**Mukta H Email: mukta.h202@gmail.com
Sr. DevOps Cloud Engineer** **Phone: (415) 735-1739**

*Around 7 years of professional experience in the field of IT industry focused on multiple*

***Cloud technologies****,* ***DevOps*** *and* ***Linux system administrator*** *that includes the principles with Continuous Integration, Continuous Delivery and Continuous Deployment.*

**PROFESSIONAL SUMMARY:**

* Experience in Installation, Configuration, Management of various flavors of Linux (**RHEL, CentOS, and Ubuntu**), built customized infrastructure for multiple applications on cloud, on-premises and established connectivity between all the resources.
* Experienced in designing & deploying applications utilizing AWS services like **EC2, EBS, Route53, S3 bucket, S3 Glacier, Lambda, RDS, DynamoDB, Elastic Load Balancer, Identity Access Management** focusing on high-availability**,** fault tolerance, auto-scaling in **AWS Cloud Formation**.
* Experienced in setting up database instances in Amazon Web Services (AWS) using **RDS**, **DynamoDB**, storage using **S3 bucket** and configuring instance backups and archives to Amazon **Glacier** archives, and configured AWS **Redshift** for Data Warehousing.
* Experienced on various **Azure Services** like **Virtual Machines** (VM’s), **Active Directory**, **Web Roles, Worker Roles**, **Azure Websites**, **Azure SQL**, **Storage**, **Azure Virtual Network** (VNet)**,** **Azure Active Directory**, **API** **Management, Load Balancing**, **Auto Scaling**, and **PowerShell** Automation.
* Experienced in configuring and deploying **Azure** Automation scripts for applications utilizing the Azure services that including **compute**, **blobs**, **Azure Data Factory** (ADF), **Azure Data Lake**, **Azure SQL**, **Azure Kubernetes service** (AKS) and **ARM** templates and utilities focusing on Automation.
* Experienced in **Google cloud platform** (GCP) services like **compute engine**, **cloud load balancing**, **cloud storage**, **cloud SQL**, **stack driver monitoring** and **cloud deployment manager**.
* Experienced in building sophisticated and highly automated infrastructure using **Cloud Formation, Azure Resource Manager** (ARM)**, Cloud Deployment Manager** and **Terraform** automation frameworks in cloud platforms like **AWS, GCP** and **Azure.**
* Experience in Designing, Architecting, and implementing scalable cloud-based web applications using A**WS and GCP.**
* Set up GCP **Firewall rules in order to allow or deny traffic to and from the VM's** instances based on specified configuration and used **GCP cloud CDN (content delivery network) to deliver content from GCP** cache locations drastically improving user experience and latency.
* Experience in Designing, Architecting, and implementing scalable cloud-based web applications using **AWS and GCP, Azure as various environments as well as good experience in Cloud Infrastructure management and Implementation (ARM and ASM).**
* Proficient in **GCP Firewall rules in order to allow or deny traffic to and from the VM's instances** based upon specified configuration and configured **GCP cloud CDN (content delivery network) to deliver the content from GCP** cache locations drastically developing user experience and latency.
* Extensive experience in the implementation of Continuous Integration (CI), Continuous Delivery and Continuous Deployment (CD) on various Java based Applications using **Jenkins, Ansible, Chef, Puppet, Maven, Git, Nexus, Docker,** and **Kubernetes.**
* Experienced in writing **Ansible Playbooks** for automating **CI/CD** pipelines, deploying, updating launch configuration for microservices, provisioning load balancers, and auto-scaling instances.
* Experienced in working with **Ansible Tower** to manage multiple nodes and inventory for different environments, automated repetitive tasks and deployed critical applications on cloud using **Ansible**, and **AWS Cloud Formation** Templates.
* Experienced in building **Docker containers** using **docker compose**, automated **Docker Image** build and pushed those artifacts to **Nexus** using **Jenkins**.
* Experienced in using **Kubernetes** to provide a platform for automating deployment, scaling, and operations of application containers across several clusters of hosts.
* Expert in writing **Jenkins pipelines** to drive all the micro services builds out to the Docker registry and then deployed the docker images to the Kubernetes, created pods and managed using Kubernetes.
* Experienced in deploying and configuring **Chef server** including bootstrapping of **Chef Client** nodes for provisioning and creating **roles**, **recipes**, **cookbooks** and uploaded them to the Chef server.
* Experienced in managing on-site OS, Applications, Services, Packages using **Chef** as well as **AWS** for **EC2**, **S3 buckets**, **Route53** and **Elastic Load Balancer (ELB)** with **Chef Cookbooks**.
* Experienced in working on **Chef** with **Knife** commands to manage **Nodes**, **Cookbooks**, **Chef Recipes**, **Chef attributes**, **Chef Templates** and used **Ruby** scripting on Chef automation for creating cookbooks comprising all resources, templates, attributes.
* Experienced in deploying **Puppet**, **Puppet Dashboard** and **Puppet DB** to existing infrastructure and created puppet modules for Protocols configuration and managing them using **Puppet automation**.
* Experienced in writing **Puppet** manifests for deploying, configuring, installing, and managing collected for metric collection and monitoring and constructed the puppet modules for continuous deployment.
* Extensively worked on **Jenkins** for **Continuous Integration (CI)** from pulling code from version control tools like **GIT**, **Subversion,** and building of deployable artifacts from source code using build tools like **Ant, Maven** and used artifacts repository managers like **Nexus**, **JFrog** for storing the builds.
* Expert in using source code version tools like **Subversion**, **Git** on **LINUX** and **Windows** environment.
* Experienced in deploying and configuring **Git** repositories with branches, forks, tags, cloning, labels, and merge requests.
* Proficient in scripting languages like **Shell**, **Python**, **Ruby**, **PowerShell**, **Bash**, **YAML**, **Groovy** scripts.
* Complete understanding of Software Development Life Cycle (**SDLC**) with **Agile Methodologies**.
* Experienced in setting up **JIRA** as a defect tracking tool and configured various plugins, workflows, customizations for **JIRA bug tracker.** Involved in planning and execution of the migration from **Bugzilla**-based bug-tracking and **Jenkins** CI tool into **JIRA.**
* Experienced with Atlassian tools like **JIRA, Bamboo** and **Confluence**, it’s installation, administration, maintenance and deployed these Atlassian products.
* Experienced on working with system health and performance Monitoring Tools like **Nagios, Splunk, Cloud Watch, Azure Monitor, Google Stack driver, ELK** (Elastic search, Log stash and Kibana) stack.
* Experienced in creative and effective front-end development using **JSP**, **JavaScript**, **HTML**, **JQuery**, **Angular Js**, **Bootstrap**, **React JS**, **AJAX**, **CSS**.
* Used databases like **RDS**, **MySQL,** and **DynamoDB** to perform basic database administration.
* Involved in Day-to-day administration of the environment systems like Development, Production and Test and provided 24x7 system **on-call** **support.**

