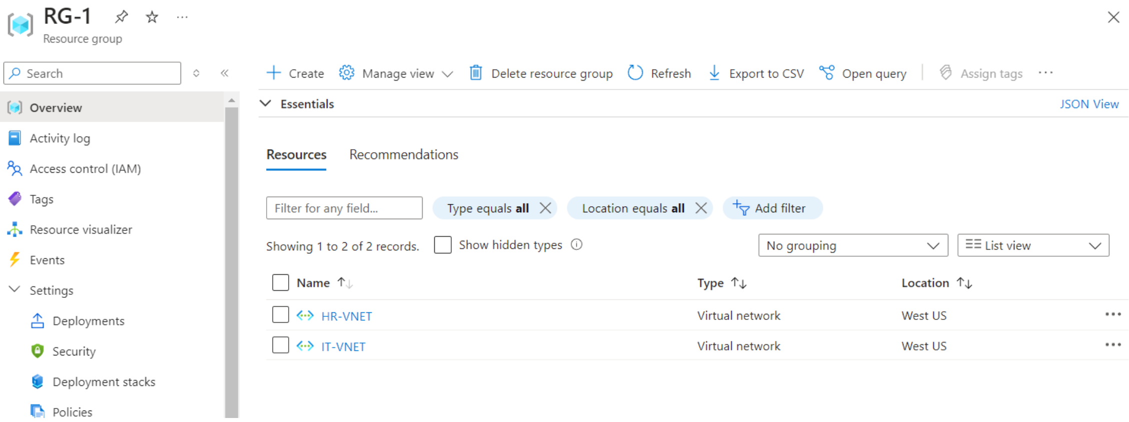
Solution Overview

- **Create virtual networks** for the IT and HR departments.
- **Deploy the resources** (VMs, DNS server, and web app) in their respective VNETs.
- **Set up VNET peering** to allow communication between the two VNETs.
- **Configure Azure Private DNS** for domain name resolution, allowing the DNS server to resolve the web app.
- Use Azure DNS to assign domain names to the DNS server
- **Test the communication** by pinging the web app from the DNS server.

---

1. Create Virtual Networks for IT and HR Departments

- Create two virtual networks, one for the **IT department** and one for the **HR department**. Each will be in its own subnet for segregation.
  - **IT-VNET** (Address Space: 10.0.0.0/16)
    - Subnet 1 (for DNS and Linux VM): 10.0.1.0/24
  - **HR-VNET** (Address Space: 10.1.0.0/16)
    - Subnet 1 (for Web App and Linux VM): 10.1.1.0/24



Type equals **all**

Location equals **all**

Showing 1 to 2 of 2 records.

No grouping

List view

Name	Type	Location
HR-VNET	Virtual network	West US
IT-VNET	Virtual network	West US

---

2. Deploy Resources in the Virtual Networks

- In the **IT-VNET**, deploy:

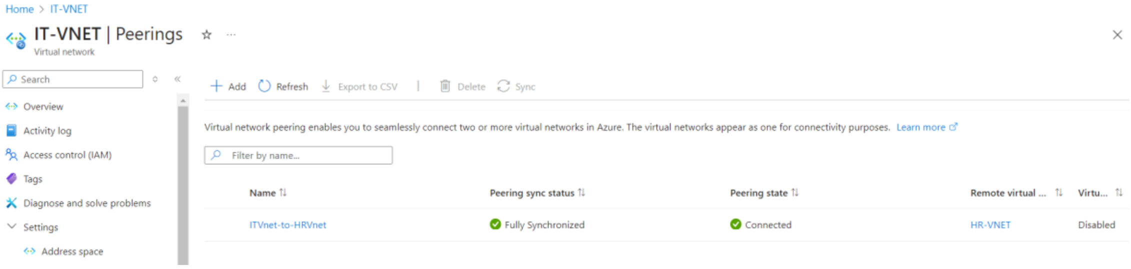
    - A **Linux-VM** for administrative purposes.
    - A **DNS Server** (could be either a VM configured with DNS services or use Azure Private DNS).
- In the **HR-VNET**, deploy:
    - A **Linux-VM-HR** for administrative purposes.
    - A **Web App** using Azure App Service.

