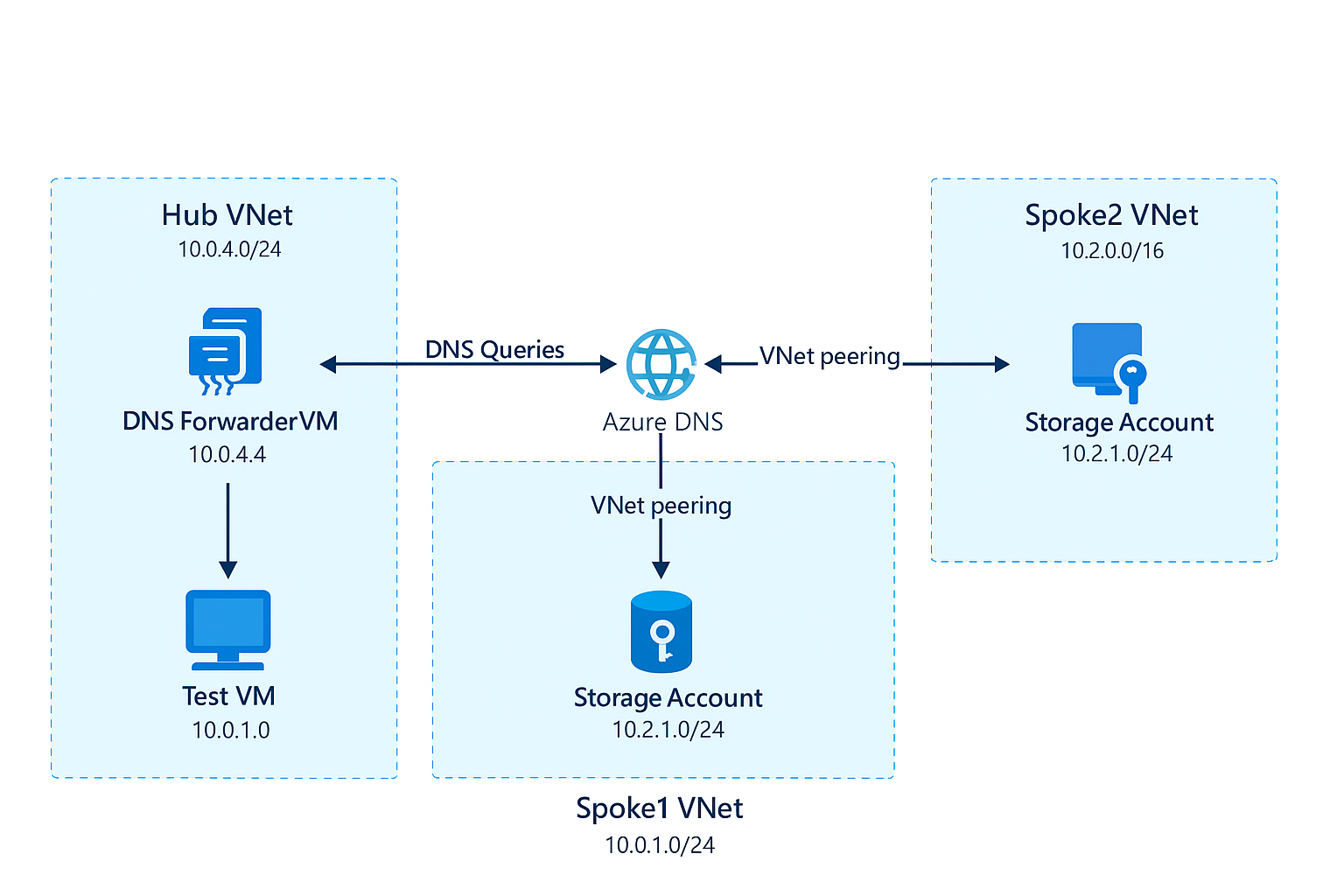
Secure Hosting of Webapp Via Azure Application Gateway

**Abstract**

This project demonstrates the implementation of DNS forwarding in Microsoft Azure by deploying a Linux virtual machine configured with dnsmasq. DNS forwarding allows resources within spoke virtual networks to resolve domain names via a centralized custom DNS forwarder hosted in the hub virtual network. The architecture follows the Azure Hub-Spoke topology. This report includes implementation steps, topology overview, command-line configuration, and troubleshooting. Screenshots at every stage validate successful execution.

