

# Dinesh P

Mobile: +1 (469-482-0033)

Email: [dineshp17560@gmail.com](mailto:dineshp17560@gmail.com)

LinkedIn: [dinesh-p-b39a09205](https://www.linkedin.com/in/dinesh-p-b39a09205)



---

*A dynamic professional 7 years of IT experience with cloud platforms like **Microsoft Azure, Amazon Web Services (AWS), OpenStack, GCP** and with strong knowledge in **Configuration Management** tools like **Ansible, Chef** and **Continuous Integration and Continuous Delivery (CI/CD), Build and Release, Linux and System Administration**, under the **Unix/Linux/Windows** platform's with sound knowledge on the principles and best practices of **Software Configuration Management (SCM)** in **Agile, Scrum, and Waterfall** methodologies.*

## Professional Summary:

- Architected and Implemented **Azure Service** Offering, such as **Azure cloud services, Azure storage, IIS, Azure active directory, Azure resource management, Azure VM, Application Gateway, Data Factory, SQL Database, and Azure Functions, Azure service bus** and experience in Azure Availability and Scalability - Configured VMs availability sets using Azure portal to provide resiliency for IAAS based solution and scale sets using **Azure Resource Manager** to manage network traffic.
- Worked in **.NET, Node.js** applications deployment on **Azure** Cloud using Azure Container and **Kubernetes** services (ACS and AKS). Excellent knowledge of **Azure** compute services, Data factory, Blob storage, **Azure** networking, identity, and Access management.
- Employed **Azure Resource Manager (ARM)** to deploy, update and delete multiple resources on Azure and in migrating the on-premises resources to **Azure** with **Azure** site recovery (ASR) and Azure backups. Worked on creating resource group of virtual machines by specifying deployment parameters in **Azure Resource Manager (ARM)**.
- Hands on experience on Backup and restore **Azure** services and in Design and configure **Azure** Virtual Networks (VNETs), subnets, **Azure** network settings, **DHCP** address blocks, **DNS** settings, security policies and routing. **Azure** cloud services, Blob storage, **Azure** Service Bus, Cosmos DBL.
- Got Opportunity to work on **Azure IaaS - Virtual Networks, Virtual Machines, Cloud Services, Resource Groups, Express Route, Traffic Manager, VPN, Load Balancing, Application Gateways, Auto-Scaling**. And Migrated existing **Active Directory** and Exchanged users to **Microsoft Azure AD** using **Azure AD Connect, Active Directory Federation Service** and **DirSync** tools.
- Expertise in adopting **DevOps** practices such as Continuous development, Continuous Integration (CI) and Continuous Deployment (CD) in runtime with various tools like **Jira, Git, Stash, Bamboo, Jenkins, Docker, Kubernetes, Terraform, Bit bucket, Hudson, Chef, Puppet, Ansible** **Kubernetes** and managing cloud services.
- Designed multiple **VPC's** and public/private **subnets** with required number of IP's using **CIDR** blocks, **Route Tables, Security groups** and **Elastic Load Balancer**. To grant granular permissions to specific **AWS Users, Groups** and **Roles** by creating **IAM policies**. Automated AWS deployment and configuration tasks using **Lambda**.
- Designed and did application deployment of AWS infrastructure utilizing services such as **EC2, RDS, VPC** and managed network and security, **Route 53, Direct Connect, IAM, Cloud Formation, AWS Ops Works** (automate operations), **Elastic Beanstalk, AWS S3, Amazon Glacier** and **Cloud Watch** monitoring Management Gateway.
- Hands on experience in setting up database in **AWS** using **RDS**, storage using **S3 bucket** and configuring instance backups to **S3 bucket** to ensure fault tolerance and high availability.
- Acted as build and release engineer, deployed the services by **VSTS (Azure DevOps)** pipeline. Created and Maintained pipelines to manage the **IAC** for all the applications across the organization.

