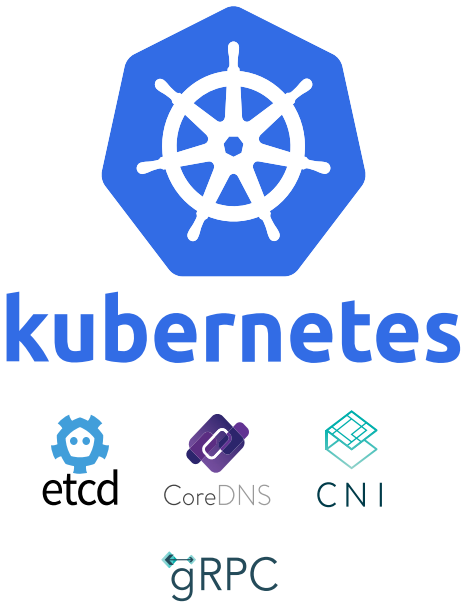
Modern Application Delivery with Kubernetes



SECURITY

APP PACKAGING + DEPLOYMENT

MONITORING

TRACING

SERVERLESS

william.hearn@canada.ca
sylus
william_hearn

zachary.seguin@canada.ca
zachomedia
zachomedia

The Base Platform

Introduction

“A secure Cloud Management Platform built using projects from the Cloud Native Computing Foundation”

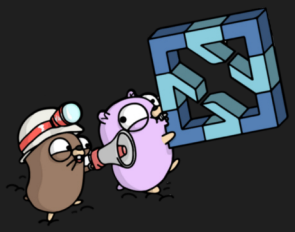
What is the Base Platform?

The base of the platform is Kubernetes which is the first graduate of the CNCF (Cloud Native Computing Foundation). Kubernetes is a platform in and of itself but in particular it is a platform for building other platforms.

Configured Ingresses

- istio-kiali.example.com
- istio-grafana.example.com
- prometheus.example.com
- alertmanager.example.com
- grafana.example.com
- ns-elastic.example.com
- ns-kibana.example.com

Table of Contents



CNCF

Introduction

“Building Sustainable Ecosystems for Cloud Native Software”

What is the CNCF?

The Cloud Native Computing Foundation, or CNCF, fosters a landscape of open source projects by helping provide end-user communities with viable options for building cloud native applications. By encouraging projects to collaborate with each other, the CNCF hopes to enable fully-fledged technology stacks comprised solely of CNCF member projects. This is one way that organizations can own their destinies in the cloud.

A total of twenty-five projects have followed Kubernetes and been adopted by the CNCF. Projects go through various phases - Sandbox, Incubation, and then finally Graduation - depending on their level of code maturity and use in production.

Kubernetes

Introduction

“Portable, extensible open-source platform for managing containerized workloads”

What is Kubernetes?

Kubernetes is a container orchestration system which abstracts the management of compute, networking and storage.

How does Kubernetes work?

Kubernetes is a series of APIs which allow you to use desired-state configuration

Community

A huge (tens of thousands) and vibrant community of both developers, businesses, and users.

Running Kubernetes

Managed Service

Many cloud service providers offer a managed Kubernetes service, such as:

- Google
- Microsoft
- Amazon
- Digital Ocean

Platform as a Service (PaaS)

There are many platform Kubernetes offerings, such as:

- Pivotal
- OpenShift

Self-hosted

Kubernetes can be self-hosted, with solutions like:

- kubeadm (official)
- kubespray

Kubernetes Continued

Declarative State

Kubernetes utilizes declarative configuration, where the developer specifies what they want and then Kubernetes reconciles how to do that always trying to enforce that state.

Hybrid Workloads

Kubernetes supports running both Linux and Windows containers in the same cluster

Container Runtimes

Kubernetes supports many different container runtimes which are anything that implements the Container Runtime Interface (CRI)

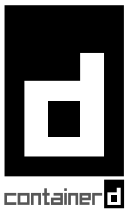
- Docker
- Containerd
- Kata Containers (FireCracker)

In use by large Enterprises

- Google
- GitHub
- Reddit
- Shopify
- CERN
- IBM
- Open AI

In use by Government Departments

- Statistics Canada
- Canadian Digital Services
- Treasury Board Secretariat
- Shared Services Canada
- Canadian Security Establishment
- Canadian Security Intelligence Service
- Royal Canadian Mounted Police



Containerd

Introduction

“An industry-standard container runtime with an emphasis on simplicity, robustness and portability”

What is Containerd?

