Microsoft + vmware

Azure Spring Cloud on Learn TV

Secure apps on Azure Spring Cloud with Managed Virtual Network

Asir Selvasingh
Principal PM Architect -- Java on Azure
@asirselvasingh

Asir Selvasingh

Principal PM Architect

Java on Microsoft Azure

On-point for everything developers need to build, migrate and scale Java applications on Azure.

Started software engineering career in the early days of Java, in 1995, and built enterprise products, applications and open source projects.

Microsoft Azure

Purpose – secure apps in managed virtual networks

01

Isolate Azure Spring Cloud (apps and service runtime) from Internet

- Place it on your corporate networks

02

Enable Azure Spring Cloud to **interact** with systems in

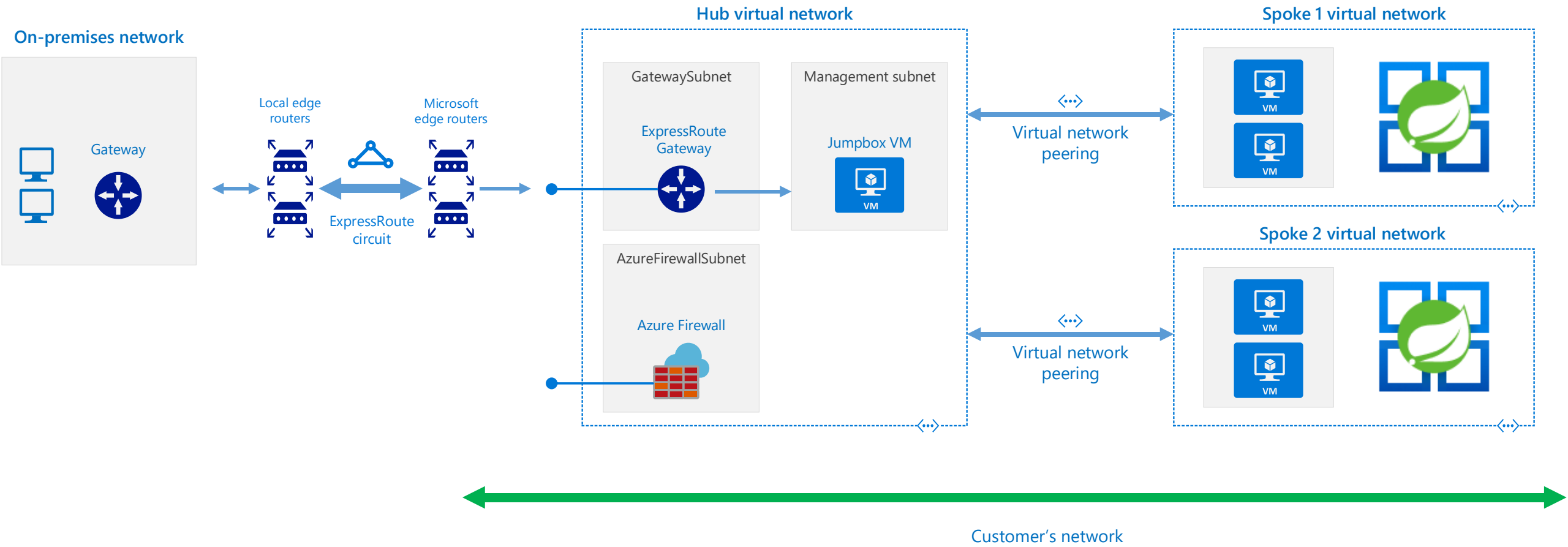
- On premises data centers
- Azure services in other VNETs