- Implemented **CI/CD** using **Jenkins** and configured **DevSecOps** strategies, used **SAST** (Static Analysis Security Testing) and **DAST** (Dynamic Analysis Security Testing) to eliminate security flaws, vulnerabilities present in the source code and running application.
- Used **JenkinsAPI** to query the Jenkins server state and change configuration and automate tasks on nodes and jobs. Experience in monitoring tools such as **Nagios, Splunk, ELK**.
- Proficiency in using **Docker Hub, Docker Engine, Docker images, Docker Weave, Docker Compose, Docker Swarm, and Docker Registry** and used containerization to make applications platform when moved into different environments.
- Responsible for all **Jenkins** Admin Tasks such as setting Continuous Integration for new branches, plugin management, securing **Jenkins**, build automation, and setting up control-commute configurations
- Created **Infrastructure as A code** (IAC) templates using **Terraform** and used them to deploy replicas in other environments like **QA, Testing, Prod**.
- Worked on setup the **Kubernetes** sandbox on **Azure Kubernetes Services** (AKS) for testing the different features and manage containerized applications using its nodes, **ConfigMaps**, selectors, services and deployed application containers as **Pods**.
- Proficient in 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**.
- Experience in deploying **Kubernetes Cluster** on **cloud/on-premises** environments with master/minion architecture and wrote many YAML files to create many services like **pods, deployments, auto scaling, load balancers, labels, health checks, Namespaces, Config Map, etc**.
- Have hands on experience using modern storage and infrastructure services like **Redis, Cassandra, MongoDB, RDS, DynamoDB**.
- Setting up/teardown of **ELK stack** (Elasticsearch, Log stash, Kibana) using ansible and troubleshoot the build issues with ELK and work towards the solution.
- Creating fully **automated build** (CI) and **deployment** (CD) infrastructure and processes for multiple projects using **Docker, Jenkins (Cloudbees), sonar cube, Kubernetes, Artifactory, Ansible & GitLab**.
- Have strong knowledge in **Ansible tool**, which is used to manage and configure nodes, Managed **Ansible Playbooks** with **Ansible roles** and used **Ansible Tower** to manage Multiple Nodes and Inventory for different Environments.
- Installed and used **Chef Server Enterprise** on premise/workstation/bootstrapped the nodes using **knife** and automated by testing **Chef Recipes/Cookbooks** with **test-kitchen/chef spec**.
- Integrated **Jenkins** with various DevOps tools such as **Nexus, SonarQube, Ansible** and used CI/CD system of Jenkins on Kubernetes container environment, utilizing **Kubernetes** and Docker for the runtime environment for the CI/CD system to build and test and deploy.
- Migrated many on-premises applications to Azure cloud. Configured required Virtual Networks and Subnets and performed **PowerShell scripting** for Patching, Imaging, and Deployments in **Azure**.

### Technical Skills Summary:

<b>Operating system</b>	Linux, Ubuntu, Windows NT, CentOS
<b>Application Servers</b>	Oracle Web Logic, IBM Web Sphere, JBOSS
<b>Automation Tools</b>	Jenkins, JFrog, Nexus, Docker, Kubernetes, Ansible, chef- solo, habitat, compliance, Kitchen, chef solo, Habitat, compliance, Puppet, Vagrant, Terraform, Cloud Foundry
<b>Version control tools</b>	GIT, GITHUB, SVN, BitBucket, GitLabs, Gerrit, IBM Rational Clearcase, CodeCommit
<b>Web servers</b>	Apache HTTP 2.x ,3.x, Nginx, IIS, Apache Tomcat
<b>Networking/protocol</b>	DNS, DHCP, CISCO ASA Firewalls Routers/Switches, WAN, TCP/IP, NIS, NFS, SMTP, LAN, FTP

<b>Scripting</b>	Perl, Python, Ruby, Bash shell, Power shell scripting, Groovy, PHP, Hashicorp, JSON
<b>Virtualization Technologies</b>	VMWare ESX/ESXi, Windows Hyper-V, Power VM, Virtual box, Citrix Xen
<b>Cloud Environments</b>	Open stack, AWS, Azure, Oracle, OpsWorks
<b>Volume manager</b>	VERITAS volume manager, LVM with Linux, Solaris Volume Manager
<b>Database</b>	AWS RDS, Aurora, DynamoDB, MongoDB, MySQL, MS SQL SERVER 2008/12
<b>Monitoring Tools</b>	Nagios XI, Wavefront, AppDynamics, Splunk, SolarWinds, Cloud Watch, ELK