Available as a daemon for Linux and Windows. It manages the complete container lifecycle of its host system, from image transfer and storage to container execution and supervision to low-level storage to network attachments and beyond.

Why use Containerd?

Containerd contains only a fraction of the capabilities of either the Docker Desktop stack or the pure Docker engine itself. Operationally Containerd makes perfect sense as an implementer of the CRI API from Kubernetes.

Operators

Introduction

“An industry-standard container runtime with an emphasis on simplicity, robustness and portability”

What is an Operator?

An Operator is a method of packaging, deploying and managing a Kubernetes application. A Kubernetes application is an application that is both deployed on Kubernetes and managed using the Kubernetes APIs and kubectl tooling.

To be able to make the most of Kubernetes, you need a set of cohesive APIs to extend in order to service and manage your applications that run on Kubernetes. You can think of Operators as the runtime that manages this type of application on Kubernetes.

Base Platform

The following Operators are leveraged by default in our base Platform.

- Cert Manager
- Elastic Cloud on Kubernetes
- Fluentd
- GateKeeper
- Prometheus Operator
- Velero

Helm

Introduction

“The Kubernetes Package Manager”

What is Helm?

Helm is the official Kubernetes package manager which allows you to define, install and upgrade complex applications with ease. It allows you to template the necessary cluster resources to easily deploy multiple instances of your application to the same cluster(s).

Helm also provides application lifecycle management through install, upgrade and delete hooks. There are many official helm charts for projects available on GitHub.

- Official Kubernetes package manager
- Helm charts help define, install, and upgrade complex applications
- Templating of your applications cluster resources
- Provides application lifecycle management

Base Platform

The following Helm charts are leveraged by default in our base Platform. Note that some of the Operators themselves can be installed via Helm charts.

Operators

- Cert Manager
- Fluentd
- Istio
- Prometheus Operator
- Sidecar Terminator
- Velero

Services

- Istio
- Sidecar Terminator

Terraform

Introduction

What is Terraform?

Terraform enables you to safely and predictably create, change, and improve infrastructure. It is an open source tool that codifies APIs into declarative configuration files that can be shared amongst team members, treated as code, edited, reviewed, and versioned.

Base Platform

The following Terraform modules are leveraged by default in our base Platform.

- terraform-kubernetes-aks
- terraform-kubernetes-aks-platform

The **terraform-kubernetes-aks** module installs the Azure Kubernetes Service along with all other related infrastructure such ass resource groups, service principals, storage accounts and container registries.

View a demo of Terraform in action at: https://www.youtube.com/watch?v=HahqsPCfTdE0

The **terraform-kubernetes-aks-platform** module configures the AKS cluster once it is up and running with all of the required cluster configuration along with all the default cluster wide applications. This module calls the following submodules:

- terraform-kubernetes-aad-pod-identity
- terraform-kubernetes-cert-manager
- terraform-kubernetes-elastic-cloud
- terraform-kubernetes-fluentd
- terraform-kubernetes-istio
- terraform-kubernetes-namespace
- terraform-kubernetes-open-policy-agent
- terraform-kubernetes-prometheus
- terraform-kubernetes-sidecar-terminator
- terraform-kubernetes-vault
- terraform-kubernetes-velero

Multi Cloud

The modules are not implemented to a specific environment, and can be re-used across clouds. For example, we have re-created our Azure AKS platform in the IBM cloud environment.

- terraform-kubernetes-iks
- terraform-kubernetes-iks-platform

Istio

Introduction

“Connect, secure, control and observe (micro) services.”

What is Istio?

Istio is a service mesh that addresses the challenges developers and operators face as monolithic applications transition to a distributed micro-service architecture.

The goal of Istio is to be separated from the applications themselves and act only by intercepting all network communication (layer 3/4 and layer 7)

How does Istio work?

Istio makes it easy to create a network of deployed services with load balancing, service-to-service authentication, monitoring, and more, with few or no code changes in service code.

Support is added to services by deploying a special sidecar proxy throughout your environment that intercepts all network communication between the micro-services.

