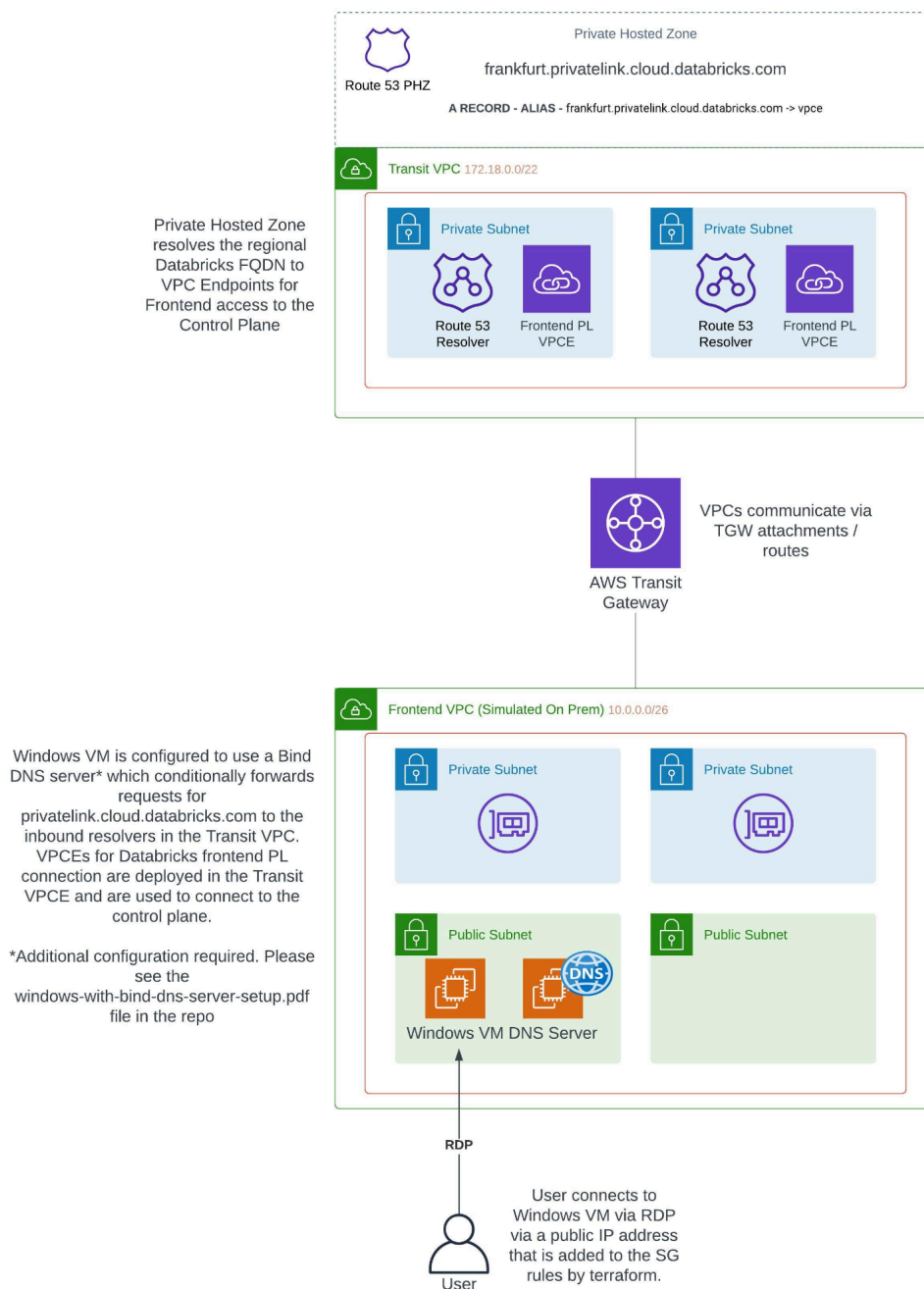


## High Level Architecture: Steps to Reproduce:

1  
2

## 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:

Route 53 > Resolver > Inbound endpoints

You are signed in to the following Region: eu-central-1 (Frankfurt). To change your Region, use the Region selector in the upper-right corner.

