Khaja **+1-646- 440-3786** mailto:shahul.hameed@ilinesoft.com

Professional Summary:

A committed IT Professional with **10+ years** of experience in **DevOps/Aws Cloud Engineer** operations in designing and implementing cloud and microservice deployments. Expertise in **Automation, Build and Release Management, Software Configuration Management** with best practices of **Software Development Life Cycle (SDLC)** and exposed to **Agile, Scrum,** and **waterfall** methodologies.

- Expertise in building and supporting major cloud platforms using cloud technologies such as **Infrastructure as a Service (IaaS), Platform as a Service (PaaS),** and **Software as a Service (SaaS).**
- Experience in **AWS** offerings which includes **EC2, ELB, Auto-Scaling, S3, IAM, VPC, RDS, DynamoDB, CloudWatch, CloudTrail, Lambda, Elastic-Cache, Snowball, Glacier, Elastic Beanstalk, EKS,** and **ECR**.
- Astute in setting up a database in **AWS** using **RDS**, **DynamoDB**, storage using **S3 bucket** and configuring instance backups and archives to **Glacier**, and configured **AWS Redshift** for Data Warehousing.
- Experience in configuring serverless applications by using **AWS Lambda with API, S3, Kinesis Data Stream and Data Firehouse, RDS services** for data transition under **CloudWatch and Cloud Trail** monitoring.
- Automated application deployment in **AWS** cloud using **Elastic Container Service (ECS)** which provides high performance and scalability service in placing the containers across the cluster accordingly.
- Meticulously worked on **Azure** Services like **VM's, App Service, Azure SQL, Blob Storage, Cosmos DB, Network Services, Subnets, VNets, Load Balancer, Azure Resource Manager, Scheduling**, **AKS, ACR**.
- Experience in Application registration using **Azure Active Directory** to create Service Principal connections between **Azure DevOps** and **Azure Cloud** for linking the web app services to make deployments.
- Skilled in migrating **VMs** from on-premises to Azure using **Azure Migrate** and build **Azure Disaster Recovery** Environment and Azure backups from the scratch using **PowerShell** script.
- Insightful in **GCP** platform Services like **Compute Engine, Cloud Functions, Container Security, App Engine, Knative, Cloud storage, Persistent Disk, Google Kubernetes Engine, Container Registry, etc.**
- Proficient in creating **GCP** firewall rules 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.
- Setup **GCP Cloud CDN** from GCP cache locations to deliver content and drastically improve user experience by reducing latency and integrating with **Cloud Monitoring** and **Cloud Logging** for getting latency metrics.
- Worked on App Containerization technology **Docker**, creating **Docker images, Containers**, **Docker Registry** to store images, cloud-based registry **Docker Hub**, **Docker Swarm** to manage containers.
- Experienced in building, configuring, and managing **Docker Containers**, **Docker Images** for Web Servers, and Applications servers such as **Apache Tomcat** and integrated it with **Amazon MySQL RDS** database.
- Extensively used **Docker and Kubernetes** to Run and Deploy the application securely to fasten the Build and Release process. Worked on creating **pods, replica sets, services, deployments** to manage the cluster.
- Extensively worked with **Scheduling**, **deploying**, **managing** container replicas onto a **node** using **Kubernetes** and experienced in creating **Kubernetes clusters**.
- Developed **Kubernetes Pod** definitions, Deployments and used **Helm Charts** to version control complete deployment strategies and worked on **Minikube** tool to run Kubernetes locally.
- Experience in deployment strategies such as **Rolling update** and **Blue-Green** in **Kubernetes** for providing updates to the applications and creating custom **ConfigMaps** and **secrets** in an encoded format for security.
- Competence in creating **Ansible Playbooks** and encrypted the data using **Ansible Vault** and maintained role-based access control by using **Ansible Tower** to manage web applications, Environment's configuration files.
- Experience in using **Ansible** as Configuration management tool, to automate repetitive tasks, quickly deploys critical applications, and proactively manage change.
- Skillful in deploying and configuring **Chef Server** including bootstrapping of **Chef Client nodes** for provisioning and created **roles, recipes, cookbooks, and database** for server configuration.
- Experience in setting up the **Chef Work station**, **Chef Repo** as well as **Chef Nodes,** created **multi-tiered**, **scalable**, and stable **chef cookbooks** to solve current and future requirements.
- Experience with **Puppet** manifests with **Jenkins** to deploy on to a Jenkins server to provide with continuous deployment and testing to automate deployment of infrastructure.
- Set-up **Jenkins**, **GitLab CI**, **GitHub** actions, **Bamboo** for Continuous Integration and for end-to-end automation for all build and deployments by setting up various pipeline jobs for various test environment.

