
kubeconf Documentation

Release 0.0.2

Zach Sailer

Oct 21, 2019

Contents

1	Contents	3
1.1	Basic Usage	3
1.2	API documentation	4
1.3	Contributing	5
1.4	Indices and tables	6
	Index	7

Lightweight Python module for creating, manipulating, and editing kubeconfig files

Why not use or wrap `kubectl config`? `kubectl config` is great and writing a Python wrapper is a fine solution. However, `kubectl config` is quite limited in functionality. I wanted more control over my kubernetes config. `kubeconfig` gives me that control. It doesn't use `kubectl` at all. Rather, it reads, edits, and writes config files entirely on its own.

CHAPTER 1

Contents

- *Basic Usage*
- *API documentation*
- *Contributing*
- *Indices and tables*

1.1 Basic Usage

1. Import the KubeConf object and create an instance. If no `path` argument is given *kubeconf* will look for a config following kubectl's method. First, it will look a `$KUBECONFIG` environment variable pointing to a config file. Otherwise, it will look for a config file in the `~/ .kube/` directory.
2. `.open()` the file.
3. Make your changes.
4. `.close()` the file to write your changes to the file.

```
# Import KubeConf
from kubeconf import KubeConf

# Initialize your file.
k = KubeConf(path='path/to/config')

# Open the file.
k.open()

# Add a cluster
k.add_cluster(
    name='mycluster',
```

(continues on next page)

(continued from previous page)

```
server='...',
certificate_authority_data='...',
)

# Commit change to the file.
k.close()
```

1.2 API documentation

class kubecconf.KubeConf (**traits)

Base object that interacts with kubeconfig file.

add_cluster (name, server=None, certificate_authority_data=None, **attrs)

Add a cluster to config.

add_context (name, cluster_name=None, user_name=None, namespace_name=None, **attrs)

Add a context to config.

add_exec_to_user (name, env, command, args, **attrs)

Add an exec option to your user.

add_to_cluster (name, **attrs)

Add attributes to a cluster.

add_to_context (name, **attrs)

Add attributes to a context.

add_to_user (name, **attrs)

Add attributes to a user.

add_user (name, **attrs)

Add a user to config.

as_yaml ()

Show data as YAML

close ()

Commit the changes to the file and close it.

cluster_exists (name)

Check if a given cluster exists.

context_exists (name)

Check if a given context exists.

get_cluster (name)

Get cluster from kubeconfig.

get_clusters ()

Get all clusters in config.

get_context (name)

Get context from kubeconfig.

get_contexts ()

Get all contexts in config.

get_current_context ()

Get the current context found in kubeconfig.

get_user (*name*)
Get user from kubeconfig.

get_users ()
Get all users in config.

open (*create_if_not_found=True*)
Open a kube config file. If the file does not exist, it creates a new file.

print_clusters (*names=False*)
Print contexts.

print_contexts (*names=False*)
Print users

print_users (*names=False*)
Print users

remove_cluster (*name*)
Remove a cluster from kubeconfig.

remove_context (*name*)
Remove a context from kubeconfig.

remove_from_cluster (*name, *args*)
Remove attributes from a cluster.

remove_from_context (*name, *args*)
Remove attributes from a context.

remove_from_user (*name, *args*)
Remove attributes from a user.

remove_user (*name*)
Remove a user from kubeconfig.

set_current_context (*name*)
Set the current context in kubeconfig.

show ()
Print the contexts of the kubeconfig.

user_exists (*name*)
Check if a given user exists.

1.3 Contributing

Download and install this repo from source, and move into the base directory.

```
git clone https://github.com/Zsailer/kubecconf
cd kubecconf
```

If you use pipenv, you can install a developement version:

```
pipenv install --dev
```

Otherwise you can install a development version using pip

```
pip install -e .
```

1.4 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

A

`add_cluster()` (*kubeconf.KubeConf method*), 4
`add_context()` (*kubeconf.KubeConf method*), 4
`add_exec_to_user()` (*kubeconf.KubeConf method*), 4
`add_to_cluster()` (*kubeconf.KubeConf method*), 4
`add_to_context()` (*kubeconf.KubeConf method*), 4
`add_to_user()` (*kubeconf.KubeConf method*), 4
`add_user()` (*kubeconf.KubeConf method*), 4
`as_yaml()` (*kubeconf.KubeConf method*), 4

C

`close()` (*kubeconf.KubeConf method*), 4
`cluster_exists()` (*kubeconf.KubeConf method*), 4
`context_exists()` (*kubeconf.KubeConf method*), 4

G

`get_cluster()` (*kubeconf.KubeConf method*), 4
`get_clusters()` (*kubeconf.KubeConf method*), 4
`get_context()` (*kubeconf.KubeConf method*), 4
`get_contexts()` (*kubeconf.KubeConf method*), 4
`get_current_context()` (*kubeconf.KubeConf method*), 4
`get_user()` (*kubeconf.KubeConf method*), 4
`get_users()` (*kubeconf.KubeConf method*), 5

K

`KubeConf` (*class in kubeconf*), 4

O

`open()` (*kubeconf.KubeConf method*), 5

P

`print_clusters()` (*kubeconf.KubeConf method*), 5
`print_contexts()` (*kubeconf.KubeConf method*), 5
`print_users()` (*kubeconf.KubeConf method*), 5

R

`remove_cluster()` (*kubeconf.KubeConf method*), 5

`remove_context()` (*kubeconf.KubeConf method*), 5
`remove_from_cluster()` (*kubeconf.KubeConf method*), 5
`remove_from_context()` (*kubeconf.KubeConf method*), 5
`remove_from_user()` (*kubeconf.KubeConf method*), 5
`remove_user()` (*kubeconf.KubeConf method*), 5

S

`set_current_context()` (*kubeconf.KubeConf method*), 5
`show()` (*kubeconf.KubeConf method*), 5

U

`user_exists()` (*kubeconf.KubeConf method*), 5