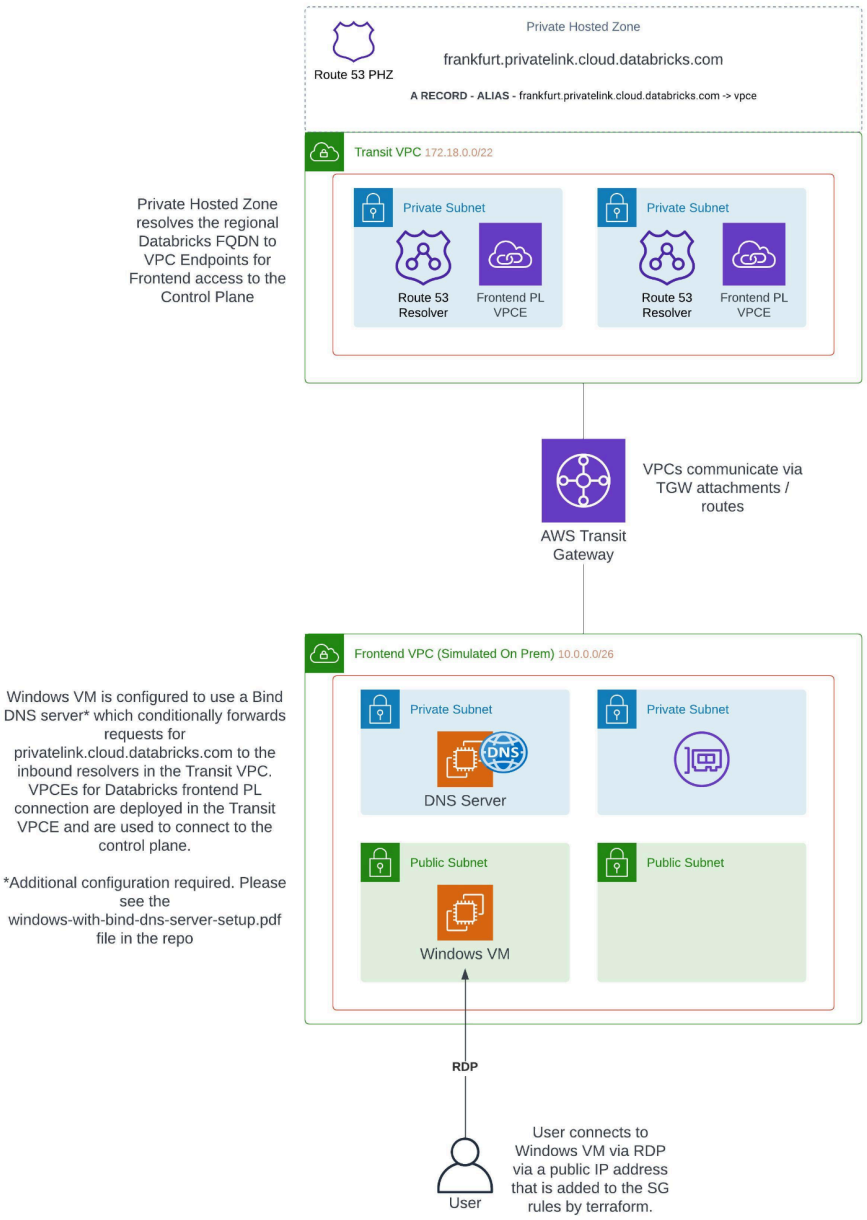
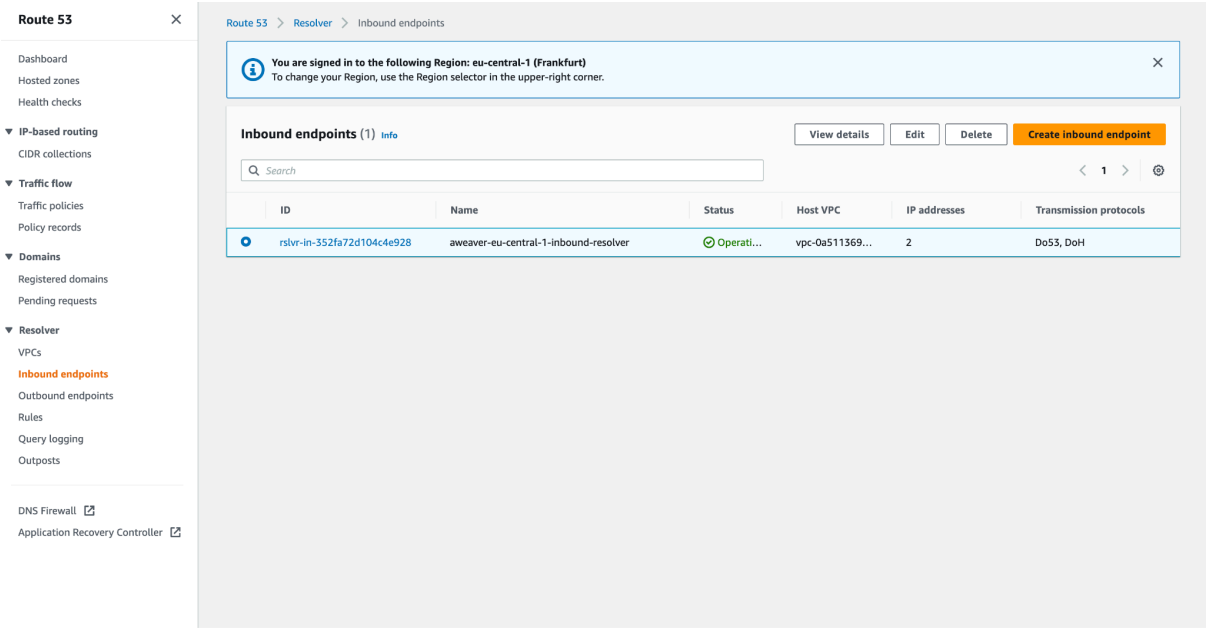
High Level Architecture:

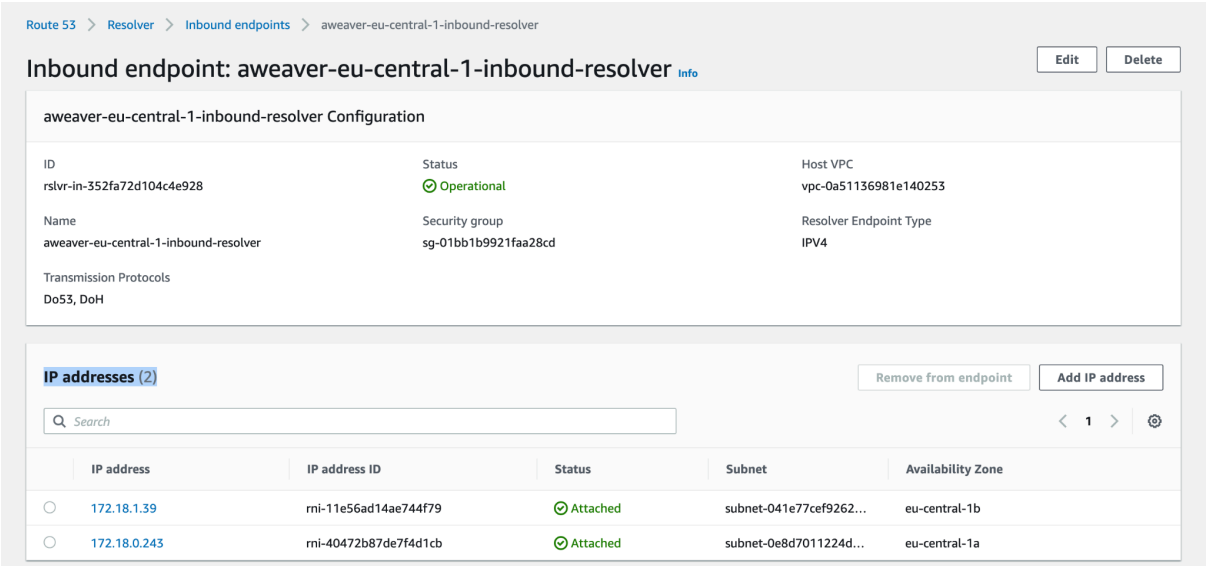


Steps to Reproduce:

