# Task 4: DNS + HTTPS Web Server Setup

This task involves setting up a local DNS server using BIND9 inside Docker and linking it to a secure Apache-based HTTPS web server. The aim is to resolve a custom domain (www.atharvapurushe.com) to the local HTTPS server and test the complete resolution + connection path.

## 🧱 Docker Setup Summary

### 1. Create a Custom Docker Network

docker network create dnsnet

### 2. Run DNS Server (BIND9)

docker run -d --name dns-server \  
 -p 53:53/tcp -p 53:53/udp \  
 --network dnsnet \  
 internetsystemsconsortium/bind9:9.18

### 3. Run HTTPS Web Server (Custom Apache)

docker run -d --name atharva-apache \  
 -p 8080:443 \  
 --network dnsnet \  
 atharva-secure-apache

### 4. Run Netshoot Container for Network Debugging

docker run -it --network dnsnet --name netshoot nicolaka/netshoot

## 🔎 Network Inspection

### Inspect Docker Network and IP Addresses:

docker network inspect dnsnet

Key Output:

"atharva-apache" IP: 172.19.0.2  
"dns-server" IP: 172.19.0.4  
"netshoot" IP: 172.19.0.3

### Inspect Running Containers

docker ps -a

Shows all containers and their statuses.

## ⚙️ DNS Configuration (Inside BIND9)

Inside the container:

docker exec -it dns-server /bin/sh

BIND config placed in:

/etc/bind/named.conf  
/etc/bind/zones/db.atharvapurushe.com

Zone config example:

zone "atharvapurushe.com" {  
 type master;  
 file "/etc/bind/zones/db.atharvapurushe.com";  
};

Zone DB:

www IN A 172.19.0.2

## 🧪 DNS Resolution Tests

### Using nslookup:

nslookup www.atharvapurushe.com 127.0.0.1  
nslookup www.atharvapurushe.com 172.19.0.4

**Common Error:**

DNS request timed out.  
\*\*\* UnKnown can't find www.atharvapurushe.com: Non-existent domain

## 🌐 HTTPS Access Tests

### Curl to HTTPS Apache Web Server:

curl https://atharvapurushe.com:8080 --insecure

Output:

<!DOCTYPE html>  
<html>...</html>

### Curl with Custom Headers (Windows escaped quotes):

curl ^"https://atharvapurushe.com:8080/^" ^  
 -H ^"User-Agent: ...^" ^  
 --insecure

Success!

### Failed attempts:

curl www.atharvapurushe.com  
# -> Could not resolve host  
curl atharvapurushe.com:8080  
# -> Bad Request (sent HTTP to an HTTPS server)

## 🛰️ Tracing Network

tracert atharvapurushe.com

Shows local resolution to:

kubernetes.docker.internal [127.0.0.1]

## 🔍 Monitoring DNS and HTTPS Requests

Inside nicolaka/netshoot container:

tcpdump -i eth0 port 53 or port 443

Captured packets like:

IP 172.19.0.1.51573 > dns-server.dnsnet.53: PTR? 1.0.0.127.in-addr.arpa.

## ✅ Key Learnings

* tcpdump helps capture low-level packet traffic.
* DNS resolution must succeed before curl or browser requests work.
* Apache running HTTPS on port 443 must be accessed via https:// (not plain HTTP).
* Use --insecure for self-signed certs in curl.
* All containers must be connected to the same custom network (dnsnet).

## 🧹 Cleanup Tips

docker stop dns-server atharva-apache netshoot

docker rm dns-server atharva-apache netshoot

docker network rm dnsnet