---

3. Establish VNET Peering for Private Communication

Now that the virtual networks are set up and the resources are deployed, you need to enable communication between the **IT-VNET** and **HR-VNET** using **VNET peering**.

1. **Peer IT-VNET and HR-VNET**:
    - Go to **IT-VNET** > **Settings** > **Peerings** > **Add**.
    - Set the peer to **HR-VNET** and enable the following settings:
        - **Allow virtual network access** (Enable this option for both VNETs to allow bidirectional traffic).
        - **Allow forwarded traffic**: Enable.
    - Confirm to create the peering.



Virtual network peering enables you to seamlessly connect two or more virtual networks in Azure. The virtual networks appear as one for connectivity purposes. Learn more

Name	Peering sync status	Peering state	Remote virtual ...	Virtu...
ITVnet-to-HRVnet	✓ Fully Synchronized	✓ Connected	HR-VNET	Disabled

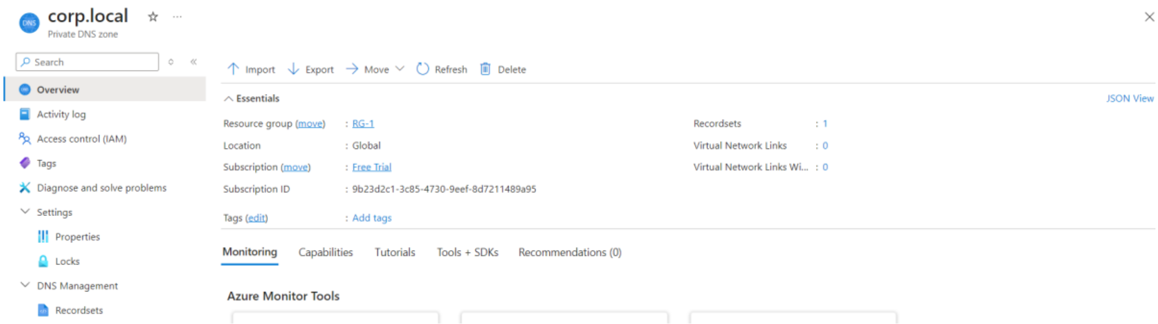
---

4. Configure Azure DNS for Name Resolution

To allow internal name resolution and map a domain name to the web app, you can use **Azure Private DNS**.

1. **Create a Private DNS Zone**:
    - Go to **Azure Portal** > **Create a resource** > **Private DNS Zone**.

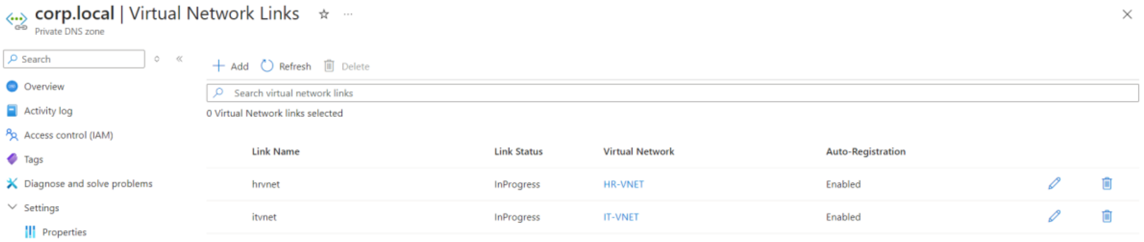
- Name it something like corp.local.
- Create the DNS zone.



Resource group (move) : RG-1
Location : Global
Subscription (move) : Free Trial
Subscription ID : 9b23d2c1-3c85-4730-9eef-8d7211489a95
Tags (edit) : Add tags
Recordsets : 1
Virtual Network Links : 0
Virtual Network Links Wi... : 0

2. **Link Virtual Networks to the DNS Zone**:

   - Open the **Private DNS Zone** (corp.local).
   - Under **Settings**, go to **Virtual network links**.
   - Add both **IT-VNET** and **HR-VNET** to the DNS zone and enable auto-registration so that any new VMs created in the VNETs automatically register their IP addresses.

