Aishwarya Reddy

Sr. Azure Cloud Infrastructure Engineer

+1 424-292-2828 | aishrddy@gmail.com | Morrisville, North Carolina

*An enthusiastic professional with around 9 years of IT experience with major focus in **Configuration Management** tools like **Ansible**, **Chef** and **Continuous Integration and Continuous Delivery** (CI/CD), **Cloud Automation**, **Version Control**, End-to-End Automation, **Build and Release, Linux** and **System Administration** with cloud platforms like **Microsoft Azure, Amazon Web Services (AWS),** under the **Unix/Linux/Windows** platforms with strong knowledge on the principles and best practices of **Software Configuration Management (SCM)** in **Agile, Scrum, and Waterfall** methodologies.*

Professional Summary

- In-depth knowledge on various **Azure services** like Compute (Web Roles, Worker Roles), Azure Security Center, **Azure AKS**, HDInsights/**Databricks**, Caching, SQL Azure, NoSQL, Azure Peering, Network services, **Azure Active Directory**, API Management**, Azure Analytic Workspace,** Event Hub, Logic App, **Auto Scaling** and Power Shell Automation.
- Handled Microsoft **AZURE** IaaS –**Virtual Networks,** Virtual Machines, Cloud Services, **Resource Groups, Express Route**, **Traffic Manager**, VPN, Load Balancing, Application Gateways**.**
- Proficient in Azure networking and security, implemented **Azure RBAC** to manage access to **Azure resources** based on users' roles and responsibilities, improving security and compliance with industry standards.
- Configured custom roles in **Azure RBAC** to provide **granular access control to Azure resources**, ensuring that users only have access to the resources they need to perform their tasks.
- Expertise in Azure security and identity management, including **Azure Service Principal** for managing application access to Azure resources like **Azure Active Directory (Azure AD)** for user authentication and authorization.
- Experience with **Azure Key Vault** for securely storing and managing cryptographic keys, secrets, and certificates.
- Integrated **HashiCorp Vault** with different cloud platforms and DevOps tools, such as **Azure, AWS, GCP, Kubernetes, or Jenkins**, using Vault plugins, APIs, or command-line interfaces, enabling seamless and automated access to secrets in various cloud environments.
- Expertise in migrating the existing **v1(classic)** Azure infrastructure into **v2 (ARM)**, scripting and templating the whole end to end process. Migrated on-prem to Windows **Azure** by building Azure **Disaster Recovery Environment**, Azure **Recovery Vault** and **Azure Backups** from the scratch using **PowerShell**.
- Proficient in using Microsoft **Azure** including **Azure CLI,** Azure Management, **Azure Portal, Azure PowerShell,** Azure Management PowerShell Cmdlets.
- Involved in **migrating** SQL Server databases to **SQL Azure** Database using **SQL Azure** Migration Wizard and used **Python** API to upload agent logs into Azure **blob** storage.
- Implemented Enterprise Integration architecture using **Azure Service Bus**, Azure AppService, **Azure Active Directory,** Azure Storage, **Azure hybrid connection manager**, **active directory** authentication for **azure SQL** server and other offerings by azure.
- Automated infrastructure provisioning for **AWS** core services such as **Ec2, VPC's, Load Balancers**, Internet Gateways, Security Groups, and platform services like **Elastic Beanstalk,** ECS**, Fargate, Auto Scaling Group**.
- Constructed **AWS CloudFormation** templates to create custom sized **VPC**, subnets, EC2 instances, **ELB**, security groups.
- Efficiently implemented tagging standard for proper identification and ownership of AWS **EC2** instances and other AWS Services like **Cloud Front,** RDS, **S3, Route53, SNS, SQS, Cloud Trail** and Worked on Simple Storage Service (S3), Simple Notification Service (SNS) and CloudWatch.
- Utilized **Cloud Watch** to monitor resources like EC2, CPU memory, Amazon **RDS** DB services, EBS volumes to set alarms for notification or automated actions to monitor logs for better understanding of the operations.
- Provisioned immutable Infrastructure through **Terraform** which includes multiple **virtual machines, databases, and blob storages.**
- Expertise in integrating **Terraform** with **Ansible**, **Packer** to create and **Version** the **Azure Infrastructure**, designing, automating, implementing and sustainment of **Azure VM images** across the **Azure Cloud environment.**
- Experienced in Configuring the updates, changes and integrating the servers with different environment and creating new machines using configuration management/provisioning tools like **Chef, Ansible** and **Puppet.**
- Worked on **Docker** to containerize the Application and all its dependencies by writing Docker file, Docker-Compose files, **Docker**