#### ACCOMPLISHMENTS

- Microsoft certified: DevOps Engineer Expert: [DevOps Expert](#)
- Microsoft certified: AZ-104: [Azure Administrator Associate: AZ-104](#)
- Microsoft certified: AZ-400: [Designing and Implementing Microsoft DevOps Solutions: AZ-400](#)
- The Linux Foundation: [Certified Kubernetes Administrator: CKA](#)

#### Work Experience:

Bank of America, Jersey City, NJ  
Senior DevOps Cloud Architect

August 2019 - Present

#### Responsibilities:

- Converted two tier **.Net** application to Microsoft **Azure** Cloud Service Project as part of cloud deployment using services like **Repos** and **Test plans** for committing codes and triggering applications respectively.
- Configured continuous integration from source control, set up builds within **Azure DevOps** and configured continuous delivery to automate the deployment of ASP.NET Model-View- Controller (**MVC**) applications to **Azure web apps**.
- Suggested and implemented **Azure policy** leading to security audit and **Azure AD** Privilege Identity management (**PIM**), which will allow just in time privilege access escalation for various **PAAS** services.
- Built a prototype **Azure Data Lake** application that accesses 3rd party data services via **Web Services**. The solution dynamically scales, automatically adding/removing cloud-based compute, storage and network resources based upon changing workloads.
- Worked on authorization and authentication in Microsoft hybrid environments and had experience working with **Azure B2B** and **Azure B2C** as well as Security aspects such as **Identity Protection Sentinel** and **Cloud App Security.F**
- Expertise in **Azure Role-based Access Control** (**RBAC**) to separate duties within our team and grant minimum amount of access to users that they need to perform their jobs. Configured **Shared Access Signature** (**SAS**) tokens and Storage Access Policies in Azure Cloud Infrastructure by setting up **Azure DevOps** pipeline with **Azure CLI** task.
- Managed Azure Storage Accounts, Containers, Blob Storage, Azure Data Lake, **Azure data factory**, **Azure SQL data warehouse**, **stretch Database**, **Machine Learning**, **Virtual Machines**, **Polybase**, **PDW**.
- Enabled communication between resources in multiple Virtual Networks by configuring **VNET Peering**. Configured **Network Security Groups** for two tiers and three tier applications to filter network traffic using **Terraform modules**.
- Designed, Developed **Azure SQL Data warehouse** Fact and Dimension tables. Used different distributions (Hash, Replicated and Round-Robin) while creating Fact Dim.
- Configured private end point for Azure Cosmos DB using Azure Private Link through automatic approval method, combined with **Network Security Groups** policies to **secure** data.
- Created and configured **Azure Functions** to work on serverless services like **HTTP Triggers** with application insights for performing load testing and monitor it on the applications using the Visual Studio Team Services (**VSTS**).
- Automated the **Azure IaaS** virtual machines and deployed virtual machine scale sets using **Terraform modules** in production environment.

