Project: Migrate WordPress to AWS

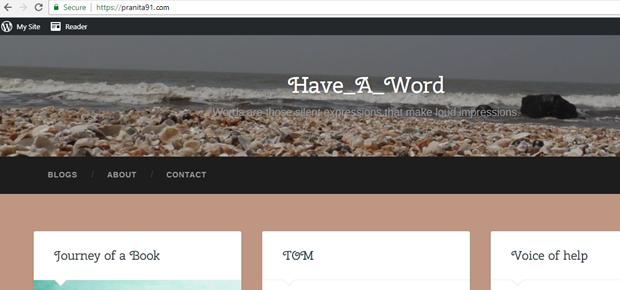
**INTRODUCTION:**

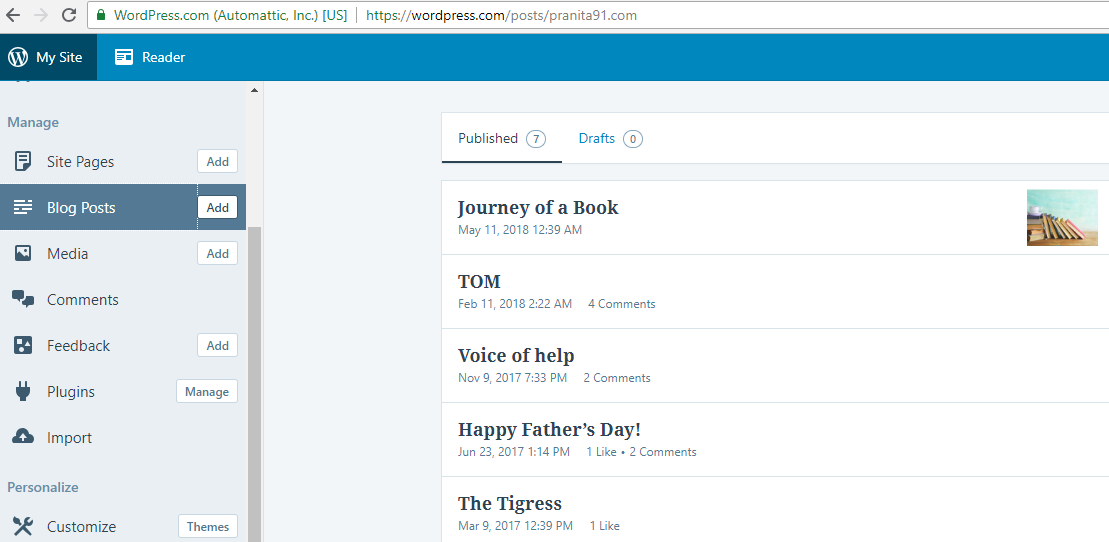
WordPress is a free and open-source content management system (CMS). It supports wide variety of web content. It also helps you promote your requirement to design a website of your own. For that purpose, it helps with providing a free domain and other facilities as according to their range of plans.

Amazon Web Services (AWS) provides with range of cloud computing platforms as per need or demand of individual, team, organization etc. The ease of implementing and managing of these services make these a booming cloud platform. It offers reliable, scalable and cost-effective solutions to your business needs.

I use WordPress.com to own a simple website and use it to publish my blogs for quite a while now. My intention here is to run and monitor my website using AWS cloud services. In this particular case study, we will be able to host a new website through AWS services, and then migrate entire content of my old original WordPress website on this newly launched website.

Please check the below screenshots to have a look at the original WordPress website that I own:





**PROCEDURE:**

The procedure is divided into three parts:

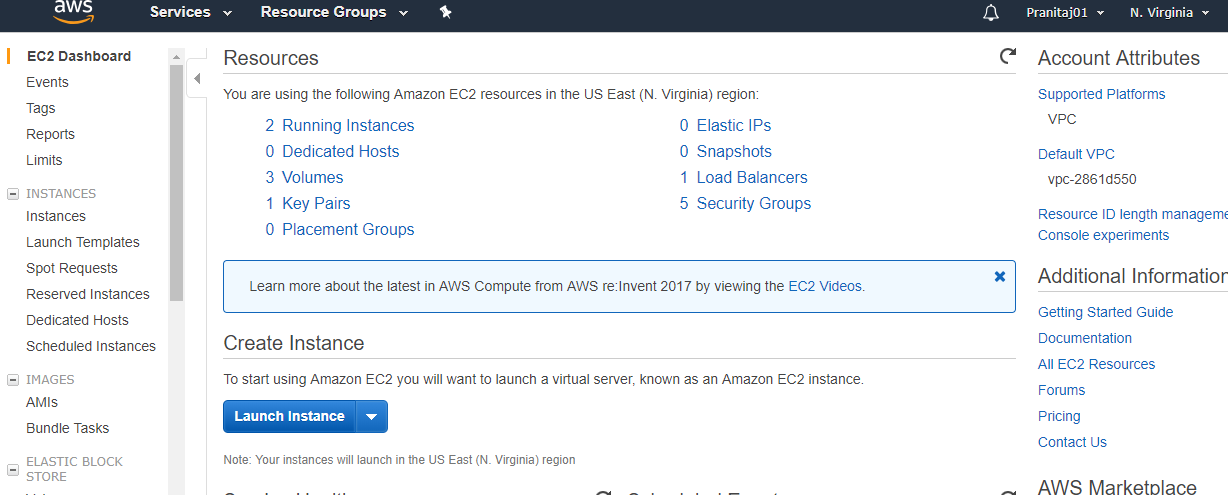
* Deploy WordPress Installation
* Transfer Domain name
* Import WordPress website

**Step 1: Deploy WordPress Installation**

In this step, we will launch a WordPress installation on an Amazon AWS virtual machine.

Login into AWS management console.

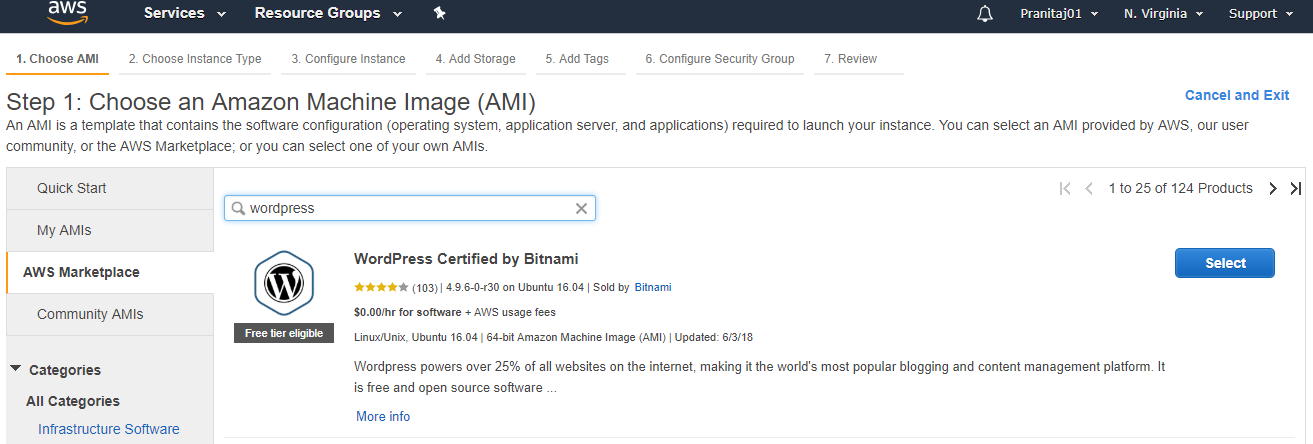
Click **services 🡪 EC2** (AWS Elastic Compute Cloud service) 🡪 **Launch Instance**