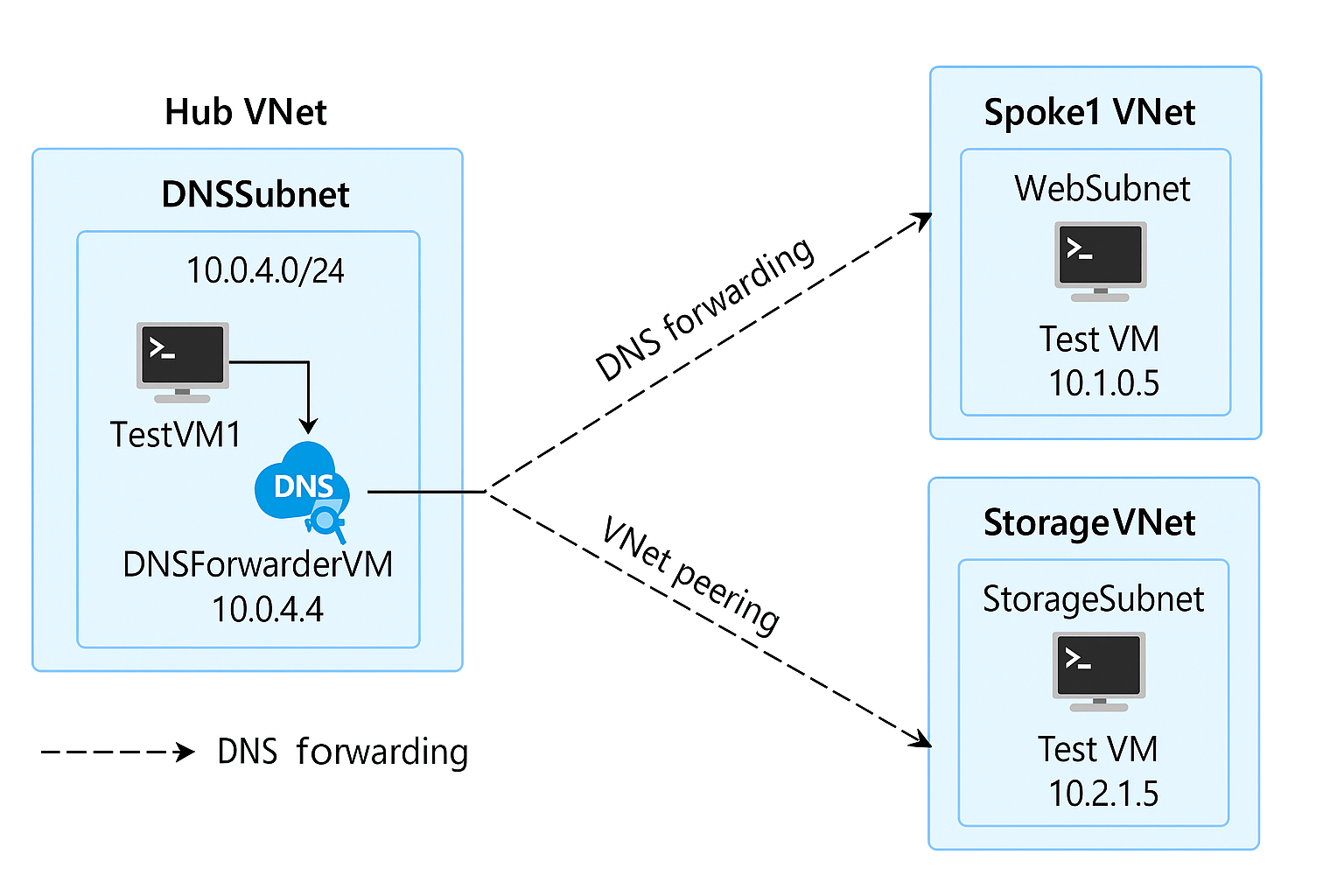
**Objective**

****To configure DNS forwarding in Azure using a custom DNS forwarder with dnsmasq on a Linux VM, enabling name resolution across spoke VNets through the hub VNet’s DNS forwarder.

