

# AWS Lightsail

<https://lightsail.aws.amazon.com/ls/webapp/home?#>

---

## Install NGINX

```
sudo apt-get update  
sudo apt-get install nginx
```

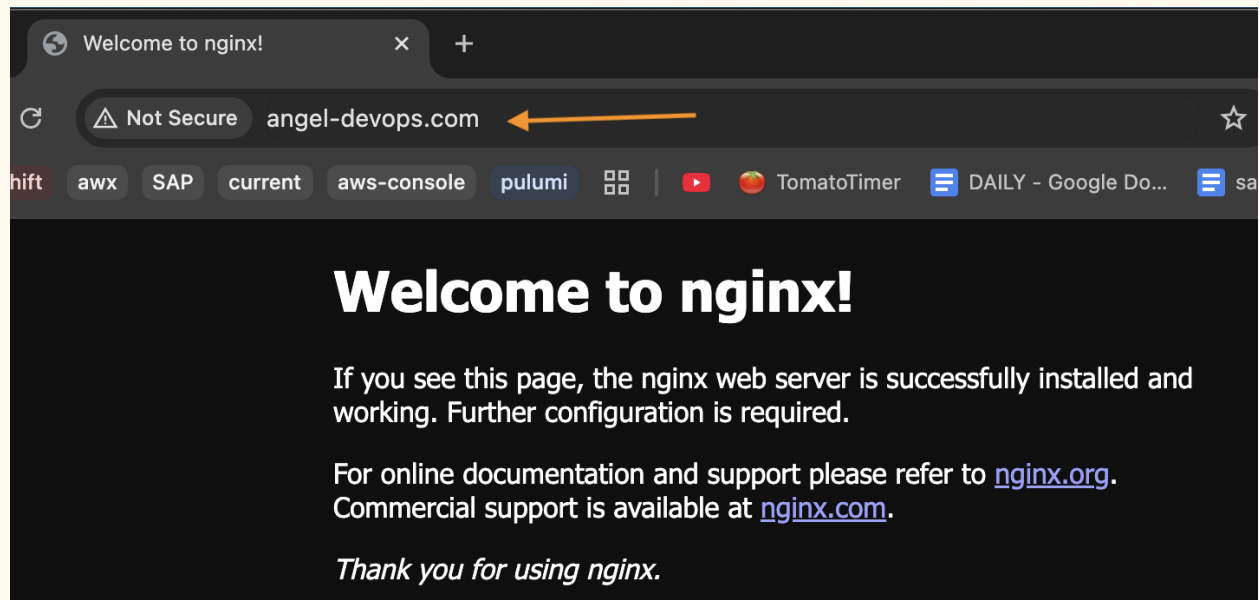
### *Configure Server Blocks:*

Create or modify Nginx server block configuration files (typically in `/etc/nginx/sites-available/` and symlinked to `/etc/nginx/sites-enabled/`) to define your website's settings, including root directory, domain names, and SSL configuration.

### *Deploy Your Website Files:*

Upload your website's static files (HTML, CSS, JavaScript, images) to the Nginx root directory (e.g., `/var/www/html/` or a custom directory defined in your Nginx configuration).

### *Your DNS name now serving NGINX:*



## ROUTE53 : A Record

Public **angel-devops.com** Info Delete zone Test record Configure query logging

► **Hosted zone details** Add 'A' record pointing to static ip address of LightSail instance Edit hosted zone

**Records (3)** | DNSSEC signing | Hosted zone tags (0)

**Records (3)** Info © Delete record Import zone file Create record

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Q Filter records by property or value Type ▼ Routing p... ▼ Alias ▼ < 1 > ⚙

<input type="checkbox"/>	Record ... ▼	Type ▼	Routin... ▼	Differ... ▼	Alias ▼	Value/Route traffic to ▼	TTL (s... ▼	Health ... ▼	E
<input type="checkbox"/>	angel-dev...	A	Simple	-	No	3.130.87.185	300	-	-
<input type="checkbox"/>	angel-dev...	NS	Simple	-	No	ns-1994.awsdns-57.co.uk, ns-1233.awsdns-26.org, ns-659.awsdns-18.net, ns-147.awsdns-18.com.	172800	-	-
<input type="checkbox"/>	angel-dev...	SOA	Simple	-	No	ns-1994.awsdns-57.co.uk. a...	900	-	-