**TECHNICAL SKILLS:**

|  |  |
| --- | --- |
| Cloud Environment | Amazon Web Services (AWS), Azure, Google Cloud Platform (GCP) |
| Platforms | C, C++, C#, Java/J2EE |
| Scripting | Shell, Python, Ruby, Power Shell, YAML, Groovy, Bash |
| Web Technologies/Frameworks | HTML, JSP, JSTL, JavaScript, CSS, Servlets |
| Version Control Tools | SVN (Subversion), GIT, CVS, GitLab, GitHub, Bitbucket |
| Build Tools | Ant, Maven |
| Configuration Management | Chef, Puppet, Ansible |
| Continuous Integration Tools | Jenkins, UDeploy |
| Ticketing Tools | JIRA, Bugzilla and Confluence |
| Monitoring Tools | Nagios, Splunk, Cloud Watch, ELK Stack 6.8, Kafka |
| Artifactory Repositories | Nexus, JFrog |
| Methodologies | Agile, Scrum, Waterfall |
| Operating Systems | Unix/Linux (Red Hat, CentOS, SUSE), Solaris, Ubuntu, Windows 2008, 2012 Server, XP, Vista |
| Databases | Oracle, MS SQL Server, MySQL, DynamoDB, MongoDB, NoSQL, PostgreSQL |
| Virtualization | Virtual Box, VMWare, Windows Hyper-V |

**WORK EXPERIENCE:**

**Sr. Cloud/ DevOps Engineer April 2020 – Present**

Accenture Federal Services, VA (remote)