Inbound endpoints (1) [Info](#)

[View details](#) [Edit](#) [Delete](#) [Create inbound endpoint](#)

Search

ID	Name	Status	Host VPC	IP addresses	Transmission protocols
<a href="#">rslvr-in-352fa72d104c4e928</a>	aws-eu-central-1-inbound-resolver	Operational	vpc-0a511369...	2	Do53, DoH

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

Route 53 > Resolver > Inbound endpoints > aws-eu-central-1-inbound-resolver

Inbound endpoint: aws-eu-central-1-inbound-resolver [Info](#) [Edit](#) [Delete](#)

aws-eu-central-1-inbound-resolver Configuration

ID	Status	Host VPC
rslvr-in-352fa72d104c4e928	Operational	vpc-0a51136981e140253
Name	Security group	Resolver Endpoint Type
aws-eu-central-1-inbound-resolver	sg-01bb1b9921faa28cd	IPv4
Transmission Protocols		
Do53, DoH		

IP addresses (2) [Remove from endpoint](#) [Add IP address](#)

Search

IP address	IP address ID	Status	Subnet	Availability Zone
<a href="#">172.18.1.39</a>	mi-11e56ad14ae744f79	Attached	subnet-041e77cef9262...	eu-central-1b
<a href="#">172.18.0.243</a>	mi-40472b87de7f4d1cb	Attached	subnet-0e8d7011224d...	eu-central-1a

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

**Route 53**

frankfurt.privatelink.cloud.databricks.com

**Hosted zone details**

**Records (3)**

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (s)
frankfurt.priv...	A	Simple	Yes	vpce-08505a6bf67bc2e12-a3bme9t9.vpce-svc-081f7850381259777.eu-central-1.vpce.amazonaws.com.	-
frankfurt.priv...	NS	Simple	No	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.	172800
frankfurt.priv...	SOA	Simple	No	ns-1536.awsdns-00.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400	900

- 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.
- Now navigate to Services / EC2 and find the Linux EC2 instance created by Terraform:

**Instances (1)**

Find Instance by attribute or tag (case-sensitive)

Running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
aweaver-eu-central-1-linux-vm-dns	i-0876648d516a28ffd	Running	t2.medium	Initializing	View alarms	eu-central-1a

- Select Connect and then use the SSH client example to SSH into the VM
- 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!

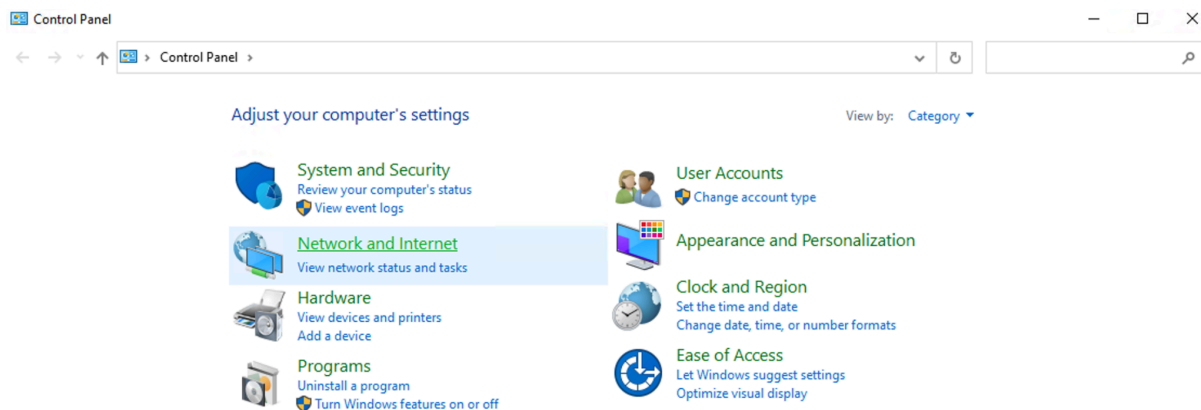
- Check our configuration file:

```
sudo named-checkconf
```

- [Update your Linux VM to use itself as a DNS server](#)

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

- Run an `nslookup` or `dig` command against the workspace FQDN. If it successfully resolves to the private IP then everything is working as expected!
- Now we're going to setup our Windows EC2 instance to simulate an on prem workstation by connecting it to our Bind DNS server
- 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.
- Ok, now we're going to configure our Windows VM to use our Linux DNS server
- Open Control Panel and then:



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

<b>Network 2</b> Public network	Access type: Internet Connections: Ethernet 2
------------------------------------	--

