**HASIKA REDDY.R** [](https://learn.microsoft.com/api/credentials/share/en-us/HASIKAREDDYRAMIDI-3751/6CA8BE061B84DFBC?sharingId=2F6A8124BC041A3E)Atlanta, GA | **Phone: +1 732-512-0009 Ext:200 Email:** [**Rajashekhar.p@logicplanet.com**](mailto:Rajashekhar.p@logicplanet.com)

**PROFESSIONAL SUMMARY:**

* 6+ of experience in IT industry as **DevOps** **Engineer, Build/Release Management, Network Engineer** and **Cloud Management**, Network Engineer and Cloud Management in all phases of **SDLC** like **Analysis, Design, Development, Deploying, Testing** and **Maintenance** of various web based applications.
* Experience in dealing with windows **Azure IaaS** – Virtual Networks, Virtual Machines, Resource Groups, Express Route, Traffic Manager, VPN, Load Balancing, Application Gateways, VM ScaleSets, Vnets, subnets
* Experience in managing Azure Storage Accounts, migrating on premise to Azure cloud using Azure Site Recovery and Azure backups, azure database management.
* Expert in several **Azure** service **like computer (worker Roles, Web Roles), Caching, Azure SQL, Storage and Network services, Azure Active Directory, API Management, Scheduling, Azure Autoscaling, PowerShell Automation, Azure Virtual Machine, Azure search, Azure DNS, gateway of Azure VPN.**
* Troubleshoot and resolve issues for customers to a great extend focused around **Azure VMs, Azure App services, Azure SQL Databases, Application Gateways, VPN Gateways, Azure AD, and Azure migrations**.
* Hands on experience on **Azure VPN** – point to site, Virtual networks, **Azure custom security end security and firewall**. Used Azure Express Route to set up a private connection to Microsoft cloud services such as a Microsoft Azure, Office 365, and Dynamic 365
* Experience in writing Infrastructure as a code (Iac) in Terraform, **Azure** resource management, AWS Cloud formation. Created reusable **Terraform modules** in both **Azure** and **AWS cloud** environments.
* Experience in **AWS Cloud Environment** which includes services like:**EC2, VPC, S3, IAM, Route 53, Load balancing,** **Cloud Formation Templates, Key management service, CloudWatch, ECS, ECR, EKS, SNS, AutoScaling, Control Tower, AWS Organizations, AWS SCP**
* Created and maintained user accounts, profiles, network security and security groups, using **AWS-IAM.**
* Proficient in designing, developing, and deploying backend solutions using Lambda, API Gateway, DynamoDB, and SQS.
* Experience in setting up **CI/CD** pipelines using **Jenkins.** Automated code quality & security scans and dependency management in pipelines.
* Experience in working with **Jenkins master – slave architecture**, plugins installation & configuration, Jenkins files a different kind of jobs in Jenkins.
* Design, wrote and maintained systems is Python scripting for administering GIT, by using Jenkins as a full cycle continuous delivery tool involving package creation, distribution, and deployment onto Tomcat application servers via shell scripts embedded into Jenkins jobs.
* Experience with Cloud automation technologies such as Cloud Formation and **Terraform** and experienced in using **Terraform** for building, changing and manage existing and cloud infrastructure as well as custom in-house solutions.
* Provisioned the highly available EC2 Instances using **Terraform** and **cloud formation** and Managed **AWS** infrastructure as code using **terraform.**
* Creating repositories, branches, tags in **GITHUB** and assist the developers to fix merging issues and creating local, virtual repositories in **Artifactory** for the project and release builds, repository management in **Maven**
* Experience in writing **Ansible playbooks**, created custom playbooks written in YAML, encrypted the data using **Ansible** vault and maintained role-based access control by using **Ansible** Tower to manage web applications.
* Worked with **Ansible** (automation tool) to automate the process of deploying/testing the new builds in each environment, setting up a new node and configuring machines/servers.
* Used Bash and python included Boto3 to supplement automation provided by **Ansible** and **Terraform** for tasks such as encrypting EBS volumes backing AMIs and scheduling Lambda functions for routine AWS tasks.
* Created **Docker file** for each micro service’s and changed some of the tomcat configuration file which are required to deploy Java based application to the docker container.
* Experience in **Kubernetes** to deploy scale, load balance and manage Docker containers with multiple names spaced versions and good understanding of **Open Shift Platform** in managing Docker containers and **Kubernetes cluster.**
* Implemented Twistlock for rule-based access control policy system for docker and Kubernetes containers to scan monitor events, runtime, vulnerabilities, compliance for containers, images, hots, registry, Jenkins Jobs.
* Used **Kubernetes** to manage containerized applications using its node, config Maps, selector, services and deployed application container as pods.
* Used Kubernetes pods and services to external traffic using Service resources, including LoadBalancer, NodePort, and ClusterIP, ensuring connectivity and accessibility. Worked on using Kubernetes Operators and custom resource definitions (CRDs) to automate the deployment and management of complex, stateful applications.
* Deployed windows **kubernetes** cluster with Azure container (ACS) from Azure CLI and Utilized kubernetes and Docker for runtime environment of the CI/CD system to build, Test and Deploy
* Creating repositories, branches, tags in **GITHUB** and assist the developers to fix merging issues and creating local, virtual repositories in **Artifactory** for the project and release builds, repository management in **Maven** to share snapshots and releases of internal projects using **JFrog Artifactory** tool.
* Good knowledge on managing and integrating code quality tools like **SonarQube**, manage Sonar rules, Quality gate.
* Experience in **Linux** Administration Installation, Configuration, Tuning and Upgrades of Linux
* Experience in **version control tools** like **Subversion**, **GIT, BIT BUCKET** and used source code management client tools like **GitHub, GitLab**
* Professional in deploying and configuring **Elasticsearch, Logstash, Kibana** (**ELK**), and AWS **Kinesis** for **log analytics** and skilled in **monitoring** servers-using **Nagios**,**Datadog**, **Splunk**, **AWS CloudWatch, Azure Monitor,** and**ELK**.
* Experienced in configuring servers to provide **Networking Services**, including **HTTP/HTTPS, FTP, SSH, DNS and LDAP.**
* Installation, Configuration, and management of **RDBMS** and **NoSQL** tools such as **MYSQL, DB2, PostgreSQL, MongoDB** and **Cassandra**.
* Good knowledge in various network services like **DNS, NFS, CIFS, FTP, NIS, LDAP, Remote access, Security management** and system troubleshooting skills.
* Good analytical, presentation & communication skills and ability to adapt to new technologies and project environments.