- Example - database, messaging, directory, FTP and mail systems

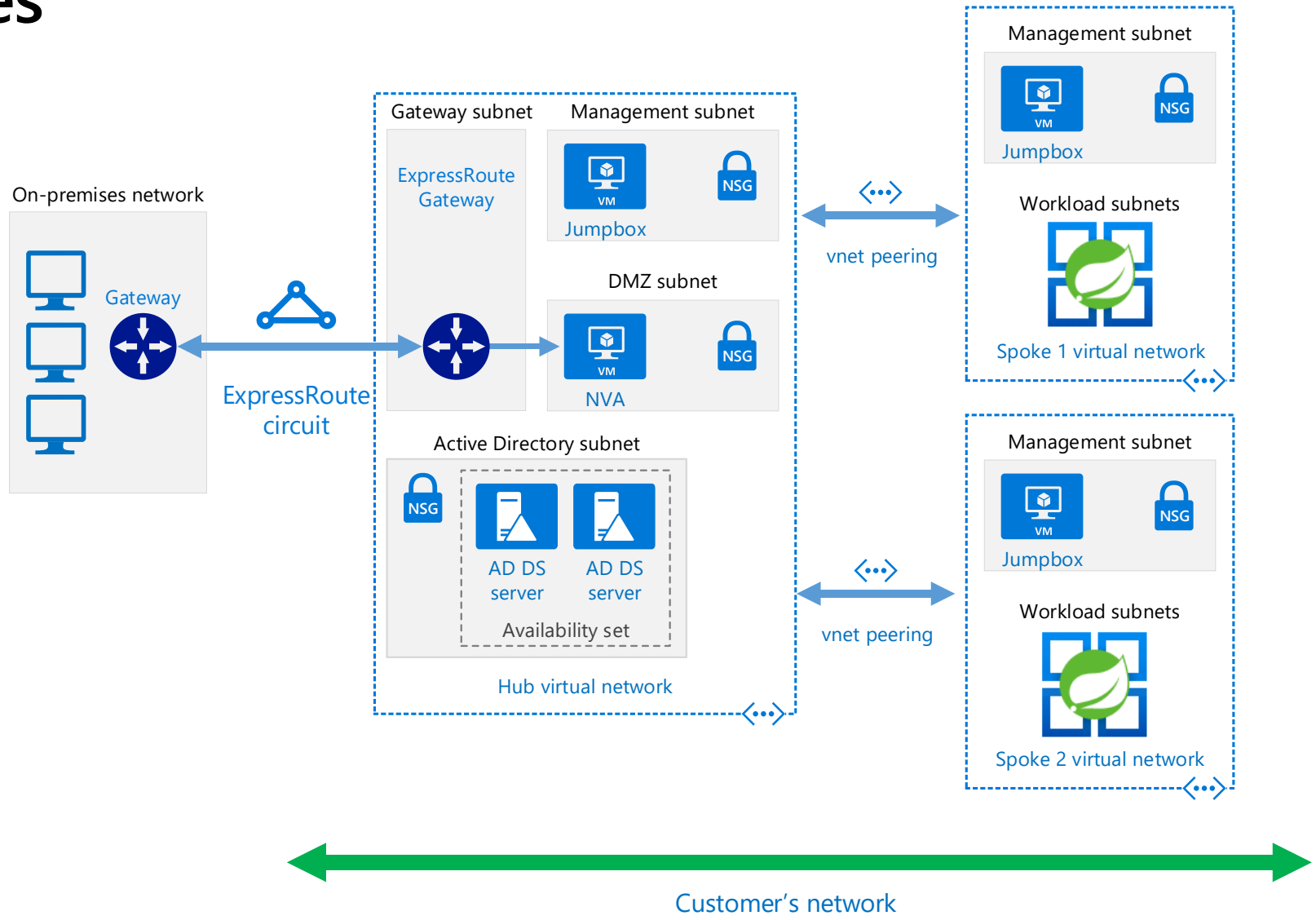
03

Control inbound and outbound network communications for Azure Spring Cloud

Example - Azure Spring Cloud in Hub + Spoke Topology



Example - Azure Spring Cloud in Hub + Spoke Topology with Shared Services



Concept – Azure Spring Cloud in VNET

