**Avinash V**Phone: 408-479-4932Email: avinash28.devops@gmail.com

**Professional Summary:**

* Over 7+ years of experience in IT industry as a **Linux** & **Windows** System Administrator, **AWS DevOps** Engineer certified by **Amazon Developer** with a major focus in the areas of **AWS**, **Azure**, **Azure DevOps**, Terraform, **Kubernetes,** **Data center/Cloud migration**, configuration management, **CI/CD** pipeline, Virtualization technologies, **Linux/Windows** administration.
* Experience setting up **CI/CD** **Azure DevOps** Pipeline and Deployment Automation for **.Net**, **Java** and **UI** based **Web Applications**.
* Developed microservices onboarding tools leveraging **Python** and **Azure DevOps** allowing for easy creation and maintenance of build jobs and **Kubernetes** deploy and services.
* Experience working on **Azure Cloud services**, **Azure storage**, **Azure CLI**, **Azure** active directory and **Azure Service Bus**. Managing Client's Microsoft Azure-based **PAAS** and **IAAS** environment.
* Expertise in working with **Terraform** key features such as Infrastructure as a code, execution plans, resource graphs, change automation and created infrastructure in a coded manner using **Terraform**.
* Deployed Pods, Services, Secrets, Configs and configured Ingress for microservice applications on top of **AKS** and **EKS** **Kubernetes** Cluster.
* Implemented a production ready, load balanced, highly available, fault tolerant, auto scaling **Kubernetes** Infrastructure and Microservice Container Orchestration.
* Hands on Experience in managing hosting plans for **Azure** Infrastructure, implementing, and deploying workloads on **Azure Virtual Machines** (VMs).
* Good experience with **Helm charts** deployments, writing manifest files to deploy K8s objects/Microservices.
* Written **Power Shell scripts** for archiving and moving of older log files to **Azure** Storage and written automation scripts using **Python Boto** framework.
* Actively worked with development and operations teams to migrate existing legacy solutions to **Azure.**
* Worked on most of the **AWS** Services EC2, IAM, S3, VPC, ELB, RDS, EBS, AMI, SNS, Cloud Watch, CloudFormation, Autoscaling, Cloud Front, Dynamo DB, Redshift and Route53
* Experience in managing and maintaining **IAM** policies for organizations in **AWS** to define groups, create users, assign roles and define rules for role-based access to **AWS** resources.
* Involved in **AWS** S3 services like creating buckets, configuring buckets with permissions, logging, versioning, and tagging & lifecycle policies to back the data from **AWS** S3 to **AWS** Glacier.
* Configuring and deploying instances on AWS, Azure, PCF, cloud environments and Data centers.
* **DevOps** role converting existing **AWS** infrastructure to server less architecture (AWS Lambda, Kinesis) deploying via **Terraform** and **AWS** Cloud formation.
* Used **Kubernetes** to manage containerized applications using its nodes, Config Maps, selector, Services & deployed application containers as Pods.
* Used **Ansible** Playbooks to setup Continuous Delivery Pipeline. Deployed micro services, including provisioning **AWS** environments using **Ansible** Playbooks.
* Experience in working on **Docker** hub, creating **Docker** images and handling multiple images primarily for middleware installations and domain configuration.
* Created **Jenkins CI/CD** pipelines for continuous build & deployment and integrated Junit and SonarQube plugins in **Jenkins** for automated testing and for Code quality check.
* Experience in working on source control management tools like **GIT** (GitHub), **Subversion** (SVN) and creating **GIT Lab** repositories and pipelines with specified branching strategies.
* Experience working on monitoring tools like **Splunk, ELK, Grafana, Cloud Watch, New Relic**.
* Created and maintained monitoring for various services provided on the platform using tools like **New Relic, Azure Monitor, Prometheus** and **AppDynamics.**
* Good knowledge on **Perl**, **Bash**, **Shell**, **Python** & **Developed Python** for automation of the build and release process.
* Involved in handling **ASP .NET** client-side state management (view state, cookies and query strings) and server-side state management (application state and session state) and installed **SQL**Server Failover clustering to meet client needs for both Development and Production environment.
* Installation and Configuration of the **APACHE** web Server on Linux platforms and worked on **PowerShell, Python, and Ruby** for purposes of task automation and configuration management.

