Assignment Documentation

Author: Jan Bujnowski

Document version: 0.2

Last Modified: 13/11/2017

Contents

[Introduction 1](#_Toc498382707)

[1. Create a Linux Host EC2 Instance 2](#_Toc498382708)

[1.1 Prerequisites: 2](#_Toc498382709)

[1.1.1 Create a key pair 2](#_Toc498382710)

[1.1.2 Install PuTTY 2](#_Toc498382711)

[1.2 Launch an EC2 instance 2](#_Toc498382712)

[1.3 Save host details as a new PuTTY 3](#_Toc498382713)

[1.4 Access EC2 Linux Instances via SSH from Windows 3](#_Toc498382714)

[1.5 Securing host access 3](#_Toc498382715)

[2. Install Nginx Web Server 4](#_Toc498382716)

[2.1 Install EPEL Repo 4](#_Toc498382717)

[2.2 Install Nginx 5](#_Toc498382718)

[3. Configure Nginx to Suit Requirements 6](#_Toc498382719)

[3.1 Root Directory and Index Files 6](#_Toc498382720)

[3.2 Create /version.txt 7](#_Toc498382721)

[3.3 Nginx config modification 7](#_Toc498382722)

[3.4 Reload the configuration 7](#_Toc498382723)

[4. Monitoring 7](#_Toc498382724)

[4.1 Diagnostic script for Our Website 7](#_Toc498382725)

[4.2 Other Test Methods 9](#_Toc498382726)

# Introduction

These notes are intended as an *aide-memoire* for anyone with some UNIX/Linux knowledge to get started with AWS EC2. They also contain some Cydar Ltd Technical Test notes.

# Create a Linux Host EC2 Instance

## 1.1 Prerequisites:

1. you need to create a key pair (minimum one)
2. you need an SSH client capable of using a private key for authentication, e.g. PuTTY, to access EC2 Linux Instances via SSH from Windows

### 1.1.1 Create a key pair

When you launch an AWS EC2 instance, you should specify the name of the key pair you plan to use to connect to the instance. If you don't specify the name of an existing key pair when you launch an instance, you won't be able to connect to the instance. When you connect to the instance, you must specify the private key that corresponds to the key pair you specified when you launched the instance.

For more information on Amazon EC2 Key Pairs, see this:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#having-ec2-create-your-key-pair>

### 1.1.2 Install PuTTY

Install the version of PuTTY suitable to your Windows version (if that’s your end client platform).

It’s best to install the whole PuTTY suite, so that you can use other tools, in addition to the terminal.

More info can be found here:

<https://linuxacademy.com/howtoguides/posts/show/topic/17385-use-putty-to-access-ec2-linux-instances-via-ssh-from-windows>

## 1.2 Launch an EC2 instance

Step 1: Choose an Amazon Machine Image (AMI) - select **Red Hat Enterprise Linux** version 7.4 (HVM), EBS General Purpose (SSD) Volume Type

Step 2: Choose an Instance Type – accept the default selection of a general-purpose instance (free tier eligible), then select **Review and Launch**

Step 7: Review Instance Launch – you can edit various machine parameters here, then press **Launch**

Select a public-private key pair – through the **Select an existing key pair or create a new key pair** dialogue box. Select Choose an existing pair (that you have created previously) and select you key pair (if you have only one, it will be selected by default). This is illustrated in *Figure 1* below:

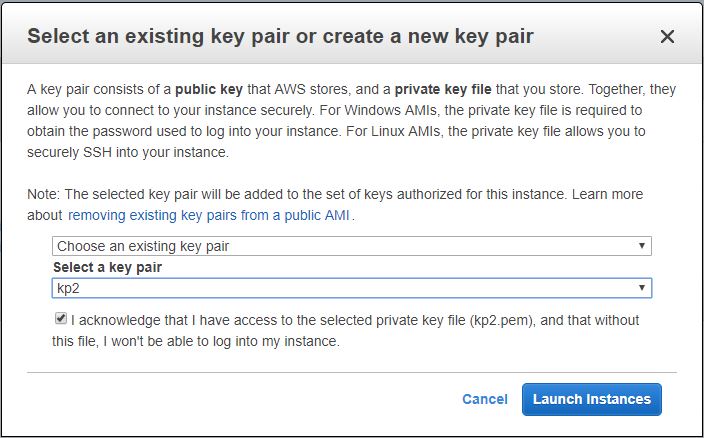


Figure 1: **Select an existing key pair or create a new key pair** dialogue box

Press the **Launch Instance** button.