**Roles & Responsibilities:**

* Using **AWS** for creating new instances, checking security group settings, adding elastic IP’s for servers, and deleting the elastic IP’s for the servers needed, applying the inbound IP addresses as needed.
* Created Users, Groups, Roles, Policies, and Identity providers in **AWS** using **Identity Access Management** for improvement in login authentication.
* Designed **AWS** **Cloud Formation** templates to create customized **VPC**, **subnets**, **NAT** to ensure successful deployment of Web applications and database templates.
* Created functions and assigned roles in **AWS Lambda** to run python scripts, and **AWS Lambda** using java to perform event driven processing. Created Lambda jobs and configured Roles using **AWS CLI.**
* Used **AWS Beanstalk** for deploying and scaling web applications and services developed with **Java, PHP, Node.js, Python** and **Ruby** on familiar servers such as **Apache**, and **IIS.**
* Created and managed policies for **S3 buckets**, Utilized and moved **S3 buckets** to **Glacier** for the storage in AWS.
* Maintained DNS records using **Route53.**Used Amazon Route53 to manage DNS zones and gave public DNS names to **Elastic load balancers** (ELB) IP's.
* Involved in setting up **Auto scaling** of instance groups using **AWS CLI** (command line interface) and implemented for **AWS** cloud environments such as Production/Development/Testing environments.
* Designed and implemented Azure **Virtual Networks** (VNets) with subnets & network security groups.
* Created and deployed VM’s on Microsoft cloud service **Azure**, created and managed the virtual networks to connect all the servers and designed **Azure Resource Manager** (ARM) templates for Azure platform.
* Converted existing Virtual Machine from Standard to Premium **Storage** Account and performed patching and validating of Virtual Machine (VM) in Azure.
* Involved in the Migration of applications from the **On-Premise** Servers to cloud server **Azure**.
* Configured Ansible control machine & wrote **Ansible playbooks** with Ansible roles. Created inventory in Ansible for automating the continuous deployment and wrote playbooks using **YAML** scripting.
* Installed & Implemented **Ansible configuration management system**. Used **Ansible** to manage Web applications, Environments configuration Files, Users, Services, Resources, Mount points & Packages.
* Provisioned the highly available **EC2** Instances using **Terraform** and **cloud formation** and wrote new **python** scripts to support new functionality in **Terraform**.
* Deployed **Ansible playbooks** in **AWS** environment using **Terraform** as well as created **Ansible** roles using **YAML**. Used **Ansible** to configure **Tomcat** servers and their maintenance.
* Worked with **Ansible** and **Ansible Tower** as **Configuration management** (CM) tool, to automate repetitive tasks, quickly deploy critical applications.
* Day to day job not only included handling **Tickets, Monitoring, Troubleshooting** and **maintenance** but also included architecting and designing Rapid deployment method using **Ansible** to auto deploy servers.
* Used **Ansible playbooks** to setup **Continuous Delivery (CD)** pipeline which primarily consists of **Jenkins** to run packages and it’s supporting software components which are from **Maven** build tool.
* Worked on **Docker container** snapshots, attaching to a running container, removing images, managing directory structures, and managing containers. Created packages in containers and committed those containers using both **Docker file** and **Docker commit** command and updated them into **Docker registry**.
* Worked on installing, configuring and managing **Docker Containers, Docker Images** for Web Servers and applications and implemented **Docker-maven-plugin** and maven pom to build Docker images for all the microservices and later used **Docker file** to build the **Docker images** from the java files.
* Used Jenkins pipelines which helped us to drive all the microservices builds out to the **Docker registry** and then deployed them to the **Kubernetes**.
* Created **Clusters** using **Kubernetes** and worked on creating many pods, replica sets, services, deployments, labels, and health checks using **YAML** files. Development of automation of Kubernetes clusters via playbooks in Ansible.
* Configured **Jenkins**, used it as Continuous Integration tool for installing & configuring Jenkins master & connecting it with multiple build slaves and performed java application builds using **Ant** & **Maven**.
* Responsible for installing & administrating **SonarQube** for code quality check and **Nexus** repository for generating the reports for different projects. Also, integrated SonarQube & Nexus with Jenkins.
* Created and automated the Jenkins pipeline using **groovy script** to check the status of the application.
* Used build tool **Maven** for the building of deployable Artifacts such as war, jar form source code and used **Nexus** for Maven builds, Configured & Administrated Nexus Repository Manager and Artifactory.
* Worked in implementation of **GIT** containing various remote repositories for a single application.
* Coordinated with developers to establish and apply appropriate branching, labelling, or naming conventions using **GIT** source control system, performed multiple pull requests and code reviews.
* Wrote **Python, Ruby**, **Bash** scripts to monitor the installed enterprise level applications and manage the configurations of multiple servers using **Ansible**.
* Migrated **PostgreSQL**, **MongoDB** and **MySQL** databases from **AWS** to **Azure** cloud.
* Designed, built, and managed the **ELK (Elasticsearch, Logstash, Kibana)** stack for centralized logging, search functionalities and monitoring applications in integration with **AWS Lambda** & **Cloud Watch**.
* Configured, Maintained, and customized Ticketing tool & Monitoring tool such as **JIRA** and **Confluence** as per requirements to promote team coordination and ticketing system.
* Installed and configured **Splunk** to monitor applications deployed on application server, by analyzing the application and server log files. Worked on setup of various dashboards, reports & alerts in Splunk.
* Promoted Continuous Integration, Continuous Delivery culture collaborating with Development, DBA, QA, Support and Business Operations.