**TECHNICAL SKILLS**

|  |  |
| --- | --- |
| Cloud | AWS, GCP, OpenStack, Azure, Google Cloud Platform. |
| Configuration Management Tools | CHEF, PUPPET, ANSIBLE. |
| Build Tools | ANT, MAVEN, GRADLE, NPM |
| Container Tools | Docker, Kubernetes, OpenShift, |
| Version Control Tools | GIT, CVS, SVN, Bit Bucket. |
| Monitoring Tools | Nagios, Splunk, ELK, Cloud watch, Grafana |
| Scripting | Bash/Shell, Perl, Python, Ruby, power shell, Jason, YAML |
| Databases | My SQL, MS Access, DB2, NoSQL (MongoDB, DynamoDB) |
| Application Servers | WebLogic 9.x/10.x, WebSphere 6.x/7.x/8.x, JBOSS 4.1 |
| Web Servers | Apache HTTP, Nginx, Apache TOMCAT |
| Networking | DNS, DHCP, SMTP, HTTP, SNMP |
| IDE Tools | ECLIPSE IDE, NETBEANS, Visio Studio |
| Operating Systems | Linux (Red Hat 4/5/6/7), UBUNTU, UNIX, WINDOWS NT/2000/2003/XP/VISTA/7/8/10. |
| Virtualization Tech | VMware, Windows Hyper-V, Xen, Virtual Box, Power VM. |
| CI/CD Tools | Jenkins, Bit bucket, Bamboo, Pipelines |