You’ll get a status page. Press **View Instances** to see your instance list.

## 1.3 Save host details as a new PuTTY

Save your new host details as a new PuTTY session, with the location of the private key file for authentication saved under Connection->SSH->Auth.

## 1.4 Access EC2 Linux Instances via SSH from Windows

After you launch your EC2 instance, you can connect to it and use it the way that you'd use a standard host running Linux. I recommend using PuTTY for command-line interface to the host.

As you connect to your instance using PuTTY for the first time, make your selection to the server host key caching dialog box – in most cases it will be **Yes**.

Provide ec2-user as your login, then the passphrase that was used to create the private key.

For full details on how to use PuTTY to access EC2 Linux Instances via SSH from Windows, see the following:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

## 1.5 Securing host access

Currently, the host is open to any IP address on the Internet. In order to control traffic to your instance, see ‘Authorizing Inbound Traffic for Your Linux Instances’ that can be found here:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/authorizing-access-to-an-instance.html>

# Install Nginx Web Server

Nginx is installed from the EPEL Repository, so ensure this is available.

To check, use

yum repolist

## 2.1 Install EPEL Repo

<https://www.cyberciti.biz/faq/installing-rhel-epel-repo-on-centos-redhat-7-x/>

follow Method #2, **Install the extra EPEL repositories from dl.fedoraproject.org**, using GNU Wget - this retrieves content from web servers and supports downloading via HTTP, HTTPS, and FTP.

**cd** **/**tmp

**wget** https:**//**dl.fedoraproject.org**/**pub**/**epel**/**epel-release-latest-7.noarch.rpm

If you get an error saying that **wget** cannot be found, install it using yum:

yum install wget

[root@ip-172-31-19-158 ~]# yum install wget

Loaded plugins: amazon-id, rhui-lb, search-disabled-repos

Resolving Dependencies

--> Running transaction check

---> Package wget.x86\_64 0:1.14-15.el7\_4.1 will be installed

--> Finished Dependency Resolution

Dependencies Resolved

===========================================================================================================

Package Arch Version Repository Size

===========================================================================================================

Installing:

wget x86\_64 1.14-15.el7\_4.1 rhui-REGION-rhel-server-releases 547 k

Transaction Summary

===========================================================================================================

Install 1 Package

Total download size: 547 k

Installed size: 2.0 M

Is this ok [y/d/N]: y

Downloading packages:

wget-1.14-15.el7\_4.1.x86\_64.rpm | 547 kB 00:00:00

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Installing : wget-1.14-15.el7\_4.1.x86\_64 1/1

Verifying : wget-1.14-15.el7\_4.1.x86\_64 1/1

Installed:

wget.x86\_64 0:1.14-15.el7\_4.1

Complete!

[root@ip-172-31-24-78 tmp]# which wget

/bin/wget

Try downloading EPEL again using wget:

|  |
| --- |
| **cd** **/**tmp  **wget** https:**//**dl.fedoraproject.org**/**pub**/**epel**/**epel-release-latest-7.noarch.rpm  **ls** **\***.rpm |

To install epel-release-7-5.noarch.rpm, type:

|  |
| --- |
| **sudo** **yum install** epel-release-latest-7.noarch.rpm |

NB. yum can also install directly from an url:

yum install https://dl.fedoraproject.org/pub/epel/beta/7/x86\_64/epel-release-7-0.2.noarch.rpm

but the 2-stage method gives you more control and visibility of what’s going on.