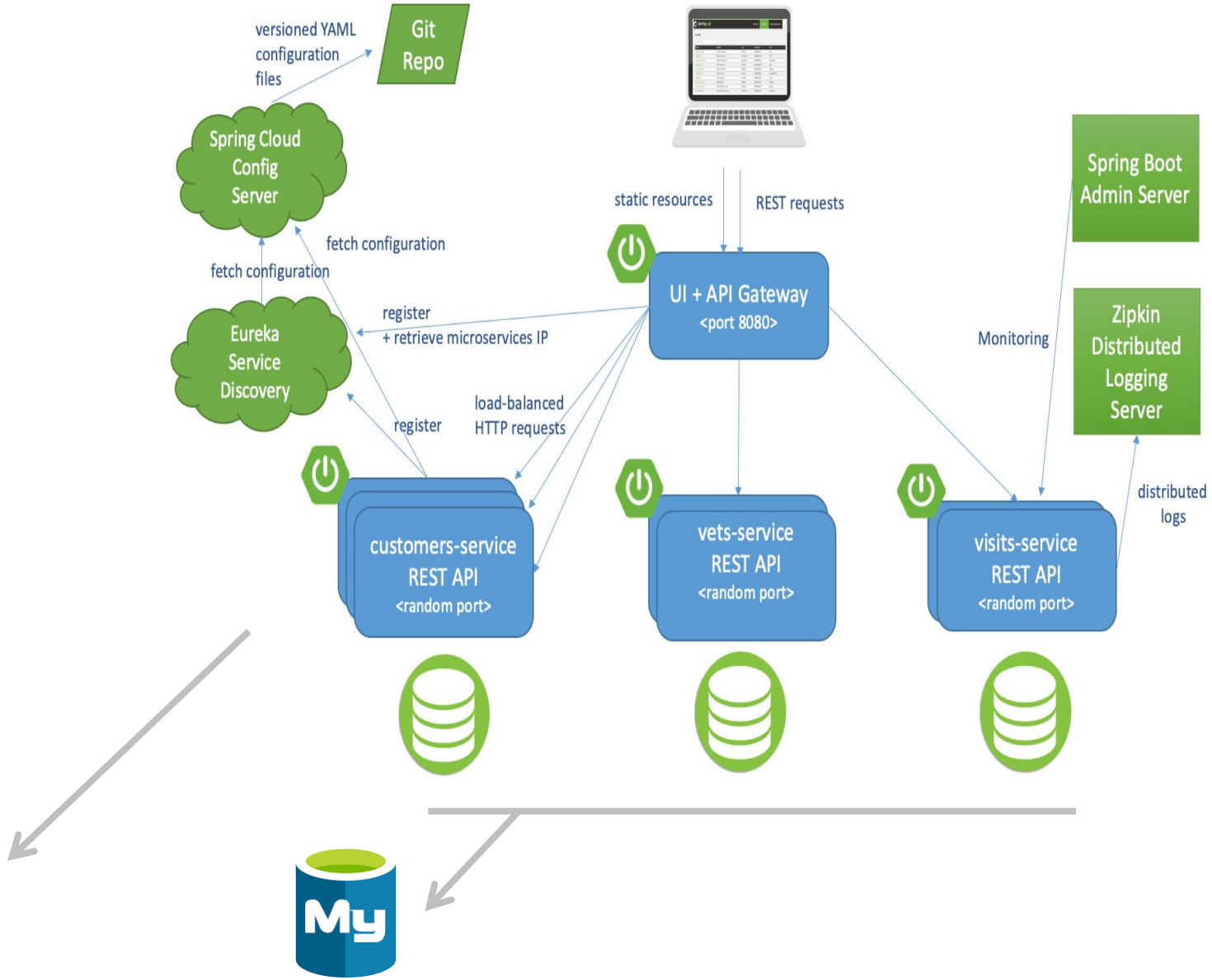
Deploy in your virtual network (also called *VNet injection*) using:

- **Subnet for service runtime** - host Spring Cloud Registry, Config Server, Storage, Azure Container Registry, log streaming, etc.
- **Subnet for Spring Boot microservice apps** - host Spring apps
- **Resource Groups** – host related Azure resources