**Cloud/ DevOps Engineer March 2019 – March 2020**

Pimco Investments, Newport Beach, CA

**Roles & Responsibilities:**

* Launched Amazon **EC2** Cloud Instances using **Amazon Images** (Linux/ Ubuntu) and Configured launched instances with respect to specific applications.
* Created a high availability and scalable AWS stacks using **EC2** auto scaling functionality. Used **AWS CloudWatch** extensively to monitor and debug the cloud-based AWS EC2 services.
* Responsible for Day-to-day Build and deployments in **Dev, Test, pre-production,** and **production** environments. Implemented AWS high availability using **AWS Elastic Load Balancing**, which performed balance across instances in multiple availability zones.
* Used **AWS Route53**, to route the traffic between different availability zones. Deployed and supported AWS Elasticache & then configured **Elastic Load Balancer (ELB)** for routing traffic between zones.
* Used **Elastic Block Store** (EBS) for persistent block level storage and performed access management using **Identity Access Management** (IAM)service.
* Created custom metrics from AWS logs to create alarms and used **CloudWatch**and **CloudTrail**for audits and the maintenance of the cloud environment and **Encrypted EBS** (Elastic Block Store)volumes through **KMS (key management service)**.
* Configured, monitored, and automated **Google Cloud Services** and involved in deploying the content cloud platform using **Google compute engine, storage buckets**.
* Created custom log metrics, monitoring, and created charts**,** alerts using **Stack driver** in **GCP.**
* Written **Terraform** templates, Chef Recipes and pushed them into Chef Server for configuring EC2 Instances, and deployed code into the required environments using AWS Code Deploy.
* Worked on **Jenkins file** with multiple stages such as checkout a branch, building the application, testing, pushing the image into Google container registry (**GCR)**, Deploying to **QA**, Acceptance testing and finally Deploying to **Prod.**
* Experience in providing highly available and fault tolerant applications and utilizing orchestration technologies such as **Kubernetes and Apache Mesos working on Google Cloud Platform (GCP).**
* Built end to end **CI/CD Pipelines in Jenkins to retrieve code, compile applications, perform tests and push** build artifacts to **Nexus Google Cloud Platform (GCP) as a part of POC .**
* Maintained and developed Docker images for a tech stack including **Cassandra, Kafka, Apache, and several in-house** written Java services running in **Google Cloud Platform (GCP) on Kubernetes.**
* Extensively used Google stackdriver for monitoring the logs of both GKE and GCP instances and configured alerts from **Stackdriver for some scenarios.**
* Used **Google Cloud Platform (GCP)** to build, test and deploy applications on Google's very adaptable and solid framework for web, portable, and backend arrangements.
* Worked with automation/**configuration management** (CM) using **Ansible** and created **playbooks** in **YAML** to automate the development processes.
* Worked on designing, installing, and implementing Ansible configuration management system and wrote playbooks for **Ansible** using YAML and deployed applications.
* Created microservices applications with integration to AWS services by using **Amazon EKS**, while providing access to the full suite of **Kubernetes** functionality.
* Worked on using **Chef Attributes, Chef Templates**, **Chef Recipes**, and **Chef Files** for managing the configurations across various nodes using **Ruby** scripting.
* Utilized Configuration Management Tool **Chef** & created **Chef Cookbooks** using **recipes** to automate system operations. Automated server with **Chef &** used **Maven** for deployment and build management.
* Used **Chef** to manage the web applications, configuring files, databases, users, and packages. Developed **Chef Recipes** using **Ruby** framework to configure, deploy and maintain corresponding software components of the existing infrastructure.
* Created additional **Docker Slave Nodes** for Jenkins, implemented Docker-maven-plugin in Maven pom.xml files to build Docker images for all microservices and later used Docker file to build the Docker images from the Java jar files and monitored using **Prometheus**.
* Managed **Kubernetes** with **Kubectl, Kubeadm** and set up **Kube-proxies, Kubelets,** and **flannel overlay networking** for logging within containerized ELK stack**.**
* Worked on implementation of new tools such as **Kubernetes** with **Docker** to assist with **auto-scaling** and **continuous integration (CI)** and Uploaded a **Docker image** to the **registry** so the service is deployable through Kubernetes. Used the **Kubernetes** dashboard to monitor & manage the services.
* Used **Kubernetes** to manage containerized applications using its nodes, Config Maps, selector, Services & deployed application containers as **Pods**.
* Built end-to-end **CI/CD** Pipelines in **Jenkins** to retrieve code, compile applications, perform tests & push build artifacts to **Nexus** & **UDeploy** to orchestrate the changes across servers and components.
* Migrated **Subversion** repositories to **GIT** and administrated **GITLAB** to manage the GIT repositories.
* Created branching & tagging strategy to maintain the source code in the **GIT** repository & coordinated with developers to establish & applied appropriate branching, labeling/naming conventions using GIT.
* Determined ways to optimize/improve **JIRA**, JIRA Agile setup and process workflows, as well as identified where functionality can/cannot meet user requests, created the dashboard for issues.
* Monitored System and Application Logs of the server using **Splunk** to detect the Production issues.
* Used monitoring tool **Nagios** to improve application performance, which helps to get the visibility & business context to meet business demands and implemented **SPARK** communication tool chatbot for the triggering alerts.