container snapshots, managing Docker volumes.

- Proficient in scheduling, deploying, and managing **Kubernetes container replicas** on a node cluster and creating clusters with frameworks running on the same resources. Experienced with building and deploying application code using **Kubectl, kubelet, kubeadm, kubespray** and schedule jobs using **kube-scheduler.**
- Expertise in App Containerization technology **Docker**, creating **Docker images**, Containers, **Docker Registry** to store images, cloud based registry **Docker Hub** and **Kubernetes** services such as **ISTIO Service Mesh**, **Flux** and **ISTIO.**
- Implemented the concept of storage class and deployed postgres database as a **stateful set** into Kubernetes cluster. Development of automation of **Kubernetes clusters** via playbooks in Ansible.
- Administered setting up the complete **Kubernetes** Dev Environment from scratch to deploy latest tools which is related to CICD using helm charts on premises BareMetals for different teams, Configured V-net integration, **Active directory**, **Encryption**, and **Security** in helm charts using configmaps.
- Responsible for **migrating** various services from a managed hosting environment to **AWS** such as automation, monitoring deployments and cutover, documentation, overall plan cost analysis and timeline, service design, network layout, and data migration.
- Approached Continuous Deployment as **GitOps** that leverages **Git** as a single source of truth for declarative infrastructure and applications, providing both revision and change control using **Flux.**
- Expertise in all areas of **Jenkins** setting up CI for new branches, build automation, plugin management and securing **Jenkins** and setting up master/slave configurations.
- Expertise in building **Jenkins** jobs to create **Azure** Infrastructure by pulling **Terraform** code from **GitHub** repositories, ability to work closely with teams, to ensure high quality, timely delivery of builds & releases.
- Expertise in using monitoring servers like **Nagios**, **Splunk**, **Grafana, Dynatrace** and **CloudWatch** for Resource & Network Monitoring and **ELK, EFK** for Log Trace Monitoring.
- Proficient with the version control tools such as **GIT**, **Subversion** (SVN) and **Bitbucket** wherein experienced with tagging, branching on platforms like Linux and Windows.
- Experience in different Issue Tracking Tools like **Remedy**, **Jira** and **ServiceNow**. Designed, deployed, and coordinated with different teams to enhance **ELK platform** and took ownership of new technologies.
- Experienced in system builds, server builds, install, upgrades, backup, performance monitoring, reliability of various flavours of Linux like **Ubuntu**, **CentOS**, RedHat Linux, **Debian**, **Fedora** and SUSE, UNIX (Solaris, IBMAIX) along with Windows.
- Exposed to all aspects of **Software Development Life Cycle (SDLC)** and In-depth understanding of the principles.
- Good understanding of the principles and best practices of **Software Configuration Management (SCM)** in **Agile** and **SAFE methodologies**.