|  |  |
| --- | --- |
| **Technical Skills** |  |
| **Cloud Environments** | Azure, AWS, PCF. |
| **Configuration Management Tools** | Chef, Ansible, Puppet |
| **Containerization Tools** | Docker, Kubernetes, Mesos |
| **Virtualization** | VMWare, Windows Hyper-V, Virtual box, Vagrant |
| **Operating Systems** | Linux (Red Hat, CENTOS), Ubuntu, Windows |
| **Automation Tools** | Jenkins, Bamboo, Azure DevOps (VSTS) |
| **Web Servers** | Apache, Nginx, IIS |
| **Database** | AWS RDS, Aurora, DynamoDB, MongoDB, MySQL, MS SQL SERVER, Oracle |
| **Scripting languages** | . Net, Java, Python, Ruby, Bash shell, Power shell, JSON |
| **Networking/Protocol** | DNS, DHCP, TCP/IP, NFS, SMTP, FTP |
| **Version Control Tool** | GIT, Bitbucket, GIT Lab, GitHub |
| **Monitoring Tools** | Nagios, Dynatrace, AppDynamics, Splunk, Cloud Watch, ELK, New Relic, Prometheus. |
| **Application Servers** | Oracle Web Logic, BM Web Sphere, JBOSS |

**Sr. Cloud Engineer/SRE Sep 2020 – Present
Client: Janus Henderson Investors, Denver, CO.**

**Responsibilities:**

* Created **Terraform** modules for **Azure Virtual Networks (VNETs), AZURE batch, Container Registry, IoT Hub, Key Vault, Content Delivery Network (CDN), IoT Hub, Notification Hub, Azure Kubernetes Service (AKS), DynamoDB.**
* Implemented solutions to forward **New Relic Insights JSON data** to **Azure Data Lake** and integrate with **Power BI** to create dashboard for tracking KPI’s.
* Installing cert-manager using **Helm**, encrypting it by using **YAML** manifest file, and updating the ingress resource to automatically request a certificate in **AKS**.
* Migrated hundreds of micro services to azure managed Kubernetes Cluster (AKS) and automated deploying of services, ingress, pods, Istio Service Mesh, Linkerd, Assigning RBAC roles, ACR, Kured etc using Helm Chats and Manifest Files.
* Experience working with **RedHat OpenShift** Infrastructure design, deployment, and operational support.
* Extensive experience implementing the **SDLC** process, deep hands-on with networking, migration, and implementation in Azure.
* Development of Azure DevOps YAML files for IAAC deployment and application build and release pipeline.
* Deployed Secure infrastructure for Azure **Big Data, Azure Data factory, Azure Cosmos DB, Databricks, ADLS Gen2, shir, encryption, firewall, App Gateway, UDR, Private Links, SQLMI, Storage Accounts, Log Analytics Workspace, App Insights.**
* Deployed **Azure Databricks** in Dev, Test, Acceptance & Prod Environments by using Terraform & **Azure DevOps** pipeline.
* Developed self-service applications by leveraging **PowerApps, Azure Logic apps & PowerShell** to increase productivity across teams and reduce ticket workload.
* Worked on Azure Automation account and **PowerShell** for infra-automation. Auto resolving the issues etc.
* Worked on **function app** and azure **automation account** scripts for environment automation like autoscaling of VM size depends on payload from http request, deletion of old blobs from the storage account.
* Worked on **Azure Key Vault** to store the application keys, secret and certificates.
* Worked on **Azure Policies** to enforce organizational standards and to assess compliance at-scale.
* Experience in deploying immutable infrastructure to reduce configuration drift utilizing **GitLab, Azure DevOps, Jenkins, Terraform, Ansible, and Packer**.
* Involved in **Splunk** upgradations, Migrating Infrastructure, Troubleshooting Splunk Enterprise and capacity planning
* Implemented cluster services using **Kubernetes** and **Docker** containers to manage deployments by building a self-hosted Kubernetes cluster using **Terraform** and **Helm** for deploying application containers.
* Used **Python** to pull **Docker images** of analysis tools from **Docker Hub** and containerizing those tools.
* **Built end to end CI/CD Pipelines in Jenkins to retrieve code, compile applications, perform tests and push build artifacts to Nexus and Orchestrate changes across servers and components.**
* Designed and implemented automated process of build and deployment for various **.Net** and **Java** applications using various tools as per the needs of projects which includes **TFS, GIT, Jenkins**.
* Responsible for weekly production deployments through our **CI/CD** process, as well as managing our environments that include: Dev/QA, Performance, and Production.

