How to deploy a meteor application to my own server?

How to deploy a meteor application to my own server?

flavour 1: the development and deployment server are the same;

flavour 2: the development server is one (maybe my localhost) and the deployment server is another (maybe a VPS in the cloud);

flavour 3: I want to make a "meteor hosting" domain, just like "meteor.com". Is it possible? How?

Update:

I'm running Ubuntu and I don't want to "demeteorize" the application. Thank you.

deployment hosting meteor

edited Jul 12 '13 at 2:02

asked Jul 12 '13 at 1:53

J. Bruni
7,290 1 31 44

2 when Galaxy will be shipped these questions will be so easy to answer... :) – imslavko Jul 12 '13 at 3:33

add comment

2 Answers

Meteor documentation currently says:

> "[...] you need to provide Node.js 0.8 and a MongoDB server. You can then run the application by invoking node, specifying the HTTP port for the application to listen on, and the MongoDB endpoint."

So, among the several ways to **install Node.js**, I got it up and running following the best advice I found, which is basically unpacking the latest version available directly in the official Node.JS website, already compiled for Linux (64 bits, in my case):

```
# Does NOT need to be root user:

# create directory
mkdir -p ~/.nodes && cd ~/.nodes

# download latest Node.js distribution
curl -O http://nodejs.org/dist/v0.10.13/node-v0.10.13-linux-x64.tar.gz

# unpack it
tar -xzf node-v0.10.13-linux-x64.tar.gz

# discard it
rm node-v0.10.13-linux-x64.tar.gz

# rename unpacked folder
mv node-v0.10.13-linux-x64 0.10.13

# create symlink
ln -s 0.10.13 current

# add path to PATH
export PATH="~/.nodes/current/bin:$PATH"

# check
node --version
npm --version
```

And to **install MongoDB**, I simply followed the instructions in the MongoDB manual available in the Documentation section of its official website:

```
# Needs to be root user (apply "sudo" if not at root shell)

apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
```

```
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | tee /
apt-get update
apt-get install mongodb-10gen
```

The server is ready to run Meteor applications! For deployment, the main "issue" is **where** the "**bundle**" operation happens. We need to run `meteor bundle` command from inside the application source files tree. For example:

```
cd ~/leaderboard
meteor bundle leaderboard.tar.gz
```

If the deployment will happen in another server (*flavour 2*), we need to upload the bundle tar.gz file to it, using `sftp`, `ftp`, or any other file transfer method. Once the file is there, we follow both Meteor documentation **and** the README file which is magically included in the root of the bundle tree:

```
# unpack the bundle
tar -xvzf leaderboard.tar.gz

# discard tar.gz file
rm leaderboard.tar.gz

# rebuild native packages
pushd bundle/programs/server/node_modules
rm -r fibers
npm install fibers@1.0.1
popd

# setup environment variables
export MONGO_URL='mongodb://localhost'
export ROOT_URL='http://example.com'
export PORT=3000

# start the server
node main.js
```

If the deployment will be in the same server (*flavour 1*), the bundle tar.gz file is already there, and we don't need to recompile the native packages. (Just jump the corresponding section above.)

Cool! With these steps, I've got the "Leaderboard" example deployed to **my custom server**, not "meteor.com"... (only to learn and value their services!)

I still have to make it run on port 80 (I plan to use NginX for this), persist environment variables, start Node.JS dettached from terminal, et cetera... I am aware this setup in a "barely naked" one... just the base, the first step, basic foundation stones.

The application has been "manually" deployed, without taking advantage of all `meteor deploy` command magic features... I've seen people published their "meteor.sh" and "meteoric.sh" and I am following the same path... create a script to emulate the "single command deploy" feature... aware that in the near future all this stuff will be part of the pioneer Meteor explorers only, as it will grow into a whole Galaxy! and most of these issues will be an archaic thing of the past.

Anyway, I am very happy to see how fast the deployed application runs in the cheapest VPS ever, with a surprisingly low latency and almost instant simultaneous updates in several distinct browsers. Fantastic!

Thank you!!!

edited May 2 at 23:57

AZ.
1,423 1 13 28

answered Jul 12 '13 at 17:35

J. Bruni
7,290 1 31 44

I'm looking forward for the meteor team to release their official deployment system. – Flavien Volken Feb 9 at 13:40

`pushd bundle/server/node_modules` should be `pushd bundle/programs/server/node_modules` – Alucard Apr 20 at 16:19

I have updated the path – AZ. May 2 at 23:51

add comment

I would recommend flavor two with a separate deployment server. Separation of concerns leads to a more stable environment for your code and its easier to debug.

To do it, there's the excellent Meteoric bash script that helps you deploy to Amazon's EC2 or your own server.

As for how to roll your own meteor.com, I suggest you break that out into it's own StackOverflow question as it's not related. Plus, I can't answer it :)

answered Jul 12 '13 at 6:27

Jonatan Littke
2,683 17 27

1 +1 Thanks, Jonatan. But I've studied the whole thing a bit, including Meteoric and former "Meteor.sh" scripts, and learned the step-by-step thing. I realized that rolling our own meteor.com, at the moment, is quite challenging... indeed. :-) ...I've added my own answer with what worked for me, and I will accept it as the answer to the question... it became a little tutorial... I upvoted your answer and thank you for the informations. – J. Bruni Jul 12 '13 at 19:54

add comment

Not the answer you're looking for? Browse other questions tagged deployment hosting meteor **or ask your own question.**