Extending K8s - kubectl plugin

Hi! This is Muhammed!

Education:

- Computer Engineering, Hacettepe University (2016 - 2021)

Experience:

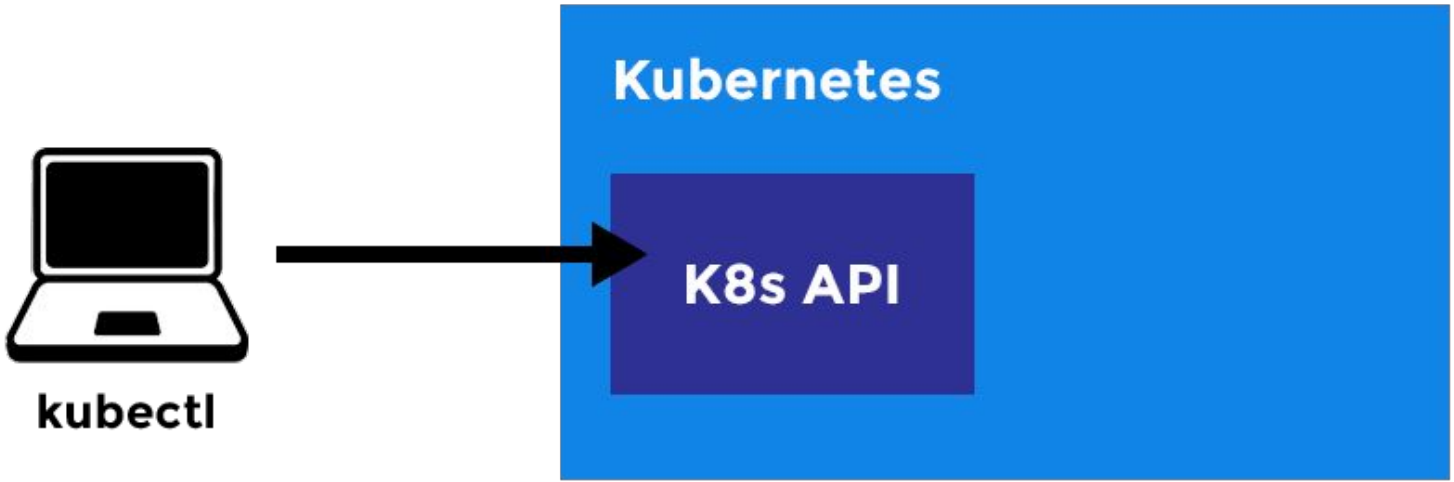
- Full Stack Developer, Tübitak Bilgem YTE (2020 - 2021)
- Platform Engineer, kloia (August 2021 -)
  - Projects: RealTyme Monitoring and E-Logo Automation
  - 2x Certified :)

Schedule

- Kubectl Overview
- Extending K8s
- Kubectl Plugin
- Hands on
- Krew - Plugin Manager

kubectl

kubectl is a client for the Kubernetes API. Kubernetes API is an HTTP REST API. Kubernetes is fully controlled through this API. This means that every Kubernetes process is exposed as an API endpoint and can be executed by an HTTP request to that endpoint. As a result, kubectl's main job is to perform HTTP requests to the Kubernetes API



Extending Kubernetes

- *kubectl* plugins
- API Extensions
- Custom Resources
- Scheduler Extensions
- Controllers
- Network Plugins
- Storage Plugins

Our Interest: Kubectl Plugin

- Users often interact with the Kubernetes API **using kubectl**. Kubectl plugins **extend the kubectl binary.** *They only affect the individual user's local environment, and so cannot enforce site-wide policies*

- Plugins extend kubectl with **new sub-commands,** *allowing for new and custom features* not included in the main distribution of kubectl..

Our Interest: Kubectl Plugin - File Type

- A plugin is a standalone **EXECUTABLE file**, whose name begins with ***kubectl-*** To install a plugin, move its executable file to anywhere **on your PATH**.

- kubectl provides a command **kubectl plugin list** that searches **your PATH** for valid plugin executables. Executing this command causes a traversal of all files in your PATH. *Any files that are executable, and begin with kubectl- will show up in the order in which they are present in your PATH in this command's output.* A warning will be included for any files beginning with kubectl- that are *not* executable. A warning will also be included for any valid plugin files that overlap each other's name.

Our Interest: Kubectl Plugin - Naming

```
# create a plugin
echo -e '#!/bin/bash\n\necho "My first command-line argument was $1"' > kubectl-foo-bar-baz
sudo chmod +x ./kubectl-foo-bar-baz

# "install" your plugin by moving it to a directory in your $PATH
sudo mv ./kubectl-foo-bar-baz /usr/local/bin

# check that kubectl recognizes your plugin
kubectl plugin list
```

If you run `kubectl foo bar baz arg1 --flag=value arg2`, kubectl's plugin mechanism will first try to find the plugin with the longest possible name, which in this case would be `kubectl-foo-bar-baz-arg1`. Upon not finding that plugin, kubectl then treats the last dash-separated value as an argument (`arg1` in this case), and attempts to find the next longest possible name, `kubectl-foo-bar-baz`. Upon having found a plugin with this name, kubectl then invokes that plugin, passing all args and flags after the plugin's name as arguments to the plugin process.

Hands on

Create **kubectl-ninfo** plugin

Commands

- **kubectl ninfo** - *Gets information of nodes*
- **kubectl ninfo arch** - *Gets architecture of nodes*

```
#!/bin/bash
kubectl get nodes -o jsonpath="{.items[*].status.nodeInfo.architecture}"
```

Create **kubectl-cns** plugin

Commands

- **kubectl cns** - *List the contexts*
- **kubectl cns c** - *List the contexts*
- **kubectl cns c docker-desktop** - *Change context*
- **kubectl cns ns** - *List namespaces*
- **kubectl cns ns kube-system** - *Change namespace on current-context*

Solution

Examples

Create **kubectl-minikube-snapshot** plugin

Commands

- **kubectl minikube snapshot save -f filename**- *Gets backup of resource configurations by using kube api server as yaml file (unique name with creationTime) and stores it in **/tmp** directory on the host.*
- **kubectl minikube snapshot list** - *Prints the backup file names from **/tmp** directory with columns (Name, Size, LastModifiedTime)*

Solution

Create **kubectl-create-pvc** plugin (Imperative)

Commands

- kubectl create pvc
  - Options:
    - --accessmode=ReadWriteOnce
    - --resources="cpu=100mi,storage=2Gi"
    - --storageclassname=slow
    - --output=yaml

Solution

What is Krew?

Krew is the plugin manager for kubectl command-line tool.

Krew helps you:

- discover kubectl plugins,
- install them on your machine,
- and keep the installed plugins up-to-date.

There are 160 kubectl plugins currently distributed on Krew.

Krew works across all major platforms, like macOS, Linux and Windows.

Krew also helps kubectl plugin developers: You can package and distribute your plugins on multiple platforms easily and makes them discoverable through a centralized plugin repository with Krew.