Here's a detailed step-by-step summary of everything we need to implement DNS-layer security using Google Cloud:

Step 1: Set Up Your Google Cloud Account

1. **Signed Up for Google Cloud:**
   - Created an account and set up billing on the Google Cloud Platform. Google Cloud offers some free credits to start.

Step 2: Create a New Project

1. **Created a New Project:**
   - In the Google Cloud Console, clicked on the project drop-down menu at the top of the page and selected "New Project."
   - Named the project `DNS-Layer-Security` and selected the billing account.
   - Clicked "Create."

Step 3: Enable Necessary APIs

1. **Enabled Cloud DNS API:**
   - Navigated to `APIs & Services > Library`.
   - Searched for "Cloud DNS API" and clicked on it.
   - Clicked "Enable."
2. **Enabled Compute Engine API:**
   - Similarly, searched for "Compute Engine API" and enabled it.

Step 4: Create a VM Instance

1. **Created a VM Instance:**
   - Navigated to `Compute Engine > VM Instances`.
   - Clicked "Create Instance."
   - Named it `dns-vm-instance` with region `europe-west3-a`.
   - Chose the machine type `e2-micro`.
   - Used the snapshot from `dns-vm-instance` to set up a new instance `dns-f1-vm-instance` with machine type `f1-micro`.
2. **Noted the External IP Address:**
   - For example, the external IP address was `34.80.4.242`.

Step 5: Create a DNS Zone

1. **Navigated to Cloud DNS:**
   - Went to `Network services > Cloud DNS`.
   - Clicked "Create Zone."
2. **Created a DNS Zone:**
   - Named the zone `my-domain-zone`.
   - Used the DNS name `urldetection.ddns.net`.

Step 6: Add DNS Records

1. **Added an A Record:**

- In the Cloud DNS page, clicked on the zone name `my-domain-zone`.
- Clicked "Add Record Set."
- Added an A record pointing `urldetection.ddns.net` to the external IP address `34.80.4.242`.

Step 7: Update Your Domain Registrar

1. **Updated Nameservers:**
   - Logged into the NO-IP account where the domain `urldetection.ddns.net` was registered.
   - Found the DNS settings section and replaced the current nameservers with the Google Cloud nameservers.
   - Saved the changes (DNS propagation might take some time, up to 48 hours).

Step 8: Set Up DNS Handling on the VM

1. **Installed BIND9:**
   - Connected to the Google Cloud VM instance using SSH.
   - Ran the following commands to install BIND9:

```
sudo apt update
sudo apt install bind9 bind9utils bind9-doc -y
```

2. **Configured BIND9:**
   - Edited the BIND9 main configuration file:

```
sudo nano /etc/bind/named.conf.options
```

   - Updated the configuration to allow queries from any IP:

```
plaintext
Copy code
options {
    directory "/var/cache/bind";
    forwarders {
        8.8.8.8; // Google's DNS
        8.8.4.4; // Google's DNS
    };
    dnssec-validation auto;
    listen-on { any; };
    listen-on-v6 { none; };
};
```

   - Saved and closed the file.
3. **Defined the Zone in BIND9:**
   - Edited the BIND9 local configuration file:

```
sudo nano /etc/bind/named.conf.local
```

   - Added the zone configuration:

```
plaintext
Copy code
zone "urldetection.ddns.net" {
```

```
    type master;
    file "/etc/bind/db.urldetection.ddns.net";
};
```

- Saved and closed the file.

4. **Created a Zone File:**
   - Created the zone file:

```
sudo nano /etc/bind/db.urldetection.ddns.net
```

   - Added the following content:

```
plaintext
Copy code
$TTL    604800
@       IN      SOA     ns1.urldetection.ddns.net.
admin.urldetection.ddns.net. (
                        2         ; Serial
                   604800         ; Refresh
                    86400         ; Retry
                  2419200         ; Expire
                   604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.urldetection.ddns.net.
ns1     IN      A       34.80.4.242
@       IN      A       34.80.4.242
```

   - Saved and closed the file.

5. **Restarted BIND9:**
   - Restarted BIND9 to apply the changes:

```
bash
Copy code
sudo systemctl restart bind9
```

Step 9: Test Your DNS Server

1. **Tested DNS Resolution:**
   - Verified the DNS setup using the `dig` command:

```
dig @34.80.4.242 urldetection.ddns.net
```

   - Received a successful response indicating correct DNS setup.

Step 10: Configure Firewall Rules

1. **Set Up Firewall Rules:**
   - Added rules to allow DNS and SSH traffic using UFW:

```
sudo ufw allow 53
sudo ufw allow 22
sudo ufw enable
```

   - Verified UFW status and rules with:

```
sudo ufw status
```

Step 11: Verify Everything is Working

1. **Checked System Logs:**
   - Monitored the syslog for any errors or blocked traffic:

     ```
     sudo tail -f /var/log/syslog
     ```

2. **Confirmed DNS Propagation:**
   - Used online tools like DNS Checker to verify DNS records propagation.

Final Configuration

- The `dns-f1-vm-instance` with machine type `f1-micro` is set up and running.
- BIND9 is configured and resolving `urldetection.ddns.net` to the correct IP address.
- Firewall rules are correctly configured to allow necessary traffic.

This detailed summary should help you follow the steps to implement DNS-layer security using Google Cloud effectively.