- Customized build process using **PowerShell** in Azure **pipelines** and designed **ARM templates** for deploying to **Azure Kubernetes service**.
- Implemented **Terraform** modules for deployment of applications across multiple cloud providers. Worked with **Terraform** key features such as **Infrastructure as code**, Execution plans, Resource Graphs, Change Automation.
- 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 comma.
- Used **Azure Kubernetes Service (AKS)**, Ingress API Gateway, MySQL, SQL Databases and Cosmo DB to deploy microservice-based applications on Azure Kubernetes for stateless external data storage and reverse proxy Nginx server setup and SSL and TLS encryption.
- Provisioned **Azure Kubernetes Service** clusters with virtual machine scale set and set up auto scaling using **terraform** and set up azure ad integration with **AKS**.
- Worked on migrating applications from **helm2** to **helm3** also set up **flux** deployments for continuous delivery.
- Configured **Argo CD** on the **Kubernetes** cluster, deployed many applications using **helm charts** and used **customization** to override site specific values as per developer requirements. Worked on poc for **flux** to deploy charts as **HelmRelease** objects and use **Tekton** to create pipelines.
- Wrote many **RBAC policies** to define user permissions in **AKS** and created **Ingress, Egress** rules for regulating the inbound and outbound traffic of the **Azure Kubernetes cluster**.
- Configured service mesh **Istio** for traffic management and query metrics from **Istio** for Http, GRPC, TCP traffic and export them to **Prometheus** and display them on **Grafana** dashboard.
- Used **Clusterapi (CAPI)** along with **CAPZ** (Cluster API provider Azure) to create **AzureMachinePools** and **AzureMachineTemplates** to leverage declarative infrastructure for Azure Kubernetes.
- Set up **Docker-in-Docker (DIND)** as a **Daemon set** to use underlying host daemon to provision **Kubernetes** pods as **Jenkins** slaves and use the dynamic pods to run **Jenkins** jobs. This provides way to utilize **docker-ce** on an old **docker** daemon.
- Managing and Monitoring **Kubernetes** clusters using **Prometheus** as a data aggregator and **Grafana** as a data visualization platform.
- Integrated existing **API's** to **Azure API** management to get all the attributes like security, usage plans, throttling, analytics, monitoring, and alerts.
- Designed Architecture for API development & deployment as Microservice including **Python** code in **Docker** container and **Azure Service Fabric**.
- Worked on **Jenkins** instance which is deployed via Helm Chart to a **Kubernetes** Cluster.
- Created a **Jenkins** pipeline to drive all micro-services builds out to **Docker Hub** and then deployed to **Kubernetes**, created **pods** and configured **Kubernetes** Replication controller. Troubleshoot build & release job failures, resolve, work with developers on resolution. Scheduled jobs to build upon every commit into SCM. Wrote **Groovy script** in Jenkins jobs to do pre-and post-build operations.
- Managed software artifacts required for development using repository managers like **Nexus** and **JFrog Artifactory** and writing **Jenkins** Pipeline Scripts for Continuous Integration (CI) and built workflows also used **Jenkins** for uploading **Artifacts** into **Nexus** Repository.
- Worked on **Ansible** Tower to automate repetitive tasks, quickly deploy critical applications and proactively manage changes. Used **Ansible Vault** and **Ansible** Tower to secure sensitive data and control the access of **playbooks**.
- Installed **Ansible Tower**, which provides an easy-to-use dashboard and role-based access control, so that it is easier to allow individual teams access to use Ansible for their deployments.
- Worked on setting up Continuous Integration (CI) by configuring Build, Code, Deploy and test **automation** Jobs in **Jenkins** for different applications and in creation and deployment using Codeship to automate branch & project creation in Git using (**Groovy** language) in **Jenkinsfile** and automating (using **Ansible**).
- Set up **Jenkins** server and build jobs to provide Continuous Automated builds based on polling the **Git** source control system during the day and periodically scheduled builds overnight to support development needs using **Jenkins, Git**, and **Maven**.