Change your networking settings

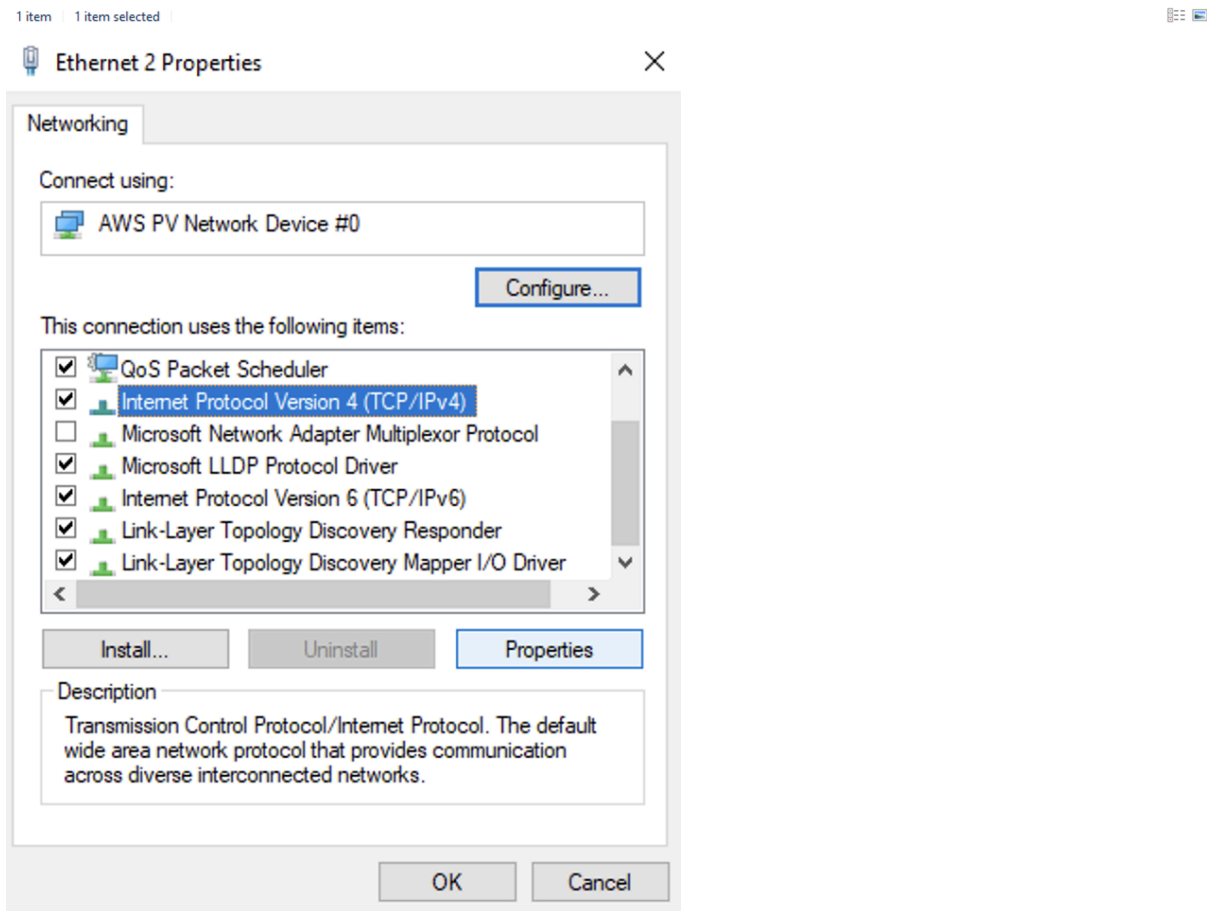
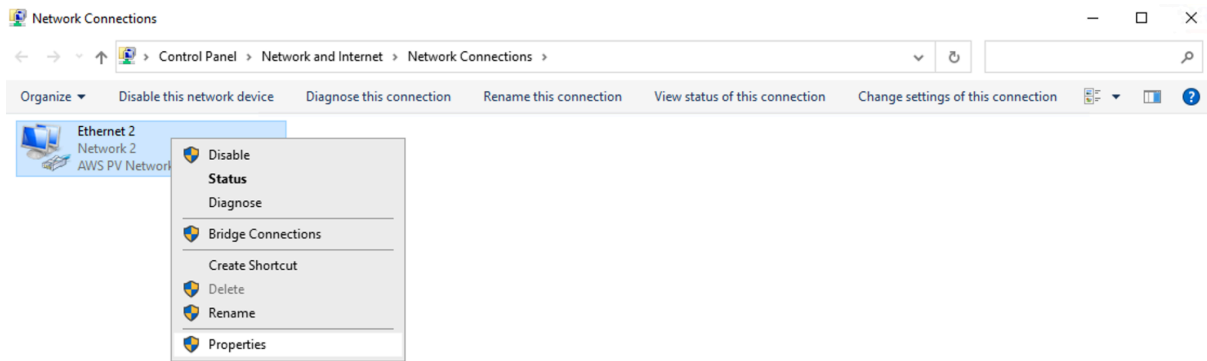
[Set up a new connection or network](#)  
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

[Troubleshoot problems](#)  
Diagnose and repair network problems, or get troubleshooting information.

