Danish Ahmed

Technical Architect

About Me

I'm a seasoned Senior DevOps Engineer with over a decade of experience, constantly honing my tech wizardry. By day, I work my magic in the world of technology, ensuring systems run seamlessly and efficiently. In my spare time, I switch gears to a different kind of adventure – playing PS4 with my kids. When it comes to technology and gaming, I'm always up for the next challenge and adventure. Let's explore the digital world together!

Skills & Proficiencies

- *Proficient in cloud technologies, with a focus on Azure*
- *Expertise in infrastructure as code (IaC) and automation*
- *Skilled in container orchestration with Kubernetes and Docker*
- *Database management and performance optimization*
- *Scripting proficiency in PowerShell and other languages*
- *DevOps practices and CI/CD pipeline implementation*
- *Disaster recovery planning and execution*
- *Strong problem-solving abilities*
- *Commitment to continuous learning and skill development*

Education & Certification

MASTERS IN INFORMATION TECHNOLOGY

Virtual University
2021

CERTIFIED KUBERNETES ADMINISTRATOR (CKA)	**IN-PROGRESS**
AZ-400 MICROSOFT CERTIFIED: DEVOPS ENGINEER EXPERT	**2022**
AZ-303 MICROSOFT AZURE ARCHITECT TECHNOLOGIES	**2021**
AZ-104: MICROSOFT AZURE ADMINISTRATOR	**2020**
AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE	**2019**
MICROSOFT CERTIFIED SYSTEMS ENGINEER	**2008**
CISCO CERTIFIED NETWORK PROFESSIONAL	**2010**

Relevant Experience

2023	**TECHNICAL ARCHITECT**
NOW	*Contour Software*
2023	**SR. DEVOPS ENGINEER**
2022	*spur solutions*
2022	**MANAGER IT**
2016	*Traffic Digital*
2016	**SYSTEMS ENGINEER**
2012	*Itsec*

Projects

Azure Infrastructure

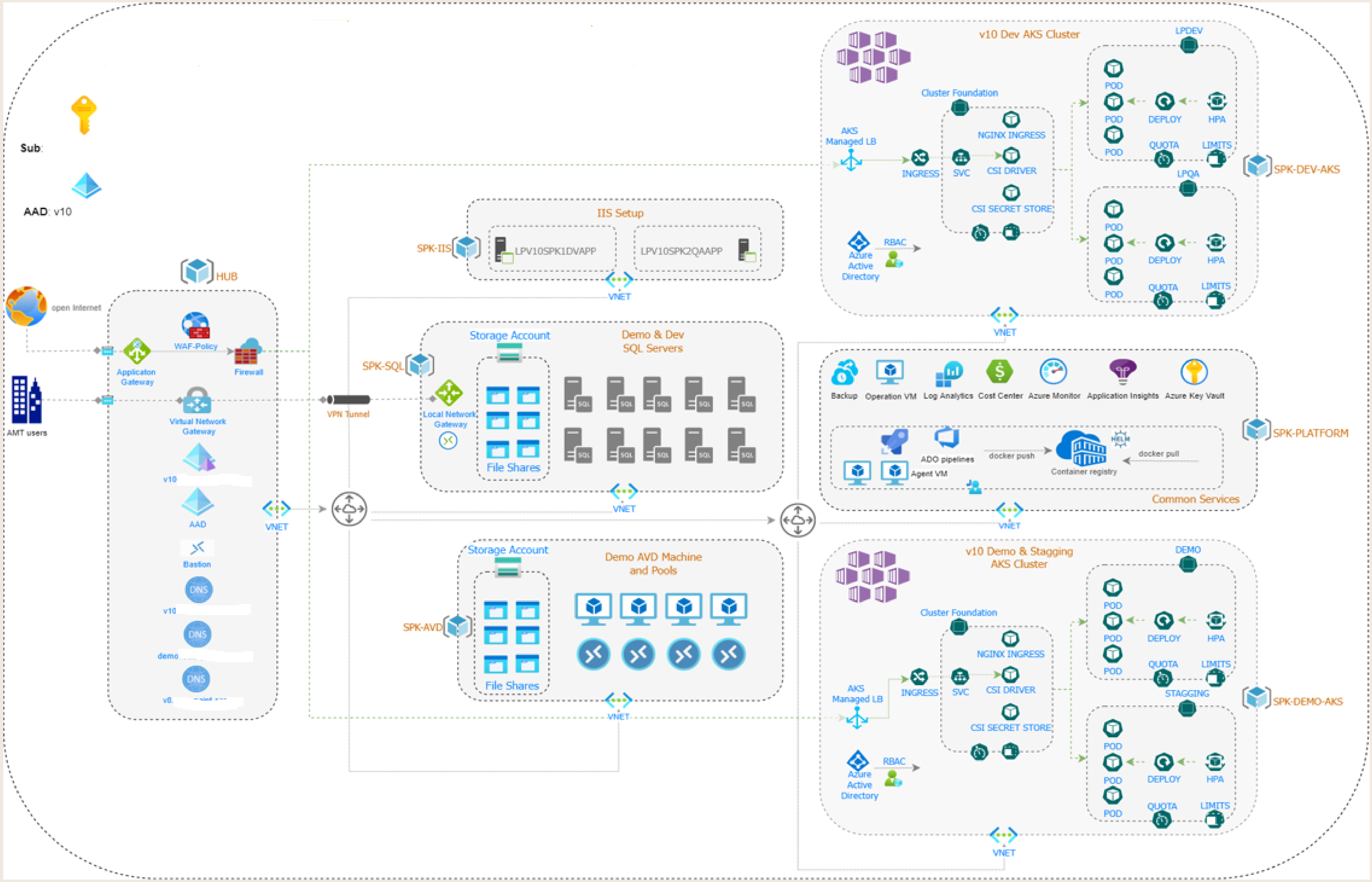
I architected and established a robust hub-and-spoke infrastructure to cater to our company's clientele. To ensure centralized administration of both inbound and outbound access, I leveraged the power of Azure's premium firewall. For efficient routing of incoming traffic, I implemented a multi-site Application Gateway.