**Responsibilities:**

- Provisioned Azure resources of different providers Compute, Network, **Service Fabric**, Application gateway using **ARM template**. Implemented and maintained Dev, Test, Staging and Production environments leveraging infrastructure as code using Azure **ARM** templates (Infrastructure as code, **JSON / YAML** config).
- Implemented firewall at DNS level by using various resources like application Gateway, Cloud Flare.
- Provisioned Application Gateway and Service Fabric Clusters using **PowerShell** and **ARM** which is a distributed systems platform to deploy and manage **Microservices** with high availability.
- Deployed Azure **AD Connect**, configuring **ADFS** authentication flow, ADFS installation using **Azure AD Connect**.
- Migrated SQL Server databases to **SQL Azure Database** using **SQL Azure Migration Wizard** and used **Python API** to upload agent logs into **Azure blob** storage.
- Recommended on how to duplicate a subset of on-premises machines to the Azure **Infrastructure** as a Service (**IAAS**) offering which will be used for Disaster Recovery. This analysis included the specifics to synchronize on-premises data with SQL Server and SharePoint instances hosted in VMs.
- Used **Azure ExpressRoute** to set up a private connection to Microsoft cloud services such as a Microsoft Azure and **Dynamic 365**. Configured Azure Virtual Networks, subnets, DHCP address blocks, Azure network settings, DNS settings, security policies and routing. Azure API Management, Security, Cloud-to-Cloud Integration (Public, Private).
- Took an initiative and converted an existing **Terraform** template for use with **Azure DevOps** and used Terraform to map more complex dependencies and identify the network issues.
- Maintained the repository management tools like **Artifactory** and **Nexus** to store the WAR, JAR files which are deployed by using chef in **Jenkins** tool.
- Minimized risks, vulnerabilities and conduct risk assessments to identify and evaluate threats during the CI/CD process, by implementing **DevSecOps** with **Jenkins**.
- Worked on implementing test cases to follow **DevSecOps** strategy model that supports microservice applications via **Jenkins** pipelines.
- Designed and installed **Hashicorp vault** for secret management. **Secrets** required for running the application are retrieved from a **vault** at a container's startup time and automatically renewed during the application's lifecycle.
- Managed deployment automation using packer-Docker, **terraform** to automate system operations.
- Used **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.
- Configured **Kube** proxy with **IPVS** dual stack and configured **Kublet** with custom extra arguments such as reserved-CPU's to isolate CPU's for schedule manager to deploy guaranteed **QoS** class.
- Built **Kubernetes** templates for various applications such as and developed Kubernetes Clusters to run micro services and pushed **Kubernetes**-backed infrastructure **micro services** into production. Using **Helm**, controlled **K8s charts** and developed reproducible builds of the Kubernetes applications.
- Adopted **DevOps** practices such as Continuous development, Continuous Integration (CI) and Continuous Deployment (CD) in runtime with various tools like **Git**, **Maven**, **VSTS**, **Chef**, **Docker**, **Kubernetes** and managing cloud services with **Azure**.
- Written **Ansible** playbooks for installing WebLogic/Tomcat application, deployment of WAR, JAR, and EAR files across all the environments.
- Implemented and supported monitoring tools like **Nagios**, **Splunk** on QA and Production servers for Resource, Network and Log Trace Monitoring. Proficient in working with network protocols like **TCP/IP**, **DNS**.



- Redesigned **docker** flies to increase **docker container** reliability to use multi-stage builds and stateless processes bringing uptime from 99.5% to over 99.95%.
- Used **Terraform** opensource tool for building, changing and versioning Infrastructure. Handled multiple providers with **Terraform** simultaneously.
- Worked on **Ansible Galaxy** to pull the reusable modules for ansible and performed the installation and configuration of middleware. Integrated Ansible with **Jenkins** to provide automation, continuous integration through Jenkins and Implemented Jenkins Workflow and Plugins for repeatable deployments of multi-tier applications, and services to Docker.
- Configured **AppDynamics** cluster agent to monitor **Kubernetes** pod failures and restarts, node starvation, missing dependencies such as Services, Config maps and Secrets.
- Introduced **DevSecOps** specific policies and procedures that enabled organizations to keep up with the pace of application development in a DevOps environment. Worked on OWASP Web/API vulnerabilities and compensating controls (CSRF, XSS, SQLI, etc.).
- Created Jenkins jobs to listen for changes in **GIT** and execute the sequence of targets including compile, Junit, created. Jar and .War file and created Jenkins Pipelines with **Jenkinsfile**. Triggering the Remote **Jenkins** Jobs.