Demo

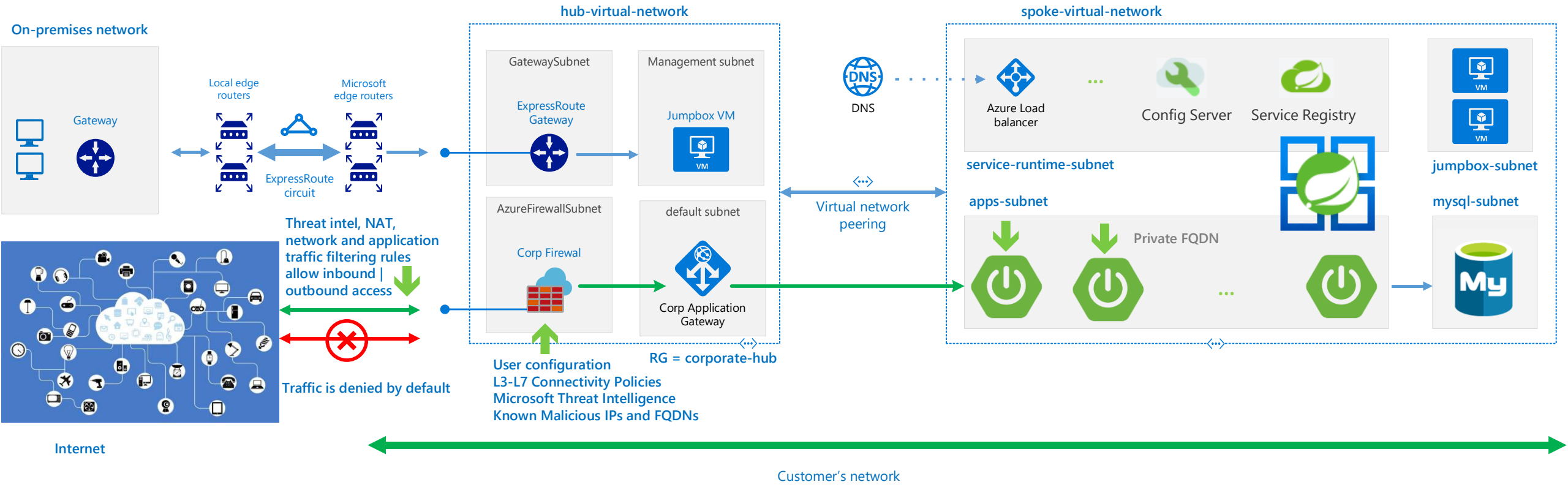
Deploy Spring Cloud app to Azure without worrying about:

- Infrastructure and scaling
- Spring Cloud middleware – config, registry, tracing and gateway, or
- Monitoring



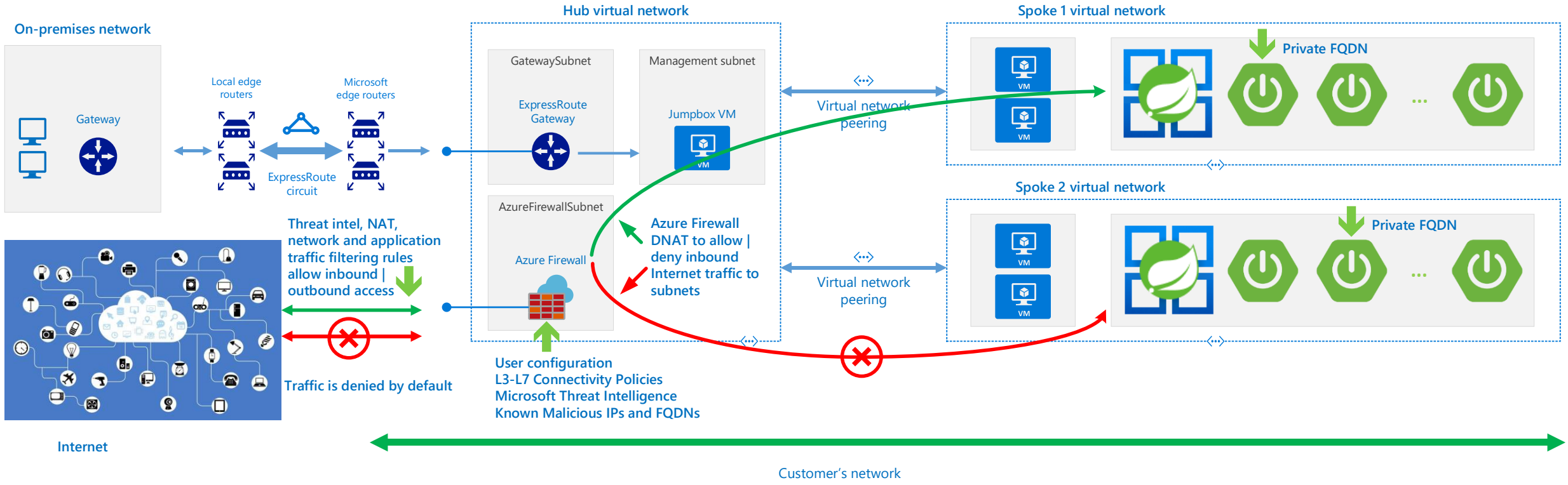
Demo – isolate apps and expose apps to Internet