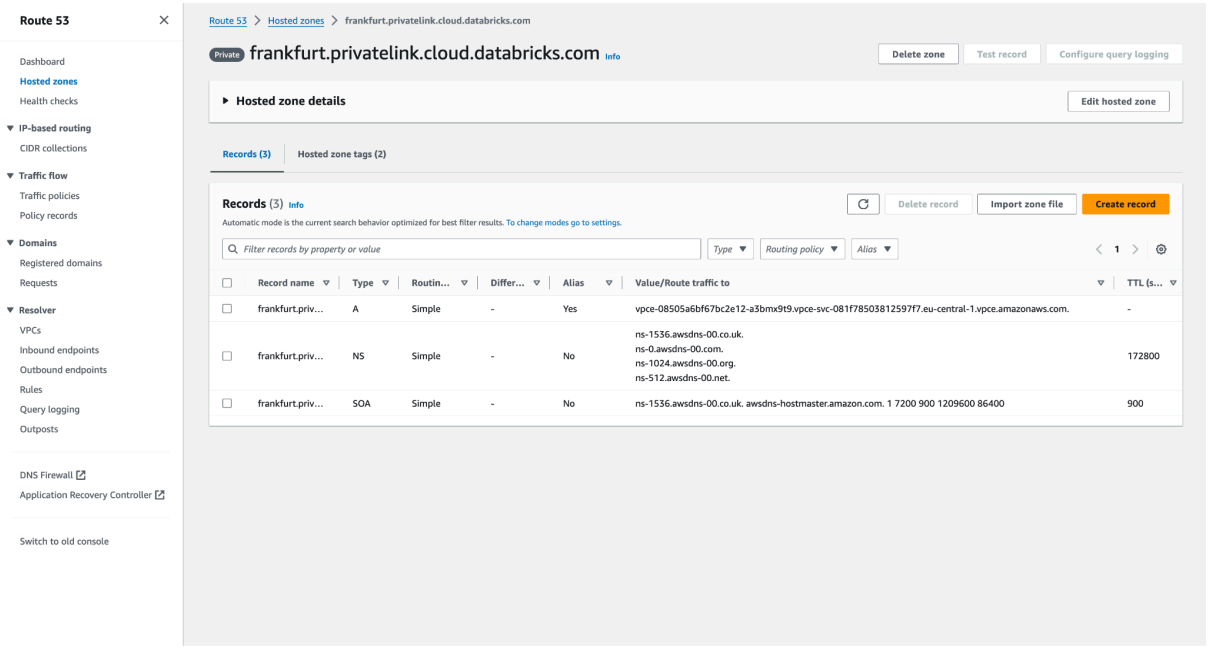
- Clone the repo
- Follow the steps in the README.md
- Login to https://accounts.cloud.databricks.com/, find and add yourself as a user of the workspace. Also make a note of the workspace FQDN
- In the AWS console, go to Route 53 and find the Inbound resolver endpoint created by Terraform:



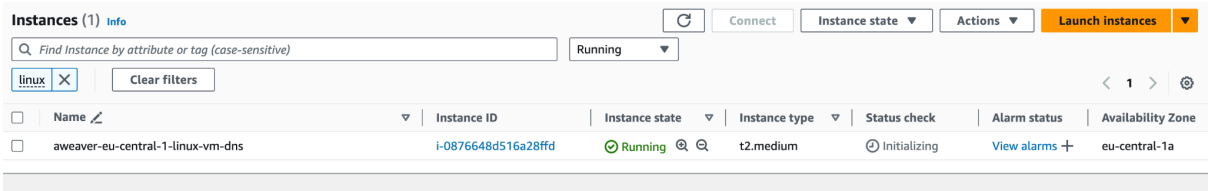
- Select View details and make a note of the IP addresses (2):



- Go to Hosted zones and find the Private Hosted Zone created by Terraform:



- Make a note of the Zone name and VPC Endpoint FQDN (the Value/Route traffic to for the Type A record). Now go to the VPC pages and find the VPC endpoint in question. Select Subnets and note the Private IP addresses allocated to the VPCEs.
- First thing we're going to do is set up our Bind DNS server, but since that is in a private subnet, we're going to need to connect to the Linux EC2 instance hosting it from our Windows VM.
- In AWS Select the Windows EC2 instance and then:
    - Connect
    - RDP client
- Use your EC2 private key to get the EC2 instance password
- Select Download remote desktop file and use it to RDP into the EC2 instance (your public IP address should be allow-listed via the `rdp_public_ip` Terraform variable.
- Now navigate to Services / EC2 and find the Linux EC2 instance created by Terraform:



- Select Connect and then copy the SSH client example. Also copy the private key you generated earlier and use it to SSH into the Linux instance:

```
ssh -i "<your-privatekey.pem>" ec2-user@<your vm
hostname>.eu-central-1.compute.amazonaws.com
```

- Install BIND DNS:

```
sudo yum install bind bind-utils
```

- Configure the name server conf:

```
sudo vi /etc/named.conf
```

- Here's my minimal working conf (replace the parts in bold):

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
acl "trusted" { <LINUX VM PRIVATE IP>; <WINDOWS VM PRIVATE IP>; };
options {
        listen-on port 53 { 127.0.0.1; <LINUX VM PRIVATE IP>; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { trusted; };
        allow-transfer { localhost ; <LINUX VM PRIVATE IP>; };
        /*
         - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
         - If you are building a RECURSIVE (caching) DNS server, you need to enable
           recursion.
         - If your recursive DNS server has a public IP address, you MUST enable access
           control to limit queries to your legitimate users. Failing to do so will
           cause your server to become part of large scale DNS amplification
           attacks. Implementing BCP38 within your network would greatly
           reduce such attack surface
        */
        recursion yes;
        forward first;
        forwarders { <AMAZON PROVIDED DNS IP FOR YOUR VPC>; };
        dnssec-validation no;
        managed-keys-directory "/var/named/dynamic";
        geoip-directory "/usr/share/GeoIP";
        pid-file "/run/named/named.pid";
        session-keyfile "/run/named/session.key";
        /* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
        include "/etc/crypto-policies/back-ends/bind.config";
};
zone "privatelink.cloud.databricks.com" {
  type forward;
  forward only;
  forwarders { <INBOUND RESOLVER ENDPOINT IP>; <INBOUND RESOLVER ENDPOINT IP>; };
};
logging {
        channel default_file {
                file "/var/log/named.log" size 10m;
                severity info;
                print-time yes;
                print-severity yes;
                print-category yes;
        };
        category default{ default_file; };
```

```
};
zone "." IN {
        type hint;
        file "named.ca";
};
include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
include "/etc/named/named.conf.local";
```

- Create the named.conf.local file and allow permissions to write to the log file:

```
sudo vi /etc/named/named.conf.local
sudo chmod 777 /var/log/
```

NB - obviously don't use 777 in production!

- Check our configuration file:

```
sudo named-checkconf
```

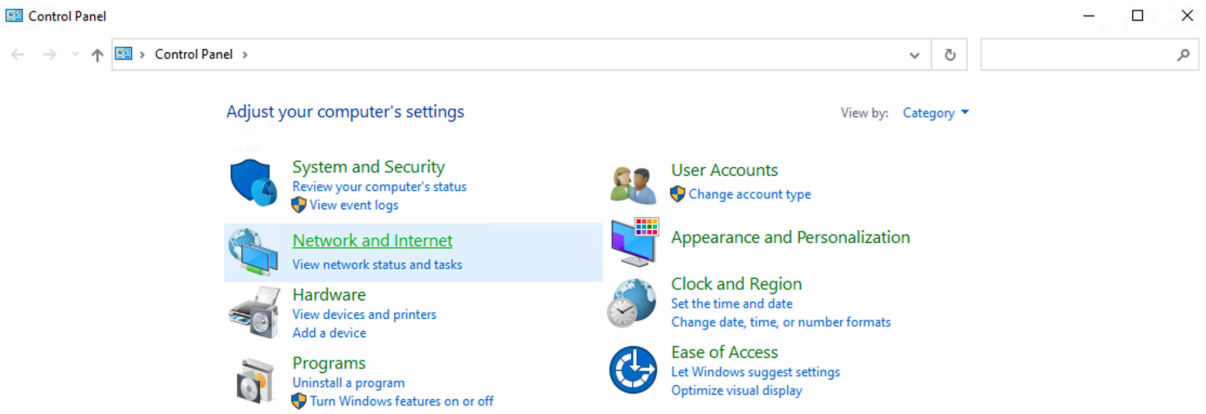
- Update your Linux VM to use itself as a DNS server

