

---

# dugaire Documentation

*Release develop*

Dec 06, 2020



# CONTENTS

<b>1</b>	<b>Install (on Linux)</b>	<b>3</b>
1.1	Using pip (recommended) . . . . .	3
1.2	From source . . . . .	3
<b>2</b>	<b>Enable autocomplete</b>	<b>5</b>
2.1	bash . . . . .	5
2.2	zsh . . . . .	5
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Features</b>	<b>9</b>
4.1	Base images . . . . .	9
4.2	Package/Dependency Managers . . . . .	9
4.3	Applications . . . . .	10
<b>5</b>	<b>Useful Docker commands</b>	<b>13</b>
5.1	List images created with dugaire . . . . .	13
5.2	Delete all images created with dugaire . . . . .	13
<b>6</b>	<b>Known issues</b>	<b>15</b>
6.1	RuntimeError: Python 3 was configured to use ASCII as encoding for the environment . . . . .	15
<b>7</b>	<b>License</b>	<b>17</b>
7.1	Product license . . . . .	17
7.2	FOSSA scan overview . . . . .	17
7.3	FOSSA Live Project report . . . . .	17



Quickly build custom Docker images for local development without having to write Dockerfiles.

Fig. 1: dugaire

### *Examples*

Install vim and curl using apt-get.

```
docker run -it --rm $(dugaire build --apt=vim,curl)
```

Install vim, python3, pip3 using apt-get and install ansible using pip3.

```
docker run -it --rm $(dugaire build --apt=vim,python3-pip --pip3=ansible)
```

Install kubectl binary version v1.15.0 (use --with-kubectl=latest to install the latest version).

```
docker run -it --rm $(dugaire build --apt=vim --with-kubectl=1.15.0)
```

Do not build the image and just print the Dockerfile:

```
dugaire build --apt=vim,curl --with-kubectl=latest --output=dockerfile --dry-run
```



## INSTALL (ON LINUX)

### 1.1 Using pip (recommended)

```
pip install dugaire
```

### 1.2 From source

Clone this repository.

```
git clone https://github.com/tadeugr/dugaire.git
```

cd to its directory.

```
cd dugaire
```

Install it (requires pip).

```
make install
```

You should have the command available.

```
dugaire build --help
```





## ENABLE AUTOCOMPLETE

To enable autocomplete for your current terminal session, run:

```
eval "$(_DUGAIRE_COMPLETE=source dugaire)"
```

*Follow the instructions bellow to permanently enable autocomplete.*

### 2.1 bash

```
echo 'eval "$(_DUGAIRE_COMPLETE=source dugaire) "' >> ~/.bashrc
```

### 2.2 zsh

```
echo 'eval "$(_DUGAIRE_COMPLETE=source dugaire) "' >> ~/.zshrc
```



**USAGE**

Usage: `dugaire build [OPTIONS]`

Build Docker images with custom packages.

Examples:

Build an image and install vim and curl using apt-get.

```
$ dugaire build --apt=vim,curl
```

Build an image and install python3 using apt-get and ansible using pip3.

```
$ dugaire build --apt=python3-pip --pip3=ansible
```

Build an image and install the latest version of kubectl.

```
$ dugaire build --with-kubectl=latest
```

Options:

<code>--from &lt;name:tag&gt;</code>	Base image (used in Dockerfile FROM). Example: <code>--from=ubuntu:20.04</code> [default: <code>ubuntu:18.04</code> ; required]
<code>--name &lt;name:tag&gt;</code>	Image name. Example: <code>--name="myimage:0.0.1"</code> [default: random]
<code>--apt &lt;pkg01 pkg01,pkg02&gt;</code>	Comma separated list of packages (no blank space) to install using apt-get install. Requires a base image with apt-get. Example: <code>--apt=curl,vim</code>
<code>--pip3 &lt;pkg01 pkg01,pkg02&gt;</code>	Comma separated list of packages (no blank space) to install using pip3 install. Example: <code>--pip3=ansible,jinja2==2.11.2</code>
<code>--with-azurecli latest</code>	Install Azure CLI. Examples: <code>--with-azurecli=latest</code> / For older versions, use <code>pip3: --pip3="azure-cli==2.2.0"</code>
<code>--with-kubectl &lt;latest semantic versioning&gt;</code>	Install kubectl. Examples: <code>--with-kubectl=latest</code> / <code>--with-kubectl=1.17.0</code>
<code>--with-velero &lt;latest semantic versioning&gt;</code>	

(continues on next page)

(continued from previous page)

	Install velero. Examples: --with-velero=latest / --with-velero=1.5.2
--force	Ignore Docker cache and build from scratch. [default: False]
--dry-run	Do not build image. [default: False]
--output [image.id image.id.short image.name dockerfile]	Command output options. [default: image.id.short]
--help	Show this message and exit.

## FEATURES

### 4.1 Base images

Distro	Tested with
ubuntu	ubuntu:18.04 ubuntu:20.04

*You may use base images that were built from the tested images.*

### 4.2 Package/Dependency Managers

#### 4.2.1 apt-get

You can install any package available in the distro repository using `apt`.

Use a comma separated (no blank space) list of packages you want to install. Examples:

```
dugaire build --apt=wget,iputils-ping
```

#### 4.2.2 pip3

**WARNING** When using `pip3`, `dugaire` will automatically use `--apt=python3-pip`.

You can install any PyPI package using `pip3`.

To install the latest versions, use a comma separated (no blank space) list of packages. Examples:

```
dugaire build --pip3=urllib3,Jinja2
```

To install specific versions, use a comma separated (no blank space) list of packages between double quotes, each package name must be followed by `==<semantic versioning>`. Examples:

```
dugaire build --pip3="urllib3==1.26.2,Jinja2==2.11.2"
```

## 4.3 Applications

### 4.3.1 azure-cli

Install Azure Command Line Interface.

#### Requirements

All requirements are solved **automatically** by dugaire:

- Azure CLI latest version: `curl ca-certificates`
- Azure CLI specific versions: `gcc python3-pip python3-dev`

#### Options

`--with-azurecli=latest` to install the latest version.

`--pip3="azure-cli==2.2.0"` (*example*) to install specific versions.

See all versions available [here](#).

#### Examples

*Covered by automated tests, you may install any package version though.*

```
dugaire build --with-azurecli=latest
dugaire build --from=ubuntu:20.04 --pip3="azure-cli==2.14.2"
dugaire build --from=ubuntu:18.04 --pip3="azure-cli==2.13.0"
```

### 4.3.2 kubectl

Install kubectl.

#### Requirements

All requirements are solved **automatically** by dugaire:

- `curl ca-certificates`

#### Options

`--with-kubectl=latest` to install the latest version.

`--with-kubectl=1.17.0` (*example*) to install specific versions.

See all versions available [here](#).

## Examples

*Covered by automated tests, you may install any package version though.*

```
dugaire build --from=ubuntu:20.04 --with-kubect1=latest
dugaire build --from=ubuntu:20.04 --with-kubect1=1.18.0
dugaire build --from=ubuntu:20.04 --with-kubect1=1.17.0
dugaire build --from=ubuntu:18.04 --with-kubect1=1.16.0
dugaire build --from=ubuntu:18.04 --with-kubect1=1.15.0
```

### 4.3.3 velero

Install velero.

## Requirements

--with-velero requires --with-kubect1.

## Options

--with-velero=latest to install the latest version.

--with-velero=1.5.2 (*example*) to install specific versions.

See all versions available [here](#).

## Examples

*Covered by automated tests, you may install any package version though.*

```
dugaire build --from=ubuntu:20.04 --with-kubect1=latest --with-velero=latest
dugaire build --from=ubuntu:20.04 --with-kubect1=1.17.0 --with-velero=1.5.2
```





## USEFUL DOCKER COMMANDS

### 5.1 List images created with dugaire

```
docker images -f label='builtwith=dugaire'
```

### 5.2 Delete all images created with dugaire

```
docker rmi -f $(docker images -aq -f label='builtwith=dugaire')
```



## KNOWN ISSUES

### 6.1 RuntimeError: Python 3 was configured to use ASCII as encoding for the environment

If you get an error like this one:

```
RuntimeError: Click will abort further execution because Python 3 was configured to
↳ use ASCII as encoding for the environment. Consult https://click.palletsprojects.
↳ com/python3/ for mitigation steps.
```

It is because `dugaire` uses Python3 and `Click`, and according to Click “in Python 3, the encoding detection is done in the interpreter, and on Linux and certain other operating systems, its encoding handling is problematic”. [Read more](#).

#### 6.1.1 Solution

Setup your locale correctly, for example if you want to use `en_US.UTF-8`, run:

```
apt update && apt-get -y install locales
locale-gen --purge en_US.UTF-8

export LC_ALL="en_US.UTF-8"
export LC_CTYPE="en_US.UTF-8"

# Alternatively you can run:
#sudo dpkg-reconfigure locales
```

Then you should be able to run `dugaire`.



## 7.1 Product license

Apache License Version 2.0, January 2004. [Read more.](#)

## 7.2 FOSSA scan overview

## 7.3 FOSSA Live Project report

The report is available [here](#)