Installing Requirements for the Kubernetes Dashboard

This document assumes you have a Linux machine (or VM), and that you have a brand new Ubuntu Linux environment setup, but does not assume familiarity with Linux. If you don't have a Linux environment and you're using Windows, you may want to read instructions on how to setup a Linux VM on Windows first.

Before you begin please make sure you can connect to your Linux machine and login. Command line instructions for Linux will be shown starting with `$`; you should only type the text following the `$`.

Basic Setup

Based on instructions from: https://docs.docker.com/engine/installation/linux/ubuntulinux/

This will update Linux and get curl and vim, which you'll need later.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install curl
$ sudo apt-get install vim
```

Unless you have another text editor you prefer, vim may be useful for beginners; instructions below use vim.

Initial checks

```
$ uname -r
```

You should get `3.2.0-23-generic`, `3.13.0-88-generic`, or something similar depending on what the current version is.

```
$ lsb_release -a
```

ˆ You should get a response like:

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 12.04.5 LTS
Release:        12.04
Codename:       precise
```

Get Kubernetes

```
$ curl -O https://storage.googleapis.com/kubernetes-release/release/v1.2.4/bin/linux/amd64/ku
```

This will take a while.

Get Vagrant on Linux

```
$ sudo apt-get install vagrant
$ cd kubernetes
$ export KUBERNETES_PROVIDER=vagrant
$ ./cluster/kube-up.sh
```

Note that the last command will throw a lot of errors.

Install Git

```
$ sudo apt-get install git
$ git --version
```

These instructions were last tested with git version 1.7.9.5.

Clone the Dashboard Repo

```
git clone https://github.com/kubernetes/dashboard.git
```

Install Kubernetes Dashboard Requirements

See the requirement list on the Getting Started page.

Install Docker

Based on instructions from: https://docs.docker.com/engine/installation/linux/ubuntulinux/

Setup

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates
$ sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912
$ cd /etc/apt/sources.list.d/
$ sudo vim docker.list
```

- i = insert

- Type `deb https://apt.dockerproject.org/repo ubuntu-precise main`
- Esc = stops inserting
- `:x` = exits and saves

```
$ sudo apt-get update
$ sudo apt-get purge lxc-docker
$ apt-cache policy docker-engine
```

Only needed for Ubuntu Precise 12.04

```
$ sudo apt-get update
$ sudo apt-get install linux-image-generic-lts-trusty
$ sudo reboot
```

Do the Docker install

```
$ sudo apt-get update
$ sudo apt-get install docker-engine
$ sudo service docker start
$ sudo docker run hello-world
```

You should receive a message that includes: `This message shows that your installation appears to be working correctly.`

Configure Docker for your user

Based on instructions from https://docs.docker.com/engine/installation/linux/ubuntulinux/#create-a-docker-group

The example below uses "username" as a placeholder. Please substitute with the user you are logged in as, which can be seen by using `$ id`. If you are running Linux in a VM using Vagrant, your username will be "vagrant".

```
$ sudo groupadd docker
$ sudo usermod -aG docker username
$ env
$ sudo reboot
$ docker run hello-world
```

You should get the same message as above, that includes: `This message shows that your installation appears to be working correctly.`

`$ status docker` –> should say "docker start/running, process [some number]"
`$ docker ps` –> should show a table of information (or at least headers)

Install Go

Get the latest download URL from https://golang.org/dl/

```
$ wget https://storage.googleapis.com/golang/go1.6.2.linux-amd64.tar.gz
```

Add Go to the path

```
$ go version
```

Should return something like `go version go1.6.2 linux/amd64`. Note that if you already had Go installed, ensure that `GO15VENDOREXPERIMENT` is unset.

Install Node and NPM

For some reason doing this...

```
$ sudo apt-get install nodejs
```

... gives a much older version, so instead we will get the more recent version:

```
$ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
$ sudo apt-get install -y nodejs
$ node -v
$ npm -v
```

Should return `v6.2.1` and `3.9.3` respectively.

Install Java 7

```
$ sudo apt-get install openjdk-7-jre
$ java -version
```

Should return `java version "1.7.0_101"`.

Install Gulp using npm

```
$ sudo npm install --global gulp-cli
$ sudo npm install --global gulp
$ gulp -v
```

Should return `CLI version 3.9.1` and `Local version 3.9.1`.

Install Other Dashboard Dependencies Automatically with NPM

```
$ cd /dashboard
$ npm install
```

This will install all the dependencies that are in the `package.json` file.

Run the Kubernetes Cluster

`$ gulp local-up-cluster`

If you need to stop the cluster you can run `$ docker kill $(docker ps -aq)`

`$ gulp serve`

Now you may continue with the Getting Started guide to learn more about developing with the Kubernetes Dashboard.