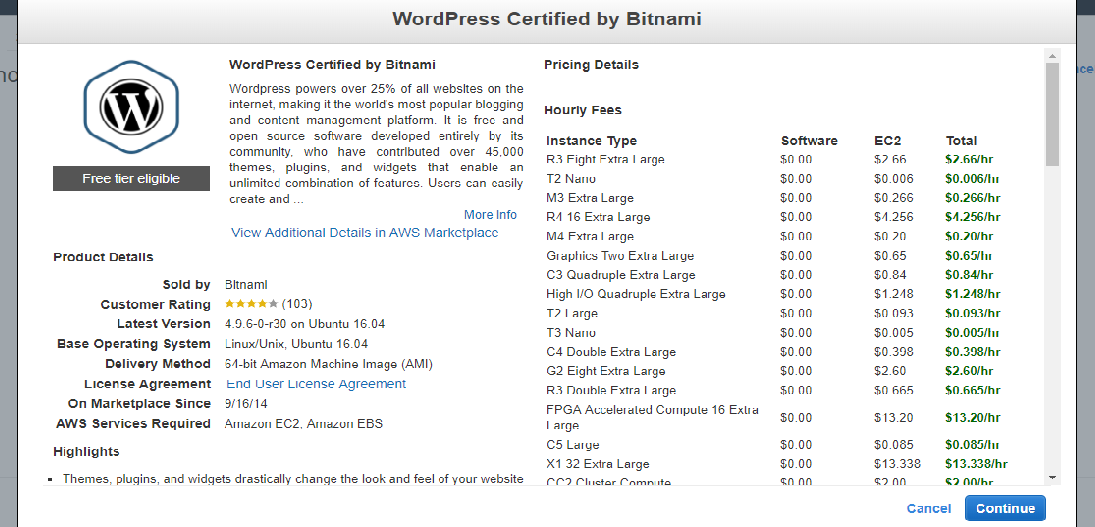
Go to **AWS Marketplace.** Select **WordPress Certified by Bitnami.**

‘WordPress Certified by Bitnami’ is a free AMI available for WordPress installation.

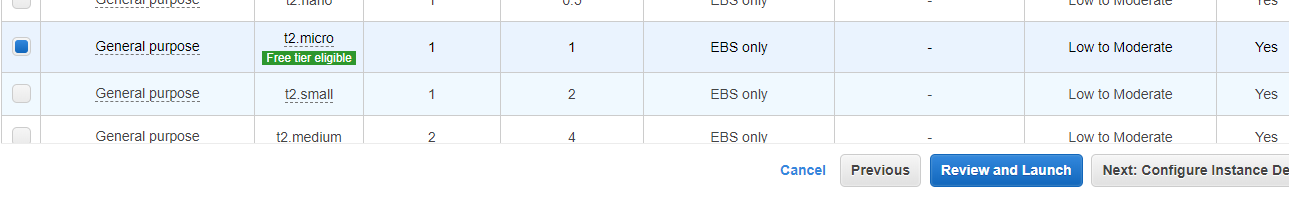
Know more about AWS AMIs: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>

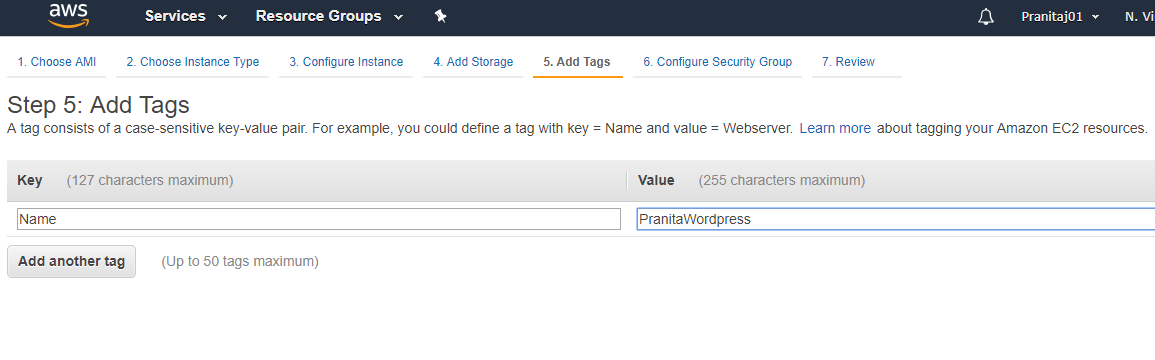


This next screen below briefs you about the pricing structure of the selected AWS AMI. For the above AMI, it will be zero. Click **Continue.**

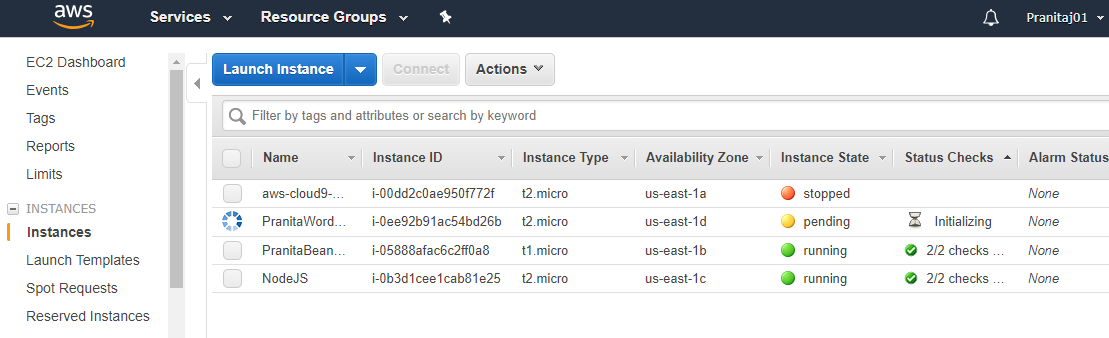


Select **te.mirco** general purpose instance. It is free tier eligible as seen below. Click **Next:Configure Intstance Details**.

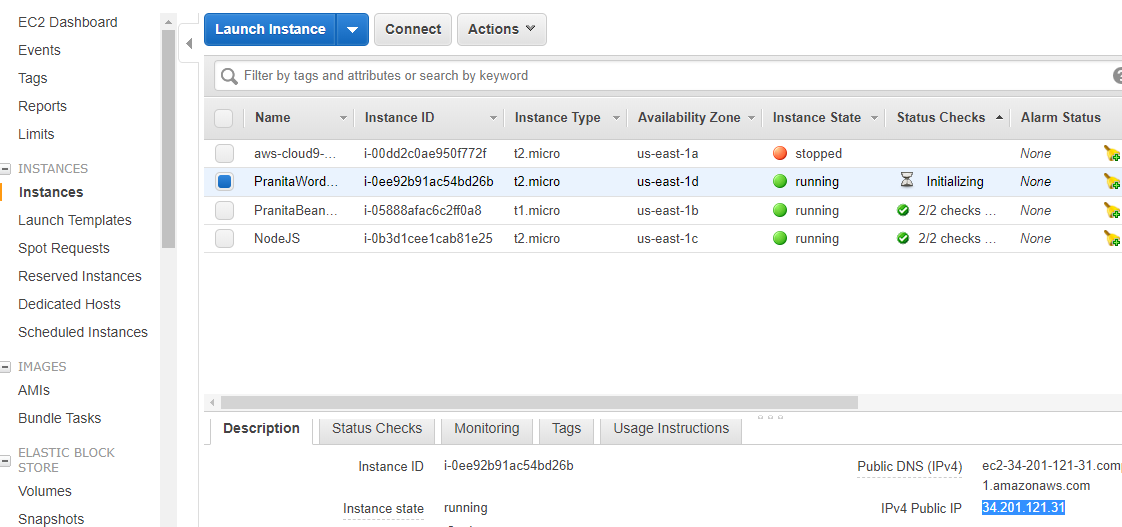


Let the Storage settings be as default. Go to next screen to add Tags to our instance. You may skip this step, however I prefer adding label to my instance. 