**Azure Topology Diagram**

Description:

* Hub VNet: Hosts a

 Linux VM (DNSForwarder)

* Spoke1 VNet: Contains a test VM
* Spoke2 VNet: Optional secondary spoke
* Peering configured between

Hub and Spoke VNets

* Custom DNS configured

on Spoke VNets to forward

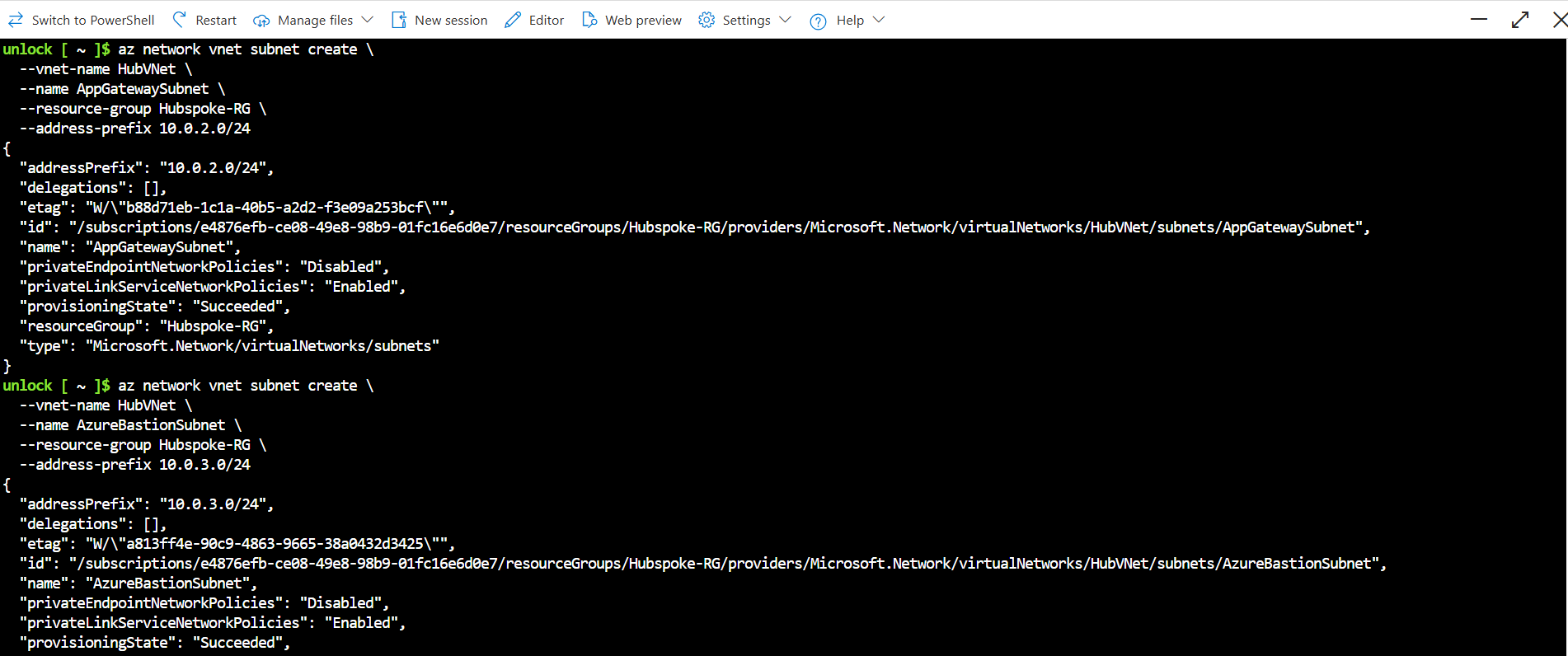