## Connect Terraform to existing Lightsail Instance

1> create [main.tf](#) file

```
main.tf U X
lightsail > main.tf
1 resource "awsLightsailInstance" "my_lightsail" {
3   name = "Ubuntu-1"
5   blueprint_id = "ubuntu_24_04"
6   bundle_id = "nano_2_0"
7 }
8
```

2> terr plan -refresh-only

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulumi-aws-1/lightsail$ terr plan -refresh-only  
No changes. Your infrastructure still matches the configuration.  
  
Terraform has checked that the real remote objects still match the result of your most recent changes, and found no differences.
```

3> terraform import aws\_lightsail\_instance.my\_lightsail Ubuntu-1

ERROR

```
aws_lightsail_instance.my_lightsail: Importing from ID "Ubuntu-1"...  
aws_lightsail_instance.my_lightsail: Import prepared!  
  Prepared aws_lightsail_instance for import  
aws_lightsail_instance.my_lightsail: Refreshing state... [id=Ubuntu-1]  
  
Error: Cannot import non-existent remote object  
  
While attempting to import an existing object to "aws_lightsail_instance.my_lightsail",  
the provider detected that no object exists with the given id. Only pre-existing objects  
can be imported; check that the id is correct and that it is associated with the  
provider's configured region or endpoint, or use "terraform apply" to create a new remote  
object for this resource.  
  
angel.cruz@angel-ubuntu:~/sandbox/repos/pulumi-aws-1/lightsail$
```

the fix

I can see you're trying to import an existing Lightsail instance into your Terraform configuration, but you're getting an error that the object doesn't exist. Let me help you investigate this issue. First, let me check your current Terraform configuration to understand how you've defined the Lightsail instance resource:

Read main.tf

Now let me check if there are any existing Lightsail instances in your AWS account and verify the correct instance name. The error suggests that either:

1. The instance name "Ubuntu-1" doesn't exist
1. The instance exists but in a different region
1. There's a configuration issue

Let me help you check for existing Lightsail instances:

```
aws lightsail get-instances  
--region us-east-2  
{
```

```
"instances": [  
  {  
    "name": "Ubuntu-1",  
    "arn": "arn:aws:lightsail:us-east-2:501005077666:Instance/392625ad-0  
a8a-4499-ba52-9  
2eeaec336f2",  
    "supportCode": "883330640267/i-0a1979345e3367098",  
    "createdAt": "2025-06-30T13:29:29.517000+00:00",  
    "location": {  
      "availabilityZone": "us-east-2a",  
      "regionName": "us-east-2"  
    },  
    "resourceType": "Instance",  
    "tags": [],  
    "blueprintId": "ubuntu_24_04",  
    "blueprintName": "Ubuntu",  
    "bundleId": "micro_3_0",  
    "isStaticIp": true,  
    "privateIpAddress": "172.26.8.176",  
    "publicIpAddress": "3.130.87.185",  
    "ipv6Addresses": [  
      "2600:1f16:2b7:3b00:7fa5:41d2:4b02:3157"  
    ],  
    "ipAddressType": "dualstack",  
    "hardware": {  
      "cpuCount": 2,  
      "disks": [  
        {  
          "createdAt": "2025-06-30T13:29:29.517000+00:00",  
          "sizeInGb": 40,  
          "isSystemDisk": true,  

```

```
        "iops": 3000,
        "path": "/dev/sda1",
        "attachedTo": "Ubuntu-1",
        "attachmentState": "attached"
    }
],
    "ramSizeInGb": 1.0
},
"networking": {
    "monthlyTransfer": {
        "gbPerMonthAllocated": 2048
    },
    "ports": [
        {
            "fromPort": 80,
            "toPort": 80,
            "protocol": "tcp",
            "accessFrom": "Anywhere (0.0.0.0/0 and ::/0)",
            "accessType": "public",
            "commonName": "",
            "accessDirection": "inbound",
            "cidrs": [
                "0.0.0.0/0"
            ],
            "ipv6Cidrs": [
                "::/0"
            ],
            "cidrListAliases": []
        },
        {
            "fromPort": 22,
```