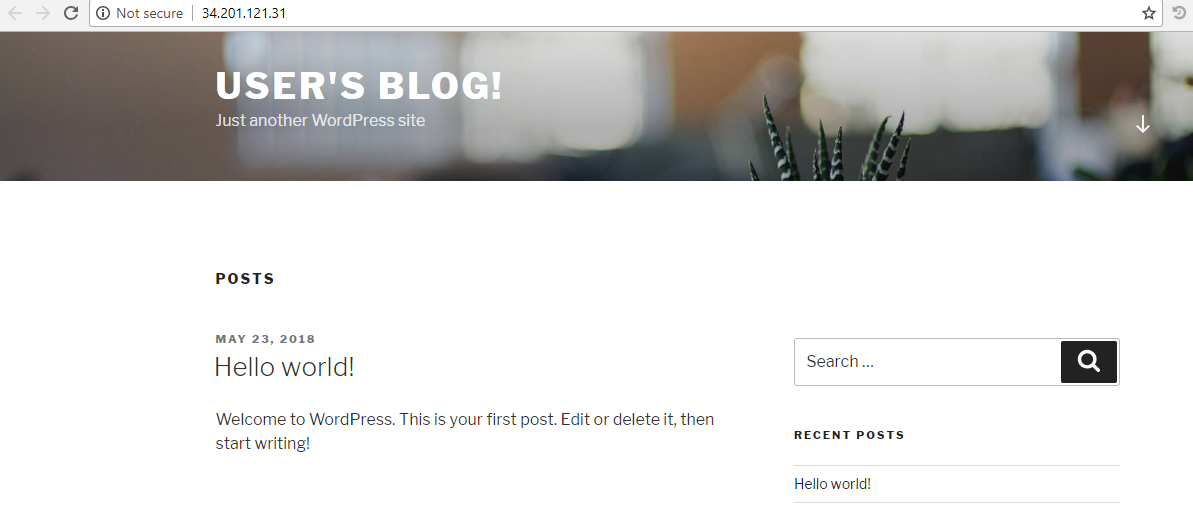
Keeping rest of the settings as they, click **Review and Launch**. When the instance is up, instance state will change from Pending to Running.



Let’s have a look at what we have launched successfully launched. Select your instance. Copy the public IP address. Paste it in your browser and hit Enter.

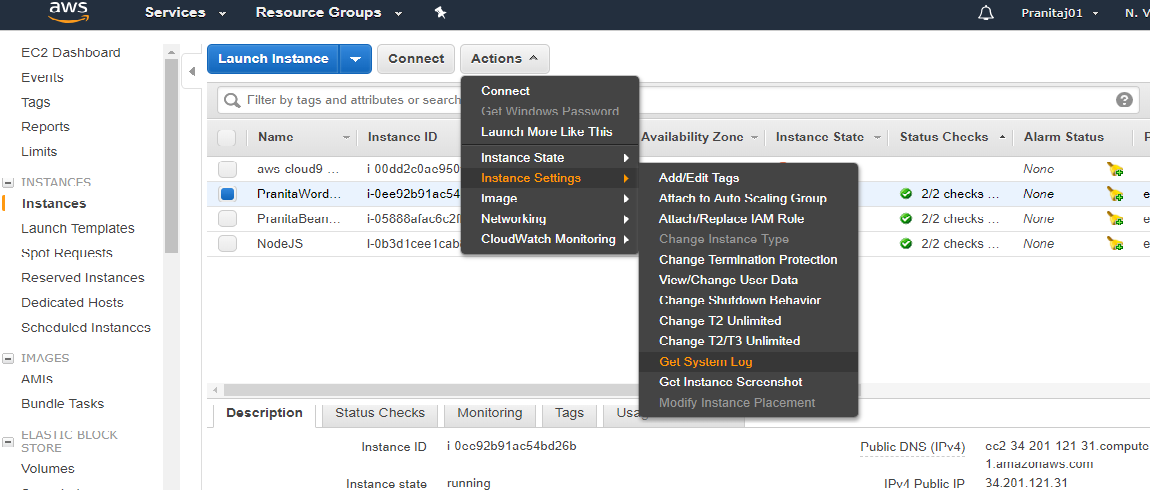


And here we are. We have successfully launched a simple WordPress website on our EC2 instance. As far as you are the admin rights of your AWS account, you have the liberty to configure your AWS service as you require.

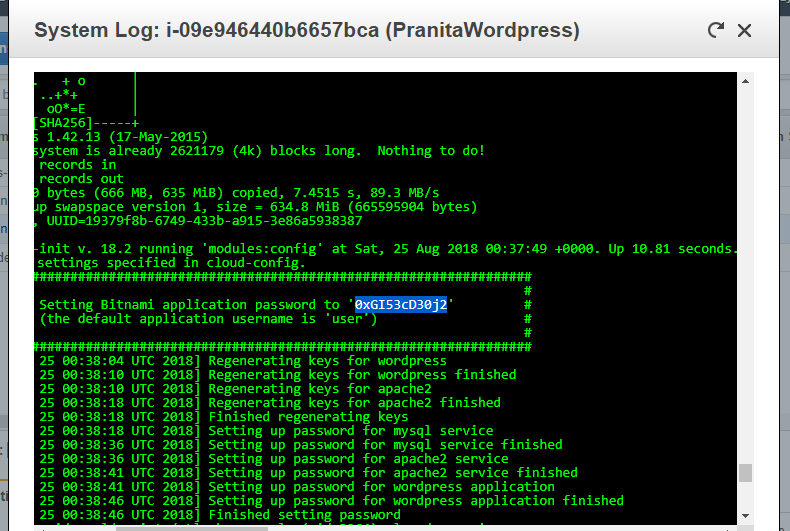


Let’s login into our newly launched website. Select EC2 instance.

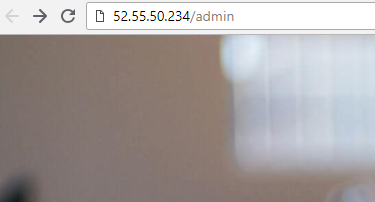
Click **Actions 🡪 Instance Settings🡪 Get System Log**



Scroll down through the System Log till you find the below shown information. That is your admin passport for your website. Select and copy your password.



Goto following URL: Public IP/admin.

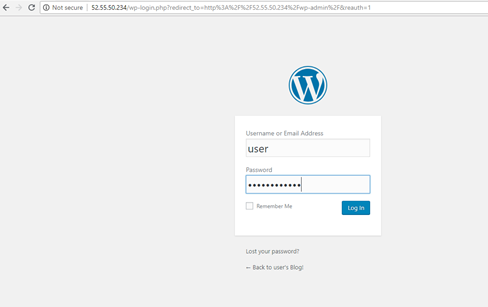


Put details:

