


Heroku Deployment Platform



To Scale a Nodejs App on Heroku


2019.11.5 Winter

Sign Up firstly!

 **HEROKU**

Jump to Favorites, Apps, Pipelines, Spaces...



 Personal ▾

New ▾

You don't have any apps yet

Every app and pipeline you create or become a collaborator on will appear here

Create new app

Looking for help getting started?

Get started by reading one of our language guides in the Dev Center

Choose a language guide... ▾

Blogs Careers Documentation

Terms of Service Privacy Cookies © 2019 Salesforce.com

Support

Pre-job (Apps Definition)

App consist of your source code and a description of any dependencies.

Dependency mechanisms vary across languages:

- gemfile in Ruby
- requirements.txt in Python
- package.json in Node.js
- pom.xml in Java

The source code for your app, together with the dependency file, should provide enough information for the Heroku platform to build your app, to produce something that can be executed.

Pre-job (to Prepare for Deployment)

1. To call API is an Alternative way.

\$curl

<https://devcenter.heroku.com/articles/build-and-release-using-the-api>

<https://devcenter.heroku.com/articles/platform-api-quickstart>

2. Heroku Client (sample)

<https://github.com/heroku/node-heroku-client>

3. To use Git as deploy way (we are using this solution)

\$git push Heroku master

Git is a powerful, distributed version control system that many developers use to manage and version source code. The Heroku platform uses Git as the primary means for deploying applications (there are other ways to transport your source code to Heroku, including via an API).

When you create an application on Heroku, it associates a new Git remote, typically named heroku, with the local Git repository for your application.

As a result, deploying code is just the familiar git push, but to the heroku remote instead.

Using Git to distribute App in Heroku

Git is a powerful, distributed version control system that many developers use to manage and version source code. The Heroku platform uses Git as the primary means for deploying applications (there are other ways to transport source code to Heroku, including via an API).

When you create an application on Heroku, it associates a new Git remote, typically named heroku, with the local Git repository for your application.

As a result, deploying code is just the familiar git push, but to the heroku remote instead.

Tips to set Heroku up !

- Install Heroku Package & cli install it
- Push Node Apps codebase in Git Repo (it can push from local git alternatively)
- Add Heroku Git Remote
- Add Procfile as Route (filename shall without file extension) in Apps Root Directory
- Listen on Port: 5000 (modify code in one line)
- Deploy Node App in Heroku platform

Heroku CLI

1. Install Heroku

<https://devcenter.heroku.com/articles/getting-started-with-nodejs#set-up>

```
$cd <app>
```

```
$brew install heroku/brew/heroku
```

2. Setup

```
$ heroku login
```

```
heroku: Press any key to open up the browser to login or q to exit: Opening  
browser to https://cli-auth.heroku.com/auth/browser/3e08a156-b445-412c-  
9fe9-041eba633149 Logging in... done Logged in as XXX@gmail.com
```

```
$ node --version
```

```
v10.16.3
```

```
$ npm --version
```

```
6.9.0
```

```
$ git --version
```

```
git version 2.20.1 (Apple Git-117)
```

Heroku CLI

3. Init Heroku

```
$ heroku create
```

```
Creating app... done, 🍄 tranquil-depths-77520
```

```
https://tranquil-depths-77520.herokuapp.com/ |
```

```
https://git.heroku.com/tranquil-depths-77520.git
```

4. Git Pushing

```
$ git push heroku master
```

```
Enumerating objects: 4, done.
```

```
Counting objects: 100% (4/4), done.
```

```
Delta compression using up to 4 threads
```

```
Compressing objects: 100% (2/2), done.
```

```
Writing objects: 100% (4/4), 1004 bytes | 1004.00 KiB/s, done. Total 4 (delta 0),  
reused 4 (delta 0) remote: Compressing source files... done.
```

```
remote: Building source:
```


Heroku CLI

5. Distribute & Deploy

```
$ heroku ps:scale web=1
```

```
>>> Scaling dynos... ! ▶ Couldn't find that process type (web). $  
heroku open
```

then Heroku is opening

at url => <https://tranquil-depths-77520.herokuapp.com>

6. View Logs

```
$ heroku logs --tail
```

Profile plays a role as Route

- The Procfile is always a simple text file that is named Procfile without a file extension.