**Environment & Tools:** Azure IAAS/PAAS, Terraform, ARM Templates, Kubernetes, Jenkins, Ansible, Docker, Java, Python, Splunk, PowerShell, .Net. Azure CLI, Azure DevOps CI/CD, YAML file, Log analytics, Networking, Azure Cosmos DB, Azure Databricks, Azure Data Factory, Linux/Unix (Red Hat, Centos, Debian).

**Sr. Azure/DevOps Engineer Jan 2019 – Aug 2020
Client: PIMCO, Newport Beach, CA**

**Responsibilities:**

* Experienced in **Azure Kubernetes** Service deployment using IAAC Terraform a managed Kubernetes cluster in **Azure** and created an **AKS** cluster in the **Azure portal**, with the **Azure CLI** used template driven deployment options such as Resource Manager Templates and **Terraform**.
* Creating dashboards in **Azure DevOps** for **CI/CD** pipelines. Set up **Azure DevOps** **CI/CD** pipelines for Micro-services on **AZURE** cloud using **App Services**.
* Developed **Terraform common modules** to deploy the infrastructure necessary to create development, test, and production environments for multiple applications.
* Using **Ansible** to manage & configure infrastructure, wrote **Playbooks** to automate the configuration setups, by using the **Ansible** best practice.
* Using **Jenkins** **Pipelines** to drive all micro services builds out to the **Docker** registry and then deployed to **Kubernetes**, created **Pods** and managed using **Kubernetes**.
* Developed new **Splunk** apps to monitor the application log volume (Event count), Indexing volume, missing events, missing hosts/source/source type from **Splunk Monitoring**.
* Development of containerized applications and deploying on **Azure AKS** Clusters using **Helm Charts** and **Azure DevOps** Pipelines.
* Using **MySQL**, **DynamoDB**, **MongoDB**, and **Elastic Cache** to perform basic database administration.
* Worked on **Serverless services**, created and configured **HTTP** Triggers in the **Azure Functions** with application insights for **monitoring** and **performing** load testing on the application using **VSTS** also called as **Azure DevOps services**.
* Automating the installation of **ELK** agent (file beat) with **Ansible Playbook**. Developed **KAFKA** Queue System to Collect Log data without Data Loss and Publish to various Sources.
* Writing **Power Shell** **Script** to do Patching, Imaging and Deployments in **Azure**.
* Creating **PowerShell** **Script** and **Azure Automation,** run books focused around automating daily tasks and quickly implementing time intensive requirements.
* Creating several **Terraform** scripts to build the infrastructure in cloud environment.
* Written **Templates** for **Azure Infrastructure as a code** using **Terraform** to build production environments.
* Developing **Ant** and **Maven Scripts** to automate the compilation, deployment, and testing of **Web** and **J2EE** Applications.
* Using **Jenkins** as Continuous Integration tools to deploy the Spring Boot Micro services to **PCF** using build pack.
* Working with **Ansible Tower**, integrating **Ansible Tower** with **Jenkins** to deploy code to different servers.
* Installing **Docker** Registry for local upload and download of **Docker** **Images** and from **Docker Hub** and created **Docker** files to automate the process of capturing and using the images.
* Working on **Kubernetes** to deploy, scale, load balance, scale and manage **Docker Containers** with multiple namespace versions.
* Configuring and administrating **Jenkins** for automated builds Responsible for installing **Jenkins** master and slave nodes.
* Expertise in release automation for **.Net** technologies to achieve continuous integration and continuous delivery (**CI**/**CD**)
* Working on **ASP. NET** web applications and **.NET** WCF services migration.
* Monitored **API’s** and **Cassandra’s** performance using **New Relic** and **Log Stash**.
* Worked on installation, configuration, performance tuning, security, backup, recovery and upgrades of **Linux/Unix** environment.

**Environment & Tools:**Azure (ARM Templets, Virtual Machines), PCF, Terraform, Kubernetes, Docker, Chef, Jenkins, GIT, Splunk, Maven, Power Shell Script, .Net, Linux/Unix (Red Hat, Centos, Debian).