queries to DNSForwarder

**Components Used**

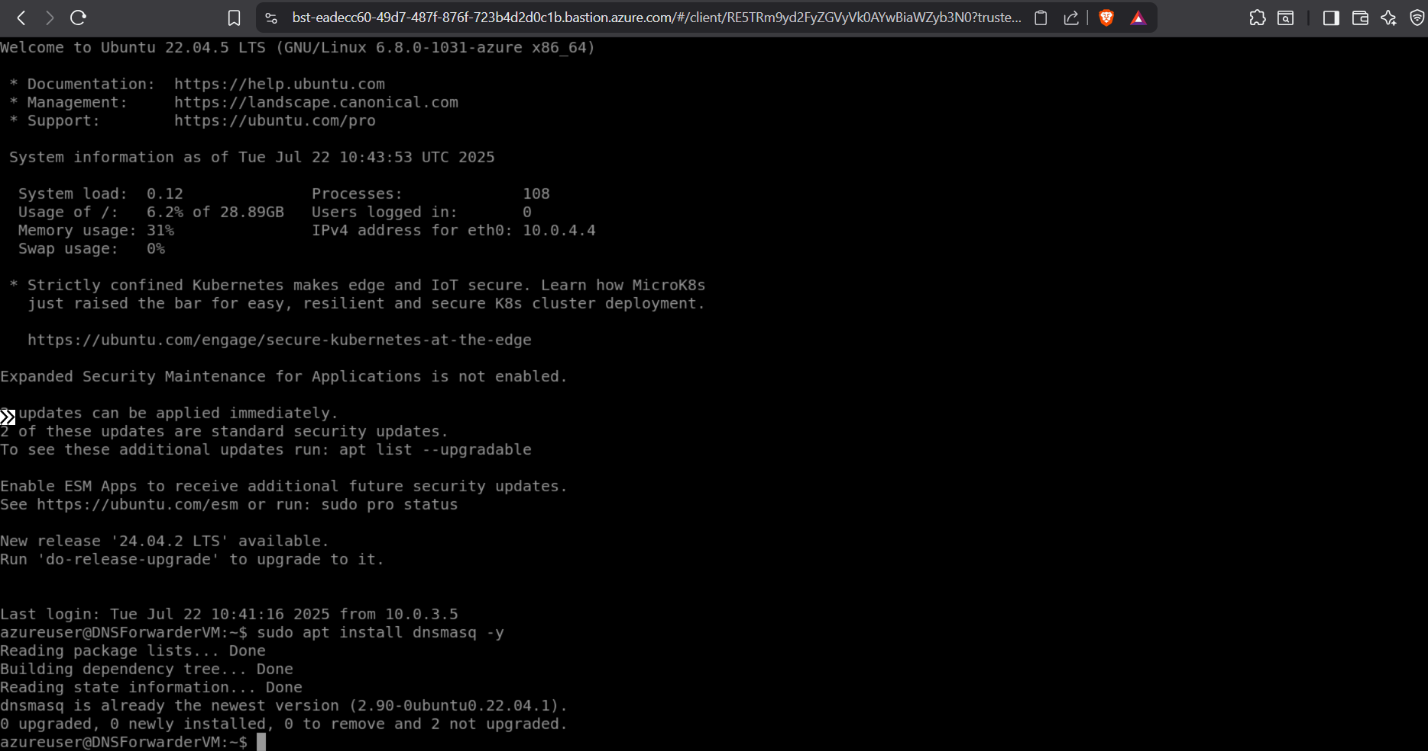
1. Azure Resource Group: HubSpoke-RG
2. Hub VNet: 10.0.4.0/24
3. Spoke1 VNet: 10.0.1.0/24
4. DNSForwarder VM (Ubuntu): IP - 10.0.4.4
5. Azure CLI for VM creation and configuration
6. dnsmasq - DNS forwarding service

