**SENIOR CLOUD ENGINEER**

**AZURE CLOUD ADMINISTRATION, AUTOMATION FROM**

**TERRAFORM & ANSIBLE, AKS WITH AZURE DEVOPS LIFECYCLE**

**Ashwani Kumar V**

**avkazure@gmail.com** [**LINKEDIN**](https://linkedin.com/in/ashwani-kumar-v-2524a6221)

**Mobile: (+91) 7989 121 975** [**GITHUB**](https://github.com/avardhineni3/mydashboard)

Summary

1. **Senior Cloud Engineer with 9+ years of experience in Azure.**
2. **Experience in working on 50+ Azure Services.**
3. **Experience in configuring single instance of VM to managing Micro-services architecture through Azure Kubernetes (AKS) with Azure DevOps life cycle.**
4. **Experience in Terraform for Infrastructure Automation and Ansible for Server Configuration management.**
5. **Building Azure Build and Release (CI/CD) Pipelines through Azure DevOps.**

Professional Experience

**Senior Cloud Engineer, Visionet Systems Pvt. Ltd. Bengaluru**

**Date: July 2021 – Till date**

**Job Profile: Cloud Infrastructure Automation with Terraform, Ansible & PowerShell**

**Cloud Engineer, TCS, Hyderabad**

**Payroll: Ivytel Technologies Pvt. Ltd. Bengaluru,**

**Date: January 2018 – July 2021**

**Job Profile: Azure Migrations, Azure Administration & Infrastructure Automation**

**Module Lead, Mindtree LTD. Hyderabad, Client: Microsoft, Hyderabad**

**Date: February 2015 – December 2017**

**Job Profile: Azure Administration & Migrations**

**Messaging Server Engineer, Aster Minds Enterprise Solutions Pvt. Ltd. Hyderabad,**

**Client: Microsoft, Bangalore**

**Date: May 2014 – February 2015**

**Job Profile: Exchange & O365 Migration**

**Server Engineer (L3), Zenith InfoTech Pvt. Ltd. Mumbai,**

**Date: July 2012 – May 2014**

**Job Profile: Exchange Server Administration**

Education

**Master of Science (MSc.)** (Innovative Technology), 2011 from UEL, London.

[Education History](https://github.com/avardhineni3/mydashboard/blob/master/Education_History.txt)

Core Competencies

**Git/Git-Hub, Linux & Windows Server Administration, PowerShell & Bash Scripting, Azure CLI, 50+ Azure Cloud Services, Terraform, Ansible, Azure Kubernetes Service (AKS), Docker, Azure DevOps, CI/CD Pipelines.**

Azure Cloud Administration

1. Configured both **Internal and Public Load balancer** by using the standard SKU to avoid overloading of resources and **seamless performance** of applications.
2. Implemented **Inbound NAT rules** to access the VMs in the Backend Pool and **Outbound SNAT to make VM’s** in the Backend Pool to access the Internet.
3. Configured **Azure Traffic Manager, Front door service** for global routing of client applications and **Application Gateway** for routing traffic to web applications within the region.
4. Enabled **WAF on Application Gateway** with both Prevention and Detection mode.
5. Identity Management through **MFA, RBAC, PIM, Conditional Access**, **Identity Protection.**
6. Assigned **fine grain access permissions for Azure Storage** using various authorization techniques and **Azure Key Vault**.
7. Configured the security of the infrastructure to control inbound and outbound access with **stringent** **rules configured on NSG and ASG**.
8. Configured **Azure Bastion Host** to secure VM access **and Azure Firewall** to filter traffic based on **FQDN and to have Microsoft Threat Intelligence** to the client architecture.
9. Configured the **Private link, Service Endpoint** to gain access to Azure storage accounts and database privately through the VNet.
10. Configured an **alert based monitoring system for capturing metrics and Log Analytics Workspace for logs** of VM and storage accounts. Also **configured Application Insights** for capturing performance related data of client live web applications.

Terraform as Infrastructure code

**Have profound experience on:**

Terraform blocks, Terraform workflow, Terraform Providers, Random Provider, Resources Meta-Arguments, Terraform Variables, Terraform Outputs, and Local Values.

Terraform Backend, Terraform state commands, Terraform Modules, Data Sources, Remote State Data Sources, Dynamic Blocks, Terraform Expressions, File Provisioners, Local & Remote-exec Provisioners and Null Resources.

