



Usage instructions:

This image is provisioned with WordPress and a SSL Certificate. **A domain name is required.**

1. Launch the product via 1-click. **Please wait until** the instance passes **all** status checks and is running. You can connect using your Amazon private key and '**ubuntu**' login via your SSH client.

To update software, use: **sudo apt-get update**

(Optional) (But Recommend)

- Allocate an “Elastic IP” to your instance under the Network & Security tab of the AWS dashboard. This will ensure that your instance **keeps its IP address during restarts.**

2. Update your DNS Records and create 2 new hosted zones.

- Create **two** different “**A**” type simple routing records.

(1) “**Record name**” with **blank entry** to create a record for the root domain &

(1) “**Record name**” with **www**

- For each record you create, change your domain’s “**Record type**” to point to “routes traffic to an IPv4 address...”
- For each record also enter your domain’s “**Value**” with your instance “**IPv4 Public IP address**”

For example:

Define simple record

Record name

Info

To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

www

.codecreator.info

Keep blank to create a record for the root domain.

Record type

Info

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

A – Routes traffic to an IPv4 address and some AWS resources

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

Value/Route traffic to

Info

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

IP address or another value, depending on the record type

34.229.186.113

<input type="checkbox"/>	Record name ▼	Type ▼	Routin... ▼	Differ... ▼	Alias ▼	Value/Route traffic to ▼	TTL (s...
<input type="checkbox"/>	www.codecreator.info	A	Simple	-	No	34.229.186.113	300
<input type="checkbox"/>	codecreator.info	A	Simple	-	No	34.229.186.113	300

For additional help:

- <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/rrsets-working-with.html>

3. Next make some domain configuration changes. Replace all 3 “YourDomain.com” with your actual domain name as identified below. Run the following commands in root:

sudo su

sudo nano /etc/apache2/sites-available/wordpress.conf

```
GNU nano 6.2
<VirtualHost *:80>
  ServerName www.YourDomain.com
  Redirect permanent / https://www.YourDomain.com/
</VirtualHost>

<VirtualHost *:443>
  ServerName www.YourDomain.com
  DocumentRoot /var/www/wordpress
```

- **Save & Exit:** After making the changes, save the file and exit the editor (in nano, press Ctrl + X, then Y to confirm, and Enter to save).

4. Next configure the wp-config file.

sudo su

cd /var/www/wordpress

sudo nano wp-config.php

- Scroll down and replace the “YourDomain.com” with your actual domain name.

```
define('WP_HOME','https://www.YourDomain.com');
define('WP_SITEURL','https://www.YourDomain.com');
```

- Save & Exit
- Restart apache2: **sudo systemctl restart apache2**

5. **Install SSL**: Replace **yourdomain.com** with your actual domain name. Run the following commands:

sudo apt install certbot python3-certbot-apache

sudo certbot certonly --apache -d yourdomain.com -d www.yourdomain.com

sudo systemctl restart apache2

Set up automatic renewal:

sudo crontab -e

0 */12 * * * certbot renew --quiet

Save and Close

Check if Certbot Timer is Enabled:

systemctl list-timers | grep certbot

Enable and Start Timer:

sudo systemctl enable certbot.timer

sudo systemctl start certbot.timer

systemctl status certbot.timer

Test Automatic Renewal

sudo certbot renew --dry-run

6. Finally go back and replace **your domain name** with the **new** SSL keys location:

sudo su

sudo nano /etc/apache2/sites-available/wordpress.conf

Important: uncomment or delete the # to enable the lines (turns white when enabled)

```

DocumentRoot /var/www/wordpress

SSLEngine on
SSLCertificateFile /etc/letsencrypt/live/yourdomain.com/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/yourdomain.com/privkey.pem

# Additional SSL configurations, logging, etc.

</VirtualHost>

```

- **Save & Exit**
- **Restart apache: `sudo systemctl restart apache2`**

8. In a browser, go to your <https://www.yourdomain.com> to set up your WordPress site. Follow the instructions.

You are now set up with a WordPress website with a HTTPS SSL.

AWS Data

- **Data Encryption Configuration:** This solution does not encrypt data within the running instance.
- **User Credentials are stored:** `/root/.ssh/authorized_keys` & `/home/ubuntu/.ssh/authorized_keys`
- **Monitor the health:**
 - Navigate to your Amazon EC2 console and verify that you're in the correct region.
 - Choose Instance and select your launched instance.
 - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.