For example, Procfile.txt is not valid.

- The Procfile must live in your app's root directory. It does not function if placed anywhere else.

//Define a Procfile

```
web: node index.js
```

```
/* This declares a single process type, web, and the command needed to run it. */
```

The name web is important here.

It declares that this process type will be attached to the

HTTP routing stack of Heroku, //and receive web traffic when deployed.

/ rocfiles can contain additional process types. For example, you might declare one for a background worker process that processes items off of a queue.

Prepare App (to edit code)

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('hi! node');
});

server.listen(port, hostname, ()=>{
  console.log(`server is running at http://${hostname}:${port}/`);
})
```

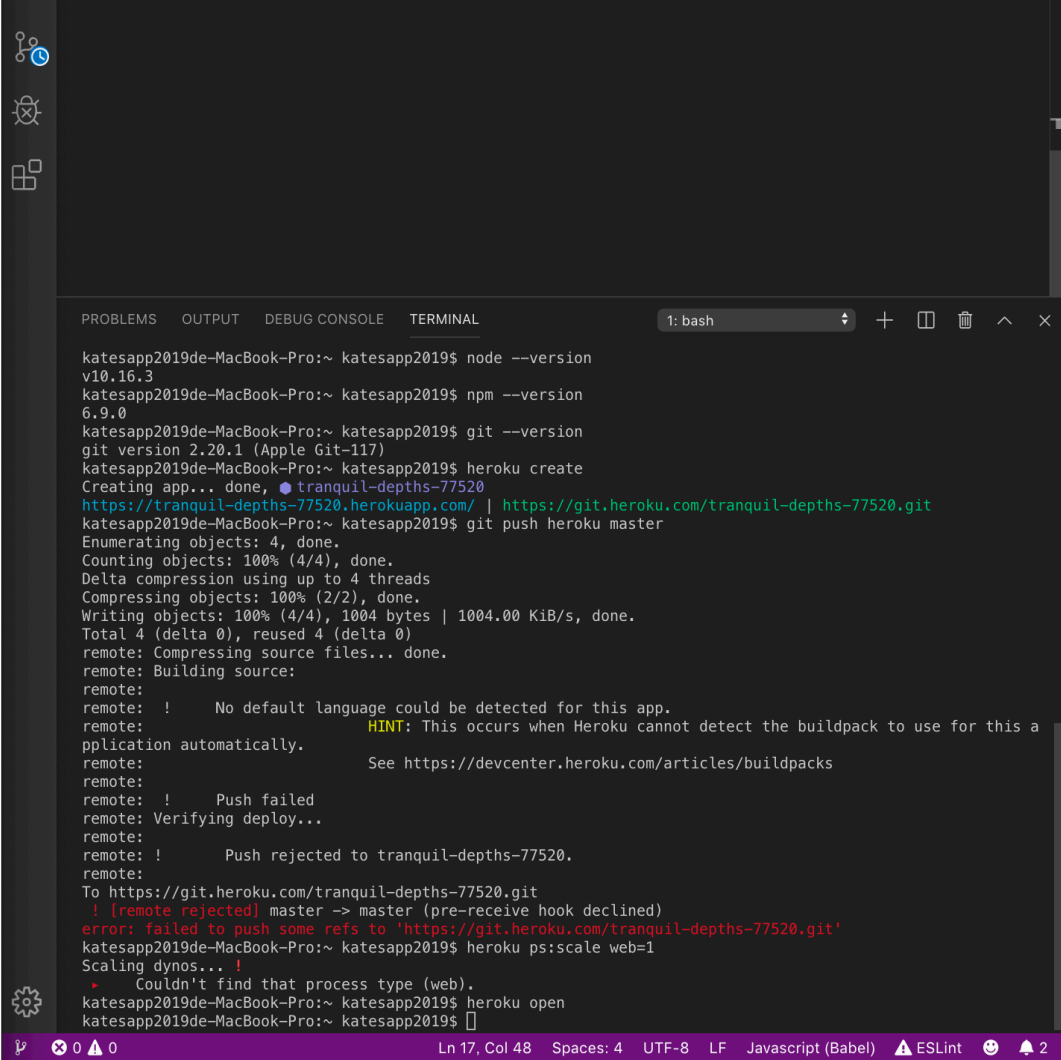
modify code as below