Benefits

1. Fine grained control of traffic behaviour:
    - Load Balancing between instances
    - Fine grained control of traffic behaviour
    - Rate limiting and quotas
    - Pluggable policy layer supporting ACL
    - Circuit Breaking, Fault Injection
2. Automatic mutual TLS encryption between services in the cluster
3. Automatic metrics, logs and traces for all traffic within the clusterd
4. Fault Tolerance knowing when traffic is unhealthy etc

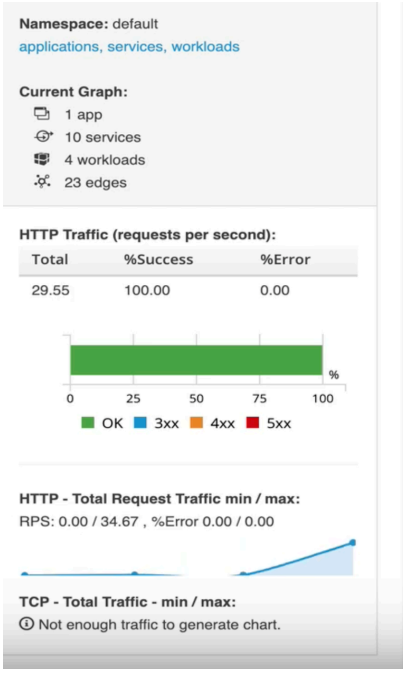
Kiali

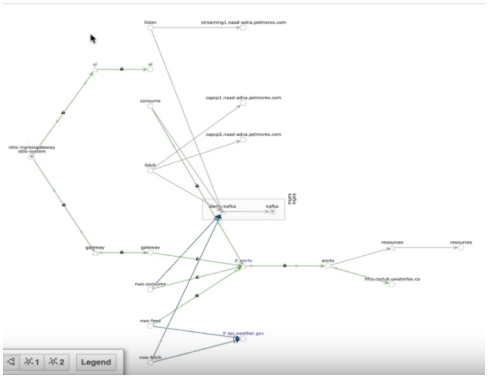
Introduction

“Service mesh observability and configuration”

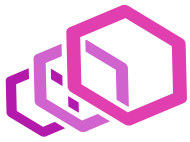
What is Kiali?

Kiali is an observability console for Istio with service mesh configuration capabilities. It helps you to understand the structure of your service mesh by inferring the topology, and also provides the health of your mesh. Kiali provides detailed metrics, and a basic Grafana integration is available for advanced queries. Distributed tracing is provided by integrating Jaeger.





View a demo of Kiali in action at: https://www.youtube.com/watch?v=oMf7jH1E3s0



Envoy

Introduction

“Connect, secure, control and observe (micro) services.”

What is Istio?

Istio is a service mesh that addresses the challenges developers and operators face as monolithic applications transition to a distributed micro-service architecture.

Prometheus

Introduction

“Real-time cluster and application metrics”

What is Prometheus?

Prometheus is an open-source systems monitoring and alerting toolkit. It enables real-time cluster and application metrics. Those metrics can be visualized at different levels such as:

- Per Application / Service
- Per Namespace / Node

How does GateKeeper work?

Prometheus scrapes metrics from instrumented jobs, either directly or via an intermediary push gateway for short-lived jobs. It stores all scraped samples locally and runs rules over this data to either aggregate and record new time series from existing data or generate alerts. **Grafana** or other API consumers can be used to visualize the collected data.

View a demo of Prometheus and Grafana in action at: https://www.youtube.com/watch?v=ZI9y7LFuW8I

Prometheus ecosystem consists of multiple components, many optional:

- Prometheus server which scrapes and stores time series data
- Alert Manager to handle customized alerting rules based on defined metrics
- Client libraries for instrumenting application code

Features

- Multi-dimensional data model with time series data identified by metric name and key/value pairs
- Flexible query language (PromQL)
- No reliance on distributed storage
- Time series collection happens via a pull model over HTTP
- Pushing time series is supported via an intermediary gateway
- Targets are discovered via service discovery or static configuration
- Multiple modes of graphing and dashboard support

Grafana

Introduction

“The analytics platform for all your metrics.”

What is Grafana?

Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored. Create, explore, and share dashboards with your team and foster a data driven culture.

Open Policy Agent

Introduction

“Policy-based control for cloud native environments.”

What is OPA?

Open Policy Agent (**OPA**), and its Gatekeeper component is a general-purpose policy engine that allows you to enforce a baseline suite of policies for all cluster resources. Uses ranging from auth / admission control to data filtering.