**PROFESSIONAL EXPERIENCE:**

Client: **Nutanix April 2022 - Current**

Role: Devops Engineer

* Implemented **Azure Service** offer such as **Azure cloud services, Azure storage, , Azure Active Directory(AD), Azure Resource Manager(ARM), Azure Blob Storage, Azure Service Fabric, Azure Monitor, and Azure Service Bus**
* Worked on Backup and restore **Azure services** and in Design and configure **Azure Virtual Networks (VNets), subnets, Azure network settings, DHCP address blocks, Azure DNS, security policies and routing. Azure cloud services, Cosmos DB**
* Involved in developing and deploying applications on **Azure cloud** using various Azure services such as **Azure Functions, Azure SQL Database, and Azure Kubernetes Service (AKS).** Proficient in using **Azure DevOps** to create and manage **CI/CD** pipelines for automating the deployment process. This includes configuring build agents, creating release pipelines, and setting up automated testing and deployment strategies.
* Managed various **Azure** resources to ensure their optimal performance, security, and cost-effectiveness. This includes monitoring resource usage, configuring auto-scaling rules, configuring backup and recovery strategies, and optimizing resource utilization. Proficient in using **Azure Portal** and **Azure CLI** to manage resources, create and manage resource groups, and configure networking settings.
* Expertise in Azure **Scalability** and Azure **Availability** – Build VMs availability sets using Azure portal to provide resiliency for IaaS based solution and Virtual Machine Scale sets (VMSS)