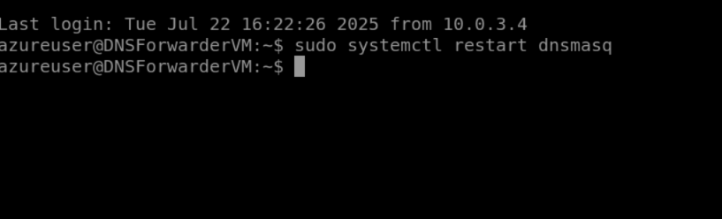
**Implementation Steps**

1. Create Hub VNet and DNSForwarder VM:
   * Command:
   * az network vnet create --name Hub-VNet --resource-group HubSpoke-RG --address-prefix 10.0.4.0/24  
     az vm create --name DNSForwarder --resource-group HubSpoke-RG --image UbuntuLTS --vnet-name Hub-VNet --subnet default --private-ip-address 10.0.4.4 --admin-username azureuser --generate-ssh-keys



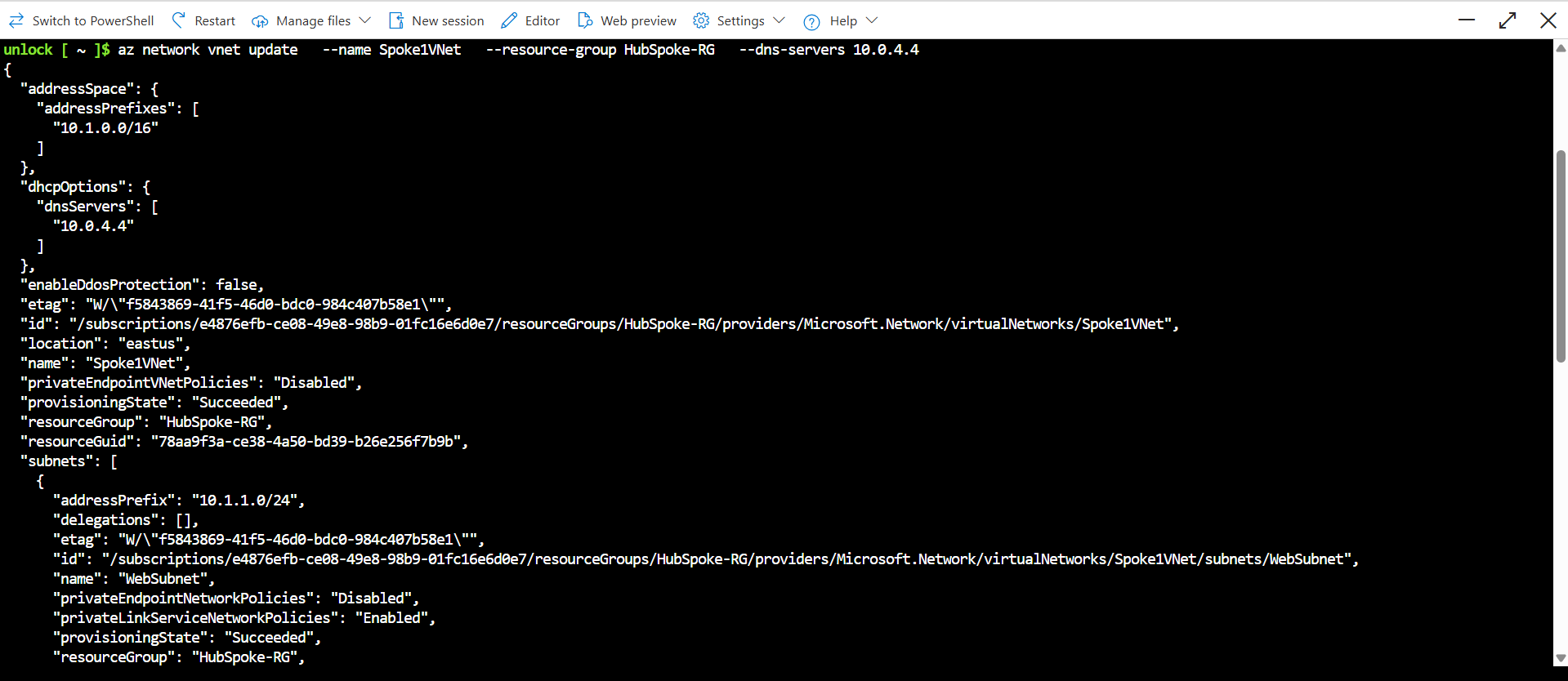
1. Install dnsmasq on DNSForwarder VM:
   * SSH into VM:
   * ssh azureuser@<public-ip>
   * Commands:
   * sudo apt update  
     sudo apt install dnsmasq  
     sudo nano /etc/dnsmasq.conf
   * Sample config:
   * server=168.63.129.16  
     domain-needed  
     bogus-priv  
     listen-address=127.0.0.1  
     listen-address=10.0.4.4