```
        "toPort": 22,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0 and ::/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound",
        "cidrs": [
            "0.0.0.0/0"
        ],
        "ipv6Cidrs": [
            "::/0"
        ],
        "cidrListAliases": []
    },
    {
        "fromPort": 8112,
        "toPort": 8112,
        "protocol": "tcp",
        "accessFrom": "Anywhere (0.0.0.0/0 and ::/0)",
        "accessType": "public",
        "commonName": "",
        "accessDirection": "inbound",
        "cidrs": [
            "0.0.0.0/0"
        ],
        "ipv6Cidrs": [
            "::/0"
        ],
        "cidrListAliases": []
    }
]
```

```

    },
    "state": {
      "code": 16,
      "name": "running"
    },
    "username": "ubuntu",
    "sshKeyName": "LightsailDefaultKeyPair",
    "metadataOptions": {
      "state": "applied",
      "httpTokens": "required",
      "httpEndpoint": "enabled",
      "httpPutResponseHopLimit": 2,
      "httpProtocolIpv6": "disabled"
    }
  }
]
}

```

Great! I can see that the Lightsail instance "Ubuntu-1" does exist in your AWS account in the `us-east-2` region. However, I notice there's a discrepancy between your Terraform configuration and the actual instance: Your Terraform config shows:

- `bundle_id = "nano_2_0"`

The actual instance has:

- `bundleId = "micro_3_0"`

This mismatch could be causing the import issue. Let me also check your AWS provider configuration to ensure it's set to the correct region:

```
aws configure get region
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulumi-aws-1/lightsail$ aws configure get region
```

```
us-west-2
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulumi-aws-1/lightsail$
```

Found the issue! Your AWS CLI is configured to use `us-west-2` region, but your Lightsail instance is in `us-east-2`. This region mismatch is causing the import to fail. Let me also check if you have a provider configuration in your Terraform files:

```
Grepped provider.*aws
```

Let me check if there's a provider configuration block:

```
Grepped provider "aws"
```

Now I understand the problem. Here are the issues and solutions:

## Issues Found:

1. Region Mismatch: Your AWS CLI is configured for `us-west-2`, but your Lightsail instance is in `us-east-2`
1. Bundle ID Mismatch: Your Terraform config specifies `nano_2_0`, but the actual instance uses `micro_3_0`
1. Missing Provider Configuration: No explicit AWS provider region is specified

## Solutions:

Let me fix your Terraform configuration to match the actual instance and specify the correct region:

```
main.tf
```

```
+14-1
```

```
resource "aws_lightsail_instance" "my_lightsail" {  
  terraform {  
    required_providers {  
      aws = {  
        source = "hashicorp/aws"  
        version = "~> 5.0"  
      }  
    }  
  }  
}
```



```
provider "aws" {
```

Now let's try the import again. First, let me initialize Terraform to ensure the provider is properly configured:

```
terraform init
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulummi-aws-1/lightsail$ terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Reusing previous version of hashicorp/aws from the dependency lock file
```

```
|
```

```
| Error: Failed to query available provider packages
```

```
|
```

```
| Could not retrieve the list of available versions for provider hashicorp/aws: locked  
| provider registry.terraform.io/hashicorp/aws 6.9.0 does not match configured version  
| constraint ~> 5.0; must use terraform init -upgrade to allow selection of new  
versions
```

```
|
```

```
| To see which modules are currently depending on hashicorp/aws and what versions are  
| specified, run the following command:
```

```
|     terraform providers
```

```
|
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulummi-aws-1/lightsail$
```

There's a version constraint issue. Let me check what version is currently locked and update the configuration accordingly:

Read `terraform.lock.hcl`

I see the issue. The locked version is 6.9.0, but I specified ~> 5.0. Let me update the version constraint to match:

`main.tf`

```
+1-1
```

```
source = "hashicorp/aws"
```

```
version = "~> 5.0"
```

```
}  
  
version = "~> 5.0"  
source  = "hashicorp/aws"  
version = "~> 6.0"  
  
