

Placement Empowerment Program

Cloud Computing and DevOps Centre

Implement DNS for Your Application: Set up a DNS record to map your web application's IP or load balancer to a domain name.

Name KEERTHIKA ANANTHAN
Department: AML

Introduction

In cloud computing, establishing a proper **Domain Name System (DNS)** configuration is essential for ensuring that applications are accessible over the internet. AWS offers **Route 53**, a highly available and scalable DNS web service, which allows users to manage domain names and point them to AWS resources like EC2 instances. This Proof of Concept (PoC) demonstrates the process of creating and configuring a DNS record in **AWS Route 53**, pointing it to a web server hosted on an **EC2 instance**, thus making the web application accessible via a custom domain name.

Overview

This PoC involves:

- 1. Launching an EC2 instance** to serve a web application.
- 2. Setting up a web server** (Apache or Nginx) on the EC2 instance to host the application.
- 3. Creating a hosted zone** in AWS Route 53 for a custom domain (e.g., myapp.local).
- 4. Configuring an A record** in Route 53 to map the domain to the EC2 instance's public IP.
- 5. Modifying the hosts file** on a local machine to test the custom domain name before making it publicly available.
- 6. Testing the configuration** by accessing the application using the custom domain name.

Objective

The main objectives of this PoC are:

- 1. Familiarize with Route 53 DNS configuration:** Understand how to use AWS Route 53 to manage domain names and map them to cloud resources.
- 2. Learn EC2 setup and configuration:** Gain hands-on experience in launching an EC2 instance and configuring a web server to serve a web application.
- 3. Enable custom domain access:** Configure a custom domain name to point to the EC2 instance, ensuring that the web application is easily accessible through the domain.
- 4. Test and verify the configuration:** Ensure that the domain correctly points to the EC2 instance by testing it in a browser and troubleshooting any issues.