- Experienced in working with **Jenkins** for creating **new jobs**, managing required plugins, **build trigger**, **build system**, **post build actions**, **scheduling** automatic builds and notifying the **build report**.
- Proficiency in developing the **Pom.xml, Gradle.xml, Build.xml** files for **Maven**, **Gradle** and **ANT** projects by integrating with the **CI** dashboard and managing the **Nexus** and **JFrog** artifactory for build artifacts.
- Hands on Experience in using **SonarQube** and **Selenium** for code quality inspection and Junit for unit testing and integrating with **Jenkins** to run unit tests and to review the code and analyze it.
- Setup code repositories like **SVN, GitHub, Bitbucket** for performing various build and release management activities like **Merging, Branching, Tagging** and managed plug-ins in **Jenkins** for automatic code checkout.
- **Installed**, **Configured**, **Managed** Monitoring Tools such as **Splunk, Nagios**, **Prometheus, Grafana** for collecting the system metrics and **logging tools** like **Kibana, Log DNA** for application-level logs.
- Worked on setting up **Splunk** to capture and analyze data from various layers **Load Balancers**, **Web servers** and **application servers** and built custom dashboards to view the **logs and metrics**.
- In Depth experience in configuring and querying the database **PostgreSQL, Oracle, MySQL, and MS SQL, Cassandra, MongoDB, Cosmos DB, Snowflake Server Database.**
- Extensive experience in deployment of applications on servers like **JBoss, WebLogic,** and **WebSphere**. Also, used **Tomcat, Wildfly, and IIS** for web application deployment.
- Skilled with **Python, Bash/Shell, PowerShell, Ruby, Perl, PowerShell, JSON, YAML, Groovy**. Developed **Shell** and **Python Scripts** to automate day-to-day administrative tasks & automation of the **build and release**.
- Experienced in using **Bug tracking tools** like **Jira, Bugzilla, Remedy and ServiceNow** to track all issues related to the software development lifecycle.

Skills Matrix:

AWS Cloud Services	EC2, VPC, S3, Route53, SNS, IAM, CloudFront, EBS, ELB, ECS, EKS, Lambda, CloudFormation, CloudFront, CloudWatch, Auto Scaling, SQS, Elastic Beanstalk, etc.
Microsoft Azure	VM, Active Directory, ARM, App Service, AKS, ACR, Blob storage, Azure SQL, Azure monitor, Azure Functions, Cosmos DB, etc.
Google Cloud Platform	Compute Engine, Cloud Functions, BigQuery, GCR, GKE, Data Proc, DataFlow, App Engine, Knative, Cloud storage, Cloud Datastore, etc.
Infrastructure as a Code (IaaC)	CloudFormation, ARM, Terraform
Containerization Tools	Docker, Kubernetes, OpenShift
Configuration Management Tools	Chef, Ansible, Puppet
Build Tools	Maven, Ant, Gradle
Continuous Integration Tools	Jenkins, Bamboo, GitLab CI, GitHub Actions, TeamCity
SCM/Version Control Tools	GitHub, SVN, Bitbucket
Artifactory Repositories	Sonatype Nexus, JFrog
Logging and Monitoring Tools	Nagios, Splunk, Prometheus, Grafana, Log DNA, ELK, AppDynamics, Dynatrace, Data Dog, Sysdig
Databases	Oracle, DynamoDB, MongoDB, SQL Server, MS SQL, MySQL, NOSQL, Cosmos DB, PostgreSQL, Cassandra, Snowflake
Scripts/ Languages	HTML, Bash, Shell Scripting, Ruby, Groovy, YAML, Python, Java, Perl, PL/SQL
Bug Tracking Tools	Jira, Bugzilla, HP-ALM, Remedy, ServiceNow, Checkov
Virtualization Tools	Oracle VM virtual Box, VMware, Vagrant
Servers	Apache Tomcat, WebLogic, WebSphere, Jboss, Wildfly, IIS, and Nginx
Operating System	UNIX, Linux (Ubuntu, RHEL, Centos), Windows

