Deployment Guide

Musketeers - Production Lifecycle Manager

This is a guide to deploy our PLM on an Ubuntu 16.04 server. This guide is tested only for Ubuntu 16.04, but should be applicable to Ubuntu v16 or newer.

This guide assumes you do not have any of the dependencies installed.

Dependencies:

- Node.js v8.x.x
- npm (node package manager)
- Meteor v1.6
- MongoDB
- Meteor Up

To Install Node.js v8 and npm

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
sudo apt-get install -y nodejs`
```

To Install Meteor v1.6:

```
run: curl https://install.meteor.com/ | sh
```

To Install and configure MongoDB

Installation guide listed below is just a copy of this guide with minor modifications: https://www.howtoforge.com/tutorial/install-mongodb-on-ubuntu-16.04/

1) Import the GPG Public Key:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927
```

2) Create a MongoDB list file:

```
echo "deb http://repo.mongodb.org/apt/ubuntu "$(lsb_release -sc)"/mongodb-org/3.2
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list
```

3) Update the repository

```
sudo apt-get update
```

4) Install MongoDB

```
sudo apt-get install -y mongodb-org
```

5) Create MongoDB service

a) Create the new service file mongod.service within /lib/systemd/system/

```
    cd /lib/systemd/system/
    sudo vim mongod.service
```

b) Paste the following into that file, save and exit

```
[Unit]
Description=High-performance, schema-free document-oriented database
After=network.target
Documentation=https://docs.mongodb.org/manual

[Service]
User=mongodb
Group=mongodb
ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

[Install]
WantedBy=multi-user.target
```

c) Update systemd service

```
  sudo systemctl daemon-reload
```

d) Start and enable mongod

```
  sudo systemctl start mongod
  sudo systemctl enable mongod
```

e) Confirm that mongodb has started using netstat

```
sudo netstat -plntu
```

Mongodb should be running on port 27017

6) Configure MongoDB username and password

a) Open the mongodb shell with the command

```
mongo
```

It is possible you wil receive a Failed global intialization error. In this case try

```
  export LC_ALL=C
```

```
mongo
```

b) Switch to the database admin

```
use admin
```

c) Create the root user

```
db.createUser({user:"admin", pwd:"admin123", roles:[{role:"root", db:"admin"}]})
```

Note: This user and password can be anything, but you must keep it safe.

Type exit to exit from MongoDB shell

d) Edit the mongodb service file '/lib/systemd/system/mongod.service'

```
sudo vim /lib/systemd/system/mongod.service
```

and edit line 9 (ExecStart) to say

```
ExecStart=/usr/bin/mongod --quiet --auth --config /etc/mongod.conf
```

e) reload the system service

```
sudo systemctl daemon-reload
```

f) Restart the mongod service and attempt to connect

If you are successfully able to enter the MongoDB shell, then your mongo database is set up correctly.

Now stop the mongod service, as our deployer Meteor Up will re-initialize the service when we are deploying.

```
sudo systemctl stop mongod
```

To Unpackage our application from the given TAR file.

```
tar -xvf plm.tar
```

To Install and configure MeteorUp

This guide has been edited from http://meteor-up.com/getting-started.html

a) Install mup with the following command:

```
npm install --global mup
```

b) Initialize your mup files

Change directories using the cd command to get into the root directory of our application (Something like cd ~/plm/)

```
mkdir .deploy && cd .deploy
mup init
```

c) Customize your configuration files.

In order to properly configure the server you MUST change the following in mup.js:

host - Should be the IP address of the server server authentication - Password or pem work. If neither of these are set, ssh-agent will be used.

In the app: section: name - A unique name, with no spaces path - A path to the meteor app, relative to the config file. Assuming you've been following this guide, that path will be ../ env.ROOT_URL - This is the url your app is accessible at. env.MONGO_URL - This is the url for your mongo database which was recently setup. ssl - Self-signed SSL certificates are included in the application. See the below example for how to use them.

Alternatively, Meteor Up claims to be able to generate SSL certificates from LetsEncrypt!. To use these, you simply have to uncomment and alter the included SSL section. This is not tested.

Example MUP file:

```
module.exports = {
  servers: {
    one: {
      // TODO: set host address, username, and authentication method
      host: 'localhost',
      username: 'vcm',
      // pem: './path/to/pem'
      password: 'password'
      // or neither for authenticate from ssh-agent
    }
  },

  app: {
    // TODO: change app name and path
    name: 'plm',
    path: '../plm',

    servers: {
      one: {},
    },

    buildOptions: {
      serverOnly: true,
    },
```

```
    env: {
      // TODO: Change to your app's url
      // If you are using ssl, it needs to start with https://
      ROOT_URL: 'https://vcm-3161.vm.duke.edu',
      MONGO_URL: 'mongodb://admin:admin123@127.0.0.1:27017/',
    },

    ssl: {
      crt: './server.crt',
      key: './server.key',
      port: 443
    },

    docker: {
      // change to 'abernix/meteord:base' if your app is using Meteor 1.4 - 1.5
      image: 'abernix/meteord:node-8.4.0-base',
    },

    // Show progress bar while uploading bundle to server
    // You might need to disable it on CI servers
    enableUploadProgressBar: true
  },

  mongo: {
    version: '3.4.1',
    servers: {
      one: {}
    }
  },
};
```

A couple notes: If you use a username and password as your authentication, that username must be added to the SUDO group:

`sudo adduser *username* sudo`

`sudo visudo`

Replace: `%sudo ALL=(ALL) ALL`

with

`%sudo ALL=(ALL) NOPASSWD:ALL`

and

`*username* ALL=(ALL) ALL`

with

`*username* ALL=(ALL) NOPASSWD:ALL`

To finally deploy, while in .deploy/ run:

```
mup setup
mup deploy
```

If SSL Certificates are giving you trouble, for whatever reason, the guide we followed to generate SSL certificates is here:

https://medium.com/@philipaffulnunoo/proxy-ssl-for-localhost-meteor-bf5a0c150ef8