AmeriHealth Caritas, Harrisburg, PA  
DevOps Engineer

October 2017 – July 2018

#### Responsibilities:

- Involved in administrative tasks, including the Build, design and deployment of Azure environments, and created Azure automation through the **Runbooks** creation, migration of existing scripts, authorization, configuration, scheduling.
- Migrated from on-prem to Microsoft azure by building **Azure disaster recovery environment and Azure backups** from the scratch using PowerShell script.
- Designed, configured implemented and supported the automated workflows for cloud and virtual environment provisioning using tools such as **Azure Blueprints**.
- Involved in Migrating existing **v1** into **v2** i.e., classic to **ARM** Azure Infrastructure. Scripting and templating the entire end-to-end phase to make it flexible for each migrating region.
- Transformed and moved large amounts of data into and out of other AWS data stores (**RDS**) and databases, such as Simple Storage Service (**Amazon S3**) and Amazon **DynamoDB** using **EMR**.
- Configured **Route 53 as a Domain Name Service across the enterprise** and configured public and private interfaces for the external clients and internal consumers.
- Managed and orchestrated complicated **CloudFormation** templates through **terraform** and migrated enterprise infrastructure standard to terraform.
- Authored **Terraform** modules for infrastructure management and published a module to the **Terraform** registry for enterprise customers to deploy our product inside their AWS environment. And Set up build and deployment automation for **Terraform** scripts using Jenkins.
- Used **AWS Fargate** in collaboration with **AWS ECS** and **AWS EKS** for container-based application running to provide high security and reliability.
- Implemented **CI/CD strategy** in AWS **CodePipeline** that tracks the Security Vulnerabilities for applications running in **AWS ECS** using **Amazon Security Hub**.
- Integrated **Terraform** with **Ansible** and automated terraform life cycles with ansible playbooks.
- Manage re-architecture of **Jenkins** and integration with **Confluence** for release management and documentation assets. Re architect a **Maven** based system reducing build times and enabled security layers by implementing FOSS and FORTIFY scans during the maven deployments.
- Configured applications that run multi-container **Docker** applications by utilizing the **Docker-Compose** tool which uses a file configured in **YAML** format. Installed **Docker Registry** for local upload and download of **Docker** images and from **Docker Hub**. Created **Docker** files to automate the process of capturing and using the images.

- Creating the **Docker** containers and **Docker** consoles for managing the application life cycle. Responsible for Build and Install Schedules using **Unix** Cron jobs.
- Deployed **Kubernetes** clusters on top of AWS 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**.
- Made use of **Kubernetes** for creating new Projects, Services for **load balancing**, and adding them to Ingress by accessing from outside, created **Pods** through new application and controlling, scaling, and troubleshooting pod failures.
- Integrated **ELK Stack** to an existing appliance framework for real time log aggregation, analysis, and querying (**Elasticsearch**, **Logstash**, **Kibana**) and using **ELK Stack**, **Elasticsearch** for deep search and data analytics. **Logstash**, **Splunk** for centralized logging, log enrichment and parsing and **Kibana** for powerful and beautiful data visualizations.
- Maintaining automated environment using **Chef recipes** and **cookbooks** and implemented bootstrap process.
- Integration of **GIT** with **Jenkins** to automate the code check-out process with the help of **Jenkins DSL Plugin**.
- Wrote **Jenkins shared libraries** for CI and CD life cycles and created generic pipelines across the cloud applications to automate application deployment as well infrastructure deployments and introduced various security compliance rules to validate the organization standards over the release work flows.
- Configuring and managing of **Chef** Master server: updating and creating modules and pushing them to **Chef Clients** and deploying application operational scripts to webserver like Apache Tomcat and JBoss using manifests.
- Developing build and deployment scripts using **ANT** and **MAVEN** as build tools in **Jenkins** to move from one environment to other environments for Continuous Integration.