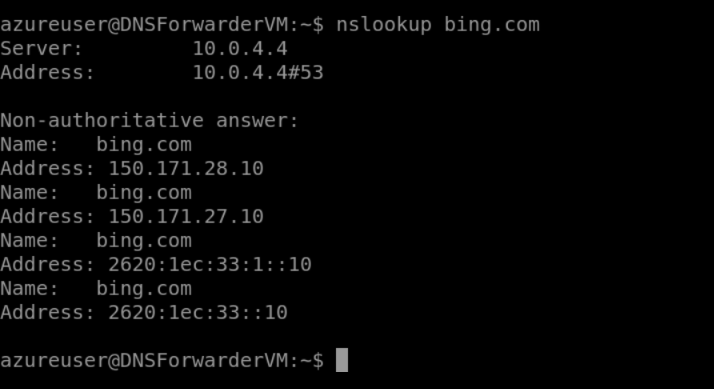
1. Restart dnsmasq and verify:

* sudo systemctl restart dnsmasq  
  sudo systemctl status dnsmasq

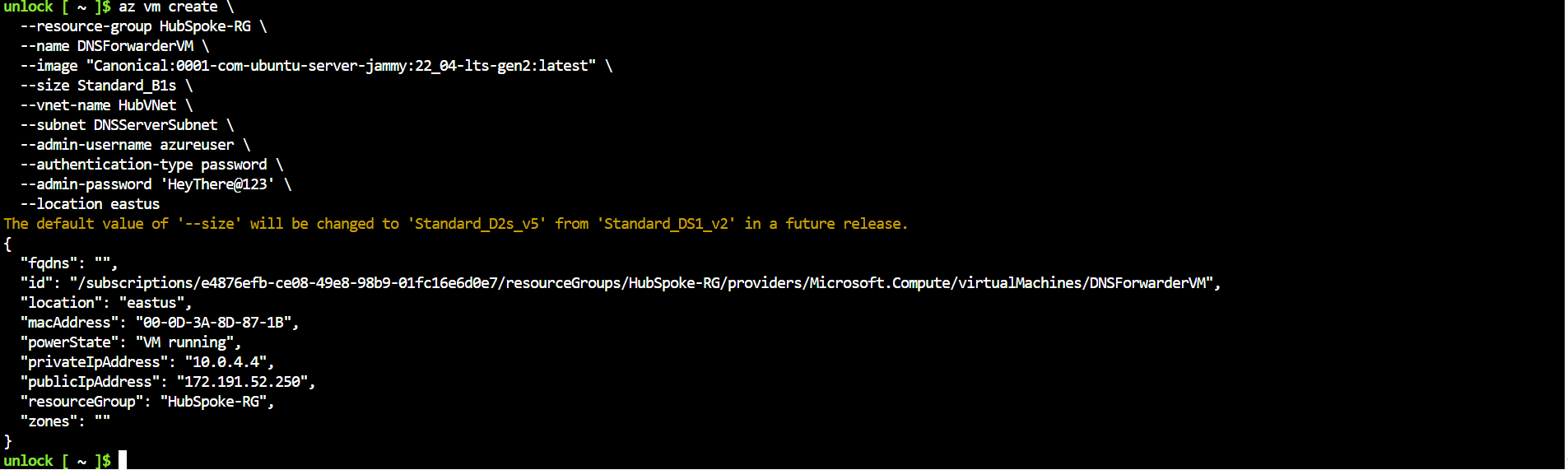
1. Update DNS settings in Spoke1 VNet:

* az network vnet update --name Spoke1-VNet --resource-group HubSpoke-RG --dns-servers 10.0.4.4