1. Created a **3-Tier Azure Virtual network architecture**. Extensively used terraform concepts to create various Azure Network Resources across Web, App, DB tier and Bastion Subnet. Configured Bastion Jump box & Azure Bastion Service to connect the VM’s in 3-Tiers.
2. Created Azure Linux VMs and boot strapped with Apache Web Server by **using terraform file(), filebase64() and base64encode() functions.**
3. Configured **Azure Standard Load Balancer, Azure Traffic Manager, Azure Front Door, Azure Application Gateway and its features such as Path Based Routing, Multisite Hosting and HTTP to HTTPS Redirect.**
4. Extensively **used resource meta-arguments depends\_on, count, for\_each with maps & strings, for loops, lifecycle, Element, Lookup functions and Splat operators** to change behavior of the resources.
5. Implemented **VMSS – Auto Scaling using Default, Recurrence** & **Fixed profiles** by using metric rules such as Percentage CPU, Available Memory Bytes and LB SYN Count Rule.
6. Standardized **the Terraform modules for all the resources in Azure** Cloud. This helped in reducing the code time and re-usability of the modules.

**Terraform Project through Azure DevOps**

1. Implementing **Terraform projects by Continuous Integration & Delivery Pipelines through Azure Build & Release Pipelines**.
2. Continuous **Integration Pipeline/Azure Build Pipeline tasks include Copy terraform configuration files from Git-Hub** to Build Artifact Directory.
3. Publish Build Artifacts to **Azure Pipelines by YAML file**, so that we can use them in Release Pipelines/Continuous Delivery Pipelines.
4. Created an **Azure Resource Manager Service Connection to Azure Cloud** and also created Storage Account for Terraform State Files. Also configured Terraform Workflow as Tasks and provided related tf.vars file for each stage.
5. Release Pipelines by **accessing the Artifacts created through Build Pipelines provision the Azure resources across 3 stages i.e. Dev, UAT and Prod**. UAT & Prod stage requires approvals to proceed with provisions.

Ansible as Configuration Management

1. Configured Servers, Application deployment, Provisioning, Orchestration, Automation of Tasks and continuous testing of already installed applications.
2. Run Ansible adhoc commands using various modules to fetch the information from remote servers. Core modules used are ping, service, yum, user, group, setup, files, copy, get\_url, shell etc.
3. Have proficient experience in creating static and dynamic inventory files.
4. Create playbooks to configure servers to a specific state.
5. Configured Apache Webserver through Ansible Playbook.
6. Extensively used Ansible Variables to retrieve the results of Ad-hoc commands and Playbooks, used various conditions to control play execution and configure error handling.
7. Used Ansible-Vault to securely encrypt the files and block unauthorized access to the Cloud environment.
8. Performed patching on Linux servers through Ansible.

Git/Git-Hub Bit Bucket as Version Control System

1. Experience in **Installing Git, Creating Local repository, different stages on Git**, and creating private repositories for the projects.
2. Created branches **Dev, UAT and Prod, and protected UAT and Prod branches**, so that only working code is pushed to UAT and Prod branches.
3. **Enabled SSH based authentication**. Added the project team members as collaborators.
4. **Enabled approvals & PR** to check-in code on UAT & Prod branches.
5. Made sure that **Build and Deploy is successful before the code is check-in onto UAT and Prod Branches**.

Linux Administration

1. Experience in navigating through the **file system and directory listing**.
2. **Assigning Permissions and Ownership, file maintenance,** file display, editing the configurations files through text editors.
3. Compress files, using **filter-text processing commands to prepare the bash scripts**.
4. User account management, **System Monitoring** and **System maintenance**.
5. Finding **System Information**, disk partition, **System upgrade and patch management**.

Azure Kubernetes Service (AKS)

1. Used Kubernetes to orchestrate the deployment, scaling and management of Docker Containers.
2. Worked on developing APIs using Kubernetes to manage and specify the copies of the containers to run the actual servers in the cloud environment.
3. Managed Kubernetes charts using Helm. Created reproducible builds of the Kubernetes applications, templatize Kubernetes manifests, provide a set of configuration parameters to customize the deployment and Managed releases of Helm packages
4. Implemented a production ready, load balanced, highly available, fault tolerant, auto scaling Kubernetes cloud infrastructure and microservice container orchestration.
5. Created Clusters using Kubernetes and worked on creating many pods, replication controllers, replica sets, services, deployments, labels, health checks and ingress by writing Yaml files.
6. Working knowledge of Docker Hub, Docker Container network, creating Image files primarily for middleware installations & domain configurations. Evaluated Kubernetes for Docker Container Orchestration.
7. Involved in development of test environment on Docker containers and configuring the Docker containers using Kubernetes.
8. Scheduled, deployed and managed container replicas onto a node cluster using Kubernetes.
9. Worked on Docked - Compose , Docker -Machine to create Docker containers for Testing applications in the QA environment and automated the deployments, scaling and management of containerized applications across clusters of hosts.
10. Building out and scaling Kafka clusters across business functions for large scale and large volume usage. Involved in installing and configuring Confluent Kafka in R&D line, also Validate the installation with HDFS connector and Hive connectors.
11. Used Kibana and Elastic search to identify the Kafka message failure scenarios. Implemented reprocessing of failure messages in Kafka using offset id.
12. Managed local deployments in Kubernetes, creating local cluster and deploying application containers.

Azure DevOps