

---

# **ekscli Documentation**

***Release 0.1.0rc5***

**Charles Z**

**May 28, 2019**



---

## Contents:

---

<b>1</b>	<b>ekscli</b>	<b>1</b>
1.1	Prerequisites . . . . .	1
1.2	Quick Start . . . . .	1
1.3	Features . . . . .	3
1.4	Roadmap . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Stable release . . . . .	5
2.2	From sources . . . . .	5
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	11
4.4	Tips . . . . .	11
4.5	Deploying . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
6.1	0.1.0rc1 (2018-06-07) . . . . .	15
<b>7</b>	<b>Indices and tables</b>	<b>17</b>



A simple and flexible command-line tool for AWS EKS management

- Free software: MIT license
- Documentation: <https://ekscli.readthedocs.io>.

## 1.1 Prerequisites

- Available AWS credentials (configured as `boto3`)
- Heptio authenticator binary (Section To install heptio-authenticator-aws for Amazon EKS in [AWS EKS User Guide](#))
- [Optional] kubectl (1.10 and later) for kubernetes cluster operations

## 1.2 Quick Start

### 1.2.1 Installation

As easy as the standard python way by using `pip`.

```
$ pip install ekscli
```

Optionally, after installation, command-completion can be achieved with:

```
$ eval "$(_EKS_COMPLETE=source eks)"
```

## 1.2.2 Use ECKCLI

Note: AWS fees will be charged in your account for the AWS resources created by ekscli.

The simplest way to create a cluster by running `ekscli` with almost everything default.

This will create an EKS cluster including the control plane (managed master by AWS), a node group and a `kubectl` configuration file (`KUBECONFIG` or `$HOME/.kube/config`).

```
$ eks create cluster --name=dev

# EKS cluster name can be set as an environment variable
$ export EKS_CLUSTER_NAME=dev
$ eks create cluster
```

To create the EKS cluster's control plane (master) only:

```
$ eks create cluster --name=dev --cp-only
```

To create the EKS cluster's control plane (master) with existing subnets of a VPC, a predefined IAM role, an existing EC2 KeyPair etc.:

```
$ eks create cluster --name=dev --cp-only \
  --subnets=subnet-1234567,subnet-abcdef1 \
  --cp-role eks-default-role \
  --region us-west-2 \
  --kubconfig ./dev.conf \
  --heptio-auth /tmp/heptio-auth-aws \
  --keyname dev \
  --node-sg-ingress port=22,cidr=10.0.0.0/8 \
  --tags Env=dev,Project=eks-poc
```

The simplest way to create a node group

```
$ eks create node-group --name=dev --node-name=workers
```

To create a node group with more options

```
$ eks create node-group --name=dev --node-name=another \
  --node-role=eks-worker-s3 \
  --node-subnets=subnet-1234567 \
  --node-min=1 \
  --node-max=10 \
  --node-sg-ingress port=22,cidr=10.0.0.0/8 \
  --node-sg-ingress protocol=tcp,from=8080,to=8088,cidr=0.0.0.0/0 \
  --region us-west-2 \
  --kubconfig ./dev.conf \
  --heptio-auth /tmp/heptio-auth-aws \
  --keyname dev \
  --tags Env=dev,Project=eks-poc
```

To help bootstrapping kubelet agent

```
# on EC2 worker instances, after copying kubelet, cni, heptio-aws-authenticator
↳ executables
$ eks bootstrap -o node-labels=gpu=enable,role=node \
  -o feature-gates=RotateKubeletServerCertificate=true,CRIContainerLogRotation=true
$ systemctl daemon-reload
$ systemctl enable kubelet.service
```

To display files created by ekscli bootstrap locally rather than on EC2 instances

```
# on local machine
$ eks bootstrap --dry-run -n poc -r us-east-1 -m 32 -i 127.0.0.1 \
  -o node-labels=gpu=enable,role=node \
  -o feature-gates=RotateKubeletServerCertificate=true,CRIContainerLogRotation=true
```

To use ekscli bootstrap as oneshot systemd unit

```
[Unit]
Description=Configures Kubelet for EKS worker nodes
Before=kubelet.service

[Service]
Type=oneshot
ExecStart=/usr/local/bin/ekscli bootstrap
RemainAfterExit=true

[Install]
WantedBy=multi-user.target
```

## 1.3 Features

- Simple and concise command line interface
- Flexible configuration
- Plain vanilla EKS cluster without unrequired resources running Kubernetes clusters
- EKS resources managed by AWS CloudFormation
- Command line auto-completion supported for Bash and Zsh
- Prepare necessary configuration for kubelet with self cluster discovery and additional options on worker nodes

## 1.4 Roadmap

- Output cluster information to different formats: yaml, json
- Update the cluster and node groups
- Create from templatable configuration files





### 2.1 Stable release

To install ekscli, run this command in your terminal:

```
$ pip install ekscli
```

This is the preferred method to install ekscli, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

### 2.2 From sources

The sources for ekscli can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/cakab/ekscli
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/cakab/ekscli/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```



## CHAPTER 3

---

### Usage

---

To use ekscli in a project:

```
import ekscli
```



Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

## 4.1 Types of Contributions

### 4.1.1 Report Bugs

Report bugs at <https://github.com/cakab/ekscli/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

### 4.1.4 Write Documentation

ekscli could always use more documentation, whether as part of the official ekscli docs, in docstrings, or even on the web in blog posts, articles, and such.

### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/cakab/ekscli/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *ekscli* for local development.

1. Fork the *ekscli* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/ekscli.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv ekscli
$ cd ekscli/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass tests, including testing other Python versions with tox:

```
$ python setup.py py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check [https://travis-ci.org/cakab/ekscli/pull\\_requests](https://travis-ci.org/cakab/ekscli/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_ekscli
```

## 4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.





### 5.1 Development Lead

- Charles Z <[charles.cakab@gmail.com](mailto:charles.cakab@gmail.com)>

### 5.2 Contributors

None yet. Why not be the first?



#### 6.1 0.1.0rc1 (2018-06-07)

- First release on PyPI.



## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`