Must-have controllers in Kubernetes

by Jim Tario
Before the fun... Intro!

• Origin: Santa Ana, CA. Hispanic (2nd gen).
• Family: of 4; wife and 2 kiddos (1 and 4 years of age)
• School: Graduated from Cal State Northridge (CSUN) 2016
• Work: SRE at Blizzard Entertainment (~5 years)
• Hobbies: Poker, Fantasy Football, Manga/Anime, Snowboarding, and of course video games.
• Social Media: @j3enx
What are controllers?

- Control loops
  - Bring the current state closer to the desired state
  - Watch the state of your cluster, and make changes where needed.
- Operators
  - Domain specific controller
    - adds an object to the k8s API
    - configure and manage the application
Ingress Controller(s)

Supported by the K8s Project
- NGINX Ingress controller\(^1\)
- AWS Load Balancer controller\(^2\)
- Ingress GCE (GLBC)\(^3\)

- Many, many, many ingress controllers..
- https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/#additional-controllers

\(^1\)https://github.com/kubernetes/ingress-nginx
\(^2\)https://github.com/kubernetes-sigs/aws-load-balancer-controller
\(^3\)https://github.com/kubernetes/ingress-gce
Monitoring

- Prometheus Operator\(^4\)
  - Creates custom resources to deploy and manage Prometheus
    - Example of CRDs created: serviceMonitors, PodMonitors, PrometheusRules, and various others
  - Operator detects changes in k8s API to any of the CRDs
- Kube-prometheus-stack\(^5\)
  - Helm chart with a collection of k8s manifest, Grafana dashboards, Prometheus rules, and Prometheus Operator

\(^4\) [https://github.com/prometheus-operator/prometheus-operator](https://github.com/prometheus-operator/prometheus-operator)
Secrets

• External Secrets

  • Operator that integrates with external secret management systems
  
  • Backends like: AWS secrets manager, HashiCorp Vault, Google Secrets Manager

  • This operator interfaces with the external APIs (backends)

  • still a control loop and tries to maintain the desired state

---

https://github.com/external-secrets/external-secrets
DNS

- External DNS\(^7\)
  - The controller interfaces with the k8s and looks at services and ingress resources
  - Those this by looking for annotations
  - Configures DNS providers to create the DNS records.

\(^7\)https://github.com/kubernetes-sigs/external-dns
Autoscaling

- Vertical Pod Autoscaler (VPA)\(^8\)
  - Sets the pod resource requests automatically based on usage
  - Use with caution, uses an Admission Controller
    - requests to the k8s API are intercepted before modifying any object
  - Consists of 3 components: Recommender, Updater, and Admission plugin
    - Start with Recommender before enabling other components
- Shoutout to Cluster Autoscaler\(^9\)

\(^8\) [https://github.com/kubernetes/autoscaler/tree/master/vertical-pod-autoscaler](https://github.com/kubernetes/autoscaler/tree/master/vertical-pod-autoscaler)

\(^9\) [https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler](https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler)
Certificates

- cert-manager\(^{10}\)
  - simplifies the process of obtaining, renewing and using those certificates
  - can issue certificates from Let's Encrypt, Vault, Venafi, even private PKI (self-signed)

\(^{10}\)https://github.com/cert-manager/cert-manager
Awesome, but how do I install?

• Many different methods to install apps onto k8s
• Helm and Kustomize are two popular open-source tools to package k8s applications
• Automation tools exist to deploy applications
  • ArgoCD
  • Spinnaker
  • Jenkins
  • Terraform/Ansible
Tech debt

• Be conscious of all the different apps that are being installed
  • Different release cycles, vulnerabilities, etc.
• Who is maintaining them?
• Set a cadence or cycles to tackle tech debt
Closing words

The Golden Rule: Treat others how you want to be treated