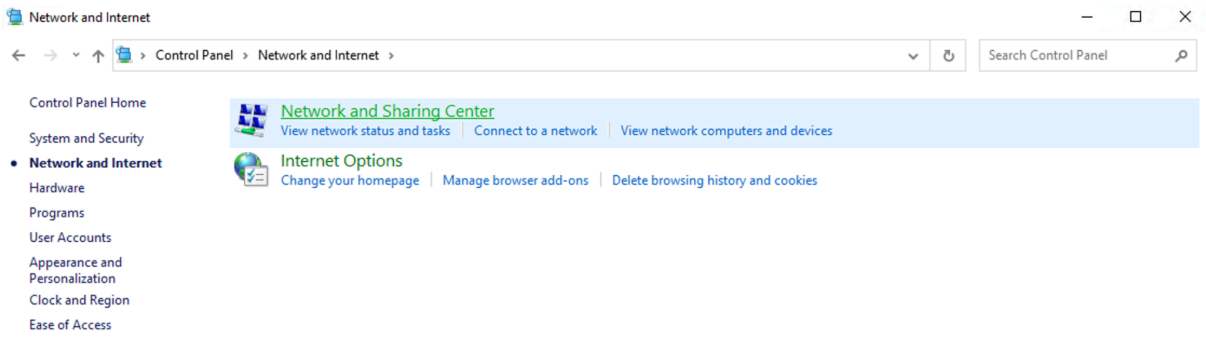
```
sudo systemctl restart systemd-resolved
```

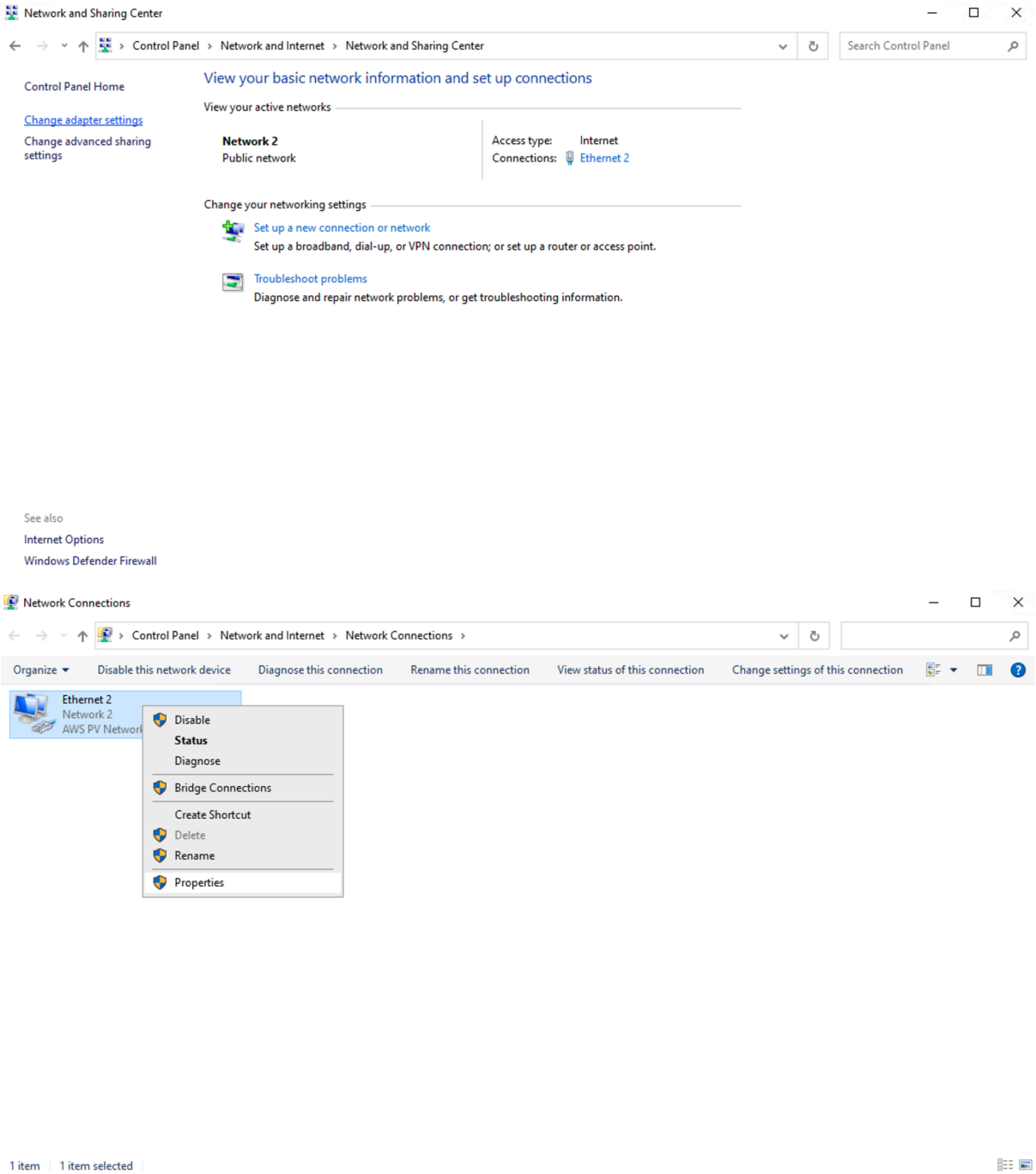
- Now start the BIND DNS service:

```
sudo systemctl start named
sudo systemctl enable named
```

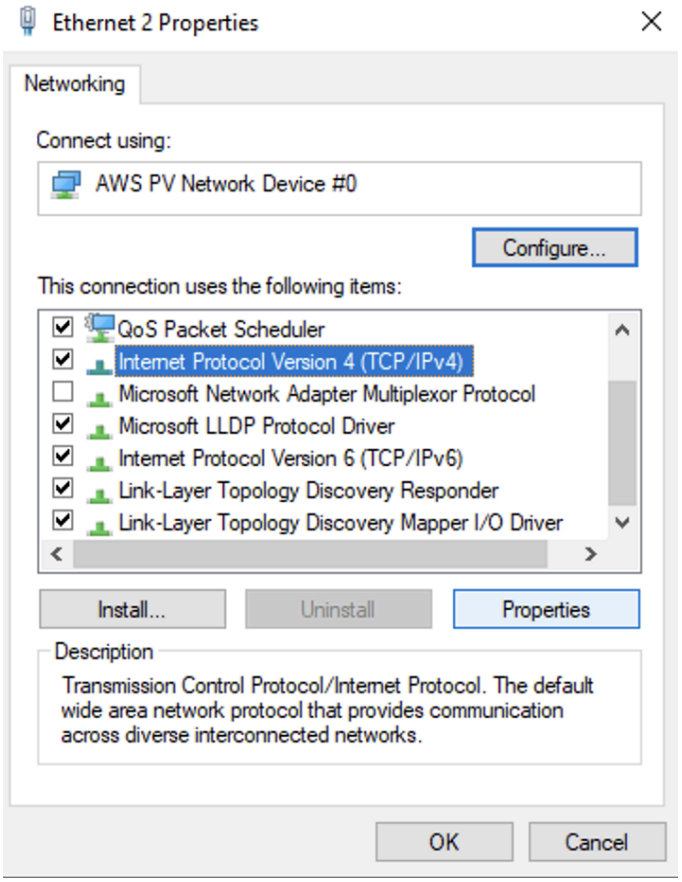
- Run an `nslookup` or `dig` command against the workspace FQDN. If it successfully resolves to the private IP then everything is working as expected!
- Ok, now we're going to configure our Windows VM to use our Linux DNS server
- Open Control Panel and then:


Control Panel
Control Panel
Adjust your computer's settings
View by: Category
System and Security
Review your computer's status
View event logs
Network and Internet
View network status and tasks
Hardware
View devices and printers
Add a device
Programs
Uninstall a program
Turn Windows features on or off
User Accounts
Change account type
Appearance and Personalization
Clock and Region
Set the time and date
Change date, time, or number formats
Ease of Access
Let Windows suggest settings
Optimize visual display

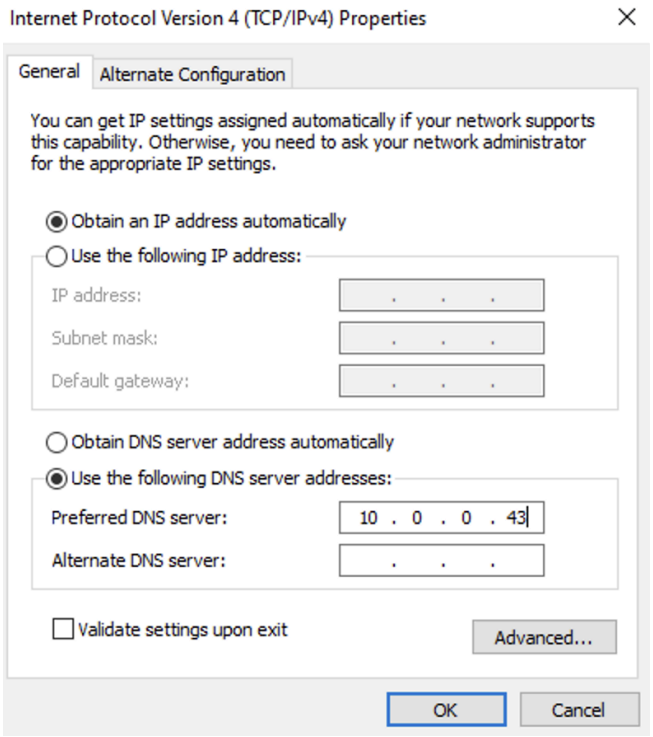

Network and Internet
Control Panel › Network and Internet ›
Search Control Panel
Control Panel Home
System and Security
Network and Internet
Hardware
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Network and Sharing Center
View network status and tasks
Connect to a network
View network computers and devices
Internet Options
Change your homepage
Manage browser add-ons
Delete browsing history and cookies


Network and Sharing Center