Integrate with Azure Firewall & App Gateway to allow | deny traffic to FQDNs



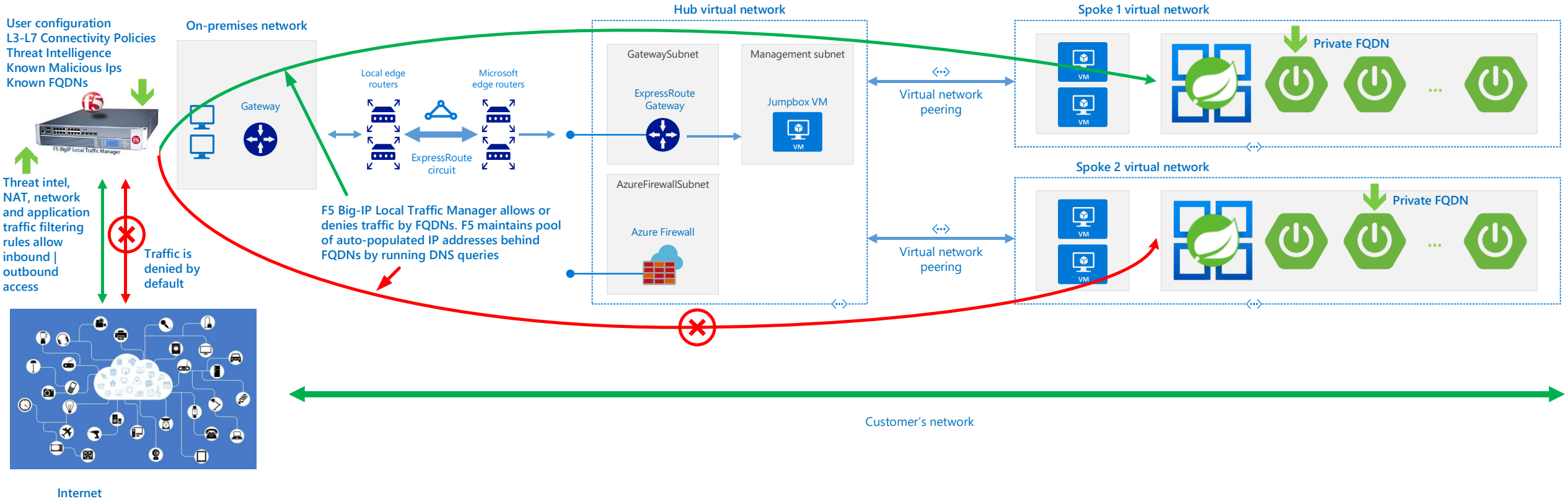
Example 2 - isolate apps and expose apps to Internet

Integrate with Azure Firewall in App Gateway to allow | deny traffic to subnets



Example 3 - isolate apps and expose apps to Internet

Integrate with F5 Big-IP Solution to allow | deny traffic to FQDNs



You can …

- **Control ingress and egress** traffic for VNET
  - Force tunnel Internet traffic (use UDR to define)
  - Define UDR for VNET
- **Assign Private FQDN** to apps
- **Compose with Azure Network** resources
  - Express Route
  - VPN
  - VNET Peering
  - Traffic Manager
  - Application Gateway
  - Azure Front Door
  - Azure Firewall

You can …

- Rely on **service-level diagnostic check** to continuously validate if VNET Injection is operational
  - Ingress, egress, resource health, subnet health, etc.