**DevOps Engineer June 2015 – Sep 2017**

CoreLogic, Irvine, CA

**Roles & Responsibilities:**

* Developed Rich User Interface using **HTML**, **JSP**, **JSTL**, **JavaScript**, **JQuery**, **CSS** using Liferay portal.
* Performed Analysis, design and development of Application based on **J2EE** using **Struts** and **Hibernate**.
* Worked on a new Portal theme for the website using **Liferay** portal and customized it for look and feel.
* Hands on experience in the usage of **AJAX** (Asynchronous JavaScript and XML)to create interactive web pages with the help of XML, HTML, CSS, and JavaScript.
* Used **ANT** to create deliverables (jar, war) & deployed the application on WebLogic application server.
* Involved in developing **JUNIT** Test Cases to validate type of data in the XML Files. Performed **SQL** tuning using SQL Trace, explained plan for optimizing query performance.
* Developed test cases and performed unit and integration testing when needed. Collaborated with users, database administrators, and team members involved in the project implementation.
* Used **AWS** for creating new **EC2** instances, checking the security group settings, adding the elastic IP’s for the servers, and deleting the elastic IP’s for the servers needed, applying the inbound IP addresses as needed.
* Worked on Setting up the **Jenkins servers** with **Maven** and installed required available **plugins**.
* Worked on **Continuous Integration (CI)** system **Puppet** for official nightly build, test and managing change list. Installed multiple plugins for smooth **Build and Release** of pipelines.
* Installed and configured automated tool **Puppet** that included the installation and configuration of the **Puppet master, agent nodes** and an **admin control workstation**.
* Used **GitHub** as a version control System and configured the **Jenkins Sona type** to generate code coverage and unit test reports.
* Created the **Jenkins jobs** for UI application and Web Services. Configured the Jenkins **CI/CD** Pipeline to run the build, Static Scan and Deploy the package on the servers, run Smoke tests and Dynamic scan, on UI applications.
* Configured **SonarQube** and integrated it with Jenkins to measure and analyze the source code quality.
* Executed and monitored builds, helped team with build issues, measures and reports results to management.
* Monitored and administered the automated build and continuous integration process to ensure correct build execution and facilitate resolution of build failures.
* Configured and maintained Jenkins to implement the **CI** process and integrated the tool with **Ant** and **Maven** to schedule the builds.
* Wrote **POM.xml** files, performed releases with the **Maven release Plugin**, and managed Maven repositories.
* Maintained and administered **GIT** source code tool, Created Branches, Labels and performed Merges in GIT.
* Created branches, performed merges in version control Systems such as **Git**, **GitHub** and **Bitbucket**.
* Implemented a Continuous Delivery pipeline with **Git**, **Jenkins**, **Dockers**, and **AWS AMI**, whenever a new GitHub branch gets started automatically.
* Created **Docker containers** with the Images and deployed the application in Dockers using those containers.
* Wrote **Python Scripts** for data manipulation and analysis to check the access token authentication by the users in the automation process.
* Used **JIRA** for story tracking, logging as part of the Scrum Process. Used **SoapUI** for web service testing.
* Involved in debugging build failures, coordinated with developers and testers to resolve related issues.
* Deployed Application in **Tomcat** Server on Unix environment.
* Provided 24/7 production on-call support as part of on-call rotation.