Technical Skills

Cloud Technologies & SDK's	AWS, Azure
Containerization Tools	Docker, Packer.
Build tools	Maven, Ant, Gradle.
SCM Tools	Git, Subversion SVN, Bit-Bucket.
CI/CD Tools	Jenkins, Bamboo, Hudson, Spinnaker.
Orchestration Tools	Chef, Ansible, Puppet, Kickstart, Kubernetes.
Infrastructure as code	Cloud Formation, Terraform, Azure Resource Manager (ARM).
Servers	Apache Tomcat, Apache HTTP server, Nginx.
Version Control	SVN, GIT, Bitbucket.
Bug Tracking Tools	Jira, Remedy, Service Now.
Monitoring tools	Nagios, Splunk, CloudWatch, ELK, Grafana, Dynatrace.
Operating Systems	Windows, MacOS, Linux, CentOS, Ubuntu.
Networking	DNS, DHCP, SMTP, HTTP, SNMP
Virtualization Tech	VMwarevSphereESXi5.x/4.x,ESX/3.x,VMwareWorkstation, Oracle Virtual box

Certifications

Certified Kubernetes Administrator - Certificate ID Number: **LF-b6pb72o6g5**
Microsoft Azure Administrator AZ 104 – Certificate ID Number: **993273842**
Microsoft Azure DevOps Expert AZ 400 - Certificate ID Number: **993273842**

SR. AZURE CLOUD INFRASTRUCTURE ENGINEER | Credit Suisse - Raleigh, NC. Aug '21 - Present

Description: "Credit Suisse Group is a leading financial services company, advising clients in all aspects of finance, across the globe and around the clock**."**

- Proficient in creating and managing **Azure virtual machines (VM)**, storage accounts, network interfaces, and other Azure resources through the **Azure portal, CLI and IAC** and written scripts and commands to automate infrastructure deployment and management.
- Experience with **IAC** tools such as **Terraform** or **ARM templates** to deploying creating and managing Azure resources using declarative code and automate the deployment of complex Azure infrastructure environments.
- Used **Azure Service Fabric** to build and manage scalable and reliable applications composed of microservices that run at high density on a shared pool of machines.
- Configured and deployed **Azure** Automation Scripts for a multitude of applications utilizing the **Azure stack** Including Compute, App Services, **Blobs**, **Resource Groups**, **Azure Data Lake,** HDInsight Clusters, **Azure Data Factory**, Azure SQL and ARM Services and utilities focusing on Automation.
- Designed and Implemented **Azure Site Recovery** in both Disaster Recovery Scenario and for **migrating** the workloads from On-Premises to **Azure** and Build SQL Data Sync job on Windows Azure to synchronize data from **SQL** databases to **SQL Azure**.
- Supervised two tier **Java, Python** web applications to **Azure DevOps CI/CD** to focus on development by using **Repos**, **Test Plans**, deployed **App Service**, **Azure Application Insights,** collects health performance and usage data of the process, stored artifacts in blob storages.
- Configured **Azure Bastion** to enable secure remote seamless **RDP** and **SSH** access to **virtual machines (VMs)** in **Azure Portal** without the need for a VPN or public IP address, ensuring secure connectivity without exposing **VMs** to the public internet in a highly available configurations across multiple Azure regions to ensure service availability and resiliency.
- Designed and implemented **RBAC** policies and role assignments that align with organizational policies and industry standards and conducted periodic reviews to ensure that users only have access to the resources they need, and that access is consistent with their roles and responsibilities.
- Configured **Azure Service Principal** to enable applications and services to access Azure resources securely and with least privilege, reducing the risk of unauthorized access and data breaches.
- Used Azure AD to manage Service Principal access to Azure resources and integrate with organizational identity systems, such as **LDAP** or **Active Directory Federation Services (ADFS).**
- Integrated **Azure Service Principal** with other Azure security features, such as Azure AD Conditional Access or **Azure AD Privileged Identity Management (PIM)**, to provide additional security controls and monitoring capabilities. ---
- Creating and Managing Virtual Machines in Windows **Azure** and setting up communication with the help of Endpoints.
- Configured **Azure Service Connection** to securely connect Azure DevOps pipelines to external services and resources, such as **Azure Kubernetes Service (AKS)** or **Azure Container Registry (ACR).**
- Used **Azure Key Vault** to securely store and manage sensitive information, such as secrets or certificates, used by **Azure pipelines** and Service Connection endpoints, providing centralized and auditable access control.
- Implemented and managed **HashiCorp Vault** for securely storing and accessing secrets, such as passwords, API keys, or certificates, across cloud environments, and integrated Vault with cloud-based DevOps tools, such as Azure DevOps, for streamlined deployment and management.
- Used **PowerShell** to automate the creation of **Virtual Machines (VMs**) using **Packer** for more efficient and quicker image creation process and made the image available such that all the required configurations and applications for the organizational standards and made it available.
- Automated and orchestrated the image build process using **Azure DevOps** or other **CI/CD tools**, enabling faster and more reliable image creation.
- Configured **Shared Access Signature (SAS)** tokens and Storage Access Policies in **Azure Cloud Infrastructure**.
- Identified, analyzed, and interpreted the pattern, trends, and conversational flow in complex data sets.
- Deployed **Azure Kubernetes Service (AKS) cluster** and configured Azure CNI networking, Azure Internal standard Load balancer through Azure Resource manager templates and **Azure DevOps** YAML pipelines.
- Involved in setting up **VPN connections** between individual servers and a Windows Azure virtual network using **Azure point to site VPN** connectivity option and between on-prem virtual network and azure virtual network using **site to site VPN** Connectivity.
- Deployed **Azure PAAS** stack which includes containerized App service hosted in **App Service Environment**, Key vault with private endpoints and **Service Bus** to the development team for Node based Application.