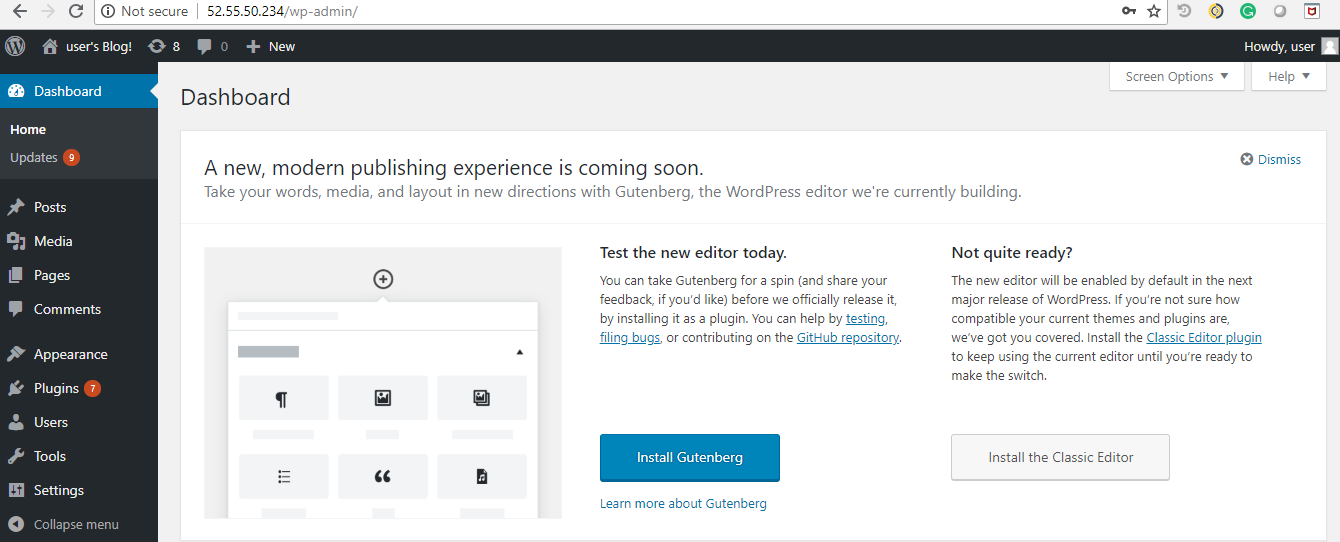
Username: User

Password: The one copied from the instance’s System Log

Click **Login**.



And here you have logged into your WordPress site.



**Step 2: Transfer Domain Name**

In the step, we will transfer/point your existing domain name to your new WordPress installation on Amazon AWS

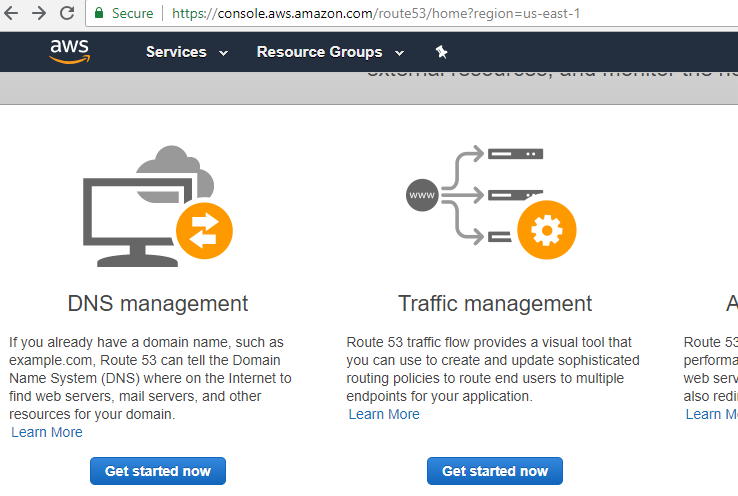
*NOTE: If you are using free plan on WordPress.com, it itself manages the DNS settings and you will not be allowed to change Name Server or any kind of Domain settings. I upgraded my WordPress plan from Free to Personal for this lab. (You can get your refund too later. Check WordPress.com refund policies.)*

*OR*

*You can do this lab using WordPress.org. You can also migrate from WordPress.com to WordPress.org.* [*https://www.elegantthemes.com/blog/resources/migrating-wordpress-com-to-wordpress-org-a-step-by-step-guide*](https://www.elegantthemes.com/blog/resources/migrating-wordpress-com-to-wordpress-org-a-step-by-step-guide)*. This way you can modify domain settings, add/remove plugins without any hassle.*

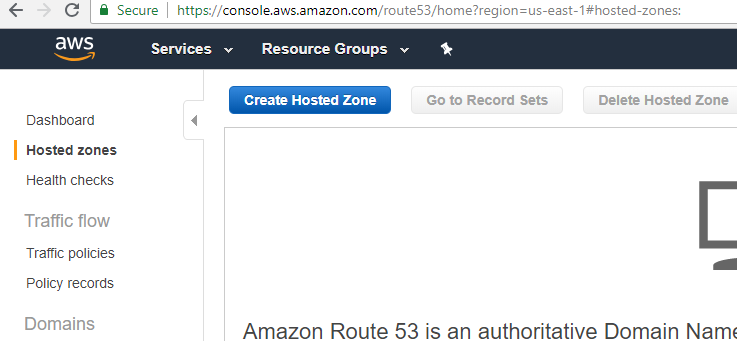
On your AWS management console, go to **services🡪 Route53.**

As according to the objective of this case study, we already have our domain name. Hence Click **Get Started** under **DNS Management.**

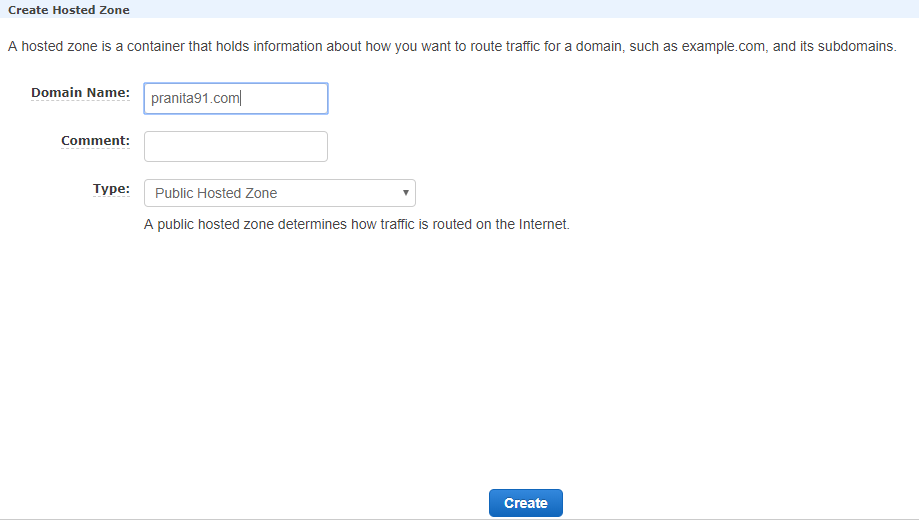


Click **Create hosted Zone.**

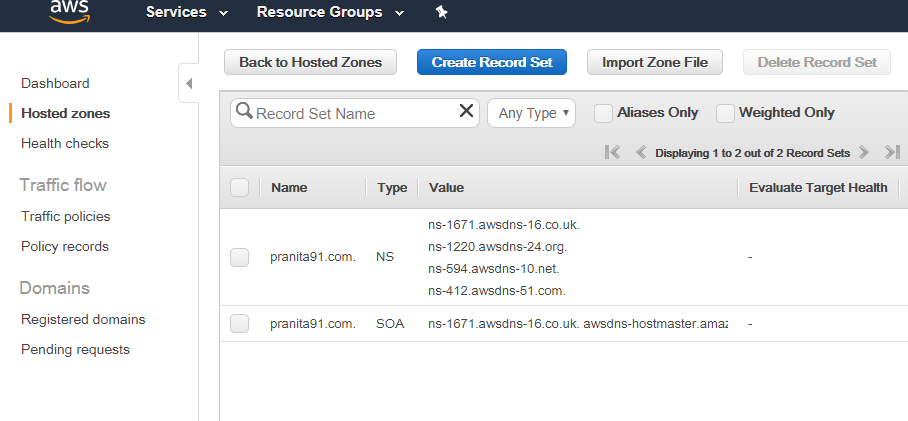
A hosted zone is a collection of resource record sets hosted by AWS Route53. In here, we will be able to add and manage our DNS records.



Enter your existing domain name (The one you are using for your WordPress website) as the Domain Name, as shown below. Click **Create**.



The NS and SOA type records are created by default for your hosted zone. We will add here couple of records of our own. Click **Create Record Set**.