Using Azure Resource Manager (ARM) to manage **network traffic.**

* Worked on migrating on-premises applications to **Azure** and configured **VNETS** and **Subnets** as per the project requirement also performed **PowerShell scripting** to do **patching**, **Imaging**, and Deployment of the migrated application on Azure VM’s.
* Knowledge of **Azure Site Recovery** and **Azure Backup** Installed and configured the Azure Backup agent and virtual machine backup agent and virtual machine backup, Enabled **Azure virtual machine** backup from the vault and configured the Azure Site Recovery (ASR).
* Leveraged **Terraform's** integration with **Azure DevOps** to enable effective collaboration and version control for infrastructure-as-code templates. Worked with **Azure Pipelines** to implement continuous integration and deployment **(CI/CD)** for infrastructure changes, ensuring automated testing and validation processes.
* Administer, configure, and maintain **PostgreSQL** databases on Azure, ensuring high availability, data integrity and optimal performance.
* Managed tools such as Terraform and ARM templates to automate the provisioning and configuration of **PostgreSQL** database resources.
* Worked on **Terraform** to define infrastructure resources in code using its own configuration language called **HashiCorp** Configuration Language **(HCL).** By utilizing **Terraform**, I am able to define and manage infrastructure resources through version control tools like **Git**, which allows for automated provisioning and management of infrastructure. This approach ensures consistency and repeatability in infrastructure and enables me to deliver robust and scalable cloud solution.
* Used **Jenkins** pipelines to drive all micro services builds out to the **Docker** registry and then deployed to Kubernetes, created Pods, and managed using **Kubernetes** and created Advanced Jenkins Pipeline with Jenkins Pipeline Scripted Syntax to Trigger Other Remote Jobs on Other Jenkins Masters and Automated the deployment of Java and .Net applications with the use of Jenkins.
* Managed **Jenkins** server and troubleshoot build & release job failures, resolve, work with developers on resolution, Expertise in the **Jenkins** plugin management areas and implemented several CI/CD plugins, build pipeline, **Docker, Git, Junit** and **Pipeline Plugins.**
* Used **Jenkins** to integrate with **Ansible** playbooks and **Terraform** templates for automated configuration of target environments and deployment of infrastructure resources, respectively. Defined playbooks and templates as part of **Jenkins** job configuration, triggering them automatically upon code changes committed to the repository. Parameters were passed to the templates, enabling creation of resources based on application-specific requirements.
* Implemented in installing, configuring, and administering **Jenkins** CI tool on Linux machines and used Jenkins Pipelines to drive all Microservices builds out to the **Docker** Registry and then deploy to **Kubernetes.**
* Worked with **Docker**- docker hub, pulling images from **docker hub**, running containers based on an image, creating Docker file to manage customized containers, exposing a container with a port redirect, container persistence volume management, docker inspect, docker commit to capturing the file changes/settings into a new image and pushing the image to docker hub.
* Managed **Docker** used with **kubernetes** to manage the deployment and scaling of containers across multiple hosts. For greater flexibility and agility in deployment and scaling applications and ensures that containers are distributed in a manner that maximizes resource utilization.
* Worked on Container management using **Docker** by writing **Docker files** and set up the automated build on **Docker Hub** and written **Docker Compose** file for **multi container** provisioning and to **build**, **run**, **tag** and **publish** a docker container to **Azure Container Registry**
* Worked on **Docker** and **Ansible** in build automation pipeline and continuous deployment of code using **Jenkins** and wrote Playbooks to automate **Ansible servers** using YAML scripting and Developed an **Ansible** role for Zabbix-agent which will be integrated into the to the CI/CD pipeline.
* Worked on Container management using **Docker** by writing **Docker files** and set up the automated build on Docker Hub and written **Docker Compose** file for **multi container** provisioning and to **build**, **run**, **tag** and **publish** a docker container to **Azure Container Registry**.
* Used **kubernetes** to manage containerized application using its nodes, config-Maps, selector, services and deployed application containers as **Pods.**
* Managed **kubernetes** using **Helm**. Created reproducible builds of the **kubernetes** applications, managed **kubernetes** **manifest files** and managed releases of **Helm packages**.
* Implementing **clusters** using **Kubernetes** and worked on creating **pods**, replication controllers, Name Spaces, deployments, Services, labels, health checks, **Ingress** resources and Controllers by writing **YAML** files. Integrated them using **weave**, **flannel**, **calico SDN networking**.
* Implemented **Kubernetes** Role-Based Access Control (**RBAC**) to define fine-grained access permissions for users and service accounts, ensuring secure and controlled cluster access by leveraging **Kubernetes** Network Policies to enforce network segmentation and traffic control between pods, ensuring granular security and isolation.
* Worked on configuring and troubleshooting **Kubernetes networking**, including the use of Service mesh, Load balancing, and Network policies to enable secure and efficient communication between applications.
* Managed in securing **Kubernetes** clusters using **RBAC, TLS, Pod Security Policies**, and other security best practices to protect against unauthorized access, data breaches, and other security threats.
* Worked on development of **APIs kubernetes** to effectively manage and specify the desired number of container replicas within a node cluster.
* Maintained **Artifacts** in binary **repositories** using JFrog Artifactory and pushed new Artifacts by configuring the Jenkins project **Jenkins Artifactory plugin**.
* Worked on configuring the **monitoring** and **alerting tools** according to the requirement like **Prometheus** and **Grafana**, setting up alerts and deployed multiple dashboards for individual applications in kubernetes.
* Creating, branching and merging strategy with multiple branches and used **Git** as source code management repository to keep track of version changes.
* Implemented a **GIT** mirror for **SVN** repository, which enables users to use both **SVN** and **GIT** because the team had the upgradation push of the stages of project in both the repository.

**Environment & Tools: Azure DevOps, Azure Functions, Kubernetes, Docker ACS & AKS, Prometheus, Splunk, Terraform, Ansible, Jenkins, Git, Azure Boards, Azure Test plans Maven, Python, Java, PowerShell, YAML, VSTS, Visual Studio Code, Tomcat, Nginx, Linux, Windows servers.**

**Client:** Oracle  **Jan21– March22**

**Role:** Cloud Engineer

* Worked in highly collaborative operations team to streamline the process of

Implementing security Confidential **Azure cloud** environment and introduced best practices for remediation.