List your new repos:

yum repolist

## 2.2 Install Nginx

NGINX is a popular alternative to the traditional Apache web server suite. A good intro can be found here:

<https://www.godaddy.com/garage/how-to-install-and-configure-nginx-on-centos-7/>

**NB** Turn Apache off, if it is active, see the above link for details.

Now that the EPEL repository is installed on the server, we can now use **yum** to install NGINX:

yum -y install nginx

#### Start NGINX:

service nginx start

#### Configure the server to start NGINX on reboot:

systemctl enable nginx

You should now be able to see an NGINX test page by going to http://<IP address or DNS name> for your server.

# Configure Nginx to Suit Requirements

## 3.1 Root Directory and Index Files

The [root](http://nginx.org/en/docs/http/ngx_http_core_module.html#root) directive specifies the root directory that will be used to search for a file. To obtain the path of a requested file, NGINX appends the request URI to the path specified by the root directive. The directive can be placed on any level within the http, server, or location contexts. In the example below, the root directive is defined for a virtual server. It applies to all location blocks where the root directive is not included to explicitly redefine the root:

server {

root /www/data;

location / {

}

location /images/ {

}

location ~ \.(mp3|mp4) {

root /www/media;

} }

Here, NGINX searches for a URI that starts with /images/ in the /www/data/images/ directory on the file system. But if the URI ends with the .mp3 or .mp4 extension, NGINX instead searches for the file in the /www/media/ directory because it is defined in the matching location block.

If a request ends with a slash, NGINX treats it as a request for a directory and tries to find an index file in the directory. The [index](http://nginx.org/en/docs/http/ngx_http_index_module.html#index) directive defines the index file’s name (the default value is index.html). To continue with the example, if the request URI is /images/some/path/, NGINX delivers the file /www/data/images/some/path/index.html if it exists. If it does not, NGINX returns HTTP code 404 (Not found) by default.

## 3.2 Create /version.txt

File /version.txt will contain the nginx version number, so let’s check that first:

[root@ip-172-31-24-78 ~]# nginx -v

nginx version: nginx/1.10.2

We’ll assume that /version.txt is a static file, created manually - although it could be created by a script, to ensure it’s content tallies with the actual version of nginx after, say, a software upgrade.

## 3.3 Nginx config modification

Modify the config file:

under server

server {

# root /usr/share/nginx/html;

root /;

location / {

index version.txt;

}

}

For more information, see ‘Serving Static Content’, Root Directory and Index Files:

<https://www.nginx.com/resources/admin-guide/serving-static-content/>

## 3.4 Reload the configuration

nginx -s reload

# Monitoring

## 4.1 Diagnostic script for Our Website

**N.B.** This is specific to our website

/home/ec2-user/test-site Bash script.

The URL is:

<http://ec2-18-217-3-112.us-east-2.compute.amazonaws.com>

From a host different than the web server, we can use **wget** or **curl**:

[root@ip-172-31-19-158 ~]# wget http://ec2-18-217-3-112.us-east-2.compute.amazonaws.com

--2017-11-13 16:59:49-- http://ec2-18-217-3-112.us-east-2.compute.amazonaws.com/

Resolving ec2-18-217-3-112.us-east-2.compute.amazonaws.com (ec2-18-217-3-112.us-east-2.compute.amazonaws.com)... 172.31.24.78

Connecting to ec2-18-217-3-112.us-east-2.compute.amazonaws.com (ec2-18-217-3-112.us-east-2.compute.amazonaws.com)|172.31.24.78|:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 8 [text/plain]

Saving to: ‘index.html’

100%[=================================================================>] 8 --.-K/s in 0s

2017-11-13 16:59:49 (1.76 MB/s) - ‘index.html’ saved [8/8]

A file called index.html will be created in the current directory:

[root@ip-172-31-19-158 ~]# pwd

/root

[root@ip-172-31-19-158 ~]# ls

anaconda-ks.cfg index.html original-ks.cfg

[root@ip-172-31-19-158 ~]# cat index.html

1.10.2

Adding -quiet parameter to wget:

[root@ip-172-31-19-158 ~]# wget -q http://ec2-18-217-3-112.us-east-2.compute.amazonaws.com

[root@ip-172-31-19-158 ~]# ls

anaconda-ks.cfg index.html original-ks.cfg

Adding -silent parameter to curl makes no difference here:

[root@ip-172-31-19-158 ~]# curl http://ec2-18-217-3-112.us-east-2.compute.amazonaws.com

1.10.2

[root@ip-172-31-19-158 ~]# curl -s http://ec2-18-217-3-112.us-east-2.compute.amazonaws.com

1.10.2

Of course, if you want it deathly quiet, just add ‘>/dev/null 2>&1’ to the end…

The /home/ec2-user/test-site can be run on demand or by cron.

