Linux Academy

Live! Lab

Configuring VPC DNS

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- Access to an AWS Console is provided on the Live! Lab page, along with your login credentials
- Ensure you are using the N. Virginia region
- Labs will automatically end once the alloted amount of time finishes

Related Courses

AWS Certified Solutions Architect - Professional

Related Videos

Network Technologies

Working with VPC DNS

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Route 53 provides internal DNS to AWS VPCs. AWS also permits outside DNS for companies expanding out to their cloud systems. This lab reviews both way of managing DNS through AWS.

Log in to the AWS console using the credentials provided on the Live! Lab page.

Create an Instance

Navigate to the **EC2 Dashboard** and select the **Instances** page. Here we can see an already-existing instance, *On Premise DNS*. This instance is to be used for the on-premise DNS configuration; we still need to create a second instance for the lab.

Press **Launch Instance**. Select *Amazon Linux AMI*, then *t2.micro* as the **Instance Type**. Press **Next: Configure Instance Details**. Ensure that the **Network** address is set to *(10.0.0.0/16)*, and the **Subnets available** are *(10.0.0.0/24)* and *(10.0.2.0/24)*. Select *Enable* for the **Auto-assign Public IP** option. Press **Next: Add Storage**, then **Next: Tag Instance**. We do not need to adjust the storage settings.

For **Tag Instance**, use *client* as the **Value**, then press **Next: Configure Security Group**. Choose the **Select an existing security group** radio button, and select the one that contains your Linux Academy username. Press **Review and Launch**.

From here, review your selections if needed, then select **Launch**. A pop-up box appears, asking for a key pair. Choose **Create a new key pair** from the drop-down list. We choose to name ours *dnslab*. Download the key pair, then select **Launch Instances**.

To ensure the instance works, open up your terminal, and navigate to the folder that contains the key pair download. Update the permissions:

```
chmod 400 dnslab.pem
```

Return to the AWS console. Select the **client** instance, then click **Connect**. Connect to the instance using the provided example.

Be aware it may take multiple minutes for the server to be ready for use.

Configure Internal DNS

VPC Settings

From the top **Services** menu, navigate to **Networking**, then **VPC (Virtual Private Cloud)**. Select **Your VPC** from the menu.

We need to confirm that our VPC is properly configured to work with DNS — this means that DNS and

hostname both need to be enabled. Check the available VPC and go to **Actions**, **Edit DNS Resolution**. Ensure is it set to *Yes*. Do the same for **Edit DNS Hostnames**.

Set Up DNS

Navigate to **Route 53**, also located under **Networking**. You may receive a permissions error, but this can be ignored. From there, move to **Hosted zones** and select **Create Hosted Zone**.

In actual practice, the domain name used would be related to the organization. In this lab, we instead use the domain *awscloud.local*, which implies this is an internal zone. Set *Private Hosted Zone for Amazon VPC* for **Type**, and select the VPC in the *US East (N. Virginia)* region for **VPC ID**. To confirm this is correct, check the VPC name against the VPC ID in the VPC Dashboard. Select **Create**.

Create an Internal Record Set

From the dashboard of the hosted zone, select **Create Record Set**.

Set the **Name** to *client*. Next, set the **Value** to the *private IP of your client instance*, which can be found under the **EC2 Dashboard**, by clicking on **Instances** and selecting the *client* instance. Press **Create**.

Now, switch to your terminal. Ensure you are logged in to your *client* instance. Run:

```
nslookup client.awscloud.local
```

It should output results similar to:

```
Server: 10.0.0.2
Address: 10.0.0.2#53

Non-authoritative answer:
Name: client.awscloud.local
Address: 10.0.0.72
```

The IP in the address field should be the client instance's private IP that was added in the Route 53 client. awscloud.local rules.

Review the */etc/resolv.conf* file:

```
sudo cat /etc/resolv.conf
```

This outputs the default search settings and nameserver information. If secondary DNS records were added, it would failover to the second, should AWS's DNS cease to work.

Using On-Premise DNS

We want to allow our AWS applications to reference servers and instances in the "On Premise DNS" server. However, this cannot be done with Route 53, since those must originate within the AWS cloud.

Connect to your *client* from your terminal and view */etc/resolv.conf*:

```
sudo cat /etc/resolv.conf
```

From here we can see that we're still looking at Amazon's internal DNS. Any secondary servers added work as a failover, so if we wish to use our on-premise internal DNS, we need to replace our internal AWS DNS.

Open */etc/resolv.conf* and change the nameserver to match the on-premise private IP, which is *10.0.2.100*. Save and exit.

There is a DNS record located "on-premise" that we have available. To check that everything works, run:

```
nslookup ns1.onpremise.local
```

You should get a resolved return:

```
Server: 10.0.2.100
Address: 10.0.2.100#53
Name: ns1.onpremise.local
Address: 10.0.2.100
```

Should we `ping ns1.onpremise.local`, we notice that it is returning the address. However, should we `ping ns1` itself, it fails, because, by default, it assumes we are trying to run `ping ns1. ec2.local`. This can be changed by altering the */etc/resolv.conf* file:

```
; generated by /sbin/dhclient-script
search onpremise.local
nameserver 10.0.2.100
```

We can now `ping ns1`.

Set All Instances to Use On-Premise DNS

We need not use Route 53 DNS at all. To only use on-premise DNS, navigate to your **VPC dashboard** and select **DHCP Options Set**. Click **Create DCHP options set**.

Set the **Name tag** to *on-premise* and the **Domain name** to *onpremise.local*. For this example, the **Domain name server** should be set to our *On Premise* server's private IP: *10.0.2.100*. Press** Yes, Create**.

From the sidebar, choose **Your VPCs**. Select your VPC and select **Edit DHCP Options Set** from the **Actions** menu. Choose the one with the *on-premise* tag, and save.

The changes have yet to take effect, and do not do so automatically. For them to do so, either type `sudo reboot` in the terminal or do it through the EC2 console. When the instance is newly-booted, return to your terminal and `ping ns1`.