Furthermore, to streamline outbound traffic, I skillfully employed VNet peering in conjunction with route tables. This strategic combination allowed us to intelligently direct traffic flows within our network, enhancing overall efficiency.

In the realm of our container orchestration, I expertly configured and deployed Azure Kubernetes Service (AKS) and Azure Container Registry (ACR) for our production and staging environments. To maintain the highest standards of security and access control, I thoughtfully segregated namespaces within AKS and applied Role-Based Access Control (RBAC) to limit client access to only what was necessary.

The AKS environment was fully operational with all its key components finely tuned, including deployments, services, config maps, secrets, Ingress controllers, Persistent Volumes (PV), and Persistent Volume Claims (PVC). This comprehensive approach ensured the seamless orchestration and management of containerized applications across our infrastructure. Below is the visual representation

Infrastructure for a client in Azure



AKS CI/CD Pipeline

I created an automatic process to build and deploy software, including .NET Core 7 APIs and an Angular frontend, using Azure DevOps. This process was defined in a structured format called YAML, ensuring it could be consistently repeated.

The software was deployed on Azure Kubernetes Service (AKS), a platform for managing containerized applications. To make sure the software could handle a variable number of users and remain reliable, I used various strategies. I also set up Nginx Ingress to manage how traffic comes into the software, and I managed important data and configurations using config maps and secrets.

This automation made our development process smoother and improved the quality and reliability of our applications.