Control Panel › Network and Internet › Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
View your basic network information and set up connections
View your active networks
Network 2
Public network
Access type: Internet
Connections: Ethernet 2
Change your networking settings
Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.
Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.
See also
Internet Options
Windows Defender Firewall

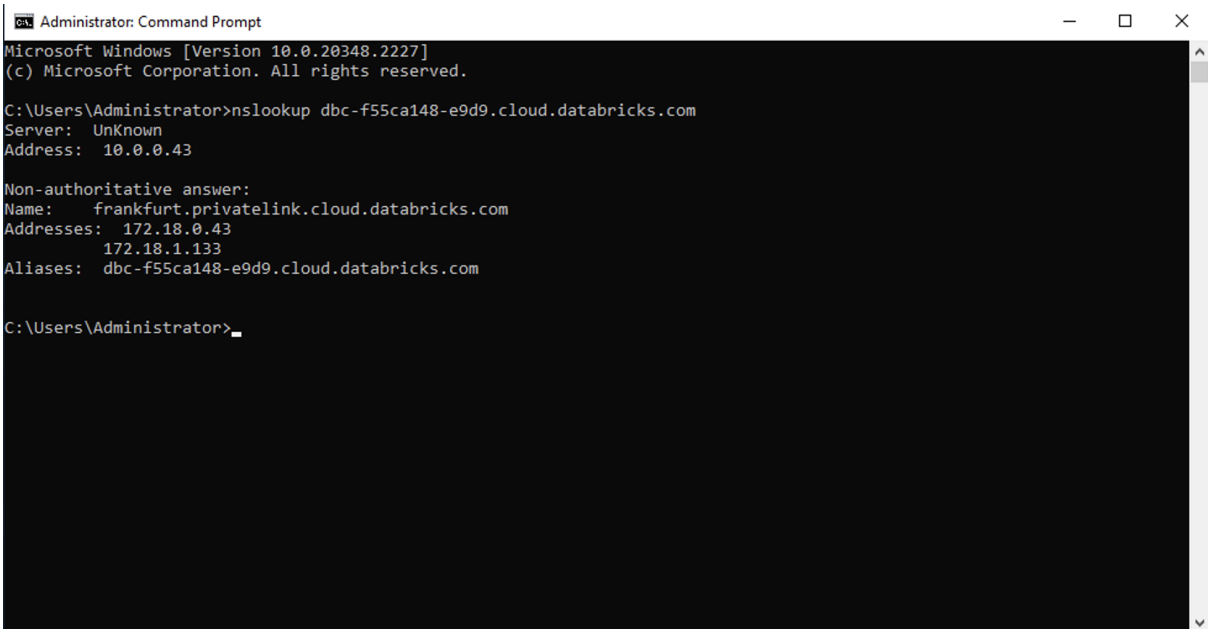
Network Connections
Control Panel › Network and Internet › Network Connections ›
Organize
Disable this network device
Diagnose this connection
Rename this connection
View status of this connection
Change settings of this connection
Ethernet 2
Network 2
AWS PV Network
Disable
Status
Diagnose
Bridge Connections
Create Shortcut
Delete
Rename
Properties
1 item
1 item selected



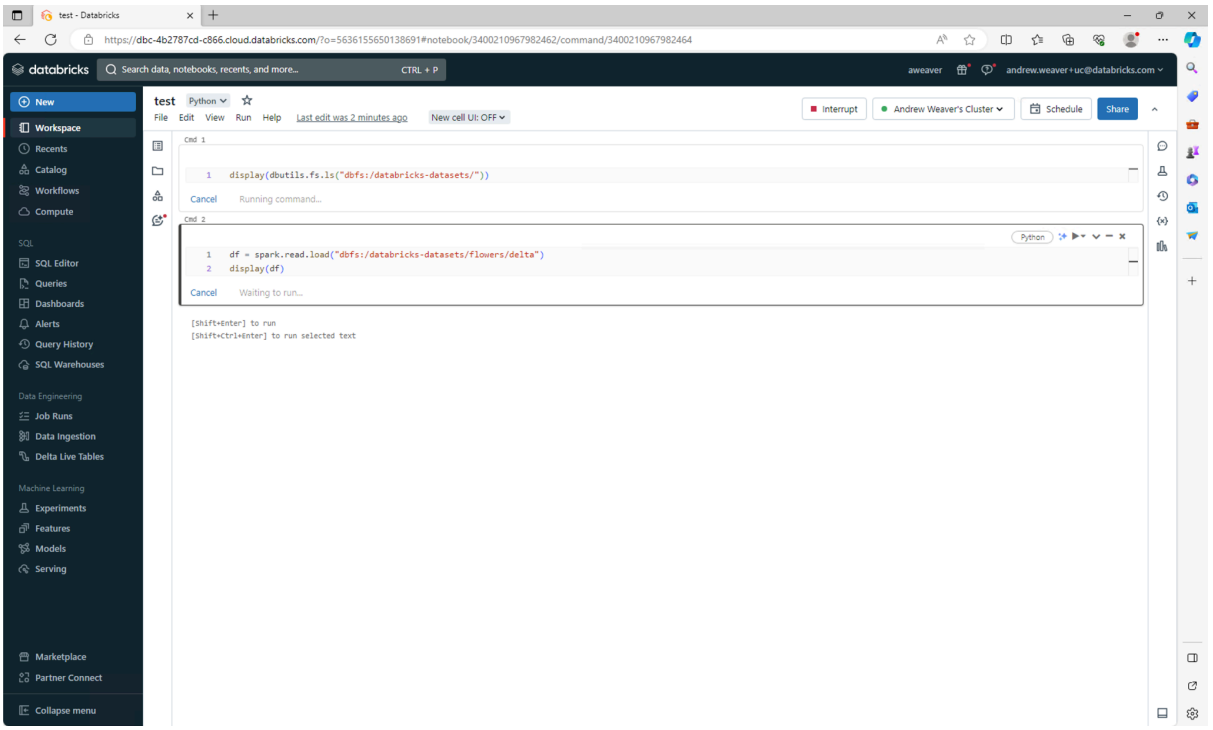
- Add the private IP of your Linux EC2 instance as your preferred DNS server:



- It’s now worth running an nslookup to test that the DNS server is configured correctly and you can resolve the workspace FQDN correctly:



- You should be set. Now you can open Edge (I know, I know) and do some testing!



NB - there’s no internet access in the data plane VPC, so you might find you need to use config like [Setup an Ephemeral Metastore for the UC-only Workspace on AWS] to get a stable environment.