Provisioning and Load Balancing

Production-grade Kubernetes infrastructure typically requires the creation of highly available, multimaster, multi-etcd Kubernetes clusters that can span across availability zones in your private or public cloud environment. The provisioning of these clusters usually involves tools such as Ansible or Terraform.

Once clusters have been setup and pods created for running applications, these pods are fronted by load balancers, which route traffic to the service. Load balancers are not a native capability in the open source Kubernetes project and so you need to integrate with products like NGINX Ingress controller, HAProxy or ELB (on an AWS VPC) or other tools that extend the Ingress plugin in Kubernetes to provide load-balancing.

#6 Security

It goes without saying that security is a critical part of cloud native applications and needs to be considered and designed for from the very start. Security is a constant

security and the security of the container and infrastructure itself. There range from using role-based access control, multifactor authentication (MFA), A&A (Authentication & Authorization) using protocols such as OAuth, OpenID, SSO etc.; Different tools that provide certification and security for what goes inside the container itself (such as image registry, image signing, packaging), CVE scans, and more.

#7 Governance

Processes around governance, auditing and compliance are a sign of the growing maturity of Kubernetes and the growing proliferation of Kubernetes-based application in large enterprises and regulated industries. Your Kubernetes infrastructure and related release processes need to integrate with tools to allow visibility and an automated audit trail — and consequently the right compliance enforcement — for the different tasks and permission levels for any updates to Kubernetes applications or infrastructure.

In summary, enabling Kubernetes for enterprise, mission-critical applications require a lot more than just deploying a Kubernetes cluster. The above key considerations would help you ensure the design of your Kubernetes infrastructure fits for production workloads — now, and in the future.

Feature image [via Pixabay](#).

CONTRIBUTED

+

THE NEW STACK UPDATE

A newsletter digest of the week's most important stories & analyses.

Email Address

Subscribe

RELATED STORIES



1

PROGRAMMING LANGUAGES

Static Site Revolution: Top Websites Built with Gatsby15 Mar 2019 3:00am, by [Weronika Nużyńska](#)

CLOUD NATIVE / DEVOPS

How DevSecOps Will Make Your Organization Future-Ready13 Mar 2019 3:00am, by [Lior Cohen](#)[View / Add Comments](#)

Please stay on topic and be respectful of others. Review our [Terms of Use](#).

SPONSORED FEED

Architecture

Development

Operations

MARCH 15, 2019



Welcome! New synthetic monitoring locations in Asia, Africa, and the US

MARCH 15, 2019



Steeltoe 2.2 Gives Your .NET Microservices a Boost

MARCH 15, 2019



Get Google Cloud Certified at Next '19: What you need to know

MARCH 15, 2019



Agile in Education: A Partnership with Promise for Oregon's Public Schools

MARCH 15, 2019



OpenShift All-in-One (AIO) for Labs and Fun

MARCH 15, 2019



Start deploying using the CircleCI deployment Orb with DeployHub

MARCH 15, 2019



Using D3 in React: A Pattern for Using Data Visualization at Scale

MARCH 15, 2019



Architecture

Development

Operations



How to install OpenEBS on OpenShift

MARCH 15, 2019



How DevOptics Helps Heidelberg See the Big Picture

MARCH 15, 2019



How to Run HA Cassandra on IBM Cloud Kubernetes Service

MARCH 14, 2019



Kubernetes AuditSync: Real-time K8s Audits and Forensics

MARCH 14, 2019



IP Masking in HAProxy

MARCH 14, 2019



NetEase's System Supports 30,000 Nodes in a Single Cluster

MARCH 14, 2019



Cut Through Complexity With Better Event Intelligence by David Shackelford

MARCH 14, 2019



CI/CD Pipeline: A Gentle Introduction

MARCH 14, 2019

Architecture

Development

Operations



Cloud Native Computing Foundation
Announces Schedule for KubeCon +
CloudNativeCon Europe 2019
MARCH 13, 2019

NS1.

Read: How Next-Gen Domain Security Can
Protect Your Business from DNS Attacks
MARCH 13, 2019



Chef Software is the First CIS Partner
Certified on AWS, Azure, and GCP
MARCH 13, 2019



Puppet joins the new Continuous Delivery
Foundation
MARCH 12, 2019



Everything I know about Integration I
learned at the Ballet
MARCH 12, 2019



How to Test Serverless Apps
MARCH 12, 2019



Introducing Interactive Slack Notifications
MARCH 11, 2019



Writing High Performance .NET Code
MARCH 11, 2019

Architecture

Development

Operations

MARCH 11, 2019



OCI New Service Roundup

MARCH 09, 2019



Webinar: Data Innovation in Financial Services

MARCH 07, 2019



Harness uses Kubernetes for On-Premises Deployment

MARCH 05, 2019



Design, develop, and deploy Reactive systems in Java using event storming and domain-driven design

MARCH 05, 2019



Configuration management: a Spring Boot use-case with Kubernetes

MARCH 04, 2019



Operationalizing Cloud Provider Managed Kubernetes

MARCH 04, 2019



Injection Attacks: Protecting Your Serverless Functions

FEBRUARY 28, 2019

Architecture

Development

Operations



Why You Should Bet on Network Monitoring Software

JULY 20, 2017

ARCHITECTURE

Cloud Native

Containers

Edge / IoT

Microservices

Networking

Serverless

Storage

DEVELOPMENT

Application Security

Cloud Services

Data

Machine Learning

Programming Languages

OPERATIONS

CI/CD

Culture

DevOps

Kubernetes

Monitoring

Tools

THE NEW STACK