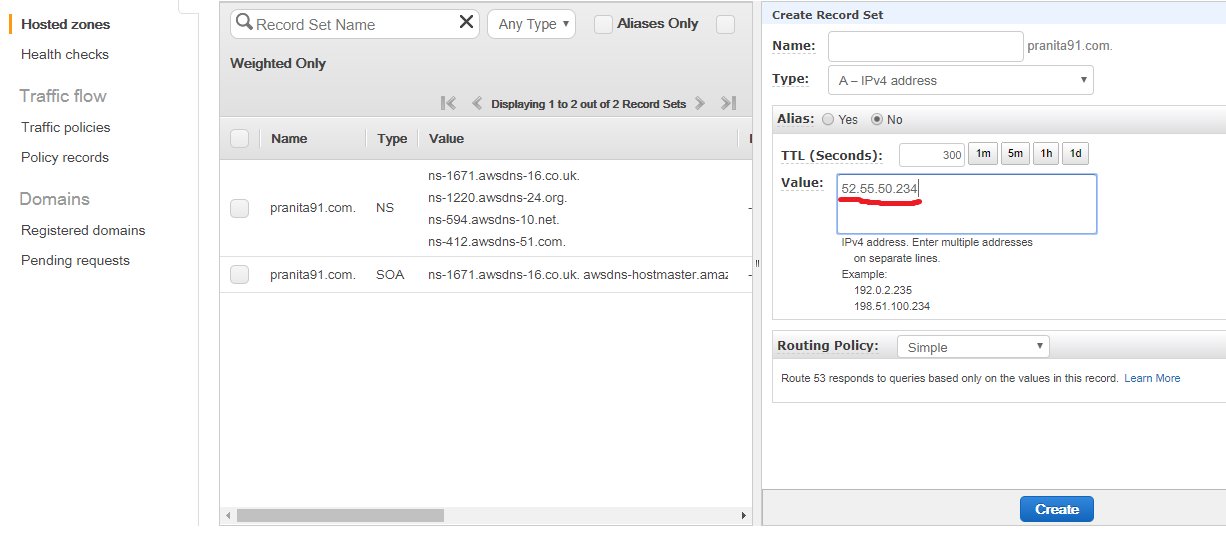


Let the Name field be empty. So it is by default ‘pranita.com’.

Select record type as **A-IPv4 address.**

Copy public IP of your launched EC2 instance, and paste it in the Value field. Click **Create**.

We are now pointing pranita91.com to our new IP address.



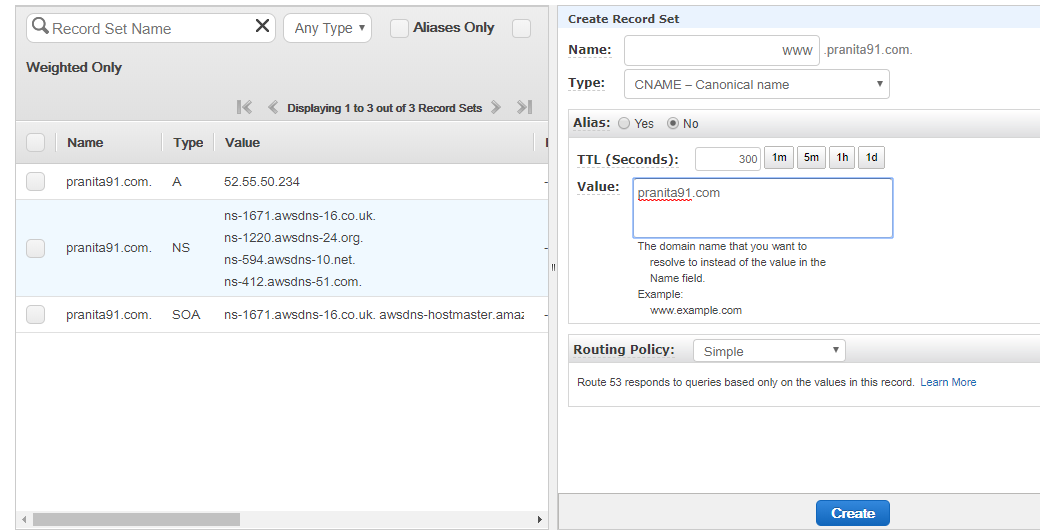
Again, click **Create Record Set**.

Add ‘www’ in the Name field. So we are adding value to ‘www.pranita91.com’.

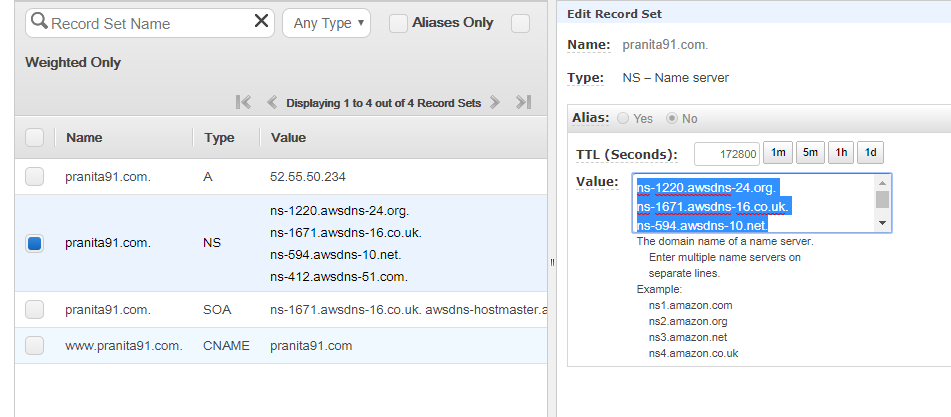
Select Type as **CNAME – Canonical name**.

Add Value as ‘pranita91.com’ (Add your domain name as value). Click **Create**.

Now we are pointing ‘www.pranita91.com’ to ‘pranita91.com’, thereby assigning an alias name.

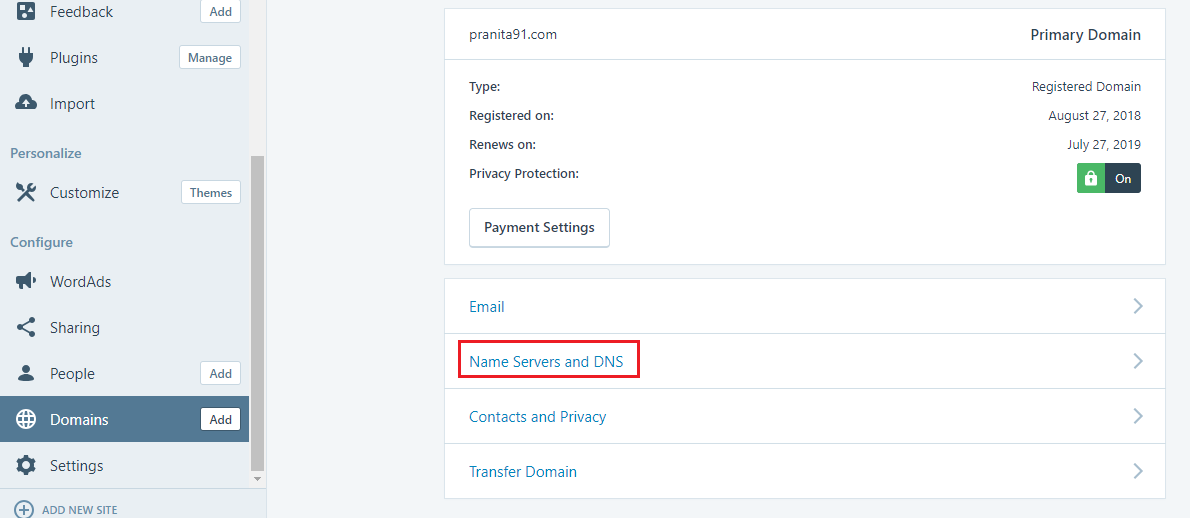


We are now going to customize our existing name servers of domain and add these new Name Server values to it.Select NS record in your hosted zone. The values are available to be copied as seen below.