* Managed configuring the **Azure Networking services** (Vnet's, Subnets, Network Security Groups, Application Security groups, **VPN gateway, Firewall, Route Tables, Network Interface**, **Load balancers, Application Gateways**)
* Managed Azure Infrastructure Azure Web Roles, **Worker Roles, VM Roles, Azure SQL, Azure Storage, Azure AD Licenses**, Virtual Machine Backup and Recover from a Recovery Services Vault using Azure PowerShell and Azure Portal.
* Used **Azure Kubernetes** service to deploy a managed **Kubernetes cluster** in Azure and created an **AKS cluster** in the **Azure portal,** with the **Azure CLI**, also used template-driven deployment options such as Resource Manager Templates and Terraform.
* Implemented security measures for **Azure Functions**, such as **Azure** Active Directory authentication, role-based access control, and network security groups.
* Utilized **Azure Load Balancer, Azure Application Gateway** and **Kubernetes** native load balancing mechanisms to distribute traffic and ensure high availability and scalability of containerized applications.
* Utilized **Azure Resource Manager** (**ARM**) templates and **terraform** to define and deploy infrastructure resources, ensuring consistent and reproducible environments for containerized applications.
* Implemented **CI/CD** pipelines using **Azure DevOps**, Jenkins for enabling automated builds, tests, and deployments of containerized applications to **Kubernetes clusters**.
* Integrated monitoring and logging solutions like **Azure Monitor, Prometheus**, and **Grafana** to capture container metrics, performance, and logs, ensuring proactive issue detection and resolution.
* Integrated **PostgreSQL** databases with other Azure services such as Azure Active Directory, to Enhance security and authentication.
* Design and implemented high-availability solutions, leveraging Azure technologies like Azure Database for **PostgreSQL**, to ensure maximum uptime.
* Utilized Infrastructure as Code (IaC) tools to automate the provisioning and configuration of PostgreSQL 16
* Implemented robust security measures in PostgreSQL 16, including access controls and encryption.
* Designed and implemented microservices architectures, leveraging **Kubernetes (K8s)** for managing containerized services, ensuring loose coupling, scalability, and resilience.
* Implemented service discovery and load balancing using **Kubernetes** Ingress Controllers or Azure Application Gateway to enable seamless communication between microservices.
* Implemented horizontal and vertical pod autoscaling based on application metrics, utilizing **Kubernetes Horizontal Pod Autoscaler** (**HPA**) or custom scaling mechanisms for efficient resource allocation and cost optimization.
* Ensured secure deployment of containerized applications by implementing **Azure** Container Registry security features, container security scanning, and access control policies.
* Implemented **Azure Network Security Groups** (**NSGs**) and **Azure Identity** and **Access Management (IAM**) roles to enforce fine-grained access controls and secure network communication within **Kubernetes** clusters.
* Conducted regular security assessments and vulnerability scanning for containerized applications, applying security patches and following industry best practices for secure container deployments.
* Successfully designed and implemented cloud-native applications using **Azure cloud services**, **Kubernetes** (K8s), and containers, ensuring scalability, high availability, and fault tolerance
* Leveraged **Azure Container Registry** to store and manage container images, enabling efficient deployment and version control of microservices-based applications.
* Orchestrated containerized workloads using **Kubernetes** (**K8s**), deploying applications to Azure **Kubernetes Service** (**AKS**) for seamless container management, scaling, and monitoring
* Integrated Docker container-based test infrastructure to **Jenkins** CI test flow and set up build environment integrating with Git and Jira to trigger builds using **Webhooks**.
* Implemented **Docker-Maven** plugin and **Maven POM** to build Docker Images for all microservices and later used **Docker file** to build the **Docker Images** from the java jar files.
* Worked with RedHat OpenShift Container Platform for **Docker** and **Kubernetes**. Used Kubernetes to deploy scale, load balance and manage Docker containers with multiple namespace versions.
* Built **docker** images, Created and deployed **docker images** to open shift. And patch update and scale OpenShift environment. Use of **Docker** and OpenShift to manage microservice for developing and testing. Implemented horizontal and vertical pod autoscaling based on application metrics, utilizing **Kubernetes Horizontal Pod Autoscaler** (HPA) or custom scaling mechanisms for efficient resource allocation and cost optimization.
* Implementing clusters using **Kubernetes** and worked on creating pods, replication controllers, Name Spaces, deployments, Services, labels, health checks, Ingress resources and Controllers by writing **YAML** files. Integrated them using weave, flannel, calico SDN networking.
* Deployed **Kubernetes** clusters on top of Servers using **KOPS**. Managed local deployments in Kubernetes, creating local clusters and deploying application containers. Building/maintaining docker container clusters managed by **Kubernetes** and **deployed Kubernetes** using **helm charts.**
* Evaluated **Kubernetes** for **Docker** container orchestration. Managed **Kubernetes** using Helm and 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.
* Implemented **CI/CD** pipelines using **Azure DevOps**, **Jenkins** for enabling automated builds, tests, and deployments of containerized applications to **Kubernetes clusters**.
* Integrated monitoring and logging solutions like **Azure Monitor**, **Prometheus**, and **Grafana** to capture container metrics, performance, and logs, ensuring proactive issue detection and resolution.
* Developed **Ansible playbooks**, inventories, and custom playbooks in **YAML**, encrypted the data using **Ansible** Vault and maintained role-based access control by using **Ansible** Tower, and implemented IT orchestration using **Ansible** to run tasks in a sequence that can work on different servers.
* Working on **Splunk** tool for monitoring all the **Kubernetes clusters** for efficient cluster visibility, proactive monitoring, and triggering action.