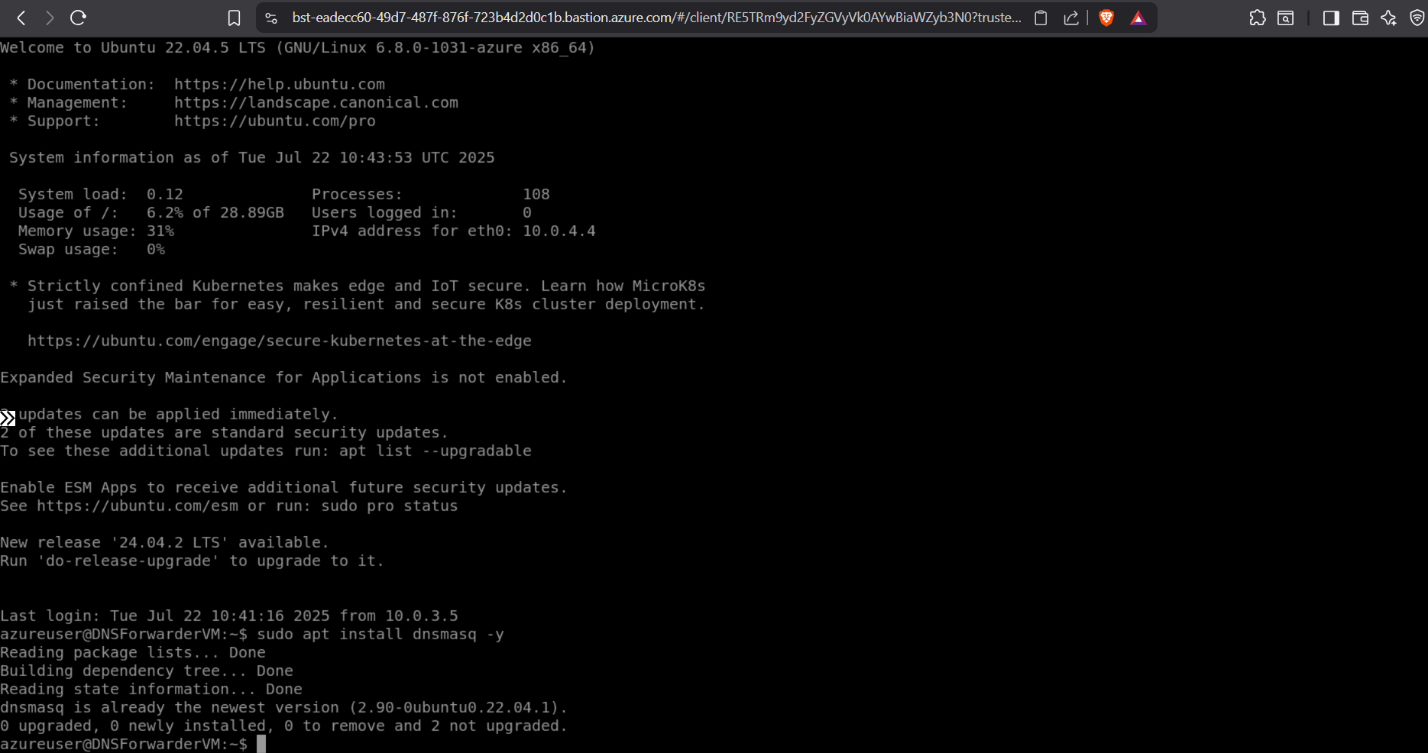
1. Test DNS Resolution:
   * SSH into Spoke1 VM:
   * nslookup bing.com