## 4.2 Other Test Methods

WebSitePulse:

<https://www.websitepulse.com/help/tools.php>

This can test the web site, DNS resolution, Service (http, smtp, telnet, etc) or network (ping, traceroute, etc).

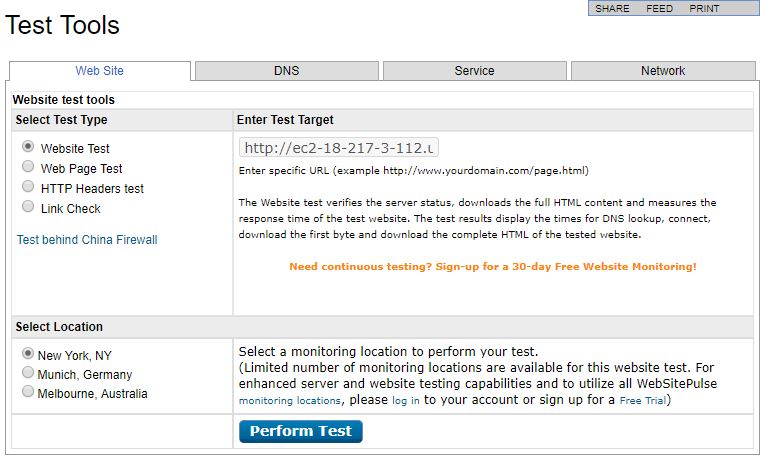


Figure 2: WebSitePulse dialogue box

Just enter the test target full DNS name (in this case ec2-18-217-3-112.us-east-2.compute.amazonaws.com ) and the location from which to test (can be left on default. The result will be something like:

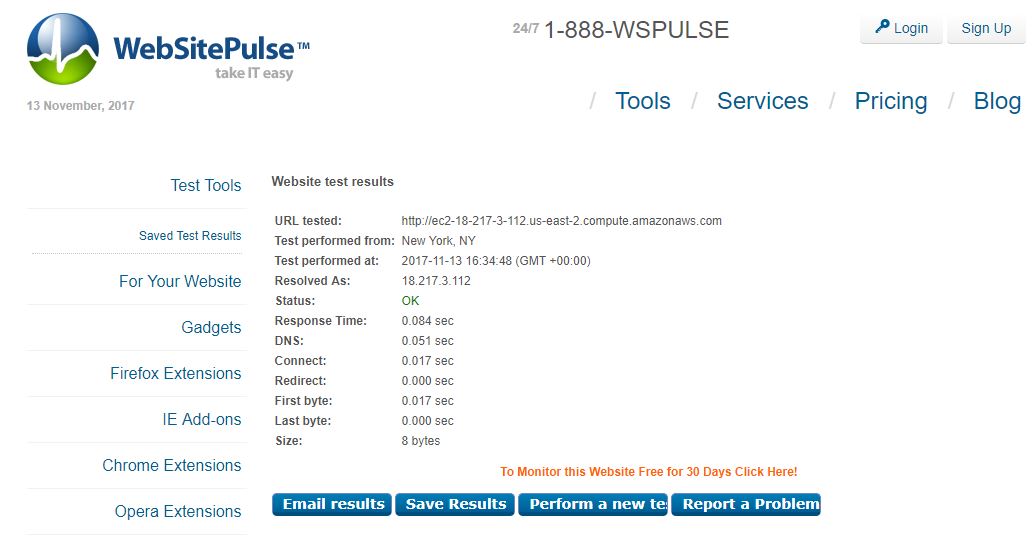


Figure 3: WebSitePulse test result