Professional Experience:
DTCC, Tampa, FL
Jan 2021 - Present
Sr. Aws Cloud DevOps Engineer
Roles & Responsibilities:

- Configured and managed various **AWS IaaS** offerings including **EC2 instance** creations using **AMIs** in a **VPC, RDS, S3 buckets, Glacier, DynamoDB, CloudWatch, CloudTrail, CloudFront, SQS, SNS,** and **Route 53**.

- Designed roles and policies for new users using **AWS Identity and Access Management (IAM)** and utilized special Security groups, **IAM roles** and edited policies for Dev and Prod users to manage access to resources.
- Worked in the **Billing and Payments** team to set up **live alert systems** using messaging queue tools such as **Kafka,** as well as providing streams **API** for real-time and Connectors API for easy integration within **AWS**.
- Implement **Re-platform migration strategy** to migrate applications from on premise infrastructure to **AWS** using **Server Migration Service,** created **CloudFormation** templates for provisioning of the resources.
- Created **Amazon machine images (AMIs)** using **Packer** and took **EBS** volume snapshots at regular intervals to store the files in **S3 buckets** using **AWS Lambda functions** which provides serverless environment.
- Used **Amazon RDS** to perform basic Database Administration and managed networks using **Security Groups** for instance and worked on **AWS** lifecycle management of data stored and later pushing it to **Glacier**.
- Worked on **Auto Scaling** for **EC2** instances to provide high availability at the lowest feasible cost, creating scaling plans based on application **load**, **metrics**, and **events** using **CloudWatch.**
- Worked on integrating **AWS DynamoDB** using **AWS Lambda** to store the values the items and backup the DynamoDB streams. Used **MySQL, DynamoDB and ElastiCache** to perform basic database administration.
- Created **Terraform** templates to automate the **Azure IaaS virtual machines** using terraform modules and deployed virtual machine scale sets in a production environment.
- Manage **Azure** Infrastructure with **Web roles**, **Worker roles**, **VM**, **Azure SQL**, **Azure Storage**, **Azure AD Licenses**, **VM Backup** and recovery from recovery services vault using **Azur PowerShell** and **Azure portal**.
- Configured **Azure VNets, subnets**, **DHCP address blocks**, **network settings, DNS, security policies**, **routing** and **deployed Azure IaaS VMs** and cloud services (**PaaS role instances**) into secure VNets and subnets.
- Created **Terraform templates** to implement **AKS** cluster in **Azure** using **Kubernetes modules** and uploaded the templates into **Azure blob**, and created **Azure DevOps pipelines** to drive all microservices builds out to images stored in **ACR** and then deployed on **AKS** configuring security groups to allow traffic to pass to **AKS.**
- Extensively used **Docker** for **containerization**, **running**, **shipping**, and **deploying** the application securely to speed up the build and release processes and automated **docker image** builds by creating **Dockerfiles**.
- Implemented build stage to build the **microservices** and push the **docker container image** to the **private docker registry and** worked on **Docker containers** to create **images** of the applications.
- **Used Labels, Selectors**, **ConfigMaps** and **Secrets** to setup communication between **Kubernetes** objects and **ReplicaSets** for the **pods** to maintain high availability and used **Kustomize** to customize the **k8s** objects.
- Managed **Kubernetes** charts using **Helm** and created reproducible builds of the **Kubernetes** applications, templatize Kubernetes manifests, provide a set of configuration parameters to customize the deployments.
- Manage local deployments in **Kubernetes**, create **local cluster** and **orchestrate** the **deployment**, **load balancing** and management of **Docker Containers** with multiple namespace versions.
- Created **CloudFormation templates** to implement **EKS** cluster in **AWS** using **Kubernetes modules** and uploaded the templates into **S3 bucket**, and created **Jenkins pipelines** to drive all microservices builds out to images stored in **ECR** and then deployed on **EKS** configuring security groups to allow traffic to pass to **EKS.**
- Automated various infrastructure activities like **configuring resources**, **application server setup**, **stack monitoring** using **Ansible Playbooks** and integrated **Ansible** with **Jenkins** for Continuous Deployment.
- Implemented **Ansible Tower** for complex network deployments by adding control and triggered **Ansible playbooks** from the **Terraform instance.tf file** as a local-exec call.
- Wrote **Ansible Playbooks** using **Python SSH** as wrapper for managing configurations of **servers, Nodes, Test Playbooks** on **Azure VMs** using Python and written Ansible **roles** to deploy an **Elastic Beanstalk application**.
- Configure **Jenkins** masters and slaves on multiple machines and integrated **Git** with **Jenkins pipelines** to automate builds using **triggers** such as **pull requests**, **tags** and **commits** for building **CI/CD pipelines**.
- Created **Gitlab-CI** environment for continuous CI/CD and implemented CI/CD pipeline involving **Jenkins, Ansible, Docker**, and **Selenium** to complete the automation from commit to Deployment.
- Built Docker Images using Gitlab-ci build automation runner, Set **GitLab** repository and Runner for build automation. Created **gitlab-ci.yml** file for kicking build process in stages that run in **Docker container**.
- Set up **SCM** polling for scheduled builds with **Maven** and used **Selenium** plugins in Jenkins for running unit tests of the code and maintained **JFrog Artifactory** to store binary **Jar, War files.**
- Worked on **Bitbucket** and **Git** for managing, hosting the source code to private **Git/Bitbucket Repositories**, and worked with **merging, Branching, Rebasing** and **Configuration Management** of **SCM**.
- Set up **CloudWatch** to monitor the **cloud infrastructure** and used the generated metrics for analyzing the performance of the applications and configured **SNS** for **alerting** the users on crossing the metric thresholds.