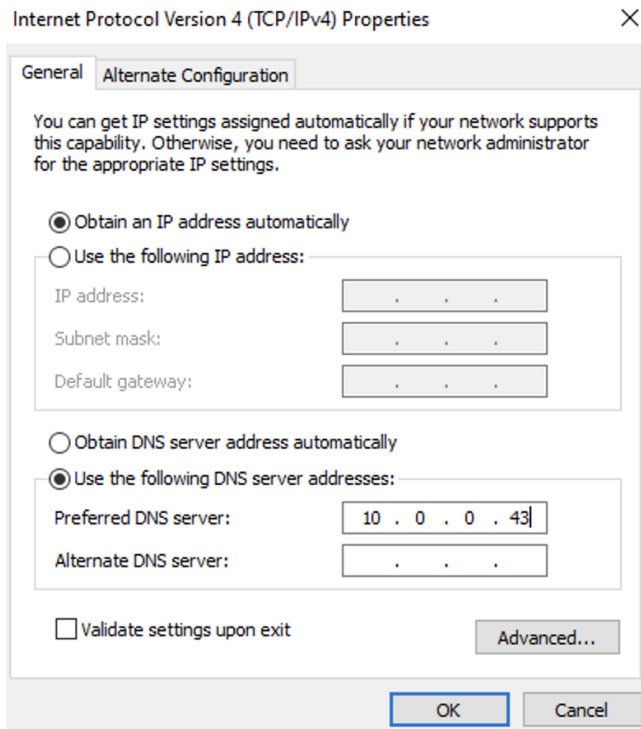
See also

[Internet Options](#)

[Windows Defender Firewall](#)



- 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:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

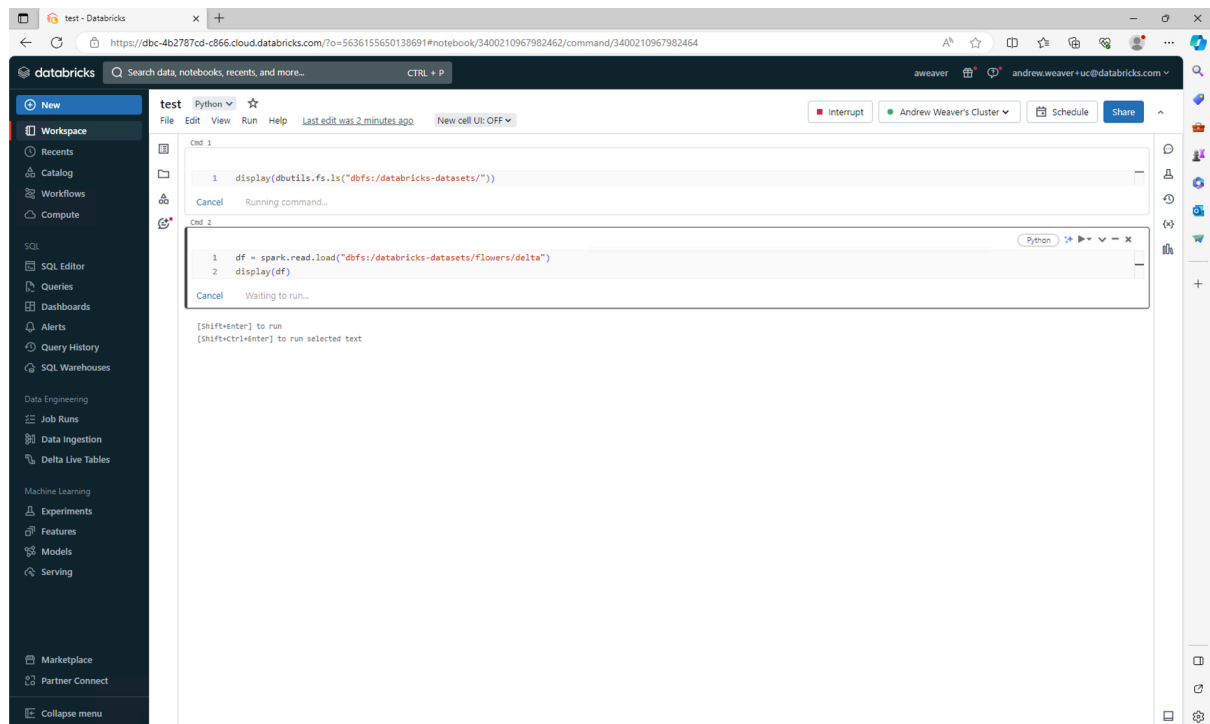
C:\Users\Administrator>nslookup dbc-f55ca148-e9d9.cloud.databricks.com
Server: UnKnown
Address: 10.0.0.43

Non-authoritative answer:
Name: frankfurt.privatelink.cloud.databricks.com
Addresses: 172.18.0.43
           172.18.1.133
Aliases: dbc-f55ca148-e9d9.cloud.databricks.com

C:\Users\Administrator>
```

- 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.