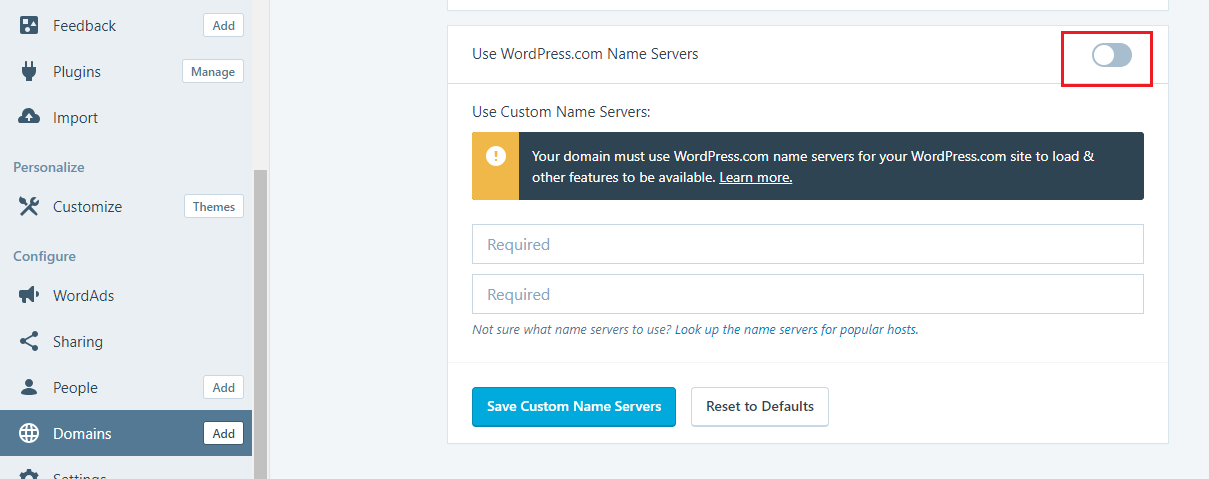


Go to your WordPress website. Select **Domains** from the dashboard of your site.

Click on your available primary domain name. You see the below screen then. Select **Name Servers and DNS.**

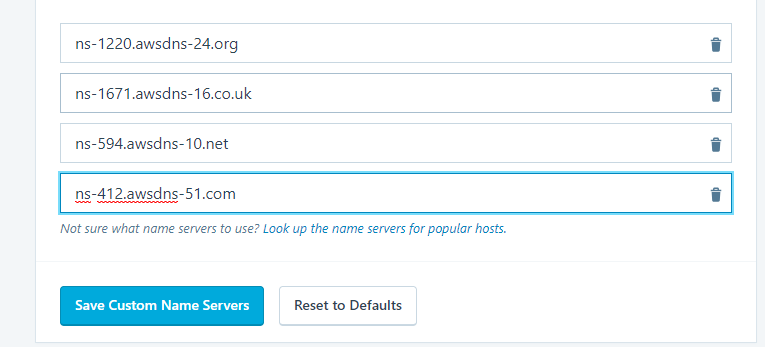


**Untoggle** the ‘Use WordPress.com Name Servers’. We don’t want WordPress Name servers as we are going to add ours.



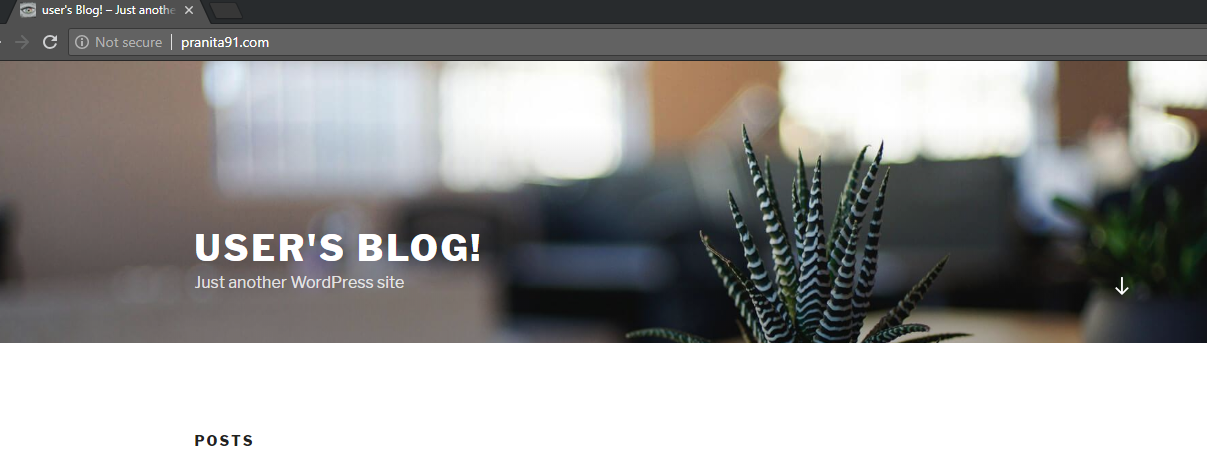
Copy each value from the NS record we selected on our AWS hosted zone. Paste them in the provided fields on WordPress.com. Make sure you don’t copy the period at the end of each value.

Click **Save Custom Name Servers**.

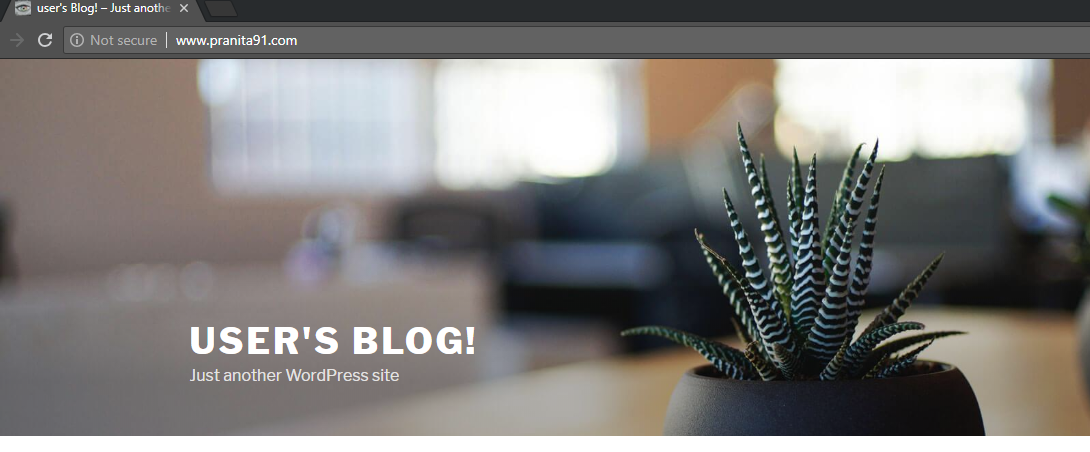


Now instead of accessing your newly launched website through the IP address, try using your existing domain name. It will no more point to your original WordPress website.

As below, ‘pranita91.com’ now points to your newly launch website through EC2 instance.



‘www.pranita91.com’ also points to newly launched website.



Congratulations! You have successfully transferred your Domain name to your new host.

These website still don’t have our original content. Let’s migrate our contents (my blogs in this case) in the next step.

**Step 3: Import from old WordPress website**

Finally, in this last step, we will migrate our existing WordPress website to our new WordPress installation on Amazon AWS.