- Deployed and Configured **ELK stack (Elasticsearch, Logstash** and **Kibana)** for log analytics and log management within **EC2/ELB** to monitor **server logs, application logs** and **click streams**.
- Used monitoring tool **Nagios** to improve performance, which helps to get the visibility and business context to meet business demands and implemented **Spark** communication tool **chatbot** for the triggering **alerts**.
- Worked on **shell scripts** for loading data from **Oracle** to **Cassandra**, and configured performance **tuning** and **monitoring** for Cassandra Read and Write processes for fast I/O operations and low latency time.
- Configure and deploy **Java and J2EE** applications into application servers such as **WebLogic** and **Apache Tomcat** and troubleshoot issues encountered during production.
- Composed **Groovy scripts** for setting up **LDAP** configuration for **Jenkins** using security matrix and developed automated scripts and frame works using **Java, Groovy** script.
- Written **Bash, Python, and Shell scripts** to automate application and system management activities. Scripts provided functionality such as **application startup and shutdown, health checks, and website management.**
- Used **Jira** as a **bug tracking** tool by triggering the issues and distributing the issues to developers to fix it. Created **workflows, custom screens** and **custom fields**.
- Performed POC on **GCP** services by Designing, Configuring, Maintaining & Managing Google Cloud Virtual Datacenter Cloud computing platform which hosted Enterprise applications. Used **Compute, Storage, Networking, Stack driver, other Google Cloud tools** for various operations.
- Leveraged Cloud **SDK** and **Google Cloud CLI** to package and deploy containerized web applications on to the Google Cloud Platform and using **Google Kubernetes Engine (GKE)** to manage clusters and implemented autoscaling conditions to accommodate traffic variance.
- Tasked with the coordination of **CI/CD** and **DevOps** teams in automating deployment of infrastructure inside **Google cloud**. Also, configured a **VPC and Database Subnet Groups** for isolation of resources.
- Created fully automated build, test and deployment processes by leveraging **Google Cloud Build** as an automated solution for deploying new versions of containerized web applications and maintaining a repository of **docker** images storage in **Google Container Registry**.
- Performed POC on **Checkov** as static code analysis tool for scanning IaC files for misconfigurations in **Terraform**, **CloudFormation**, **Kubernetes**, **Azure Resource Manager**, and Serverless Framework.