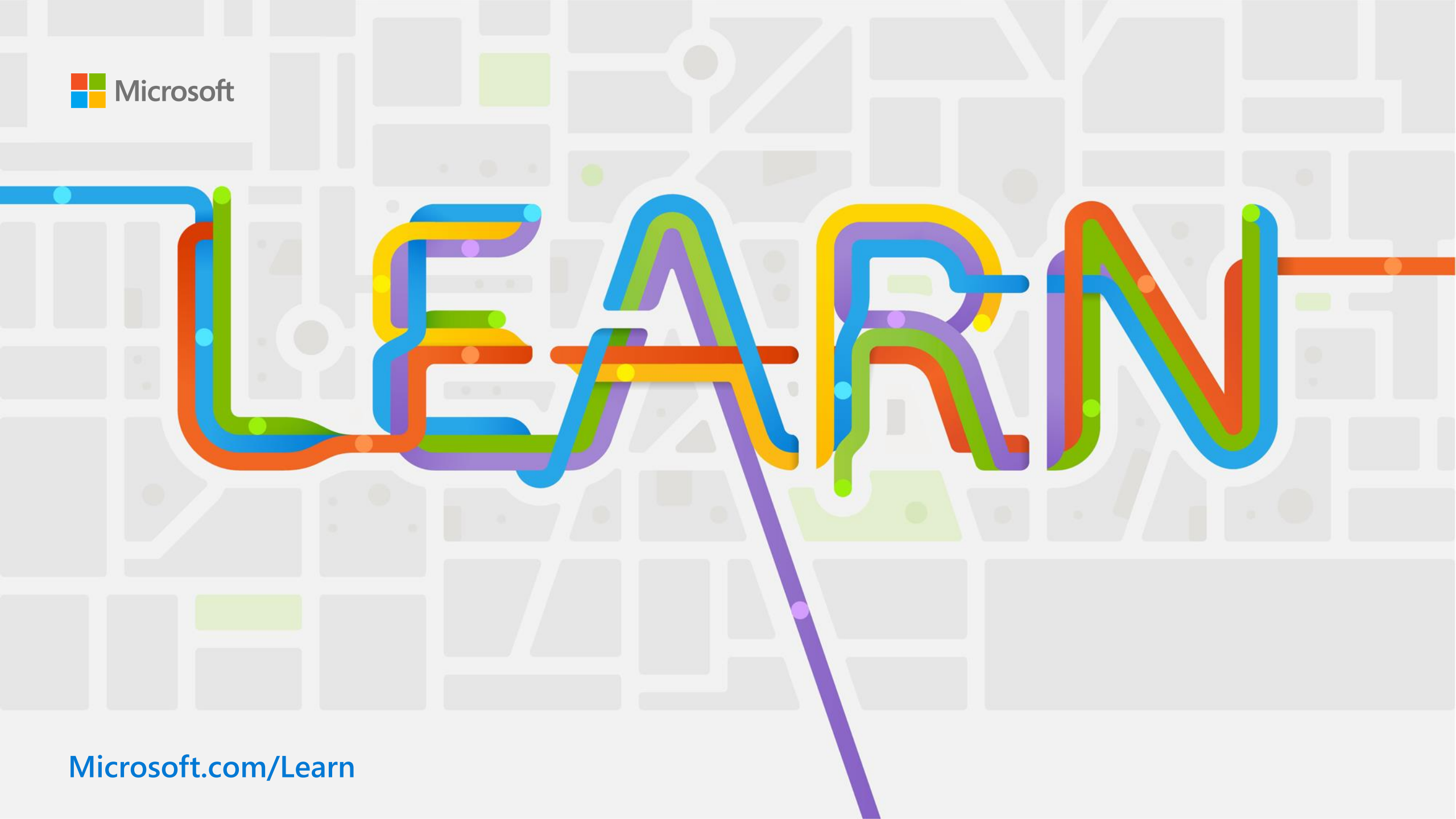
- Upload **certs**, bind **custom domains** to apps, and use certs for **TLS** communications
- Bring your own certs from **any Certificate Authority**
- **Deploy** apps and apply config – just like they do without VNET
  - Using Azure CLI, Maven, IntelliJ, Azure Pipelines, GitHub Actions, Jenkins Pipelines, etc.
- Enable **apps to interact** with Azure Services
  - Using Private Endpoints or Service Endpoints

Microsoft + vmware

Get started – build your cloud-native solutions today!

- Get started with Azure Spring Cloud using quickstart: aka.ms/azure-spring-cloud-start
- Learn using a self-paced workshop on GitHub: aka.ms/azure-spring-cloud-github
- Learn about implementing solutions on Azure Spring Cloud: aka.ms/azure-spring-cloud-docs
- Migrate your apps Azure Spring Cloud
  - Spring Boot: aka.ms/azure-spring-cloud-migrate-springboot
  - Spring Cloud: aka.ms/azure-spring-cloud-migrate-springcloud
  - Tomcat: aka.ms/azure-spring-cloud-migrate-tomcat
- Wire Spring apps to interact with Azure services: aka.ms/spring-integrations

- For feedback and questions, please reach out to spring-team@microsoft.com


Microsoft
LEARN
Microsoft.com/Learn