Abbott, Minneapolis, MN  
AWS Cloud Administrator

October 2016 – September 2017

#### Responsibilities:

- Managed the user accounts (**IAM**) for AWS cloud resources and managed **ELB**, **EC2**, **RDS**, **ROUTE53**, **VPC**, **RDB**, **Dynamo DB**, **SES**, **SQS** and **SNS** services in **AWS** cloud.
- Worked in core **AWS services** (**S3**, **EC2**, **ELB**, **EBS**, **Route53**, **VPC**, **auto scaling** etc.) and deployment services (**OpsWorks** and **Beanstalks**, **Lambda**) and security practices (**IAM**, **KMS**, **SSM**, **Cloud watch** and **Cloud trail**) and services **Kinesis**, **SNS**, **SQS**, **EMR**, **RedShift**, **ECS**, **Elastic Bean Stalk**, **X-ray** etc. .
- Built servers using **AWS** includes importing necessary volumes, launching **AWS EC2 instances**, **AWS S3** for object static webpages and created **security groups**, **auto scaling**, **AWS load balancer**, **Route 53** and **SNS** as per architecture using **Cloud formation JSON Templates**.
- Worked directly with **AWS** architects to help reproduce and resolve customer issues.
- Created monitors, alarms, and notifications for **EC2 hosts** using **Cloud Watch** Monitored System Performance Managed Disk Space **LVM (Logical Volume Manger)** and performed system Backup and Recovery.
- Worked on **AWS Cloud Watch**, **Cloud Formation**, **Cloud Trail services** and **Cloud Front** to set up and manage cached content delivery.
- Automatically spin- up API's and database servers in each layer of the **OpsWorks** Stack in AWS cloud using boto3 python scripting.
- Integrated **SVN** and **Maven** with **Jenkins** to implement the continuous integration process.
- Used **Atlassian Product** such as **JIRA** to track issues and Change Management.
- Coordinate/assist developers with establishing and applying appropriate branching, labeling/naming conventions using **Subversion (SVN)** and **GIT** source control.
- Wrote JUnit test cases for unit, integration, and functional tests, run automatically by **Hudson/Jenkins** in the builds triggered by each push to **GIT**.
- Worked on creation of **Docker** containers and **Docker** consoles for managing the application life cycle.

Cisco, San Francisco, CA  
Build and Release Engineer

October 2015 – September 2016

**Responsibilities:**

- Participated in the release cycle of the product which involved environments like **Development, QA, UAT** and **Production**.
- Build servers using **AWS**, importing volumes, launching **EC2, RDS**, creating security groups, auto-scaling, load balancers (ELBs) in the defined virtual private connection (**VPC**), provided high availability of applications of **EC2** instances by taking the advantages of AWS High availability options.
- Configured **S3** versioning and lifecycle policies in **AWS**, to backup files and archive files in **Glacier**.
- Migrated and automated cloud deployments using **API's, Puppet, python** (Boto & fabric) and **AWS Cloud Formation Templates**.
- Installing and configuration of an automated tool **Puppet** that included installation and configuration of **Puppet Master, Agent nodes** and an admin control **workstation**.
- Wrote **Docker** files to build Micro service applications and created docker images to reused to create the servers and created the lightweight deployment workflow by adopting container strategies.
- Worked on **Jenkins, Git Hub, Puppet, Artifactory** and all internal build systems for the core development team on an enterprise-level Java-based cloud **orchestration/automation** tool.
- Involved in Configuration of **LDAP, NFS** and **NIS** in Linux and implemented **Samba** for sharing of resources between Linux and Windows environments.

Accenture Hyderabad, INDIA

June 2014 – September 2015

Linux System Engineer

**Responsibilities:**

- Involved in design, configuration, installation, implementation, management, maintain and support for the Corporate **Linux** servers **RHEL 4.x, 5.x, SLES 9, CENTOS 5.x, Solaris** for business critical with Disaster Recovery.
- Managed and administrated of all **UNIX** servers, includes **Linux** operating systems by applying relative patches and packages at regular maintenance periods using **Red Hat Satellite** server, **YUM, RPM** tools.
- Customized and compiling the **Linux kernel** according to the application/user requirements and various communication protocols by creating and maintaining new **VM boxes** using **Linux virtual machine** templates.

Infosys Hyderabad, INDIA  
Dot Net(.NET) Developer

January 2014 to May 2014

**Responsibilities:**

- Working as a **C#.NET** developer in an Agile environment and developed the company's flagship product named AmpliFund, built primarily using the C#.NET **MVC** framework and developed static websites using **ASP.NET** interactive web applications with languages such as C#, JavaScript, Razor, LINQ, HTML5, CSS3.
- Worked on routing to load controller type using Asp.Net MVC for reaching appropriate action in the web application.
- Prime responsibilities include developing new features/module and debugging, fixing/patching existing bugs to face-lift and modernize the current application.