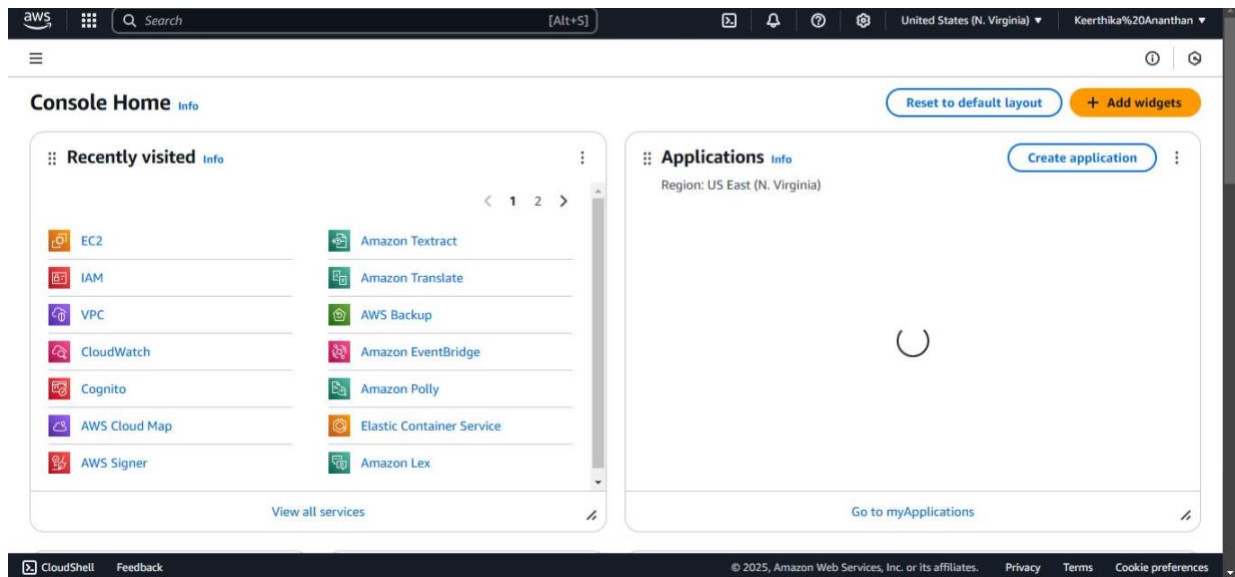
Importance

- 1. Improves Web Access:** Provides a user-friendly way to access applications via a custom domain.
- 2. Scalable DNS Solution:** Route 53 offers scalable and reliable DNS management.
- 3. Hands-on Cloud Skills:** Essential for cloud architects and developers to work with AWS services.
- 4. Cost-Effective Testing:** Utilizes AWS Free Tier for testing without incurring costs.

Step-by-Step Overview

Step 1:

1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.



Step 2:

Launch an instance named **route instance** .

Configure Security Group:

Add a **new security group** with a rule for **HTTP** (port 80) and **SSH** (port 22).

For HTTP, set the source to **Anywhere (0.0.0.0/0)** to allow access via the web.

For SSH, set the source to your **IP** (recommended for security), or use **Anywhere** for now.

Create a Key Pair (or use an existing one) and download the key file (.pem).

Review and click on launch

aws

Search

[Alt+S]

United States (N. Virginia)

Keerthika%20Ananthan

EC2 > Instances

EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Instances (1) Info

Last updated 1 minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

Instance ID = i-0351c2f3661ec46ba

Clear filters

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	unk1234	i-0351c2f3661ec46ba	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d

Select an instance

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates.

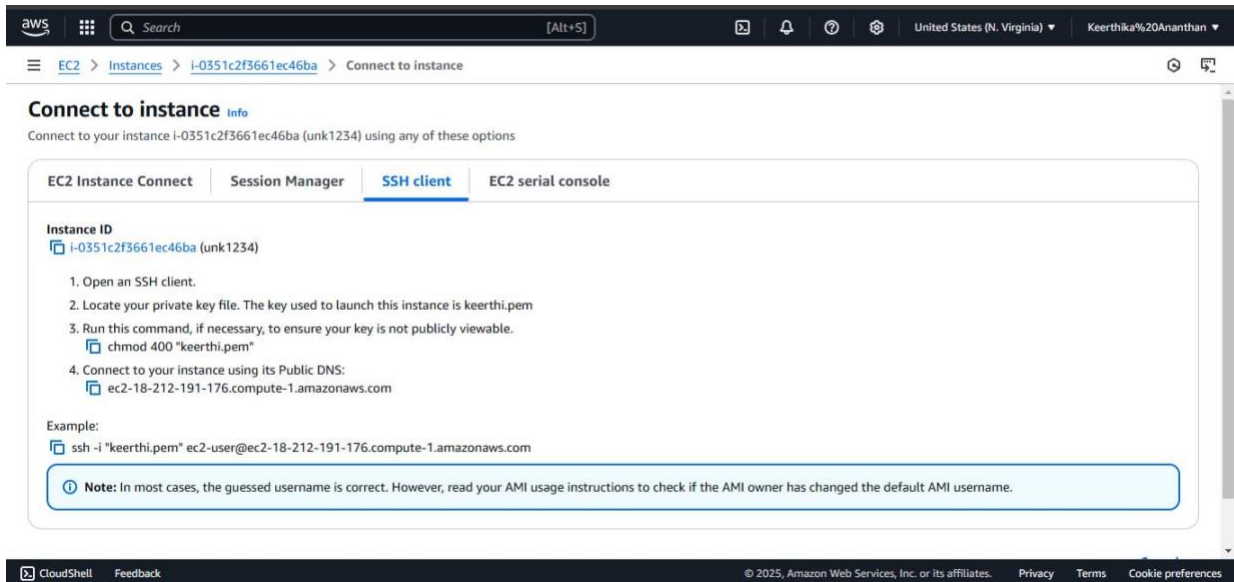
Privacy

Terms

Cookie preferences

Step 3:

Click the 'Connect' option on your launched instance, go to the SSH client section, and copy the command provided under the 'Example' section.