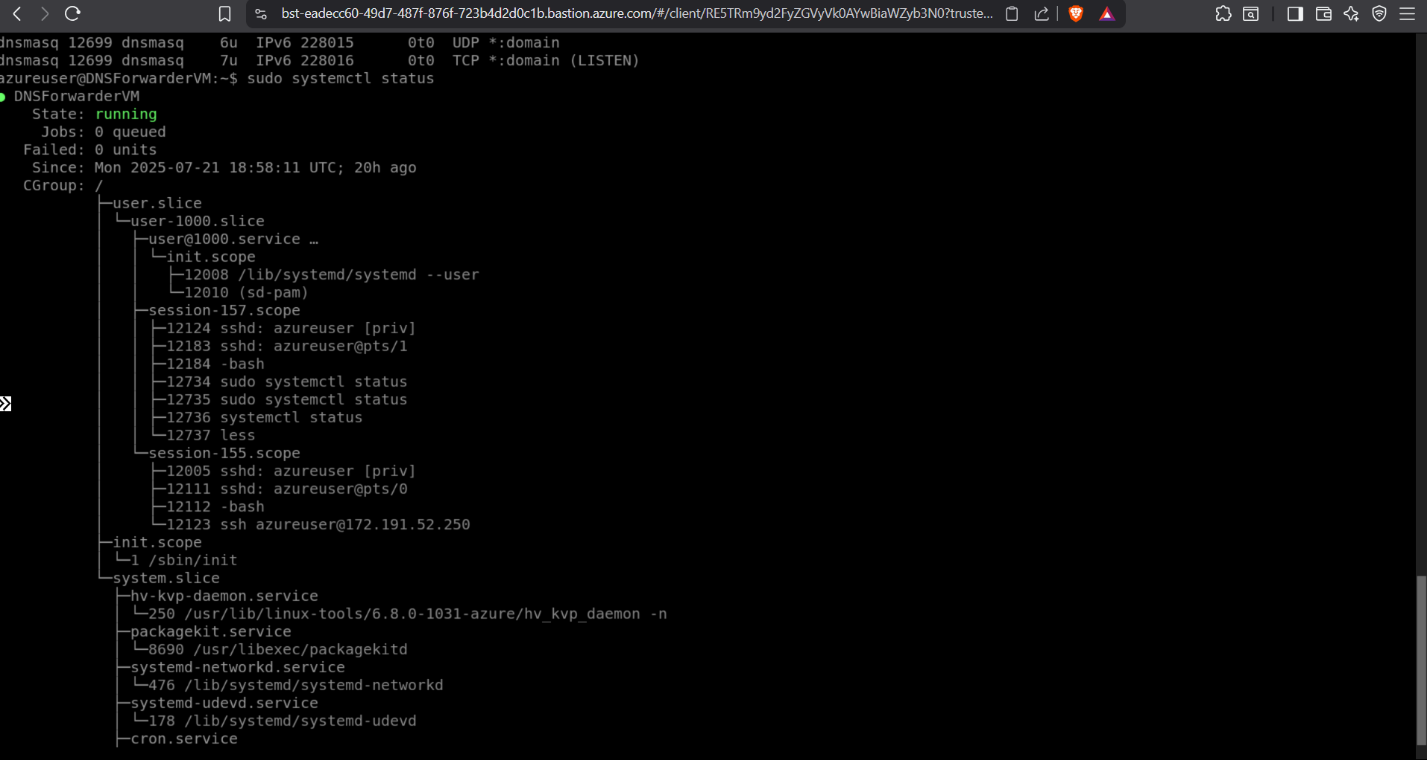
**Screenshots with Labels**

1: VM Created Successfully

2: dnsmasq Installed and Configured



3: dnsmasq Service Running]



4: DNS Server Updated in Spoke1 VNet

A screenshot of a computer

AI-generated content may be incorrect.

5: Successful nslookup from Spoke1 VM



**Challenges & Troubleshooting**

1. dnsmasq failed to start:
   * Checked status with:
   * systemctl status dnsmasq  
     journalctl -xeu dnsmasq.service
   * Solution: Corrected config file syntax, ensured no port conflicts.
2. SSH hostname resolution error:
   * Resolved by using public IP instead of hostname.
3. nslookup failed initially:
   * Fixed by ensuring correct DNS IP in VNet and restarting dnsmasq.

**Conclusion**

DNS forwarding in Azure using a Linux VM and dnsmasq enables efficient and centralized name resolution across VNets. This implementation is scalable, customizable, and suitable for enterprise-grade network architectures following the Hub-Spoke model. The project successfully demonstrates DNS forwarding in a simulated environment.

Appendix:

* Azure CLI Version
* OS: Ubuntu 20.04 LTS
* Azure Subscription: Free Tier