Environments: AWS, Azure, GCP, Kafka, Terraform, Docker, Kubernetes, Ansible, Jenkins, Gitlab, Maven, Git, Selenium, JFrog, Bitbucket, NodeJs, Nagios, Spark, Oracle, Cassandra, WebLogic, Apache Tomcat, Groovy, Java, bash, Python, Shell, Jira, Checkov etc.

Caterpillar Inc., MI
May 2019 – Dec 2020
Sr. Aws Cloud DevOps Engineer
Roles & Responsibilities:

- Configured and managed **Google Cloud Platform (GCP)** services like **compute engine**, **cloud load balancing**, **cloud storage**, **cloud SQL**, **stack driver monitoring** and **cloud deployment manager** etc.
- Leveraged **GCP** cloud services such as **Compute**, **auto-scaling** and **VPC (Virtual Private Cloud)** to build secure, highly scalable and flexible systems that handled expected and unexpected load bursts.
- Used **GCP App Engine** for deploying and scaling **web applications** and services developed with **Java.** Also, Optimized the application for speed by using Google cloud and app engine.
- Ensured, successful architecture and deployment of enterprise grade PaaS solutions using **Private Cloud Foundry (PCF)** as well as proper operation during initial Application migration and net new development.
- Designed and built highly available private cloud and migrating large production systems on **PCF**. Installed **Pivotal Cloud Foundry (PCF)** on instances to manage the containers created by PCF.
- Set-up and build **AWS** infra. using resources **VPC**, **EC2**, **S3**, **RDS**, **Dynamo DB**, **IAM**, **EBS**, **Route53**, **SNS**, **SES**, **SQS**, **CloudWatch**, **CloudTrail**, Security-Group, Autoscaling & **RDS** using **CloudFormation** templates.
- Designed **CloudFormation templates** to create custom sized **VPC**, to set up **IAM** policies for users, **subnets**, **NAT** to ensure successful deployment of **Web applications**, **database templates** and **Security groups**.
- Deployed and monitored scalable infrastructure on **AWS** and configuration instances and managed servers using **Chef** and Created instances in AWS as well as migrated data to AWS from data Center.
- Implemented and maintained the **monitoring and alerting** of production and corporate servers and storage in **AWS using CloudWatch** and assigned **elastic IP addresses** to work around host or availability zone failures by quickly re-mapping the address to another running instance.