*NOTE: Wordpress.com will not allow adding or installing plugins unless you have business plan active on your WordPress account. I have mentioned another alternative in the upcoming details in this step which can serve the purpose of migration for now. But I highly recommend installing ‘All in one WP migration’ plugin to import/export the website content. (Reference)*

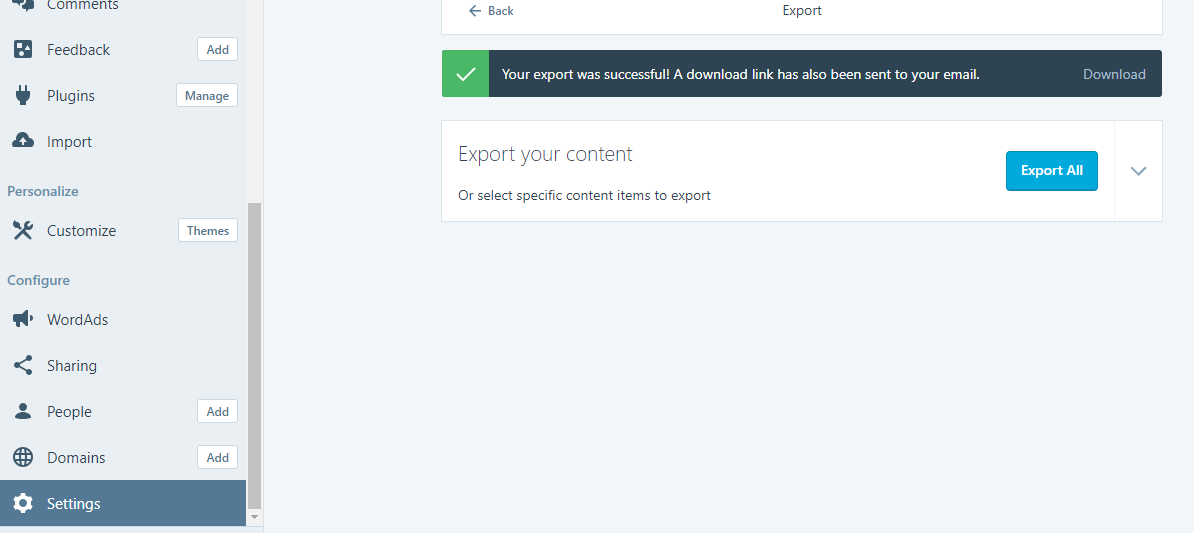
*OR*

*You can do this project using WordPress.org. You can also migrate from WordPress.com to WordPress.org.* [*https://www.elegantthemes.com/blog/resources/migrating-wordpress-com-to-wordpress-org-a-step-by-step-guide*](https://www.elegantthemes.com/blog/resources/migrating-wordpress-com-to-wordpress-org-a-step-by-step-guide)*. This way you can modify domain settings, add/remove plugins without any hassle.*

As per my Personal plan on WordPress.com account, I am not allowed to add/install any plugin. I am using inbuilt tool to export.

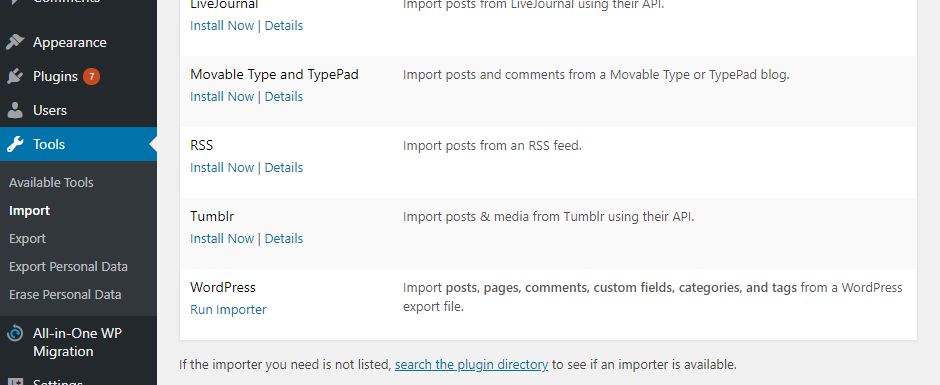
Login in your WordPress account; go to dashboard. CLick **Settings🡪 Export🡪 Export All**.

Download the prompted zipped file. We will now import this to our WordPress installation on AWS.



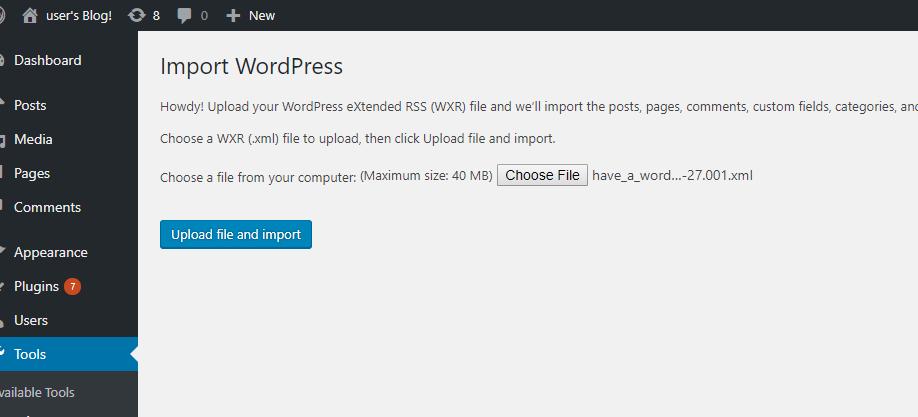
Login as admin in the newly launched Website (as we did at the end of Step1). Only this time you don’t need to copy the EC2 public IP. Instead type your domain name in URL.

Go to **Tools🡪 Import🡪** see **WordPress** somewhere at the end of list**🡪 Run importer**.



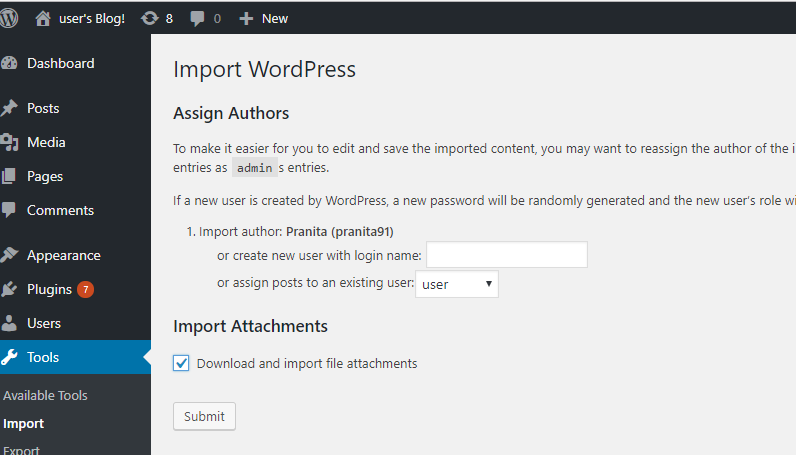
Click **Choose File**. Select the downloaded export file (xml file).

Click **Upload file and import**.



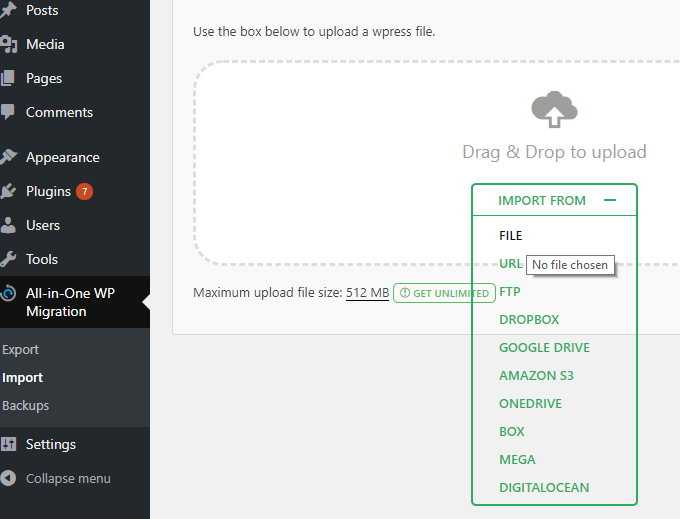
Either create new user which will be added as subscriber to this admin account. Or. As shown below select **user** in the second field. This way all the imported content will be assigned to existing user of this account.