```
//const port = 3000;
const port = process.env.PORT || 5000;
```

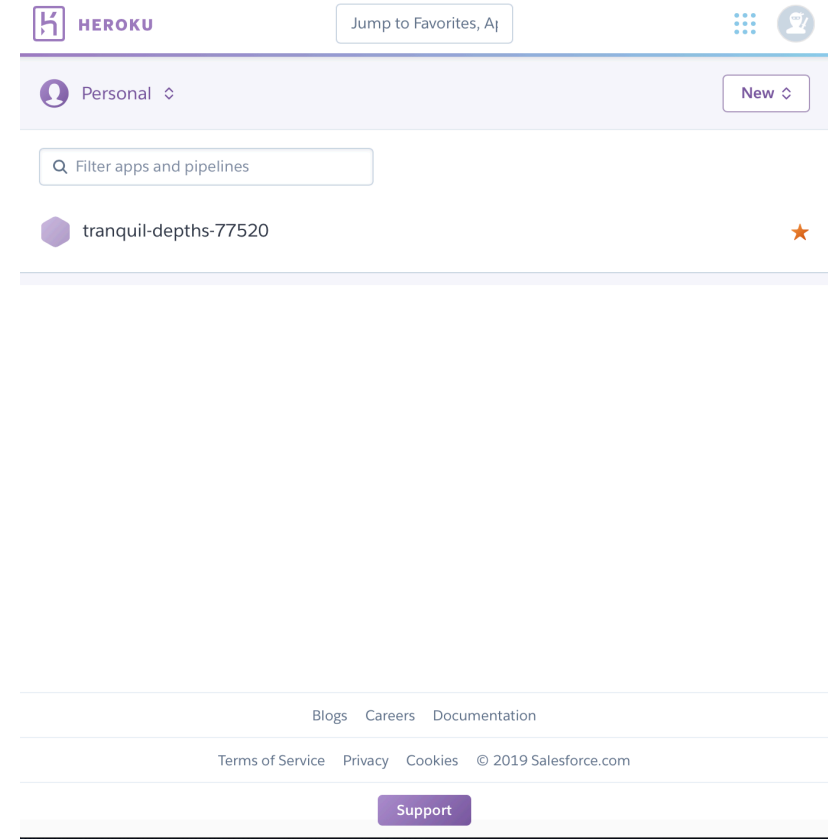
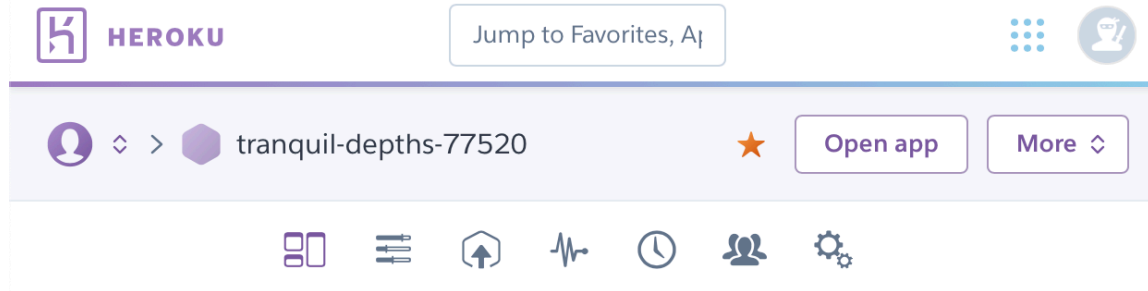
Deploy Node/Express App in Heroku

Heroku | Welcome to your new app!

Refer to the [documentation](#) if you need help deploying.