- Installed, configured and managed the **RDBMS databases like MySQL, DB2, PostgreSQL, and NoSQL databases like MongoDB, DynamoDB.**
- Worked on installing, configuring and managing **Docker Containers**, **Docker Images** for web-servers and applications and implemented **Docker-maven-plugin** to build **Docker images** for all microservices and later used **Docker file** to build the Docker images from the **java jar files**.
- Created **Docker images** using a **docker file**, worked on **Docker container snapshots**, removing images and managing **Docker volumes** and also virtualized servers in Docker as per QA and Dev-environment requirements and configured automation using Docker containers.
- Worked on **Docker-Compose, Docker-Machine** to create **Docker containers** for testing applications in the QA environment and automated the **deployments**, **scaling** and **management** of containerized applications.
- Used **Kubernetes** to orchestrate the deployment, scaling and management of **Docker Containers**. Also, managed local deployments in Kubernetes, creating **local cluster** and deploying application containers.
- Hands-on experience setting up **Kubernetes (k8s)** Clusters for running microservices. Took several microservices into production with Kubernetes backed Infrastructure.
- Created **Clusters** using **Kubernetes** and created many **pods, replica sets, services, deployments, labels**, and **health checks** using **YAML files**. Managed multiple Kubernetes pod containers scaling, and auto-scaling.
- Implemented **Chef** configuration management to manage **web applications, environment configuration files, users, mount points,** and **packages** and created **cookbooks** in **YAML** to automate the development processes.
- Configured support scripts to **Chef Recipes** and **AWS server** provisioning using **Chef Recipes**. Developed Chef Recipes to configure, deploy and maintain software components of the existing infrastructure.
- Configured and administered **Jenkins pipelines** for automated builds and deployments, and created deployment workflows in which includes pre-release, release and post release steps.
- Integrated **Jenkins** with **GitHub** and various plugins like **SonarQube** for source code analysis and **Maven** for building the applications to **.jar and .war files**, pushed and maintained the built artifacts in **Nexus artifactory**.
- Created **Splunk reports, dashboards, forms, visualizations, alerts**. Configured **Splunk** with forwarder configuration, deployment, search heads, indexing, and data source configurations for **log monitoring**.
- Integrated **Splunk** Enterprise with **Dynatrace** to monitor the performance of the application, managed Splunk forwarder on centralized deployment server.
- Supported large-scale web applications, indexed database queries using **MYSQL server** by writing **SQL** queries and set-up **MongoDB** client and wrote queries to validate data against **Mongo Collections, MySQL**.
- Wrote **AWS Lambda** functions in **python** for **AWS's Lambda** which invokes **python scripts** to perform various transformations and analytics on large data sets.
- Configured, administered and troubleshoot **WebLogic** and **JBoss** servers on Linux and Windows environment and deployed application builds for various **java and .net** applications.
- Performed **synchronous and asynchronous** file operation and event driven programming using **NodeJs**. Installed and configured **JBoss server** developer and test environments.
- Created, maintained and automated the scripts by using **Bash, Python** and **Ruby.** Developed **Cron jobs** and **Shell Scripts (Shell, Python)** for automating installations to extract logs and administration tasks like file system management, process management, backup and restore. for automated**.**
- For IT Service Management **Used ServiceNow as a ticketing tool and to create workflows**. Creating Reports in Service Now for better tracking and process improvements.

Environment: GCP**,** AWS, Terraform, Maven, Nexus, Docker, Kubernetes, Chef, Jenkins CI/CD, GitHub, SonarQube, Splunk, Dynatrace, MYSQL, SQL, MongoDB, WebLogic, Jboss, NodeJs, Bash, Python, Ruby, Shell ServiceNow etc.

Client: Auto Desk, CA
July 2017 – April 2019
Role: Aws DevOps Engineer
Responsibilities:

- Deployed and monitor scalable infrastructure on Amazon web services (AWS) & configuration management using **puppet**.
- Involved in maintaining user accounts **(IAM), SQS, SNS** Services on AWS cloud.
- Create develop and test environments of different applications by provisioning Kubernetes clusters on AWS using **Docker**, OpenShift, Ansible, and Terraform.
- Designed puppet modules to manage configurations and automate install process.

- Wrote **Puppet** manifests for deployment and configurations. Deployed Puppet DB.
- Extensive experience in implementing Continuous integration systems for Java based projects using **Ant, Maven, Jenkins and Hudson**.
- Experience in Azure Cloud on ASM portal and all Infrastructure modules.
- Consume a Restful API developed by the API team.
- Worked with development/testing, deployment, systems/infrastructure and project teams to ensure continuous operation of build and test systems.
- Built application and database servers using AWS **EC2** and used **RDS** for **Oracle DB**.
- Experienced with version control tools like **GIT** and **SVN** and integrated build process with Jenkins.
- Hands on experience in automating builds and deployment process using Shell scripting.
- Utilized CloudFormation and **Puppet** by creating DevOps processes for consistent and reliable deployment methodology.
- Migrated from **SVN** to GitLab Server and internal to **GitLab** servers.
- Setting up schedules for backups and monitoring processes.
- Diagnosed and resolved systems related tasks in accordance with priorities setup in dealing with trouble tickets.
- Responsible for maintaining standard operation procedures for storage environment.
- Updated **YUM** Repository and **RedHat Package Manager (RPM)**.
- On-call support for 24/7 for troubleshooting production issues.
- Configured Elastic Load Balancers with EC2 Auto scaling groups
- Configured servers to host Team Foundation Server (TFS) instance, build controllers and build agents.
- Scripting in multiple languages on UNIX, LINUX and Windows - **Batch, Shell script**.