**AWS Cloud DevOps Engineer Oct 2017 – Dec 2018
Client: Schneider, Green Bay, WI**

**Responsibilities:**

* Involved in complete cycle on migrating physical **Linux/Windows** machines to cloud (**AWS**) and configured Apache webserver in the **Linux AWS Cloud Environment** using **Chef** Automation.
* Worked on **AWS** Auto Scaling in VPC for providing high availability of applications and horizontal scaling of **EC2** instances based on a load of applications by using **AWS Cloud Watch**.
* Managing Cloud security using **AWS IAM** by providing limited access to users, roles, groups and providing restricted access to AWS resources.
* Worked on cloud watch to monitor the resources such as **EC2** instances CPU memory, Amazon to design high availability applications on **AWS** across availability zones.
* Converted existing **Terraform** modules that had version conflicts to utilize cloud formation during **Terraform** deployments to enable more control or missing capabilities.
* Deployed all **Kubernetes Clusters** in datacenter with **Kubespray** and participated in the redesign of a list management system to run as a scalable micro service within the **Amazon cloud (AWS)** using **Amazon Kinesis** and **Lambda**.
* Managed a **PAAS**for deployments using **Docker, Ansible** and **Mesos** which reduced considerably deployment risks.
* Collaborated with development and Support teams to setup a Continuous Delivery environment with the use of **Docker**, continuous build and Delivery.
* Implemented and Configured AppDynamics APM tool for various .**Net, Java, IOS, Android** applications.
* Built **Jenkins** pipeline to drive all Microservices builds to **Docker** registry and deploy to **Kubernetes**. Created **Jenkins** jobs to create **AWS** Infrastructure from **GitHub** repositories containing **Ansible Playbooks**.
* Written **Terraform Templates**, **Chef Cookbooks**, **and Recipes** and pushed them onto **Chef Server** to build and configuring EC2 Instances.
* Developed **Kubernetes** charts using **Helm**. Created reproducible builds of the **Kubernetes** applications, Kubernetes manifest files and releases of **Helm** packages.
* Written Templates for **AWS** infrastructure as a code using **Terraform** to build staging and production environments.
* Created and troubleshooted **Ansible Scripts** and **Kubernetes YAML** files to deploy **SCM**/**CI**/**CD** applications Automatically and Idempotently to reduce time costs.
* Deploying **Java Enterprise** applications to **Apache Tomcat** server cluster and **APIs** to WildFly cluster.
* Worked on Container management using **Docker** by writing **Docker files**, set up the automated build on **Docker HUB**, installed, and configured **Kubernetes**.
* Extensively worked on **Jenkins CI/CD** pipeline jobs for end-to-end automation to build, test and deliver artifacts and troubleshoot the build issue during the **Jenkins** build process.
* Worked on Branches, tagging, merges, cherry-picks, user permissions, release process in **GIT**.
* Developed build and deployment scripts using **Maven** build tools in **Jenkins** to move from one environment to other environments.
* Created SLA/SLO, Overview, operational Dashboards in **Splunk**. Installed **Splunk** agents on on-prem servers using chef and also configured alerts and integrated with PagerDuty

**Environment & Tools:** Aws (EC2), Amazon Kinesis, Lambda, Kubernetes, Jenkins (CICD), Git Hub, Ansible Playbooks, Open Stack, Chef (cookbooks, Recipes), Terraform, Docker, Splunk, Maven. Java, Linux/Unix (Red Hat, Centos, Debian).