```
katesapp2019de-MacBook-Pro:~ katesapp2019$ node --version
v10.16.3
katesapp2019de-MacBook-Pro:~ katesapp2019$ npm --version
6.9.0
katesapp2019de-MacBook-Pro:~ katesapp2019$ git --version
git version 2.20.1 (Apple Git-117)
katesapp2019de-MacBook-Pro:~ katesapp2019$ heroku create
Creating app... done, ● tranquil-depths-77520
https://tranquil-depths-77520.herokuapp.com/ | https://git.heroku.com/tranquil-depths-77520.git
katesapp2019de-MacBook-Pro:~ katesapp2019$ git push heroku master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 1004 bytes | 1004.00 KiB/s, done.
Total 4 (delta 0), reused 4 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: !    No default language could be detected for this app.
remote:                                     HINT: This occurs when Heroku cannot detect the buildpack to use for this a
pplication automatically.
remote:                                     See https://devcenter.heroku.com/articles/buildpacks
remote:
remote: !    Push failed
remote: Verifying deploy...
remote:
remote: !    Push rejected to tranquil-depths-77520.
remote:
To https://git.heroku.com/tranquil-depths-77520.git
! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'https://git.heroku.com/tranquil-depths-77520.git'
katesapp2019de-MacBook-Pro:~ katesapp2019$ heroku ps:scale web=1
Scaling dynos... !
  • Couldn't find that process type (web).
katesapp2019de-MacBook-Pro:~ katesapp2019$ heroku open
katesapp2019de-MacBook-Pro:~ katesapp2019$
```



Heroku Apps Dashboard

Domains

Add your custom domains here then [point your DNS to Heroku](#).

Your app can be found at <https://tranquil-depths-77520.herokuapp.com/>

Add domain

🔍 Filter domains

Custom domains will appear here

Custom domains allow you to access your app via one or more non-Heroku domain names (for example, `www.yourcustomdomain.com`)

Info

Region	 United States
Stack	heroku-18
Framework	No framework detected
Slug Size	No slug detected
Heroku Git URL	<code>https://git.heroku.com/tranquil-depths-77520.git</code>

Heroku Deploy Info & Domain

Buildpacks

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment. [Find new buildpacks on Heroku Elements](#)

Add buildpack

Buildpacks will appear here

Buildpacks are used to install dependencies for your app and configure your environment.

Add Buildpack

Enter Buildpack URL

heroku/nodejs

Or select from our officially supported buildpacks



nodejs



python



php



ruby



java



go



gradle



scala



clojure

Save changes

Buildpacks

Buildpacks are scripts that are run when your app is deployed. They are used to install dependencies for your app and configure your environment. [Find new buildpacks on Heroku Elements](#)

Your new buildpack configuration will be used when this app is next deployed.

Add buildpack

 heroku/nodejs



Heroku's Features

Build Packs (all the dependencies matter here)

Heroku's Features

SSL certificate

SSL certificates

Apps with paid dynos use [Automated Certificate Management](#).

If you wish to manually acquire your own certificate, you can find out [more information here](#).

SSL Certificate

[Configure SSL](#)

Upgrade to paid dynos to configure Heroku SSL

Heroku's Features

Modify Code in Remote

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

Hide Config Vars

There are no config vars for this app yet
[Learn about config vars](#) in the Dev Center.

Add

Installed add-ons **\$0.00/month**

[Configure Add-ons](#) ➔

There are no add-ons for this app

You can add add-ons to this app and they will show here. [Learn more](#)

Dyno formation **\$0.00/month**

[Configure Dynos](#) ➔

This app has no process types yet

Add a Procfile to your app in order to define its process types. [Learn more](#)

Collaborator activity ⓘ

[Manage Access](#) ➔

There is no recent activity on this app

Collaborator activity will be shown when there are recent deploys

Add-Ons

Show logs

Latest activity

[All Activity](#) ➔



katesapp2019@gmail.com: Build failed

Today at 2:08 PM · [View build log](#)



katesapp2019@gmail.com: Enable Logplex

Today at 2:04 PM · v2



katesapp2019@gmail.com: Initial release

Today at 2:04 PM · v1