

---

**neo-cli**

**Sep 19, 2021**



---

## Contents

---

<b>1</b>	<b>Overview</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Features . . . . .	3
1.3	Take the tour . . . . .	3
<b>2</b>	<b>Usage</b>	<b>5</b>
2.1	Authentication . . . . .	5
2.2	Creation . . . . .	5
2.3	List your stuffs . . . . .	5
2.4	Remove your stuffs . . . . .	6
2.5	Attach . . . . .	6
<b>3</b>	<b>Troubleshooting</b>	<b>7</b>
3.1	Unable to locate package python3-venv . . . . .	7
3.2	No command 'neo' found . . . . .	7
<b>4</b>	<b>Project Information</b>	<b>9</b>
4.1	Contributing . . . . .	9
4.2	License . . . . .	10
4.3	Changelog . . . . .	10
<b>5</b>	<b>How To's</b>	<b>13</b>
5.1	Deploy Kubernetes . . . . .	13
5.2	neo DSL . . . . .	15



Don't let all those templates and commands burden your head. Sit back, relax and let `neo-cli` asks what you need.

`neo-cli` lower the barrier of understanding complicated templates, memorizing long command and parameters to create your cloud infrastructure. `neo-cli` will interactively ask what you need.

In addition, `neo-cli` aims to support diverse cloud platforms in near future. So no matter what your cloud platforms is, `neo-cli` will always be your friend.



# CHAPTER 1

## Overview

## 1.1 Installation

```
$ pip install neo-cli
```

## 1.2 Features

- Support common OpenStack operation: creating vm, listing vm, stack, network, floating ip, and removing and updating stacks.
- Auto-login for previous account.
- Support attaching local standard input, output, and error streams to a running machine.
- Can be used as library for your OpenStack application.

## 1.3 Take the tour

### 1.3.1 List your virtual machine

```
$ neo ls vm
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
↪+-----+-----+-----+-----+
| ID      | Name    | Key Pair | Image    | Flavor  | RAM (GiB) | vCPU | ↪
↪Addresses          | Status  |
+=====+=====+=====+=====+=====+=====+=====+=====+
| 1bf4d720 | my-vm-1 | vm-key   | CentOS 7.5 | SS2.1   | 2        | 1    | ↪
↪network : default          | ACTIVE  |
```

(continues on next page)

(continued from previous page)

```

| | | | | | | |
↪fixed IP : 192.168.68.5 | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+
| 9ca3e7d10 | my-vm-2 | vm-key | CentOS 7.5 | SS2.1 | | 2 | 1 |
↪network : default | ACTIVE | | | | | |
| | | | | | | |
↪fixed IP : 192.168.68.12 | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+

```

Some info omitted.

### 1.3.2 Attach local terminal to running machine

```
$ neo attach vm c57a477b-84dc-4ae3-1234-5678
2020-04-02 01:22:34 INFO Check your key pairs
2020-04-02 01:22:38 INFO Done...
2020-04-02 01:22:38 INFO Check username
Username : centos
Last login: Wed Jul 31 04:52:07 2019 from 123.456.68.101
[centos@foo-vm ~]$
```

### 1.3.3 Creating stuffs

```
$ neo create
2020-04-02 01:25:57 ERROR Can't find neo.yml manifest file!
Do you want to generate neo.yml manifest?  [y/n]? y

Select Stack  :
- clusters
- networks
- instances
- others
Enter your choice :
```



You can list all NEO command with `help`

```
$ neo --help
```

## 2.1 Authentication

Use `neo login` to log in. `neo logout` to do the opposite.

## 2.2 Creation

```
$ neo create
```

NEO creates `neo.yml` for you if it doesn't find one. Then it will guide you through questions to do the right job for you.

It will ask you the 'stack' and 'template' you want to create. Then fill 'key-pairs' and 'network' configuration. The last step is to setup your 'vm' where you are asked to choose 'image name' and 'flavor'.

When you are sure with the configuration. Hit 'y/yes' to continue to deploy.

## 2.3 List your stuffs

To see all available commands to list your stuffs:

```
$ neo ls --help
```

Some of them are `stack`, `vm`, and `network`.

## 2.4 Remove your stuffs

```
$ neo rm
```

It will delete your stack, network and machine

### 2.4.1 Update

```
$ neo update
```

Use `update` to see your changes.

## 2.5 Attach

Attach local standard input, output, and error streams to a running stack or virtual machine

```
$ neo attach vm
```

`neo attach` will read `neo.yml` configuration automatically if you didn't pass the of your vm.

You can also specify your running vm id manually with

```
$ neo attach vm <your-vm-id>
```

### 3.1 Unable to locate package python3-venv

Try to check the python3 venv module name provided by your distro

```
$ apt-cache search python3 | grep venv
```

The results

```
python3-venv - pyvenv-3 binary for python3 (default python3 version)
python3.5-venv - Interactive high-level object-oriented language (pyvenv binary, ↵
↵version 3.5)
```

It might be differ on your machine. So please make sure you get the correct name.

### 3.2 No command 'neo' found

Make sure you virtual environments is activated



### 4.1 Contributing

We'd be happy for you to contribute to neo-cli.

To learn more about support questions, project organization, opening a new issue and submitting a pull request, see [BiznetGio Project Contribution Guide](#).

#### 4.1.1 First time setup

Please refer to installation from source installation-from-source guide.

#### 4.1.2 Running the tests

You can run the test with your own credentials

Run the basic test suite with:

```
$ pytest
```

You can add more parameter to get more details.

```
$ pytest --cov=neo -vv -s
```

If your test script get 'aborted' by the server. Try login manually with `neo login` before running test.

