Kubernetes: Setting up a highly-available cluster

Learning Objectives

- To discover and learn how to build a highly available Kubernetes cluster
- To learn and implement observability and logging on a Kubernetes cluster and the underlying infrastructure
- To experience writing runbooks that include troubleshooting steps for handling an incident in a production environment

Case Description

Create a single-zoned highly-available Kubernetes cluster using AWS as your infrastructure. This cluster is expected to run workloads and is treated as production.

Solution Must-haves

Design and setup

1. There are multiple layers that make up a single Kubernetes cluster, and having high-availability on each layer introduces robustness to your platform. Discover and learn how to build a highly-available Kubernetes platform by implementing HA on the following layers:
    a. Data plane (etcd)
    b. Control plane
        i. API
        ii. Scheduler
        iii. Control Manager
    c. Application
2. Your Kubernetes cluster should also have running apps to simulate a production environment. To simulate this, deploy at least three (3) different apps, each having characteristics that they are "owned" by different teams using the cluster

3. Observability and logging is set up and available using Prometheus and Splunk. Observability should cover the following (but not limited to):
    a. Host (EC2) metrics and logging
    b. Kubernetes objects metrics and logging
    c. Time series graphs that show the status of high-availability of cluster
4. OS packages needed to build the Kubernetes platform must be available in an instance of an Elastic Container Registry (ECR). Setting up a highly-available ECR is optional.
5. A separate section in your GitHub repo must be dedicated for operational runbooks which contains incident scenarios and troubleshooting steps to resolve or remediate each failure scenario. Some examples of incident scenarios are the following but not limited to: worker node is down, API server is down, etcd was corrupted, ...