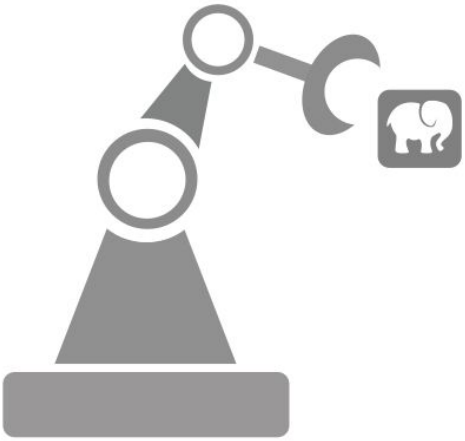
Kubectl Plugin
for
Postgres Custom Resource Definitions

By

Vineeth Reddy



Kubectl Plugin For Postgres Custom Resource Definitions

Project Proposal for Google Summer of Code 2019

1. Student Information

Name:	Vineeth Reddy
Email:	vineethbfhs@gmail.com
Phone:	+917386161718
Github username:	VineethReddy02
Website:	http://vineethweb.com/
Twitter:	https://twitter.com/VineethReddy02
University:	BV. Raju Institute of Technology, Narsapur, Medak
Major:	Information Technology

2. Introduction

The Postgres Operator is a project to create an open-sourced managed Postgres service for Kubernetes. The Postgres operator manages Postgres clusters on Kubernetes. kubectl plugins enable extending the Kubernetes command-line client `kubectl` with commands to manage custom resources. The task is to design and implement a plugin for the `kubectl postgres` command. My project aims to simplify and ease the usage of postgres clusters using the kubectl plugin. As the postgres operator is capable of many features having a kubectl plugin will ease in running the clusters and understanding the resources way better.

My GSoC work aims to solve the above-mentioned issues.

3. Impact on the Community

The need for the new kubectl plugin is well known. Using postgres in kubernetes now is not an easy task as the postgres clusters cannot be invoked directly with specific commands having a kubectl plugin helps in managing the postgres resources and in provisioning them. Using postgres in kubernetes was not so common previously, with the advent of containers and Kubernetes, it is becoming fairly common. The new kubectl plugin helps in managing postgres in kubernetes in a much better way. By thinking of core **kubectl** commands as essential building blocks for interacting with a Kubernetes cluster, a cluster administrator can think of plugins as a means of utilizing these building blocks to create more complex behavior. Plugins extend **kubectl** with new sub-commands, allowing for new and custom features not included in the main distribution of **kubectl**.

1. Basic commands like create, delete, update, get, a clone for a particular cluster using the manifest can be done.
2. Also creating postgres cluster right from the command line without having to write a manifest i.e specifying required properties through flags.
3. List all the clusters is also expected. Listing based on particular property and printing particular properties of the clusters would also be useful.
4. All the properties in the manifest of a particular cluster can be added as a command to change the property of a particular cluster right from the command line without having to change the manifest and apply.
5. Commands will be divided into two groups.

    - The first group is convenience aliases for things one can already do through kubectl.
    - The second group consists of commands that encapsulate the user experience with running the operator in production.

4. Deliverables

This section presents the outcomes that I plan to achieve in the course of this project. *Must-Haves* define the central objectives of my work. In addition, *Nice-to-Have* denotes goals which are not likely to be fully completed in the course of this project.

Must have:

1. Command to create, update and delete postgres clusters.
2. A Command that lists all the information about the pg cluster.
3. Interacting with master/Replica pod by shell prompt and underlying psql prompt.
4. Auto Completion for most commands and the cluster names, etc would be great.
5. kubectl cmd to fetch the logs of the operator with --follow flag to fetch real-time logs.
6. Increase the volume size of the cluster using kubectl command.
7. List all the kubernetes resources created by the postgres operator.
8. Adding this plugin to the official package manager for plugins. Which is https://github.com/kubernetes-sigs/krew

Good to have:

1. Extract Patroni logs.
2. Extract Postgres / WAL-E logs; the write to the $PGLOG dir inside the container.
3. Diff command to compare manifest with the actual state of the cluster stored in the operator.
4. Logs command to show the logs of a specific cluster based on cluster name provided.
5. As the operator also exposes an endpoint, where the status of the operator can be checked. It would be great to have a *check* command that checks if the operator is healthy.