Step 4:

Open PowerShell, navigate to the 'Downloads' directory where the downloaded key pair is located using the **cd Downloads** command

Paste the command copied from the EC2 Connect's SSH client section, replace the key pair name with your downloaded key (e.g., new.pem), press Enter, and type 'yes' when prompted.

```

C:\Users\admin>cd downloads
C:\Users\admin\Downloads>ssh -i "instance.pem" ec2-user@ec2-98-82-121-112.compute-1.amazonaws.com
The authenticity of host 'ec2-98-82-121-112.compute-1.amazonaws.com (98.82.121.112)' can't be established.
ECDSA key fingerprint is SHA256:VWntr7fdbUhzA26NgijNADT0+V4mZ67Qxe0P+mDFUag.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-98-82-121-112.compute-1.amazonaws.com,98.82.121.112' (ECDSA) to the list of known hosts.

#_
~\##### Amazon Linux 2023
~\#####
~\#####
~\#####
~\#/
~\V' -> https://aws.amazon.com/linux/amazon-linux-2023
~\_
~\_
~\m/'
[ec2-user@ip-172-30-5-175 ~]$

```

Install Apache :

```
sudo yum install httpd -y
```

```
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd downloads

C:\Users\admin\Downloads>ssh -i "instance.pem" ec2-user@ec2-98-82-121-112.compute-1.amazonaws.com
The authenticity of host 'ec2-98-82-121-112.compute-1.amazonaws.com (98.82.121.112)' can't be established.
ECDSA key fingerprint is SHA256:VWnNr7fdbUhZa26NgijNADT0+V4mZ67Qxe0P+mDFUag.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-98-82-121-112.compute-1.amazonaws.com,98.82.121.112' (ECDSA) to the list of known hosts
```

The screenshot shows the initial steps of establishing an SSH connection from a Windows command prompt to an Amazon Linux EC2 instance. The user navigates to the 'downloads' directory and runs an ssh command with a private key file named 'instance.pem'. The system prompts for confirmation to add the host's ECDSA fingerprint to the local known hosts database.

Start Apache:

Start Apache:

```
sudo systemctl start httpd
```

Make Apache start on boot:

```
sudo systemctl enable httpd
```



```

Installing      : mod_lua-2.4.62-1.amzn2023.x86_64          10/1
Installing      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 11/1
Installing      : httpd-2.4.62-1.amzn2023.x86_64          12/1
Running scriptlet: httpd-2.4.62-1.amzn2023.x86_64          12/1
Verifying       : apr-1.7.5-1.amzn2023.0.4.x86_64         1/1
Verifying       : apr-util-1.6.3-1.amzn2023.0.1.x86_64    2/1
Verifying       : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/1
Verifying       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 4/1
Verifying       : httpd-2.4.62-1.amzn2023.x86_64          5/1
Verifying       : httpd-core-2.4.62-1.amzn2023.x86_64     6/1
Verifying       : httpd-filesystem-2.4.62-1.amzn2023.noarch 7/1
Verifying       : httpd-tools-2.4.62-1.amzn2023.x86_64    8/1
Verifying       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64   9/1
Verifying       : mailcap-2.1.49-3.amzn2023.0.3.noarch    10/1
Verifying       : mod_http2-2.0.27-1.amzn2023.0.3.x86_64  11/1
Verifying       : mod_lua-2.4.62-1.amzn2023.x86_64        12/1

Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.62-1.amzn2023.x86_64          httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch httpd-tools-2.4.62-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64   mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.27-1.amzn2023.0.3.x86_64  mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[ec2-user@ip-172-30-5-175 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-30-5-175 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-30-5-175 ~]$

```

Step 7:

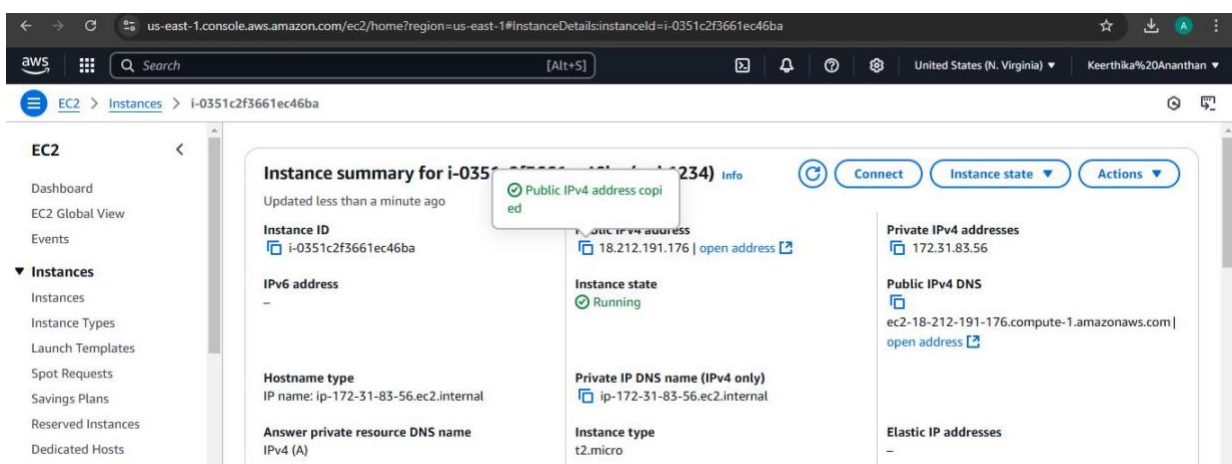
Verify Apache is running:

In your browser, enter the **EC2 public IP** (e.g., <http://<your-ec2-public-ip>>).

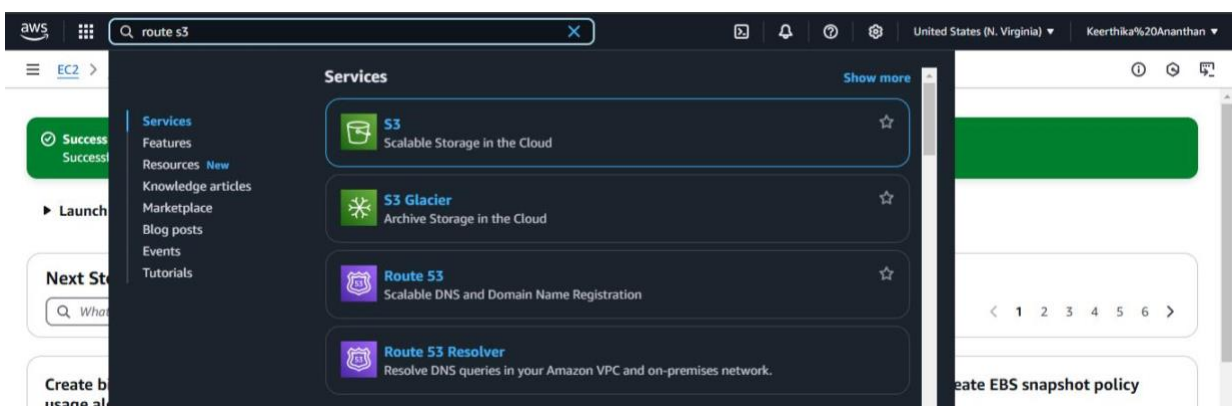
You should see the **Apache default page**. This means your EC2 instance is set up to serve websites.



It works!