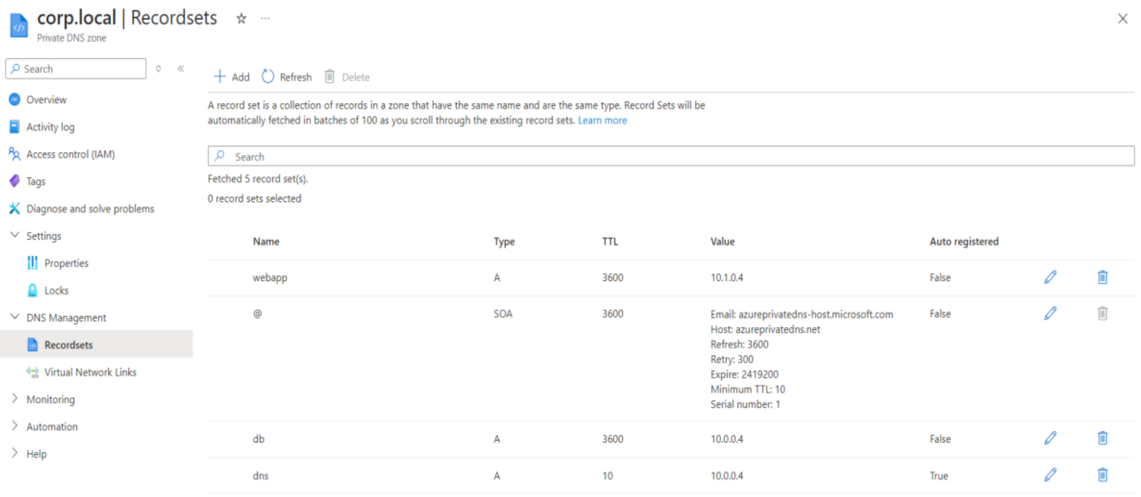


0 Virtual Network links selected

Link Name	Link Status	Virtual Network	Auto-Registration
hrvnet	InProgress	HR-VNET	Enabled
itvnet	InProgress	IT-VNET	Enabled

3. **Create DNS Records**:

   - Inside the **Private DNS Zone** (corp.local), create DNS records for the resources:
     - **DNS Record for Web App**: Create an A record like webapp.corp.local and point it to the private IP of the web app in HR-VNET.



A record set is a collection of records in a zone that have the same name and are the same type. Record Sets will be automatically fetched in batches of 100 as you scroll through the existing record sets. Learn more

Fetched 5 record set(s).

0 record sets selected

Name	Type	TTL	Value	Auto registered
webapp	A	3600	10.1.0.4	False
@	SOA	3600	Email: azureprivatedns-host.microsoft.com Host: azureprivatedns.net Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 10 Serial number: 1	False
db	A	3600	10.0.0.4	False
dns	A	10	10.0.0.4	True

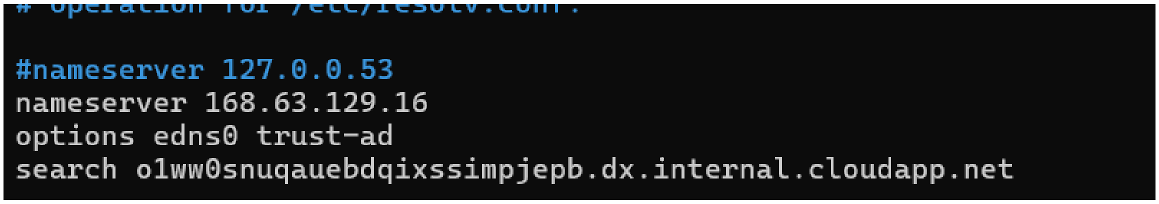
5. Set up the DNS Server to use the Azure Private DNS Zone.