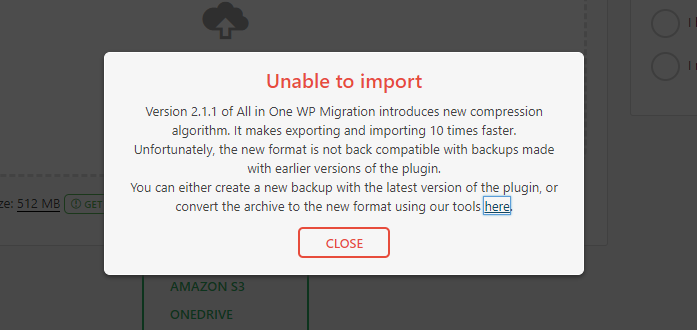
Check **Download and import file attachments.** Click **submit.**



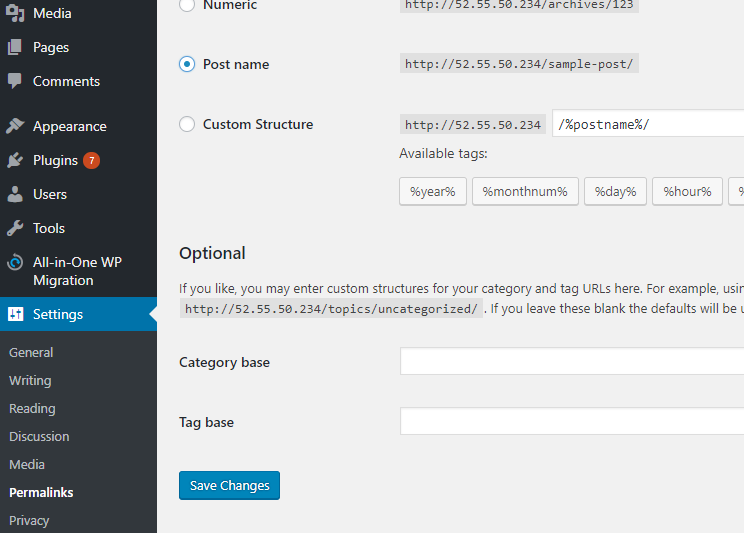
*Note: If you try to import the downloaded export file (not exported using the plugin), it will throw a file compatibility issue. Just to try, please see below two screenshots.*

*This is not a step, just a note.*





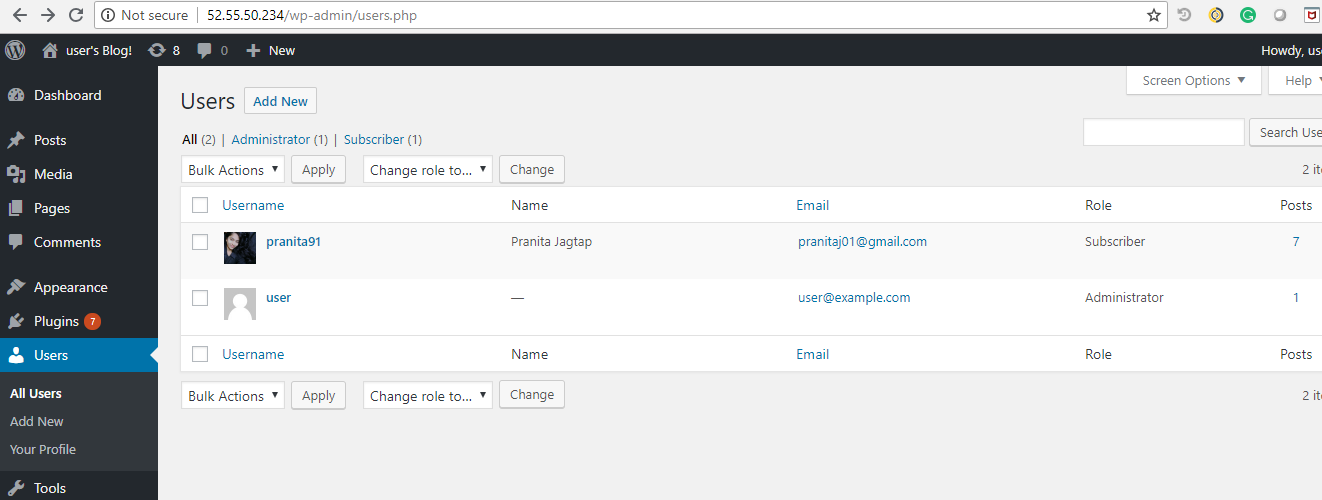
After successfully import, go to **Settings🡪 Permalinks🡪** **Post name🡪 Save Changes**



**FINAL OUTPUT:**

In Step 3 while importing, if you import data as a new user, your previous content will be added along with your previous username as a subscriber. Please see screenshot below.

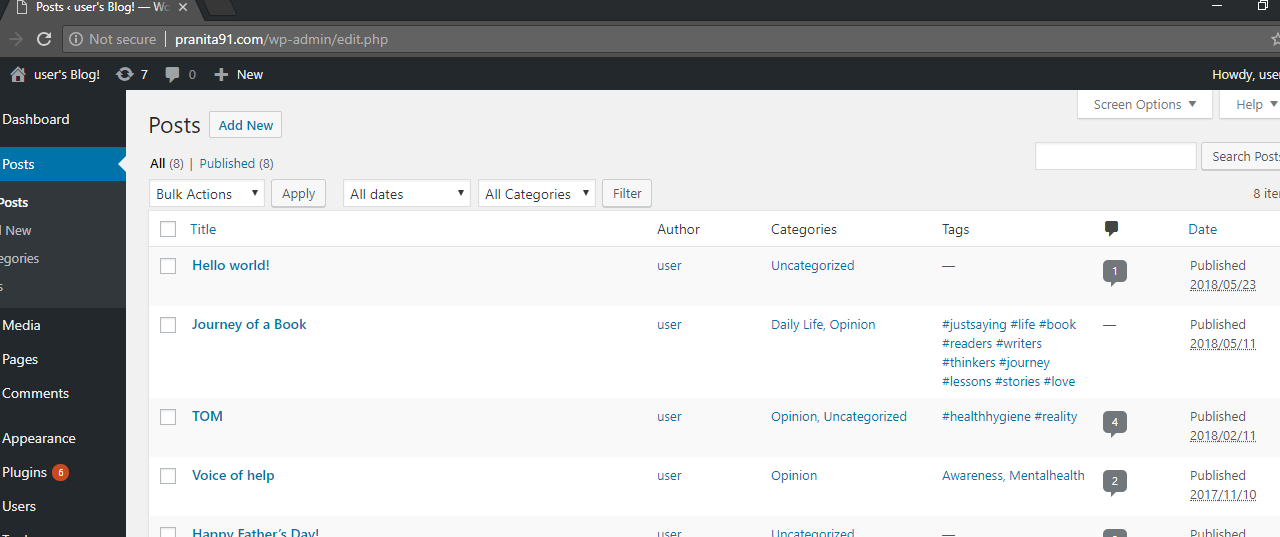
Which means **user** is the owner of this WordPress website, and the blogs and content belong to imported new user/subscriber **pranita91.**



**OR**

In Step 3 while importing, if you select to assign all the imported data to existing user, you can see the list of blogs of your previous WordPress website successfully listed on this new WordPress installation, as below.

Which means **user** is the owner of this WordPress website, and the blogs and content belong to **user** itself**.**



**Conclusion:**

We have successfully installed WordPress installation on AWS EC2, transferred existing domain to our new IP address using AWS Route53, imported previous content on our new website.

Thereby, we have successfully migrated WordPress on AWS.

**References:**

**Issues:**