- Utilized **Cosmos DB** for a flash sale requirement, where the hits-per-second requirement is very high, where the application was storing the users and orders details from multiple regions.
- Set up **Azure DevOps** CI/CD pipelines for Micro-services on an **AZURE cloud** using Azure **App services** and Created Web Apps on an **Azure** portal for deploying web applications.
- Implemented a unified infrastructure security management system (**Azure Security Center**) to strengthen the security posture of the data centers and to provide advanced threat protection across hybrid workloads in the cloud.
- Involved in setting up **Azure Functions**, created and configured **HTTP** Triggers in the **Azure** Functions with Application insights for monitoring and performing the load testing on the Applications in QA and Pre-prod using **Artifacts**, **Test Plans** and **Release pipelines** in Azure DevOps**.**
- Written **Terraform** reusable modules to deploy the required infrastructure for applications in **Azure** on multiple environments such as development, test, and production environments.
- Modified existing **Terraform Modules** that had version conflicts to utilize **cloud formation** during Terraform deployments to enable more control or missing capabilities.
- Written **Ansible playbooks** to automatically install packages from a repository, to change the configuration of remotely configured machines and to deploy new builds.
- Worked with **Ansible** to manage Web Applications, Config Files, Data Base, Commands, Users Mount Points, and Packages. **Ansible** to assist in building automation policies.
- Configured and Integrated **Docker** container orchestration framework using **Kubernetes** deployments and used **Python** Scripts to schedule, deploy and manage container replicas onto a node cluster using Kubernetes.
- Developed **Kubernetes** templates for various applications like **Cassandra**, **Grafana**, **Kafka** and setting up Kubernetes Clusters for running micro services, pushed micro services into production with Kubernetes Infrastructure.
- Configured **Kubernetes Replication controllers** to allow multiple pods such as Jenkins master server in multiple minions. Managed **Kubernetes** charts using **Helm** and created reproducible builds of **Kubernetes** applications, templatized **Kubernetes manifests**, configured parameters to customize the deployment and releases of Helm packages.
- Provided consistent environment using **Kubernetes** for deployment **scaling** and **load balancing** to the application from development through production, easing the code development and deployment pipeline by implementing **Docker containerization.**
- Implemented **ISTIO service mesh** to decentralize application networking infrastructure that transparently enables applications to communicate securely and reliably and adds observability and traffic/policy control.
- Defined **Ingress & Egress** routing rules connecting external HTTP and HTTPS requests with internal services and individual pods using **ISTIO**, Configured default backend as part of the **Ingress controller.**
- Implemented **HTTPS Ingress controller** and use TLS certificate on **AKS (Azure Kubernetes Service)** to provide reverse proxy, configurable traffic routing for individual **Kubernetes** services with multiple ingressclasses.
- Migrated all **Kubernetes** container logs, application logs, event gs and cluster logs, activity logs and diagnostic logs into logstash and various azure blob storage activities using filebeat and configured kibana dashboard for accessing logs.
- Configured **Nexus** as enterprise artifactory solutions by writing **snapshot** and released policies and Managed application deployments by integrating with maven build cycles and applied **semantic** versioning concepts to achieve stability across application releases.
- Implemented **New Relic's** suite of monitoring and observability tools, such as APM, Infrastructure, Browser, Mobile, Synthetics, and Logs, and configured and used them to gain insights into application and infrastructure performance, identify and resolve issues, and optimize system behavior along with integrating **New Relic** with other tools and systems using APIs and webhooks, for advanced automation and customization of monitoring and reporting.
- Demonstrated ability in designing, developing, testing, and deploying cloud-based solutions and manage and prioritize tasks, track progress, and adapt to changing project requirements using **Azure Agile** tools and methodology.
- Created **branching** and **merging** strategy with multiple branches and used **Bitbucket** as source code management repository to keep track of version changes.