5. Implementation

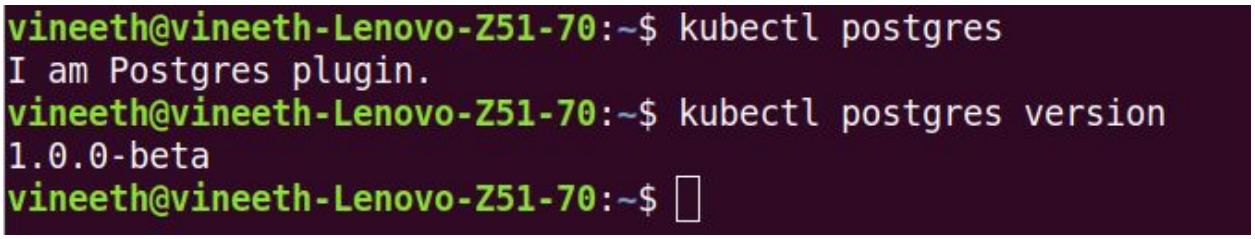
There are several ways to get information about Kubernetes cluster and its resources. You can do it using Dashboard, kubectl, or using programmatic access to Kubernetes APIs. Client-go is the most popular library used by the tools written in Go. Kubernetes is written in Go, and I find it easier to develop a plugin in the same language the main project is written.

The Postgres operator plugin will be implemented using the client-go library to patch an existing KUBECONFIG file.

These utilities exist under the k8s.io/cli-runtime repository and can be used by plugins written in Go to parse and update a user's KUBECONFIG file, obtain REST clients to talk to the API server, and automatically bind flags associated with configuration and printing.

Plugins have to use Go in order to take advantage of the tools and utilities in the CLI Runtime repository.

As a part of the initial implementation. Created a basic kubectl plugin to understand how plugins are configured and used in kubernetes cluster.



Going forward kubectl plugin for Postgres-Operator will be implemented a a new project in go. Which leverages from the above-mentioned resources.

6. Timeline

Timespan	Activity
7 May - 26 May	*Community Bonding Period:* Sketch out all the details as design docs and present to the community via a public channel.
Actual Start of the Work Period	
27 May - 02 June	**4.1** Creating the plugin, Initial setup of the plugin and Implementing the kubectl command to check Postgres CRD registered or not.
03 June - 09 June	**4.1** Create, Update, delete commands to Postgres clusters.
10 June - 16 June	**4.1** Overview of a pg cluster: list pods with their age, labels, status and add space/users/DBs.
17 June - 23 June	**4.1** "kubectl postgres --help" help command for the plugin and Documentation for 4.1, testing, mentors reviews.
Evaluation due	
24 June - 30 June	**4.2** Connect to the master or replica pod, with options to connect to the pod itself (shell prompt) and the DB within a pod (psql prompt).
01 July - 07 July	**4.2** Autocompletion for commands and resources Scale Up/Down a PG cluster using kubectl commands.
08 July - 14 July	**4.2** kubectl commands for fetch logs of the operator pod, fetch operator version, reload operator configmap / CRD without restarting the operator pod.
15 July - 21 July	**4.3** Documentation for 4.2, testing, mentors reviews.
Evaluation due	
22 July - 28 July	**4.3** Commands to extract Patroni/Postgres/WAL-E logs
29 July - 04 August	**4.3** List all k8s objects created by the operator for the cluster so far and on time permitting the support for kubectl explain for both operator and Postgres CRDs.
05 August - 11 August	**4.3** Documentation for 4.3, testing, mentors reviews and Adding this plugin to the official package manager krew.
12 August - 18 August	Final improvements and evaluation preparation.
Final evaluation due	

7. Availability and Communication

- Will be available **35-40 hrs per week** during holidays.
- Will be online for > 8hrs a day on postgresteam.slack.com with nick Vineeth.
- In a doc, I will be updating the progress weekly.

8. About Me

I am a 4th-year Undergraduate student pursuing in Information technology major at BV Raju Institute of Technology. I love dev/ops. I am passionate about system design and infrastructure. To know more about me reach out to my website http://vineethweb.com.

I have been working on kubernetes for six months. It was an amazing experience interacting with the community and sharing my views on various features. Postgres operator will definitely impact the way Postgres clusters are running in kubernetes with it's extended functionalities and having kubectl plugin will add more value in managing and provisioning the Postgres resources.

Experience

- I have been coding in Go from the last 6 months.
- I have also been using and trying Kubernetes for my personal projects and fun experiments through AKS.
- I'm a Microsoft Student Partner from India and have conducted various events and workshops specific to Kubernetes and Microsoft Azure throughout my college.
- I have also completed the following certification courses in Golang from Coursera.
    1) Getting Started with Go
    2) Functions, Methods and Interfaces in Go
    3) Concurrency in Go

Contributions to Postgres Operator

I have also given a patch to Postgres-operator in passing the go reports. which can be found here.