

App Commands

create app	<code>dokku apps:create <name></code>
clone some app	<code>dokku apps:clone <name></code>
destroy app	<code>dokku apps:destroy <name></code>
list existing apps	<code>dokku apps:list</code>
rename app	<code>dokku apps:rename <name></code>
get report for app	<code>dokku apps:report <name></code>

Process management

list process inside container	<code>dokku ps:report <name></code>
rebuild container	<code>dokku ps:rebuild <name></code>
restart process inside container	<code>dokku ps:restart <name></code>
scale process inside container	<code>ps:scale <name> <proc>=<count> [<proc>=<count>]</code>
rename app	<code>dokku apps:rename <name></code>
get report for app	<code>dokku apps:report <name></code>

SCALING

Dokku allows you to run multiple process types at different container counts. For example, if you had an app that contained 1 web app listener and 1 background job processor, dokku can, spin up 1 container for each process type defined in the Procfile. By default, dokku will only start a single web process (if defined.) However, if you wanted, for example, 2 job processors running simultaneously, you can modify this behavior in one of the following ways.

proxy NGINX and ports configuration

enable proxy for app	<code>dokku proxy:enable <app></code>
disable proxy	<code>dokku proxy:disable <app></code>
show proxy status for app	<code>dokku proxy:report <app></code>
list proxy ports	<code>dokku proxy:ports <app></code>
add proxy port	<code>dokku proxy:ports-add <app> <scheme http>:<host-port 80>:<container-port 500-0></code>
remove proxy port	<code>dokku proxy:ports-remove <app> <scheme http>:<host-port 80>:<container-port 5000></code>
clear ports	<code>dokku proxy:ports-clear <app></code>

Dokku will extract all tcp ports exposed using the EXPOSE

Postgres integration

create db	<code>dokku postgres:create <name></code>
link db to app	<code>dokku postgres:link <db-name> <app-name></code>
list db's	<code>dokku postgres:list</code>
connect via psql	<code>dokku postgres:connect <name></code>
expose port for external connection	<code>dokku postgres:expose <name></code>
export dump	<code>dokku postgres:export <name> > <file></code>
destroy db	<code>dokku postgres:destroy <name></code>
get info for db	<code>dokku postgres:info <name></code>

after link your db to your app, the path for you db will be accesible through DATABASE_URL env var

more commands here

[postgres dokku](#)

you need install first postgres plugin with:

```
dokku plugin:install https://github.com/dokku/dokku-postgres.git postgres
```

other commands

upload public key	<code>cat /home/<user>/<public-key>.pub ssh <user>@<ip> "sudo sshcommand acl-add dokku <key-name>"</code>
execute command inside docker container	<code>dokku --rm run <name> <command:pyt-hon node npm > [args]</code>

useful links

dokku plugins	link github
CD gitlab	gitlab-ci-git-push
useful dokku ports plugin	dokku ports plugin

```
FROM ubuntu:14.04
EXPOSE 1234
RUN python -m SimpleHTTPServer 1234
```

Logs,info and ENV

```
get report for app dokku apps:report [<app>]
```

```
get logs dokku logs <name>
```

```
set env variable dokku config:set [--no-restart] <name> <VARIABLE>=<VALUE>
```

```
remove env variable dokku config:unset [--no-restart] <name> <VARIABLE>=<VALUE>
```

```
get DB info dokku postgres:info <name>
```

```
get proxy/-ports info dokku proxy:ports <name>
```

```
know proxy is enable dokku proxy:report <name>
```

```
get NGINX errors dokku nginx:error-logs <name>
```



By **Carlos L** (jejete)
cheatography.com/jejete/

Published 4th October, 2019.
Last updated 4th October, 2019.
Page 1 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopad.com>