#### 4.1.3 Running test coverage

You can generate coverage report with:

```
$ coverage report -m
# or
$ coverage html
```

## 4.2 License

MIT License

Copyright (c) 2017 PT Biznet Gio Nusantara

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 4.3 Changelog

### 4.3.1 Unreleased

#### 4.3.2 0.9.3 (2021-09-19)

- Add region wjv2, jkt2, btn1
- Add vm cli to manage instances

#### 4.3.3 0.9.0 (2020-05-19)

- Add multiple region login
- Fix login session on windows

#### 4.3.4 0.8.2 (2020-03-31)

- Add partial windows support
- Fix incompatible git API changes

#### 4.3.5 0.8.0 (2019-11-04)

- Add the ability to specify heat version.
- New documentation page at <https://biznetgio.github.io/neo-cli/>.
- Reword Kubernetes size label.
- Fix logout process by removing neo .env file after logout.





## 5.1 Deploy Kubernetes

```
$ neo create kubernetes
```

### 5.1.1 Dashboard

#### Tunneling

```
$ neo attach -t 8001:127.0.0.1:8001
[k8s@k8s-test-controller-2hojdpb5a22a ~]$ kube-token
Name:      admin-user-token-qt8dr
Namespace: kube-system
Labels:    <none>
Annotations: kubernetes.io/service-account.name=admin-user
             kubernetes.io/service-account.uid=1dc769a0-4679-11e8-829f-fa163ebedac7

Type: kubernetes.io/service-account-token

Data
====
ca.crt:      1090 bytes
namespace:   11 bytes
token:       eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.
             eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2t
             eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2t
             XdWFIA49ckETvBSEA

[k8s@k8s-test-controller-2hojdpb5a22a ~]$ kubectl proxy
Starting to serve on 127.0.0.1:8001
```

open url <http://127.0.0.1:8001> and then login with your token access

### 5.1.2 Create simple user

In this guide, we will find out how to create a new user using Service Account mechanism of Kubernetes, grant this user admin permissions and log in to Dashboard using bearer token tied to this user.

Copy provided snippets to some `xxx.yaml` file and use `kubectl create -f xxx.yaml` to create them.

#### Create Service Account

We are creating Service Account with name `admin-user` in namespace `kube-system` first.

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kube-system
```

#### Create ClusterRoleBinding

In most cases after provisioning our cluster using `kops` or `kubeadm` or any other popular tool admin Role already exists in the cluster. We can use it and create only RoleBinding for our ServiceAccount.

**NOTE:** `apiVersion` of `ClusterRoleBinding` resource may differ between Kubernetes versions. Starting from `v1.8` it was promoted to `rbac.authorization.k8s.io/v1`.

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kube-system
```

#### Bearer Token

Now we need to find token we can use to log in. Execute following command:

```
kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep_
↪admin-user | awk '{print $1}')
```

It should print something like: ““bash Name: admin-user-token-6gl6l Namespace: kube-system Labels: Annotations: kubernetes.io/service-account.name=admin-user kubernetes.io/service-account.uid=b16afba9-dfec-11e7-bbb9-901b0e53

## 5.2 neo DSL

### 5.2.1 Structure

```
<category>:
  <name_entity>:
    parameters:
      <parameter_1_key> : <paramater_1_value>
      <parameter_2_key> : <paramater_2_value>
    template: <template>
```

### 5.2.2 Templates

#### Cluster

#### Kubernetes

parameters :

key	value	requirement	description
image	string	no	•
controller_flavor	string	no	•
master_flavor	string	yes	•
worker_flavor	string	yes	•
public_network	string	no	•
master_size	number	yes	•
worker_size	number	yes	•
private_network	string	yes	•

example :

```
clusters:
  neo-k8s:
    template: "kubernetes"
    parameters:
      master_flavor: SS2.1
      worker_flavor: SS2.1
      master_size: 1
      worker_size: 1
```

(continues on next page)

(continued from previous page)

```
deploy:
- clusters.neo-k8s
```

## Networks

### Private (Private network)

parameters :

key	value	requirement	description
gateway	string (example :192.168.0.1)	yes	•
cidr	string (example :192.168.0.0/24)	yes	•
dns	string (example :8.8.8.8)	no	•

## Instances

### vm (Virtual Machine)

parameters :

key	value	requirement	description
private_network	string	yes	•
key_name	string	yes	•
image	string	yes	•
flavor	string	yes	•
username	string	yes	•

example :

```
deploy:
- others.key-coba
- networks.neowork-coba
- instances.vm-coba
instances:
  vm-coba:
    parameters:
      flavor: SS2.1
      image: CentOS 7.3
```

(continues on next page)

(continued from previous page)

```

    key_name: key-coba
    private_network: neowork-coba
    username: ibnu
    template: vm
networks:
  neowork-coba:
    parameters:
      cidr: 192.168.3.0/24
      gateway: 192.168.3.1
    template: private
others:
  key-coba:
    template: key-pairs

```

## plesk (Plesk Bundle)

parameters :

key	value	requirement	default value	description
private_network	string	yes	.	.
key_name	string	yes	.	.
image	string	yes	Image-Plesk Bundle-1.0	.
flavor	string	yes	.	.
username	string	yes	.	.
email	string	yes	.	.
password	string	yes	.	.
activation_key	string	yes	.	.
floating_desc	string	no	PLSK	set description to floating ip
neo_type	{ "metadata": { "neo_type": }}	no	{ "metadata": { "neo_type": "PLSK"}}	add metadata to neo vm
package	enum (wordpress, business)	no	wordpress	add metadata to neo vm

## Others

## key\_pairs

parameters : no parameters