Environment: Azure, Azure DevOps, Terraform, Ansible, Shell, Python, Linux, Jira, BGP routes, Apache Tomcat 7.x, Docker, Virtualization, Kubernetes, NGINX, Data Dog, Splunk, Pager Duty, Slack, Prometheus, ELK

Sr. Azure Cloud ENGINEER | BCBS – Detroit, Michigan **Sept '19 – July '21**

Description: "*Blue Cross Blue Shield Association is a federation, or supraorganization, of 34 independent and locally operated BCBSA companies that provide health insurance in the United States***"**

- Experienced in **Microsoft Azure IAAS, SAAS** and **PAAS** services such as **Virtual Networks**, **Virtual Machines**, **Blob storage**, **Azure Data Factory**, **Cloud Services**, **Resource Groups**, **Express Route**, **Traffic Manager**, **VPN**, **Load Balancing**, **Application Gateways**, **Auto-Scaling**, deploying **Web Apps,** monitoring by **Azure Application Insights,** and creating **Web-Jobs**.
- Expertise in Azure Development and worked on **Azure web application**, **App Services**, **Azure SQL Database**, **Fabric Controller**, **Azure Search**, and **Notification Hub**.
- Configured Azure Virtual Networks, subnets, **DHCP address blocks**, Azure network settings, **DNS** settings, **security policies** and routing. Also, deployed **Azure IaaS** virtual machines and Cloud services (**PaaS role instances**) into secure Virtual Networks and **subnet**.
- Creating and Managing Virtual Machines in Windows **Azure** and setting up communication with the help of Endpoints and **VM Migrations** from Transitional hosts on VMWare/ Boxes.
- Migrating existing **Active Directory** and Exchange users to Microsoft Azure AD using **Azure AD Connect**, Active Directory Federation Service **ADFS** and **DirSync tools**.
- Configured private end point for Azure Cosmos DB using Azure Private Link through automatic approval method, combined with NSG policies to secure data.
- Provide high availability with **Azure Classic** and **Azure Resources Manager** deployment models. Setup **Azure Virtual Appliances** (VMs) to meet security requirements as software-based appliance functions (Firewall, WAN optimization and intrusion detections).
- Configured the Kubernetes provider with Terraform to interact with resources supported by Kubernetes to create several services such as Deployments, services, **ingress rules**, **Config Map**, **secrets** etc.
- Designed and implemented **Continuous Integration** (CI) using tools **Azure DevOps** over multiple environments to facilitate an **agile development** process which is automated and repeatable, enabling teams to safely deploy code many times a day while ensuring **Azure Kubernetes Services (AKS)** are supported.
- Deployed and optimized two tier **.NET** web application to **Azure DevOps** to focus on development by using services such as **Repos** to commit codes, **Test Plans** to trigger application and unit test, deployed artifacts to **App Service**, **Azure Application Insight** collects health performance and usage data of the process.
- Designed and implemented Azure Repo branching strategies, CI builds and CD releases using **Azure DevOps / VSTS** and Azure Platform and Designed Azure Infrastructure Architecture for globally distributed PAAS web applications.
- Experience in using **Kubernetes** for creating new Projects, Services for load balancing and adding them to Routes by accessing from outside, created **Pods** through new application and controlling, scaling, and troubleshooting pods through **SSH**, managed manifest files and releases of **Helm charts.**
