If your DNS provider is **AWS Route 53**, you can use the **DNS-01 challenge** with Certbot by leveraging the **Route 53 DNS plugin** to automate the DNS validation process. This means that Certbot can automatically create the necessary DNS records in your AWS Route 53 DNS configuration to prove that you own the domain.

Here’s how you can set it up:

**1. Install Certbot and the AWS Route 53 Plugin**

If you haven't already installed Certbot, do so using the following command:

sudo apt update

sudo apt install certbot

Next, install the Certbot Route 53 plugin, which allows Certbot to interact with Route 53 and automatically create DNS records.

sudo apt install python3-certbot-dns-route53

**2. Set Up AWS IAM Credentials**

You need to provide Certbot with AWS IAM credentials that have permission to create DNS records in Route 53. This can be done using an **AWS IAM user** with the necessary permissions.

**2.1. Create an IAM User with Route 53 Permissions:**

1. Go to the **AWS IAM Console** and create a new IAM user with **Programmatic access**.
2. Attach the following policy to the IAM user to grant permissions to modify Route 53 DNS records:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"route53:ChangeResourceRecordSets",

"route53:ListResourceRecordSets",

"route53:GetChange",

"route53:ListHostedZones"

],

"Resource": "\*"

}

]

}

1. After the IAM user is created, **save the access key and secret key**.

**2.2. Configure AWS Credentials for Certbot:**

To configure Certbot to use your AWS credentials, you can either:

* Set the environment variables for the AWS credentials:

export AWS\_ACCESS\_KEY\_ID="your-access-key-id"

export AWS\_SECRET\_ACCESS\_KEY="your-secret-access-key"

export AWS\_DEFAULT\_REGION="us-east-1" # or your desired region

* Or, you can create a **~/.aws/credentials** file with your credentials:

[default]

aws\_access\_key\_id = your-access-key-id

aws\_secret\_access\_key = your-secret-access-key

**3. Request the Wildcard Certificate with DNS-01 Challenge**

Now, you can use Certbot to request a wildcard certificate with DNS-01 validation using Route 53.

Run the following command to request the wildcard certificate:

sudo certbot certonly --dns-route53 -d "\*.m-khamisi.com" -d "m-khamisi.com"

This command will do the following:

* **--dns-route53**: Instructs Certbot to use the Route 53 DNS plugin for DNS-01 validation.
* **-d "\*.m-khamisi.com" -d "m-khamisi.com"**: Requests a wildcard certificate for \*.m-khamisi.com as well as the base domain m-khamisi.com.

Certbot will automatically:

1. Create the necessary DNS TXT records in Route 53 for domain validation.
2. Wait for DNS propagation and validate ownership of the domain.
3. Once validation is successful, Certbot will issue the certificate and store it in /etc/letsencrypt/live/m-khamisi.com/.

**4. Verify the Wildcard Certificate**

Once the certificate has been issued, you can verify the details of the certificate with:

openssl x509 -in /etc/letsencrypt/live/m-khamisi.com/fullchain.pem -text -noout

In the output, you should see the **Subject Alternative Name** (SAN) section, which will list \*.m-khamisi.com and m-khamisi.com.

**5. Install the Certificate on Your Servers**

You can now use the generated certificate on your Docker registry, GitLab, or any other services running under \*.m-khamisi.com.

For example, to use the certificate with Docker registry, you would set it up as follows:

services:

registry:

image: registry:2

environment:

REGISTRY\_HTTP\_TLS\_CERTIFICATE: /path/to/your/fullchain.pem

REGISTRY\_HTTP\_TLS\_KEY: /path/to/your/privkey.pem

volumes:

- /path/to/your/certificate:/etc/ssl/certs:ro

ports:

- "5000:5000"

**6. Test Certificate Renewal**

Certbot automatically sets up **auto-renewal** for your certificates. However, it's always a good idea to test the renewal process to ensure it works properly:

sudo certbot renew --dry-run

This command simulates the renewal process to ensure that Certbot can renew your wildcard certificate without any issues.

**Summary:**

1. **Install Certbot and Route 53 Plugin** on your server.
2. **Create an IAM user in AWS** with Route 53 permissions and configure the credentials on your server.
3. **Use Certbot to request the wildcard certificate** with the --dns-route53 flag for DNS-01 challenge.
4. **Verify the certificate** and ensure it's valid for both \*.m-khamisi.com and m-khamisi.com.
5. **Install the certificate on your servers**, and ensure auto-renewal is set up.

By using a wildcard certificate, you can cover all subdomains under m-khamisi.com, making it easier to manage your SSL certificates in the long term.

Let me know if you have any questions or need further assistance! 😊