crane Documentation

Release 0.6.3

Globo.com

January 19, 2017

Contents

crane is a command line for service providers/administrators on tsuru.

There are several ways to install **crane**:

- *Downloading binaries (Mac OS X and Linux)*
- *Using homebrew (Mac OS X only)*
- *Using the PPA (Ubuntu only)*
- *Using AUR (ArchLinux only)*
- *Build from source (Linux and Mac OS X)*

CHAPTER 1

Downloading binaries (Mac OS X and Linux)

We provide pre-built binaries for OS X and Linux, only for the amd64 architecture. You can download these binaries directly from the releases page:

- **crane**: https://github.com/tsuru/crane/releases

CHAPTER 2

Using homebrew (Mac OS X only)

If you use Mac OS X and homebrew, you may use a custom tap to install **crane**. First you need to add the tap:

```
$ brew tap tsuru/homebrew-tsuru
```

Now you can install **crane**:

```
$ brew install crane
```

Whenever a new version of **crane** is out, you can just run:

```
$ brew update
$ brew upgrade crane
```

For more details on taps, check homebrew documentation.

Note: **crane** requires Go 1.4. Make sure you have the last version of Go installed in your system.

CHAPTER 3

Using the PPA (Ubuntu only)

Ubuntu users can install tsuru clients using `apt-get` and the tsuru PPA. You'll need to add the PPA repository locally and run an `apt-get update`:

```
$ sudo apt-add-repository ppa:tsuru/ppa
$ sudo apt-get update
```

Now you can install **crane** clients:

```
$ sudo apt-get install crane
```

CHAPTER 4

Using AUR (ArchLinux only)

Archlinux users can build and install tsuru admin from AUR repository, Is needed to have installed yaourt program.

You can run:

```
$ yaourt -S tsuru
```

CHAPTER 5

Build from source (Linux and Mac OS X)

Note: If you're feeling adventurous, you can try it on other systems, like FreeBSD, OpenBSD or even Windows. Please let us know about your progress!

tsuru admin source is written in Go, so before installing tsuru from source, please make sure you have installed and configured Go.

With Go installed and configured, you can use `go get` to install **crane**:

```
$ go get github.com/tsuru/crane
```

After installing, you must set the target with the tsuru server URL, something like:

CHAPTER 6

Managing remote tsuru server endpoints

The target is the **tsuru** server to which all operations will be directed to.

```
$ crane target-add <label> <address> [--set-current|-s]
$ crane target-list
$ crane target-set <label>
$ crane target-remove <label>
```

With this set of commands you are be able to add a new labeled target, set a target for usage, list the added targets and remove a target, respectively.

CHAPTER 7

Check current version

To see the current version of **crane** you should use the *version* command:

```
$ crane version
crane version 0.6.3.
```

CHAPTER 8

Authentication

8.1 login

```
$ crane login [<email>]
```

Login will ask for the password and check if the user is successfully authenticated. If so, the token generated by the **tsuru** server will be stored in ${HOME}/.tsuru_token.

All crane actions require the user to be authenticated (except *login* and *version*).

8.2 logout

```
$ crane logout
```

Logout will delete the token file and terminate the session within tsuru server.

CHAPTER 9

Create an empty manifest file

Usage:

```
$ crane template
```

Template will create a file named “manifest.yaml” with the following content:

```
id: servicename
username: username_to_auth
password: ****************
team: team_responsible_to_provide_service
endpoint:
  production: production-endpoint.com
```

Change it at will to configure your service. Id is the id of your service, it must be unique.You must provide a production endpoint that will be invoked by tsuru when application developers ask for new instances and are binding their apps to their instances.

You should have a role with **service.create** permission to be able to create a new service in tsuru.

For more details, see the text “Services API Workflow”: http://tsuru.rtfd.org/services-api-workflow.

CHAPTER 10

Create a new service

Usage:

```
$ crane create <manifest-file.yaml>
```

Create will create a new service with information present in the manifest file. Here is an example of usage:

```
$ cat /home/gopher/projects/mysqlapi/manifest.yaml
id: mysqlapi
endpoint:
  production: https://mysqlapi.com:7777
$ crane create /home/gopher/projects/mysqlapi/manifest.yaml
success
```

You can use "crane template" to generate a template. Both id and production endpoint are required fields.

When creating a new service, crane will add all user's teams as administrator teams of the service.

CHAPTER 11

Update a service

Usage:

```
$ crane update <manifest-file.yaml>
```

Update will update a service using a manifest file. Currently, it's only possible to edit an endpoint, or add new endpoints. You need to be an administrator of the team to perform an update.

CHAPTER 12

Remove a service

Usage:

```
$ crane remove <service-id>
```

Remove will remove a service from crane server. You need to be an administrator of the team to remove it.

CHAPTER 13

List services that you manage

Usage:

```
$ crane list
```

`crane list` will list all services that you manage, and the instances of each service, created by application developers.

Update service's documentation

Usage:

```
$ crane doc-add <service-id> <doc-file.txt>
```

doc-add will update service's doc. Example of usage:

```
$ cat doc.txt
mysqlapi

This service is used for mysql connections.

Once bound, you will be able to use the following environment variables:

        - MYSQL_HOST: host of MySQL server
        - MYSQL_PORT: port of MySQL instance
        - MYSQL_DATABASE_NAME: name of the database
        - MYSQL_USER: MySQL user for connections
        - MYSQL_PASSWORD: MySQL password for connections
$ crane doc-add mysqlapi doc.txt
Documentation for 'mysqlapi' successfully updated.
```

Warning: You need to be an administrator of the service to update its docs.

CHAPTER 15

Retrieve service's documentation

Usage:

```
$ crane doc-get <service-id>
```

doc-get will retrieve the current documentation of the service.