GateKeeper Features

- Extensible, parameterized policy library
- Native Kubernetes CRDs for instantiating / extending policy library (Constraints / Constraint Templates)
- Auditing / Reconciliation functionality

Policies

Policies are written in a language called Rego and can refer to many data sources:

- Objects that are being created and/or updated in the cluster
- External data sources fed into OPA (users/group memberships, etc)

Deployed Policies

Validating policies that are deployed into the base platform which block non complaint objects from being created.

- Containers within the limits threshold
- No duplicate ingress hosts
- Liveness and Readiness probes are set
- Deny external load balancers
- No deployed pods run under privileged
- Only allowed Container image pulled from specified registries

Mutating policies that are deployed into the base platform which can add or remove information from resources.

- Add necessary taints / tolerations for Linux / Windows Nodes
- Billing codes / Owner annotation from namespace object to sub resources

View a demo of Open Policy Agent in action at:
https://www.youtube.com/watch?v=ZI9y7LFuW8I

Velero

Introduction

“Backup and migrate Kubernetes resources and persistent volumes.”

What is Velero?

Velero is an open source tool to safely backup and restore, perform disaster recovery, and migrate Kubernetes cluster resources and persistent volumes.

It takes snapshots for your cluster resources and the data stored on persistent volumes, and allows you to restore those in the same cluster, across different clusters and, planned, across cloud providers.

Useful Commands

```
# Backup an app + persistent volume
> velero backup create app --selector release=app \
                           --snapshot-volumes

# Backup an app based on cron schedule
> velero schedule create app --selector release=app \
                             --snapshot-volumes \
                             --schedule “30 15 * * Sat”
```

View a demo of Velero in action at: https://www.youtube.com/watch?v=2BQqgDuKkT0

Cert Manager

Introduction

“Automatically provision and manage TLS certificates in Kubernetes”

What is Cert Manager?

Cert Manager is a Kubernetes add-on to automate the management and issuance of TLS certificates from various issuing sources. It will ensure certificates are valid and up to date periodically, and attempt to renew certificates at an appropriate time before expiry.

Elastic Cloud on K8S

Introduction

What is Elastic Cloud on K8S?

Elastic Cloud on Kubernetes automates the deployment, provisioning, management, and orchestration of Elasticsearch and Kibana on Kubernetes based on the operator pattern.

Current Features

- Elasticsearch and Kibana deployments
- TLS Certificates management
- Safe Elasticsearch cluster configuration & topology changes
- Persistent volumes usage
- Dynamic local persistent volumes provisioning
- Custom node configuration and attributes
- Secure settings keystore updates

Why use Elastic Cloud on K8S?

Elasticsearch allows us to centralize all applications and cluster logging into a single indexed and searchable database.

A good model is to use this in conjunction with Azure Log Analytics. Elasticsearch being used primarily for application level logs and log analytics for infrastructure logs.

Using Kibana, we can also generate visualizations of the information contained within our logs. For example, we could visualize HTTP requests and responses by request URLs, response status codes.

- Centralized logging for app / cluster
- Indexed and can be searched
- Can be visualized with Kibana

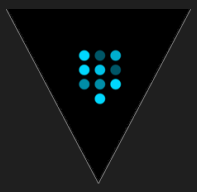
Fluentd

Introduction

“Build Your Unified Logging Layer.”

What is Fluentd?

Fluentd is an open source data collector, which lets you unify the data collection and consumption for a better use and understanding of data.



Hashicorp Vault

Introduction

Hashicorp Vault is a cloud-agnositc secret management system.

What does Hashicorp Vault?

Vault is a cloud-agnostic secret management and storage system. It has support for static secrets and the generation of dynamic secrets on the fly. Support secret engines include:

- Key-value store
- Azure/Google/Amazon service principals
- Database credentials
- PKI Certificate authority
- SSH Certificate authority
- and more!

What are the benefits?

Dynamic secrets generated by vault are intended to be valid for a short lifetime, reducing the impact of an accidental leak of credentials. Dynamic secrets are also automatically revoked after their lease expires.

Access to credentials in Vault is deny-by-default, requiring explicit policies for access to credentials.

Additionally, Vault supports multiple authentication methods for both user and machine authentication:

User:

- OpenID Connect
- LDAP

Machine:

- Token
- Kubernetes service account
- Cloud machine identity

Finally Vault maintains a full audit log of all secrets created and accessed.

View a demo of Vault in action at: https://www.youtube.com/watch?v=HahqsPCfTdE0