**Environment & Tools: Azure DevOps, Azure Functions, Kubernetes, Docker, Terraform, Jenkins, GIT,**

**Prometheus, Artifactory, Ansible, Visual Studio Code, Nginx, Linux, Windows servers.**

Client: Unisys Aug2019 – Dec2020

Role: Cloud Administrator

* Configured **AWS** **Route53** for sending transactional and marketing emails at scale and to manage **DNS** zones globally, create record sets, **DNS** failover and health checks of domains, assign domain names to **ELB** and **CloudFront**.
* Implemented AWS solutions using **EC2**, **S3**, **RDS**, **EBS**, **Elastic Load Balancer**, **Auto scaling groups** and maintained access to **AWS** resources through users and groups on an **IAM** console.
* Implemented event buses, rules, and targets within EventBridge to enable decoupling and flexibility in system components.
* Implemented step functions to coordinate the execution of lambda Functions, API calls.

Leveraged CDK to provision and manage AWS resources programmatically and Deployment processes using CDK to increase efficiency and reduce manual errors.

* Performed the automation deployments using **AWS** by creating the **IAM**s and used the code pipeline plugin to integrate **Jenkins** with **AWS** and created the **EC2** instances to provide the virtual servers.
* Integrated **Jenkins** with **AWS services** such as **CodeCommit**, **CodeBuild**, and **CodeDeploy**, establishing robust and automated software delivery pipelines for cloud-native applications.
* Orchestrated containerized applications using **Amazon ECS**, leveraging **AWS** Auto Scaling and Elastic Load Balancing to ensure scalability, high availability, and fault tolerance.
* Implemented infrastructure automation and configuration management using **CloudFormation**, enabling rapid provisioning and consistent management of **AWS** resources.
* Established robust **CI/CD** pipelines using **Jenkins**, integrating with **AWS** services like **CodeCommit**, **CodeBuild**, and **CodeDeploy**, ensuring efficient software delivery and continuous deployment of Dockerized applications.
* Implemented a load balanced, highly available, fault tolerant, auto-scaling **Kubernetes** **AWS** infrastructure and **microservice** container orchestration.
* Leveraging **kops** implemented a **Kubernetes** Container Orchestration solution within **AWS** allowing for easy management, creation and recovery of **AWS assets**.