Environment: AWS (SQS, SNS, IAM), Linux distribution (Ubuntu, Red Hat Enterprise Linux), RPM, Shell scripting, Jenkins, Kubernetes, Maven, Ant, VMware, Puppet, Git, GitLab, SVN and Oracle.

Client: Intuit, San Diego, CA
Mar’13 - Oct’17
Role: DevOps / Build & Release Engineer
Responsibilities:

- Setting up continuous integration and formal builds using **Jenkins** with Artifactory repository. Installed and administered Artifactory repository to deploy the artifacts generated by **ANT** and to store the dependent jars which are used during the build.
- Managed Dependency Management within the teams by installing and configuring **Nexus Repository Manager** and used **JIRA** to track issues and Change Management.
- Configured and maintained Jenkins to implement the **CI** process and integrated the tool with **ANT** to schedule the builds. Developed build and deployment scripts using **ANT** as build tools in Jenkins to move from one environment to other environments.
- Managed site wide various application's build life cycle using **Jenkins** and setting up **Jenkins** master, adding the necessary plugins and adding confidential slaves to support scalability and agility.
- Worked recently on tools migration from old tools like **PVCS** (version control system) to **SVN**, Tracker to **Jira** and finally CI tool **Hudson** to **Jenkins**. Used Hudson/Jenkins to perform automated Builds and Deployments.
- Performed automation of **CI / CD** pipeline in private cloud using Jenkins shared libraries and multibranch pipeline and automated the static code analysis through **SonarQube** in **Jenkins** pipeline for checking code quality.
- Worked on creating **ANT** scripts to build and deploy Java/J2EE, Oracle ADF and Oracle SOA based enterprise applications to **WebLogic** Application Servers.
- Involved in editing existing **ANT** build files as per changes in the project requirements. And maintained build related scripts developed in ANT. Modified build configuration files including Ant's build.xml.
- Developed **GIT** hooks for the local repository, code commit and remote repository, code push functionality and on GitHub.
- Experienced in branching, tagging, and maintaining the version control systems across the environments using SCM tools like GitHub, Subversion (**SVN**), **Atlassian**, **GIT lab** and **TFS** on **Linux** and windows platforms.
- Administered Linux servers for several functions including managing **Apache/Tomcat** server, mail server, **MySQL** databases in both development and production.

- Worked on Implementing and setting up **Route 53** (**AWS DNS** service) for **AWS Web Instances**, **ELB**, **Cloud Front** and **Elastic Beanstalk** on **AWS** Environment and worked on **Route 53** for Failover and Latency Based Routing to the client's websites with weighted routed policies.
- Created Python scripts to totally automate **AWS** services which includes **Web servers, ELB, Cloud Front distribution, database, EC2** and database security groups and application configuration, this script creates stacks, single servers, or joins web servers to stacks.
- Managing **AWS** infrastructure and automation with **CLI** and **API**. working on Inbound and Outbound services with automation **CHEF**. Deployed multiple resources simultaneously, using **Cloud Formation** templates in AWS.
- Good experience in creating Splunk apps, navigations, interfaces and good experience on **Splunk** lookups, macros, Pivot, data models, lookup files and their publication into Splunk. Network Monitoring, bandwidth and traffic monitoring through Splunk and solar wind.
- Used **Splunk Deployment Server** to manage Splunk instances and analysed security-based events, risks & reporting and monitored and track Splunk performance problems, administrations and open tickets with Splunk if there is need.
- Created the **Docker** file built various containers using **Docker Engine** and **Docker Machine** environments and working on several Docker components like **Docker Engine**, **Hub, Machine, Compose and Docker Registry**.

Environment: AWS, Puppet, Bitbucket, JIRA, Artifactory, PVCS, SVN, