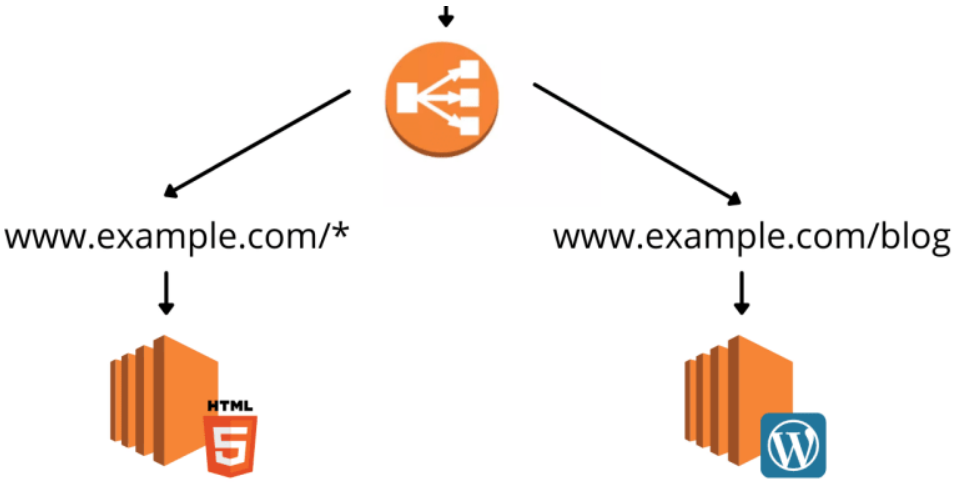
Static website + WordPress blog - Hosting on AWS



In a startup, where you don't have much time to deal with all these. You might choose one of the three following options:

1. Build your entire website with WordPress using a theme so your content writers feel at home
2. Create static pages for each blog and deploy them with the static site
3. Create a custom blogging tool

But hold on,

You have another option

But first, look at an example

- www.teamingway.com
- www.teamingway.com/blog/

We will set up the same thing.

Deploy static website

Create a t2.micro EC2 instance with the following Nginx config:

```
# filename: example.com
# location: /etc/nginx/sites-enabled

# redirect http to https
# redirect example.com to www.example.com
server {
   listen 80;
   server_name nanotechx.org.com;
   return 301 https://www.example.com$request_uri;
}

# main block
server {
   listen 80;

   server_name www.example.com;

   # path to the static website folder
   root /srv/www/your-website-folder;
   index index.html index.htm index.nginx-debian.html;

   location / {
      # serve static website files, throw "nginx 404 error" if file not found
      try_files $uri $uri.html $uri/ =404;
   }
}
```

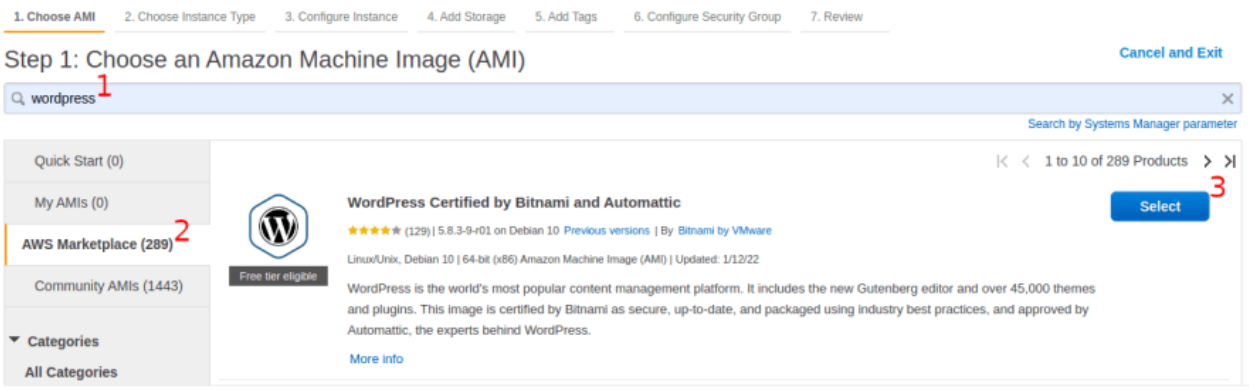
Notice, we only handled "HTTP" and not "HTTPS", because we will handle HTTPS at Load Balancer level and attach ssl certificate there too.

Restart Nginx:

sudo service nginx restart

Our static website is now up and running.

Deploy WordPress site

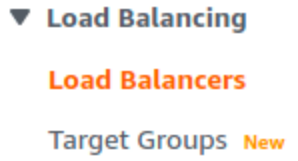


Or you can manually set up one, it's all up to you. Once your WordPress instance is up and running, move to the next step.

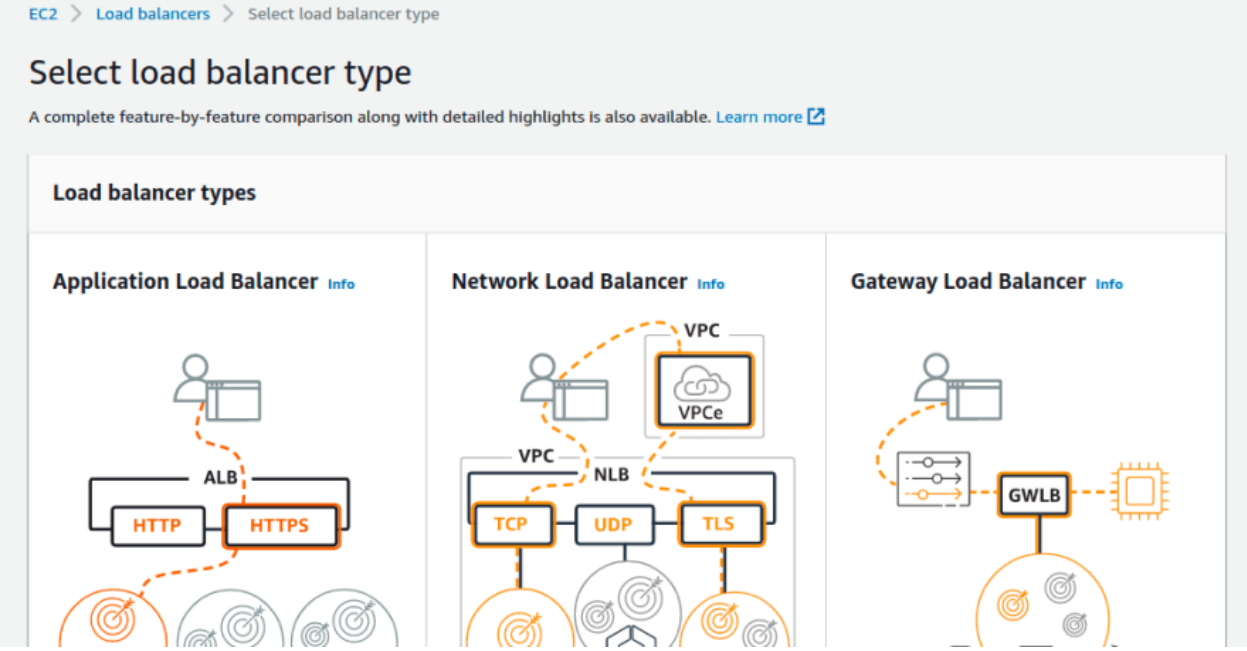
Set up redirection using "AWS Load Balancer" and "Route 53"

1) Create Load Balancer

On your EC2 dashboard, go to **Load Balancers** from the left sidebar:



and then create a new load balancer, make sure you select **Application Load Balancer**

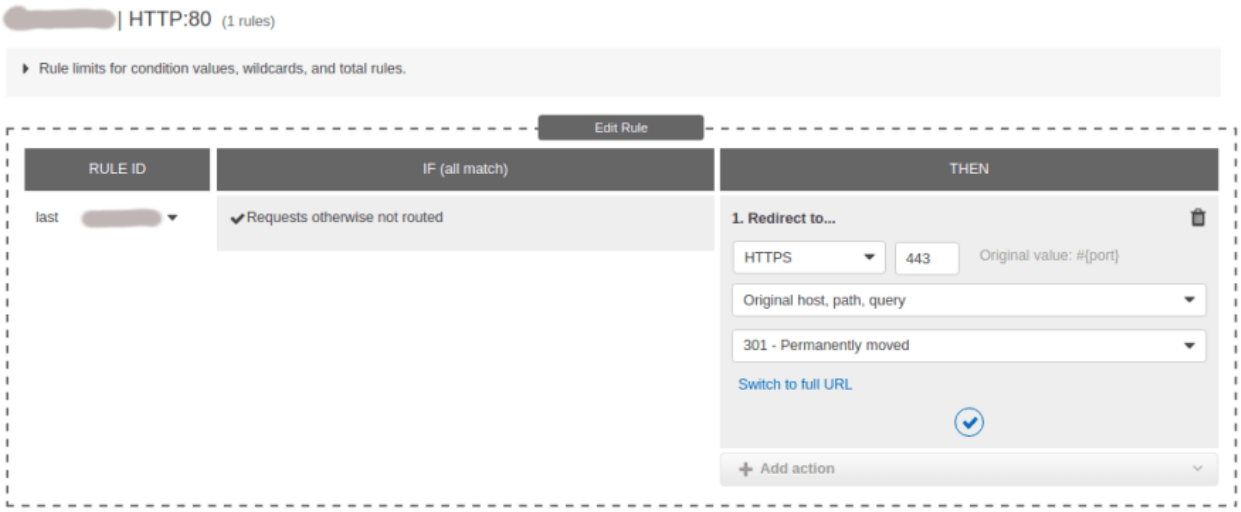


It should be:

1. Internet-facing
2. In the same VPC as your website and WordPress instance
3. In the same Availability zone as your website and WordPress instance

Add **default listener** as your website instance with port 80 and complete the load balancer creation. We'll change it later anyway.

2) Set up ALB routing



Add the second listener for port 443. Now there will be two rules (See the image below for reference)

Rule 1: Default action will be to forward all the traffic to our static website.

Rule 2: If the path matches the pattern /blog*, forward the request to our WordPress instance.

[redacted] | HTTPS:443 (2 rules)

▸ Rule limits for condition values, wildcards, and total rules.

1 [redacted]	**IF** ✔ Path is /blog*	**THEN** **Forward to** website-blogs-wordpress : **1** (100%) Group-level stickiness: Off
last	**HTTPS 443: default action** *This rule cannot be moved or deleted*	**IF** ✔ Requests otherwise not routed	**THEN** **Forward to** website-static : **1** (100%) Group-level stickiness: Off

Listener ID	Security policy	SSL Certificate	Rules
HTTP : 80 [redacted]	N/A	N/A	Default: redirecting to HTTPS://#{host}:443/#{path}?#{query} View/edit rules
HTTPS : 443 [redacted]	ELBSecurityPolicy-2016-08	Default: [redacted] (ACM) View/edit certificates	Default: forwarding to website-static View/edit rules

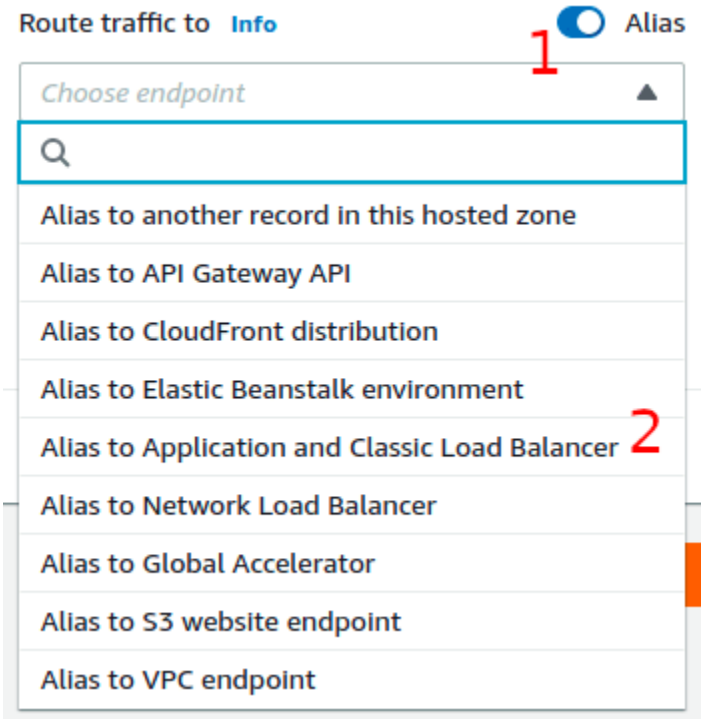
3) Add Load Balancer to Route 53

We have everything ready now:

1. Static website instance
2. WordPress site instance
3. Load Balancer with https and rules to redirect based on URL

Choose Record type = A

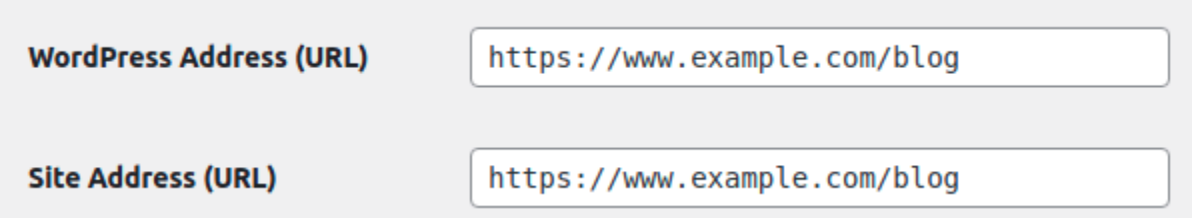
Turn on the Alias toggle in the Route Traffic to section and select Alias to Application and Classic Load Balancer



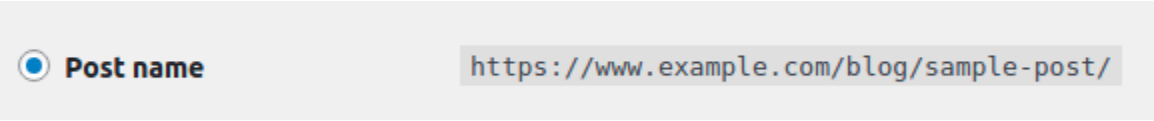
Select your zone i.e. us-west-1 and your load balancer will automatically appear, select and create the record.

Add sub path to your wordpress posts

Change your site name and url, add `/blog` at the end:



Change permanent links by going into `Settings > Permalinks`



---

and that's it