}
```

Now let's try initializing Terraform again:

```
terraform init
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulum-i-aws-1/lightsail$ terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Reusing previous version of hashicorp/aws from the dependency lock file
```

```
- Using previously-installed hashicorp/aws v6.9.0
```





```
Terraform has been successfully initialized!
```

```
You may now begin working with Terraform. Try running "terraform plan" to see  
any changes that are required for your infrastructure. All Terraform commands  
should now work.
```

```
If you ever set or change modules or backend configuration for Terraform,  
rerun this command to reinitialize your working directory. If you forget, other  
commands will detect it and remind you to do so if necessary.
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulum-i-aws-1/lightsail$
```

Perfect! Now let's try the import again. The configuration should now work since we've:

1.  Specified the correct region (us-east-2) in the provider
1.  Updated the bundle\_id to match the actual instance (micro\_3\_0)
1.  Fixed the version constraint to match the locked version
1.  Initialized Terraform successfully

```
terraform import
```

```
aws_lightsail_instance.
```

```
my_lightsail Ubuntu-1
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulum-i-aws-1/lightsail$ terraform import  
aws_lightsail_
```

```
angel.cruz@angel-ubuntu:~/sandbox/repos/pulumi-aws-1/lightsail$ terraform import
aws_lightsail_
instance.my_lightsail Ubuntu-1
aws_lightsail_instance.my_lightsail: Importing from ID "Ubuntu-1"...
aws_lightsail_instance.my_lightsail: Import prepared!

  Prepared aws_lightsail_instance for import
aws_lightsail_instance.my_lightsail: Refreshing state... [id=Ubuntu-1]
Import successful!

The resources that were imported are shown above. These resources are now in
your Terraform state and will henceforth be managed by Terraform.

angel.cruz@angel-ubuntu:~/sandbox/repos/pulumi-aws-1/lightsail$
```

Excellent! The import was successful. Let's verify the current state and run a plan to see if there are any differences between your configuration and the actual instance:

```
terraform plan
```

[Follow Allowlist](#)

[Open in Terminal](#)

Perfect! 🎉





## Summary

The import was successful! Here's what was causing the issue and how we fixed it:

### Root Causes:

1. **Region Mismatch:** Your AWS CLI was configured for `us-west-2`, but the Lightsail instance exists in `us-east-2`
1. **Bundle ID Mismatch:** Your Terraform config had `nano_2_0`, but the actual instance uses `micro_3_0`
1. **Missing Provider Configuration:** No explicit region was specified in the Terraform provider

### What We Fixed:

1.  Added explicit AWS provider configuration with `region = "us-east-2"`
1.  Updated `bundle_id` from `"nano_2_0"` to `"micro_3_0"` to match the actual instance
1.  Fixed the provider version constraint to match your locked version (`~> 6.0`)
1.  Successfully imported the existing Lightsail instance


## Current Status:


- The instance "Ubuntu-1" is now successfully imported into your Terraform state
- `terraform plan` shows no changes needed - your configuration matches the actual infrastructure
- You can now manage this Lightsail instance through Terraform going forward

Your Lightsail instance is now under Terraform management! 🚀

---

## Create Static IP for existing instance





### Create a static IP address

A static IP is a fixed, public IP address that you can assign and reassign to your instances.

#### Select your static IP location

Select a Region


☒  **Ohio**  
us-east-2


☐  **Virginia**  
us-east-1

☐  **Montreal**  
ca-central-1

## Attach to an instance

Attaching a static IP replaces that instance's dynamic IP address.

 **Ubuntu-1**  
1 GB RAM, 2 vCPUs, 40 GB SSD  
Ubuntu

Cancel 

## Identify your static IP

Your Lightsail resources must have unique names.

**Static IP addresses are free only while attached to an instance.**  
You can manage five at no additional cost.

Create


## Public static IP address


This static IP is available for public connection worldwide.

3.130.87.185

## Attach to an instance

Attaching a static IP replaces that instance's dynamic IP address.

 **Ubuntu-1**  
1 GB RAM, 2 vCPUs, 40 GB SSD  
Ubuntu

Detach 

**IP ADDRESS:**

**3.130.87.185**