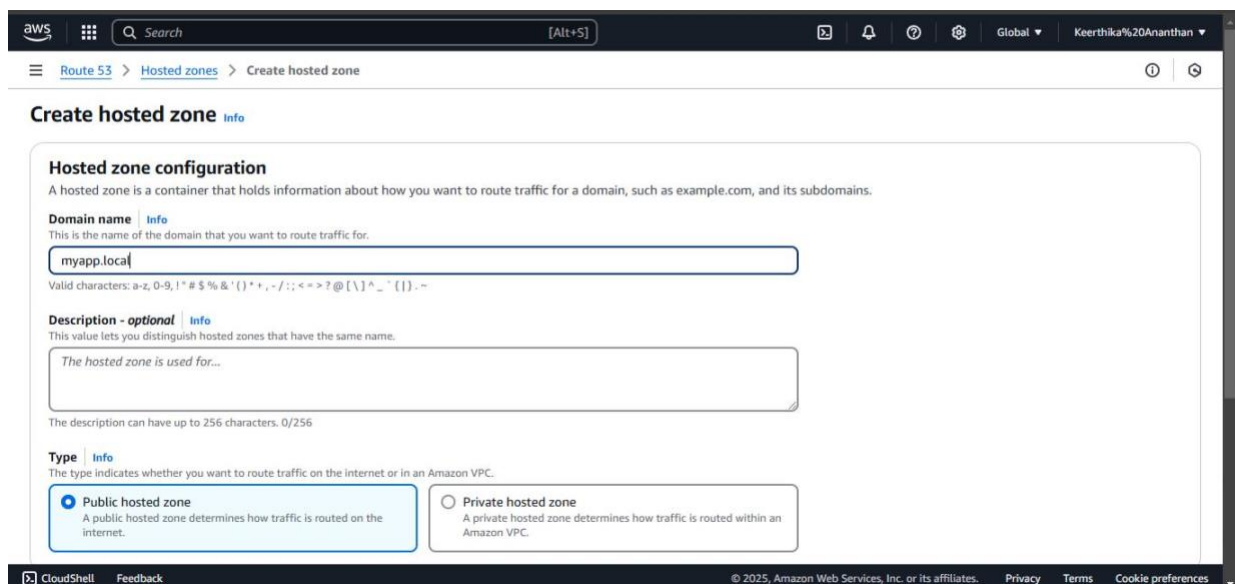


Step 8 In the AWS Console, search for **Route 53** and select it.



Step 9:

1. Click on **Create hosted zone**.
2. Enter a **Domain Name** (e.g., myapp.local).
3. Set the **Type** to **Public Hosted Zone**.
4. Click **Create hosted zone**.



The screenshot shows the AWS Management Console interface for creating a hosted zone. The breadcrumb navigation at the top indicates the path: Route 53 > Hosted zones > Create hosted zone. The main heading is 'Create hosted zone' with an 'Info' link. Below this is a section titled 'Hosted zone configuration' with a descriptive paragraph. The 'Domain name' field is required and contains 'myapp.local'. Below it, a note specifies valid characters. The 'Description - optional' field is empty, with a note about its purpose and a character limit. The 'Type' section has two radio buttons: 'Public hosted zone' (selected) and 'Private hosted zone'. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services.

Create hosted zone [Info](#)

Hosted zone configuration
A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name [Info](#)
This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! * # \$ % & ' () ^ + , - / : ; < = > ? @ [\] ^ _ ` { } . ~

Description - optional [Info](#)
This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 0/256

Type [Info](#)
The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

☒ **Public hosted zone**
A public hosted zone determines how traffic is routed on the internet.

☐ **Private hosted zone**
A private hosted zone determines how traffic is routed within an Amazon VPC.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 10:

1. In your hosted zone, click **Create Record**.
2. **Record Name:** Leave it empty for the root domain (myapp.local),
3. **Record Type:** Select **A – IPv4 address**.
4. **Value:** Enter the **Public IP** of your EC2 instance.
5. **TTL:** Set to 60 seconds.
6. Click **Create records**.

The screenshot shows the AWS Route 53 console's 'Create record' page. The breadcrumb trail at the top indicates the path: Route 53 > Hosted zones > myapp.local > Create record. The page is titled 'Record 1' and includes a 'Delete' button. The 'Record name' field contains 'subdomain' and the 'Record type' is set to 'A – Routes traffic to an IPv4 address and some AWS resources'. The 'Value' field contains the IP address '192.0.2.235'. The 'TTL (seconds)' is set to '60', with buttons for '1m', '1h', and '1d' available. The 'Routing policy' is set to 'Simple routing'. A note below the TTL field states: 'Recommended values: 60 to 172800 (two days)'. At the bottom right, there is an 'Add another record' button. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

Record 1 Delete

Record name [Info](#) myapp.local

Keep blank to create a record for the root domain.