**DevOps/Java Developer Mar 2014 - May 2015**
Progress Rail - Albertville, AL

**Roles & Responsibilities:**

* Involved in designing the design documents for the project using UML data model representation by Use case diagrams, Sequence diagrams, Class diagrams.
* Worked with **XML**, **Servlets**, **JavaScript**, **JSP**, **JSF**, **EJB**, **AJAX**, **JDBC**, **Angular JS**.
* Involved in developing J2EE applications using **Spring MVC Framework**.
* Developed **UI** using **HTML, CSS** and **JSP** and developed Scripts where users have all the services listed.
* Developed **Servlets** and java files for control of business processes in the middle-tier.
* Worked in an Agile environment and participated in **SCRUM** meetings.
* Implemented code coverage tool to identify the dead code in the application.
* Experienced in using the Object Relational Mapping Tool Hibernate, strong in using XML mappings and Java Objects.
* Used **SPRING** framework that handles application logic and makes calls to business objects and services mainly **SOAP** Web services.
* Worked on **Eclipse IDE** for development, configuration setup and deployment of the application on to **WebLogic** using **Maven,** involved in design and development of **RESTful API**.
* Built and deployed the deliverable files (jar, war, ear) on test, stage and production systems in **JBoss Application Server.**
* Used **SVN** as a version control system for the source code and project documents.
* Used **LOG4J** & **JUnit** for debugging, testing and maintaining the system state. Performed monitoring of log files and resolved issues.
* Worked with team members from many groups to come up with timely, quality decisions and technical solutions.
* Worked on Integrating the modules, bug fixing and collaborated to identify needs of end users for improved user interface.
* Worked on production support and implemented unit testing, system testing and integration testing.

**Jr. Linux Administrator Dec 2012 - Aug 2013**

Param Technologies – Hyderabad, India

**Roles & Responsibilities:**

* Installed, upgraded, and configured **RHEL, CentOS, Ubuntu, and Solaris**.
* Monitored creating and managing user accounts, security, Access Control lists, permissions, disk space and process monitoring in RHEL.
* Installed and configured **Apache** and supported them on Linux production servers.
* Performed software package installation, upgrades and updates using **yum** and **rpm** commands. Applied system patches and updates.
* Hands on experience working with production servers at multiple data centers.
* Automated server building using **Kickstarting RHEL** and **Jumpstarting Sun Solaris**.
* Involved in troubleshooting of production servers and configuring standalone production servers for testing.
* Involved in developing dynamic web-based user interfaces using **HTML** and **JavaScript**.
* Configured yum repository server for installing packages from a centralized server.
* Created a local repository in yum.repos.d to run packages using yum, rpm and up to date.
* Installed and configured **DHCP server** in Linux to give IP leases to production servers.
* Management of Red Hat Linux user accounts, groups, directories, and file permissions.
* Performed Kernel tuning with the sysctl and installed packages with yum and rpm.
* Performed Disk management with the help of **LVM** (Logical Volume Manager).
* Configuration and Administration of **Apache Web Server** and **Secure Socket Layer** (SSL).
* Backup management using **CRONTAB**, running the backup based on client requirements.
* Maintain file system and host security using chmod, chown, chgrp, immutable bit, iptables and tcp wrappers.
* Managing User Administration - Creating users and groups, modifying, and deleting user accounts/groups and managing permissions.
* Password-less setup and agent-forwarding done for **SSH** login using ssh-keygen tool.
* Established and maintained network users, user environment, directories, and security.
* Documented testing procedures and steps involved in data migration on production servers.

**EDUCATION:**

Master’s in Computer Science – Texas A&M University, USA
Bachelor’s in Mechanical Engineering – Mahatma Gandhi Institute of Technology, India.