App service and IIS CI/CD Pipeline

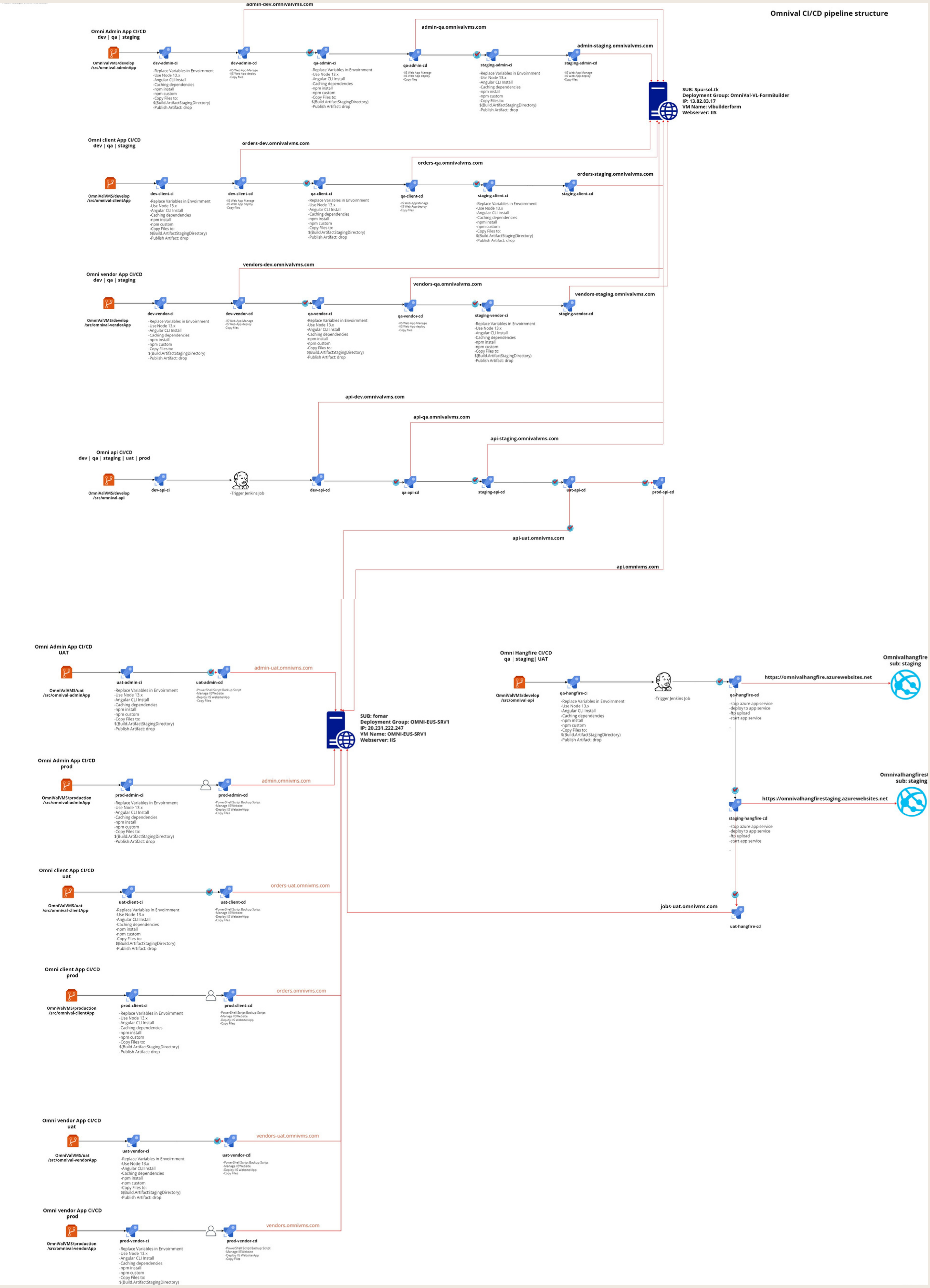
I designed a sophisticated multistage pipeline using a combination of Azure DevOps and Jenkins. This pipeline was structured to deploy our application across various environments, including development (dev), quality assurance (QA), staging, and the final production stage.

Our application, which consisted of both .NET Core and Angular components, required an efficient and cost-effective approach. To achieve this, we harnessed the power of Jenkins, which allowed us to run multiple tasks concurrently, minimizing expenses and speeding up the deployment process.

To facilitate this collaboration between Azure DevOps and Jenkins, we established a service connection. This connection acted as a bridge, enabling seamless communication and data exchange between the two platforms.

Our deployment strategy was versatile, encompassing both Azure App Service and Internet Information Services (IIS). This flexibility ensured that our application could be hosted in a way that best suited the specific needs of each deployment stage.

Pipeline for a product with multiple stages



Disaster Recovery

I designed and put into action a comprehensive disaster recovery (DR) plan that proved its effectiveness during real-world scenarios. This DR plan was orchestrated to safeguard critical assets, including MSSQL databases, Virtual Machines, and Redis caches.

To ensure a swift and efficient recovery, I incorporated PowerShell scripts and runbooks. These automation tools streamlined the DR process, reducing downtime and human error. This approach allowed us to restore operations promptly, safeguarding data integrity and minimizing potential losses in case of unforeseen incidents.

The successful execution of this DR plan underscored our commitment to data resilience and business continuity, providing peace of mind to both our team and stakeholders.

Database Migration

I orchestrated a seamless migration of databases from on-premises infrastructure to Azure Elastic Pools. This strategic move had a profound impact on our operations, enhancing the availability of crucial applications and, most importantly, leaving our customers satisfied.

The migration to Azure Elastic Pools played a pivotal role in bolstering high availability, ensuring that our applications remained accessible and responsive even during unforeseen challenges. It significantly reduced the risk of downtime, contributing to a positive customer experience.

Furthermore, this transition empowered our development team, enabling them to work with greater efficiency. By leveraging Azure's cloud capabilities, developers could streamline their workflows, test applications more effectively, and collaborate more efficiently.

Overall, this migration marked a significant step forward in our commitment to delivering reliable, high-quality services and enhancing the collaborative potential of our development team.

Connect with me

Danish Ahmed

dan.ahmed@live.com
+92 345 239 5630

Find my resume here

CONNECT WITH ME