Steps:

1. SSH into the DNS server in IT-VNET.
2. Modify the DNS server configuration:

   Open /etc/resolv.conf

   Add Azure's DNS server IP



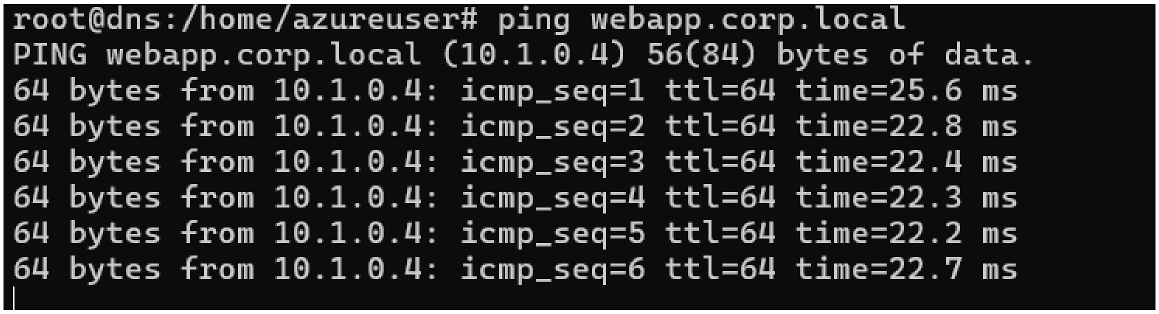
```
# operation for /etc/resolv.conf.

#nameserver 127.0.0.53
nameserver 168.63.129.16
options edns0 trust-ad
search o1ww0snuqauebdqixssimpjepb.dx.internal.cloudapp.net
```

6. Test the Communication

Now that both VNETs are peered and DNS is set up, you can test the private communication between the DNS server and the web app.

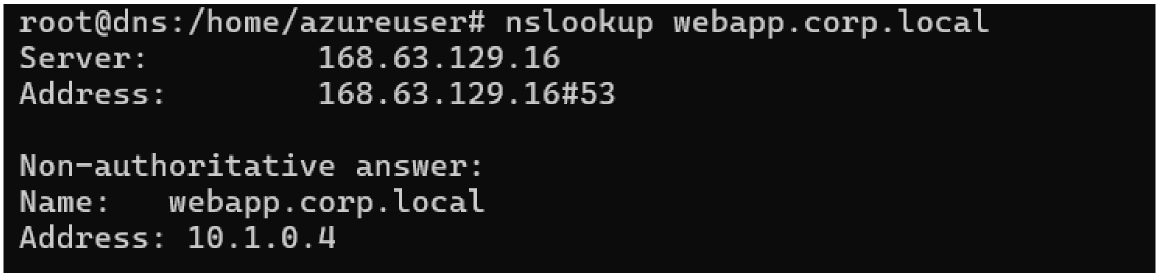
1. **SSH into the DNS Server VM** in IT-VNET.
2. Run a **ping** or **nslookup** command to test DNS resolution and connectivity:

ping webapp.corp.local



```
root@dns:/home/azureuser# ping webapp.corp.local
PING webapp.corp.local (10.1.0.4) 56(84) bytes of data.
64 bytes from 10.1.0.4: icmp_seq=1 ttl=64 time=25.6 ms
64 bytes from 10.1.0.4: icmp_seq=2 ttl=64 time=22.8 ms
64 bytes from 10.1.0.4: icmp_seq=3 ttl=64 time=22.4 ms
64 bytes from 10.1.0.4: icmp_seq=4 ttl=64 time=22.3 ms
64 bytes from 10.1.0.4: icmp_seq=5 ttl=64 time=22.2 ms
64 bytes from 10.1.0.4: icmp_seq=6 ttl=64 time=22.7 ms
```

nslookup webapp.corp.local



```
root@dns:/home/azureuser# nslookup webapp.corp.local
Server:         168.63.129.16
Address:        168.63.129.16#53

Non-authoritative answer:
Name:   webapp.corp.local
Address: 10.1.0.4
```