**Sr. Site Reliability Engineer Sep 2016 – Oct 2017
Client: T-Mobile, Seattle**

**Responsibilities:**

* Created Environments with **ARM** Templates and migrated current **Azure SQL** Data Center environment using **ADR** and also worked on the **Azure Cloud** including automated deployment of resources using **ARM** templates and the **Azure CLI**.
* Setup repos on **TFS Git**, merging code from develop branch to master branch and make it ready for deployment.
* Installed **Chef Workstation**, Enterprise **Chef** Server from scratch, bootstrapped nodes using knife, wrote recipes and cookbooks and uploaded them to **Chef Server** and used **Chef** Attributes, Templates, and Recipes for managing the configurations across various nodes using **RUBY**.
* Used **Ansible Tower** to enhance the scheduling of the multiple configurations and also to scale the run-time of the clusters and wrote **Ansible Playbooks** and Modules to automate our Build/deployment process and do an overall process improvement to any manual processes.
* Administered and Implemented CI tool **Bamboo** for automated builds migrated all projects from perforce to sub version.
* Integrated **Jenkins** with repositories like **Bit Bucket** to build Automation tools (Maven and Ant), and Artifact repository for pushing successful build code.
* Installed and configured **Nagios** for monitoring the network bandwidth and hard drive status and alert solutions for the server disk management.
* Experience in writing the automation scripts in **Python** and **Ruby** Source Code Management.
* Created **BASH Shell** scripts to automate Cron Jobs and system maintenance. Worked on NoSQL databases including HBase, MongoDB, and Cassandra, Configuring Apache Mesos and Marathon using cloud formation templates on **Ubuntu**.
* Involved in Building and configuring **Red Hat Linux** Servers using Kickstart for the project and maintained maximum uptime, performance for enterprise production, QA and UAT staging.

**Build Release Engineer Sep 2015 – Aug 2016
Client: DAVITA, Denver, CO**

**Responsibilities:**

* Extensively worked on **Jenkins** by installing, configuring and maintaining for the purpose of **CI** and for End-to-End automation for all build and deployments.
* Designed and built automation tools, in **Jenkins** Pipeline logic, to automate code deployment for **Windows & Linux** Application servers and Database instances in **MySQL**.
* Created tagging standards for proper identification and ownership of **EC2** instances and other **AWS** Resources.
* Installing, configuring and administering **Jenkins Continuous Integration** (CI) tool on Linux machines.
* Implemented automations tools like **Jenkins** and deployed monitoring tools like **Nagios**, wrote custom monitors and alerts based of **Nagios**.
* Developed **Python Scripts** to automate various system tasks, OS patches for Dev environments, and deployment of applications to testing/prod environments using **Python** OS modules.
* Modifying and Writing scripts in **Bash Shell** for optimizing day-to-day administration and designed & executed automation test scripts using **Java** as a programming language.
* Worked on Cruise Control **.Net** automation tool for **.Net** code deployments and also worked on **DynamoDB** and **Cassandra** Database for decentralization and replication.
* Experienced in installation and configuration of various OS (Windows & Linux) and maintenance of various distros of Linux like **RHEL**, **CentOS**, **Debian** and **Ubuntu**

**Linux/Unix Administrator** **Aug 2014 - Aug 2015**
**Client: DE Shaw & CO, Hyderabad**

**Responsibilities:**

* Performed all **LVM** and file system tasks like the creation of a logical volume and file systems (ext2, ext3, ext4), mounting and unmounting files, modifying the size of the logical volume by resizing and removing the file system in **Linux environment**.
* Local and Remote administering of servers, routers and networks using **Telnet and SSH** login using **SSH Keygen** tools for password-less setup and agent forwarding on **Linux.**
* Written **Bash** and **Perl Scripting** for automated processes in managing disk space, deleting old logs, and CRON jobs to perform backups using local archiving utilities using tar, GZIP, and bzip2.
* Experienced in configuring the repositories for package installations based on various Linux flavors like **RHEL**, **CentOS**, Debian, Arch Linux etc.,
* Wrote **Shell Scripts** to automate the periodic tasks like removing old logs and files, applying patches and solving day-to-day technical issues on system integration, managing user and security administration related to **Windows and Linux OS**.
* Worked on Setting up **Splunk** to capture and analyze data from various layers **Load Balancers** Web and application servers.

**System Administrator June 2013 - July 2014
Client: Global Nest, Hyderabad**

**Responsibilities:**

* Create user accounts and group policies on **Windows and Linux Servers**.
* Day to day **Linux** hardware support, problem and change ticket implementations, OS builds & upgrades along with day-to-day Storage.
* Analyzed and resolved problems associated with **Linux server**/desktop hardware, application software.
* Configured and monitored **JBOSS** instances on **CentOS** and **Linux Red Hat Servers** and also responsible for administration of **GIT** version control and **Perform activities branching, tagging, backup, restore, access permissions**.
* Set up and configure **Network, switches, routers, WAN and Firewall**.
* Remotely copying files using **SFTP, FTP, SCP and FileZilla**.
* Wrote **UNIX Shell Scripts** for automated installations, to extract logs using **C, Bash, Perl, and Python**.