☐ Alias

Record type [Info](#) A – Routes traffic to an IPv4 address and some AWS resources

Value [Info](#)

Enter multiple values on separate lines.

TTL (seconds) [Info](#) 1m 1h 1d **Routing policy** [Info](#) Simple routing

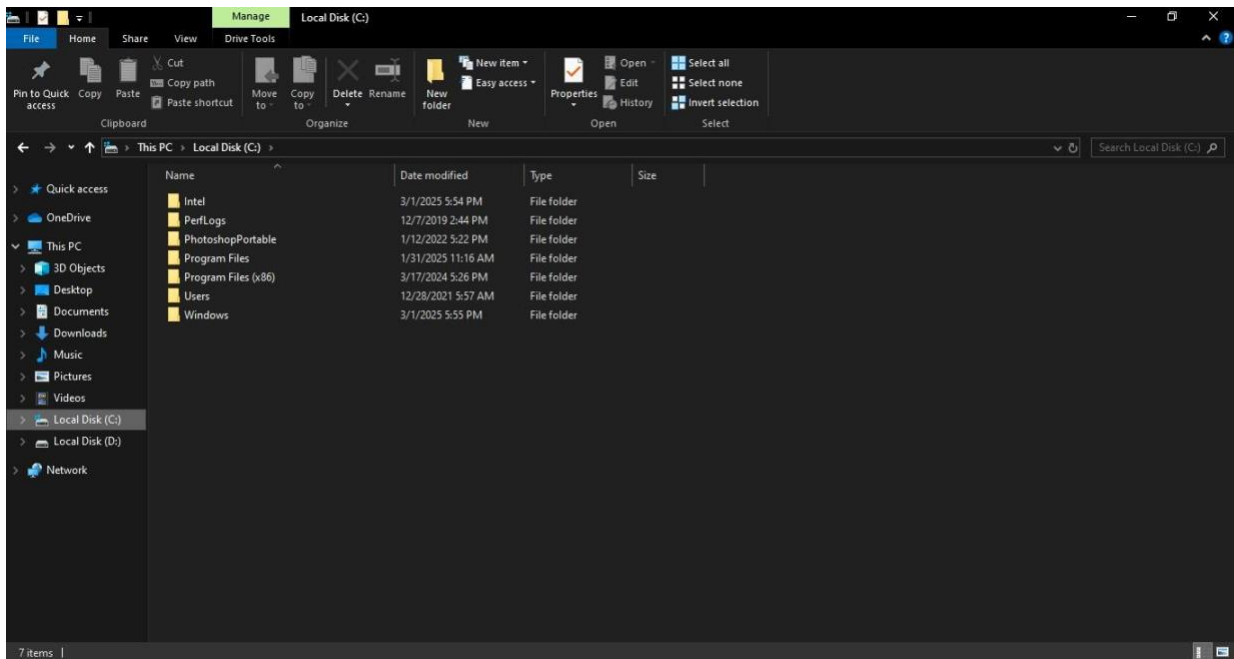
Recommended values: 60 to 172800 (two days)

Add another record

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 11:

1. Go to **File Explorer > Open**.
2. Navigate to: `C:\Windows\System32\drivers\etc`.
3. In the file name field, type `hosts` and press **Enter**.



Step 12:

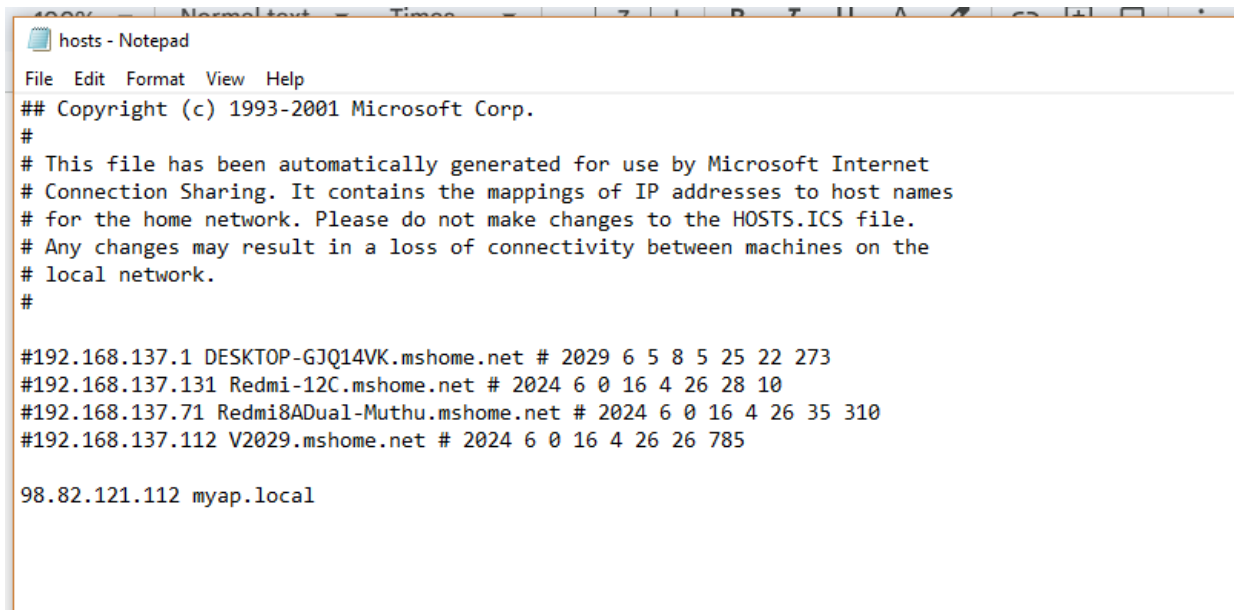
1. At the bottom of the file, add:

`<Your EC2 Public IP> myapp.local`

Replace `<Your EC2 Public IP>` with the public IP you copied.

(Eg: `52.91.79.69 myapp.local`)

2. Save the file and close Notepad.



```
hosts - Notepad
File Edit Format View Help
## Copyright (c) 1993-2001 Microsoft Corp.
#
# This file has been automatically generated for use by Microsoft Internet
# Connection Sharing. It contains the mappings of IP addresses to host names
# for the home network. Please do not make changes to the HOSTS.ICS file.
# Any changes may result in a loss of connectivity between machines on the
# local network.
#
#192.168.137.1 DESKTOP-GJQ14VK.mshome.net # 2029 6 5 8 5 25 22 273
#192.168.137.131 Redmi-12C.mshome.net # 2024 6 0 16 4 26 28 10
#192.168.137.71 Redmi8ADual-Muthu.mshome.net # 2024 6 0 16 4 26 35 310
#192.168.137.112 V2029.mshome.net # 2024 6 0 16 4 26 26 785
98.82.121.112 myap.local
```

Step 12:

Open your **web browser**.

Type myapp.local in the address bar and press **Enter**.

You should see the Apache default page



Outcome

By completing this PoC of configuring DNS for your application using AWS Route 53 and EC2, you will:

1. Launch and configure an EC2 instance with a web server (e.g., Apache or Nginx).
2. Deploy a sample web application on the EC2 instance and ensure it is accessible via the instance's public IP.
3. Create a hosted zone in AWS Route 53 for DNS management.
4. Set up an A record in Route 53 to map your custom domain (e.g., myapp.local) to the EC2 instance's public IP.
4. Modify the local hosts file on your system to resolve the custom domain name locally for testing.
5. Successfully access your web application using the custom domain name in a browser.