**Environment and Tools: AWS ,GIT, Visual Studio Code, Nginx, Linux, Windows servers**

Client: Mastek June 2018 – July2019

Role : Cloud Architect

* Involved in managing **Private Cloud** and **Hybrid cloud** configurations and practices in Windows **Azure**, **SQL Azure**, **Azure Web** and **Database** deployments. **Upgraded** and **Migrated** web applications to latest **.Net framework** versions and **Azure platforms**.
* Created **Azure** automated assets, **Graphical runbooks**, **PowerShell** run books that will automate specific tasks. Expertise in deploying **Azure** **AD** connect, configuring ADFS installation using **Azure** AD connect.
* Created **ARM** templates for **Azure** platform and in migrating on premise to **Windows Azure** using **Azure** Site Recovery and **Azure** backups and other **Azure** services.
* Prepared capacity and architecture plan to create the **Azure** **Cloud** **environment** to host migrated **IaaS,** **VMs** and **PaaS** **role** **instances** for refactored applications and databases.
* Implemented high availability with **Azure Classic** and **Azure Resource Manager** deployment models and worked on Azure access controls, RBAC to manage privileges on Azure resources.
* Creation and Maintenance of **MS Azure Cloud** Infrastructure and **Virtual** **Network** between **MS** **Azure** Cloud and on-premises network for backend communication.
* Deployed and configured **Elasticsearch**, **Logstash**, and **Kibana** (**ELK**) for log **analytics**, and application **monitoring** in integration with AWS **Lambda** and **CloudWatch**. Then store those logs and metrics into an **S3 bucket** using Lambda **function**.
* Integrated AWS **Dynamo DB** using AWS lambda to store the values of items and backup the **Dynamo DB streams,** implemented load balanced, highly available, fault tolerant, auto-scaling **Kubernetes** **AWS** infrastructure and **microservice** container orchestration.
* Deployed **AWS** Cloud services (**PaaS** role instances) into secure **VNets**, **subnets** and designed Network Security Groups (**NSG**s) to control Inbound & Outbound access to Network Interfaces (**NIC**s), **VM**s & **subnets.**
* Wrote scripts using **ANT** tools and automated the build and deploy process using **Jenkins** to move from one environment to other environments. Also edited the existing **ANT** files in case of **errors.**
* Implemented a Continuous Delivery pipeline with **Docker**, **Jenkins** and **GitHub**, whenever there is a change in **GITHUB**, our Continuous Integration server automatically attempts to build a new **Docker** container from it.
* **Environment & Tools: Azure DevOps, Azure Functions, Kubernetes, Prometheus, Splunk, Terraform, Ansible, Jenkins, Git, Azure Boards, Azure Test plans Maven, Python, Java, PowerShell, YAML, VSTS, Visual Studio Code, Tomcat, Nginx.**
* **Client**: Lumen Technologies Feb2017 - May 2018
* **Role**: Linux Administrator
* Day to day duties involved Linux server maintenance and support to developer's team for their issues application, tuning, troubleshooting, and software running.
* Installed the latest patches for, Oracle on Red hat Linux servers, Configured and administered Send mail, Samba, Squid servers in Linux environment.
* Set up the Linux Cron jobs for automating various build related jobs and application data synchronization jobs.
* Responsible for building of Linux OS servers using kickstart automation application.
* Configured Kickstart and Jumpstart servers to initiate installation of RedHat Linux and Solaris on several machines at once.
* Updated previous LDAP tools to work with version of Ruby Rails.
* Involved in Installing, Configuring and Upgrading of RedHat Linux AS 4/5, Solaris 9/10 operating systems.
* Performed automated installations of Operating System using kickstart for Red Hat Enterprise Linux5/6 and Jumpstart for Solaris 9/10 Linux.
* Administered and supported distributions of Linux, including Linux Enterprise Desktop, SUSE Linux Enterprise Server, RedHat and Solaris.
* Install, maintain and upgrade Drupal and Word press on LAMP stack and Configured LAMP Stack on Unix/Linux servers.
* Configured the NIS, NIS+ and DNS on RedHat Linux 5.1 and update NIS maps and organize the RHN Satellite Servers in combination with RHN Proxy Server.
* Worked on Linux Package installation using RPM and YUM, provisioned system with LVM.
* Developed, customized and build packages on Solaris and rpms on Linux for deployment on various servers through Software Development Life Cycle.

**Environments: Oracle on Red hat Linux, Samba, Squid, RedHat Linux AS 4/5, Solaris 9/10, Linux Enterprise Desktop, SUSE Linux Enterprise Server, RedHat and Solaris, LDAP.**