- Functioned as a key player/coach in the implementation of **DevSecOps** model across the organization and implemented security/vulnerability scans at multiple phases.
- Worked on **Kubernetes** Containerization, Orchestration, **Pod configuration**, Kubernetes **API Objects**, Federated Deployments, setting and creation of **Custom Resources**.
- Created **Docker Swarm** using **Docker CLI** to Orchestrate, schedule and deploy the services to Swarm and managed the Swarm behavior and created virtual networks to connect Docker containers across multiple hosts using Docker weave.
- Configured **Sentinel policies** in **terraform enterprise** and making them available across the workspaces to perform compliance checks prior to provisioning of the cloud resources.
- Worked with **Security Compliance** squads to configure **Sentinel** policies (**Policy as a code**) to perform pre provisioning security compliance rule validations prior to infrastructure provisioning and extended scope to perform post provisioning infrastructure scans.
- Provisioned the high availability of Azure VMs using **Terraform** and **ARM** templates, wrote new plugins to support new functionality in Terraform and involved in using terraform to migrate legacy and monolithic systems to Azure.
- Configured **VNET Peering** to enable communication between resources across multiple Virtual Networks and configured **NSGs** for two tiers and three tier applications to filter network traffic using **Terraform modules**.
- Implemented jobs to create **Azure** Infrastructure from **GitHub** repositories containing **Terraform** code and created on-premises active directory authentication using automation with **Ansible playbooks** and migrated the servers with the required configuration changes and testing and deploying the machines using **Ansible** Commands.
- Implemented **Ansible Tower** and written ansible roles, playbooks maintained in SCM to create workflows and schedule nightly jobs to automate patching's from Ansible Tower.
- Configured **Jenkins** on **Kubernetes** container environment, utilizing **Kubernetes** and **Docker** for the runtime environment for the **CI/CD** system to build and test and deploy.
- Used **.Net** API for uploading all the agent logs into **Azure blob storage**. Managed internal deployments of monitoring and alarm services for the **Azure** Infrastructure (**OMS**). Developed micro-services onboarding tools leveraging **.Net** and **Jenkins** for easy

creation and maintenance of build jobs and **Kubernetes** deploy and services.

- Configured **RBAC** and **Azure Monitor** for adding security in Azure Cloud and **Network Security Groups** for Subnet and **NIC** level for Azure Virtual Machine.
- Created and optimized **Node.js** web app and deployed by **Jenkins** into Kubernetes and used **Azure Container Registry** and automated development by authenticating new codes and deploying into staging or production platforms.
- Modified application's code and push the updated code to automatically build and deploy your changes to production by using the **canary testing pipeline**.
- Used different **Prometheus** operations to collect metrics from monitored targets by scraping metrics from http endpoints on these targets and to store the backend data and used **Grafana** dashboards as the interface for analysis and visualization.
- Worked on azure **log analytic workspace** where bulk logs are available once the Kubernetes is enabled, then used **logic app** to filter the required logs and used event hub to transfer the logs to **Splunk**.
- Used **Jira** to track the activities of tasks done by the developers and **Confluence** for documentation purpose.

CLOUD DEVOPS ENGINEER | Comcast - NYC, NY **Feb '18 – Aug '19**

Description: "Comcast is a telecommunications conglomerate that provides cable television, internet, telephone, and other related services to both residential and commercial customers in the United States".

- Involved in designing and deploying a multitude of applications utilizing almost all the **AWS** stack including **EC2, Route53, S3, RDS, Dynamo DB, SNS, SQS, LAMBDA, REDSHIFT,** focusing on high-availability, fault tolerance and auto-scaling in **AWS cloud formation.**
- Implemented various services like **AWS Device Farm**, **Fargate**, **SES**, **TimeStream**, **IoT Core**, **Aurora** etc., and integrated with the existing tools and authentication methods.
- Configured **Docker Swarm Cluster** and setup Jenkins inside it to reduce the failover downtime.
- Managed data warehouse using **Amazon RedShift** to analyze all data using **SQL** & existing business intelligence tools. Converted existing **AWS** infrastructure to serverless architecture (**AWS Lambda, Kinesis**) deployed via **Terraform** & **AWS Cloud formation.**
- Worked on **Terraform** for automating **VPCs**, ELBs, security groups, **SQS** queues, S3 buckets, and continuing to replace the rest of our infrastructure and managed different infrastructure resources Cloud, VMware, Bare Metal Servers, and Docker containers.
- Utilized **Terraform** as **IaC** for the project and used **terraform modules** for different services like **Application Load Balancer**, **EMR**, **ECR** clusters for the project we use in AWS for Infrastructure provisioning.
- Created **terraform** templates for provisioning virtual networks, subnets, VM Scale sets, Load balancers, and **NAT** rules and used terraform graph to visualize execution plan using the graph command.
- Utilized **AWS Cloud Formation** and **AWS Opss Works** to deploy the infrastructure using **Chef** that is necessary to create development, test, and production environments for a software development project.
- Launched **EC2 instances** with various AMI's and Configured Application Servers on those instances by deploying a code in **AWS Elastic Beanstalk.**
- Used **AWS Beanstalk** for deploying and scaling web applications and services developed with **Java, Node.js**, **Python** and **Ruby** on familiar servers like **Apache, Nginx, Tomcat**.
- **Data Migration** projects such as **Teradata** to **AWS Redshift, Teradata** to **Snowflake Migration**, AWS from on-premises.
- Created **Jenkins** jobs for continuously building the projects and integrated Jenkins with **SonarQube** for code inspection and stored the artifacts in **Nexus**. Wrote **Groovy** script in **Jenkins** jobs to do pre-and post-build operations.
- Integrated **Ant**, **Nexus**, **Jenkins**, **JIRA,** and Used **Git** with **Jenkins** to integrate to automate the code checkout process.
- Configured and maintained the **AWS Lambda function** to trigger when there is a **Jenkins** build kicked off, which internally gets stored on **AWS S3 bucket** for everybody to access. Created **Route53** to route traffic between different regions. Worked with **Security groups** & **Network ACLs.**
- Developed **AWS Lambda** functions in **Python** to automate AWS Cloud Watch Scheduled Events between services like AWS SNS and AWS SES to deliver notifications regarding **AWS EC2** instances to the team.
- Designed and configured **bastion host** in **AWS** cloud by using **Terraform modules** and controlled Network Access to **EC2** Instances to withstand attacks.
- Extensively worked on **Chef Roles, Cookbooks, Recipes**, Templates, Resources**,** Attributes & Data bags. Proficient in the setup of **Chef Servers,** workstations & **bootstrapping** infrastructure **Chef Nodes** for configuration management. Written **Chef Recipes** to automate build/deployment process and **data bags** in **Chef** for better environment management.
- Configured the **Weave Net** such that it creates the Virtual networks which connects **Docker Containers** to multiple hosts.
- Designed strategies for optimizing all aspect of the continuous integration, release and deployment processes using container and

virtualization techniques like **Docker** and **Kubernetes.**

- Expertise in automating **Jenkins** to build code based on **Ruby**, **YAML, Python, Shell**, PowerShell, **JSON**, PHP and Perl triggered from GitHub to run web applications on AWS **Elastic Beanstalk** including EC2 build server for continuous delivery with less failover.
- Configured and maintained **Jenkins** to implement the CI/CD process and integrated the tool with **Maven** to schedule the builds. Took the sole responsibility to maintain the CI Jenkins server.
- Configured the users accounts for continuous integration and created Upstream and Downstream jobs using build Pipeline in **Jenkins**. Configured **SVN** with **Jenkins** and scheduled periodic builds. Involved in Branching and Merging of the code in **Tortoise SVN**.
- Experience in writing **Jira API** Tools to auto-move Service Desk tickets of one issue type to a **Jira** project of another issue type and to extract the list of Jira users with the respective Jira Groups and Project Roles.

CLOUD SUPPORT ENGINEER | Accenture -Hyderabad, India **June '16 – Jan '18**

Description: "Accenture is a multinational professional services firm that offers services in strategy, consulting, digital, technology, and operations."

- Responsible for all software builds, including continuous integration **builds, nightly builds** and **release builds** using **Jenkins.**
- Configured **GIT** with **Jenkins** and utilized the **POLL SCM** option to schedule jobs, enabling the automation of the code checkout process. **JUnit** test cases for unit, integration, and functional tests were written and set to run automatically by **Jenkins** triggered by each push to **GIT**.
- Scheduled both **Linux Cron** and **Jenkins jobs** for build automation, developing **Perl** and **python**/**shell** scripts to automate the build and release process and facilitate deployments to **JBoss** and **Tomcat Web servers**.
- Integrated Ansible with **Jenkins** to provide Automation and Continuous Integration through Jenkins-managed nodes over SSH/PowerShell using **Ansible** and also extensively worked on other tools like **Bamboo** and **Hudson** for continuous integration and for all end-to-end automation functions with all build and deployments.
- Collaborated on integrating **Ansible Tower** with a cloud environment, implementing role-based access control, establishing job monitoring, email notifications, scheduling jobs, and utilizing multi-playbook workflows to chain playbooks.
- Leveraged **Ansible Galaxy** to pull reusable modules for Ansible and successfully completed the installation and configuration of middleware.
- Configured and integrated the servers for different environments to automatically provision and configuration management of **Linux** instances using **Puppet**.
- Developing **MAVEN** and Shell scripts to automatically compile, package, deploy and test **J2EE** applications to a variety of **Web Logic** platforms.
- Worked in **Agile Project management** Process, **JIRA** for Issue Tracking and monitoring and with the Architects on SDLC process being the owner of post-development environments.

BUILD & RELEASE ENGINEER | Citi Bank – Hyderabad, IN **Apr '15 – May '16**

Description: "Citibank, also known as Citigroup, is a global financial institution that offers a wide range of financial products and services to individuals, corporations, and governments."

- Worked with **AWS API** to manage **EC2, S3, VPC, CloudWatch, ELB**, **Auto-scaling** and **SNS.** Created **python scripts** using **AWS API** calls to manage all **AWS** resources**.** Integrated **AWS DynamoDB** using **AWS Lambda** to store and backup the **DynamoDB streams.**
- Configuration and administration of **LDAP**, **NFS** and **NIS** in **Linux** and implemented Samba for sharing of resources between Linux and Windows environments.
- Coordinate/assist developers with establishing and applying appropriate branching, labelling/naming conventions using **GIT** source control and analyse and resolve conflicts related to merging of source code for **GIT**.
- Worked on **GIT (GIT Hub)** repositories as **Distributed Version Control System**. Extensively worked on integrating **GIT** into the **Continuous Integration (CI)** environment along with **Jenkins** configured the services using modern **DevOps** tools.