



LOGGING AND TROUBLESHOOTING

Logging and Troubleshooting

Using krew

We have been using the **kubectl** command throughout the course. The basic commands can be used together in a more complex manner extending what can be done. There are over seventy and growing plugins available to interact with Kubernetes objects and components.

At the time this course was written, plugins cannot overwrite existing **kubectl** commands, nor can it add sub-commands to existing commands. Writing new plugins should take into account the command line runtime package and a Go library for plugin authors.

As a plugin the declaration of options such as namespace or container to use must come after the command. For example:

```
$ kubectl sniff bigpod-abcd-123 -c mainapp -n accounting
```

Plugins can be distributed in many ways. The use of **krew** (the **kubectl** plugin manager) allows for cross-platform packaging and a helpful plugin index, which makes finding new plugins easy.

Install the software using steps available in [krew's GitHub repository](#). See the following command:

```
$ kubectl krew help
```

You can invoke **krew** through **kubectl**:

```
kubectl krew [command]...
```

Usage:

```
krew [command]
```

Table: Available Commands

COMMAND	DESCRIPTION
help	Help about any command
info	Show information about kubectl plugin
install	Install kubectl plugins
list	List installed kubectl plugins
search	Discover kubectl plugins
uninstall	Uninstall plugins
update	Update the local copy of the plugin index
upgrade	Upgrade installed plugins to newer versions
version	Show krew version and diagnostics

More information can be found in the Kubernetes documentation, ["Extend kubectl with plugins"](#).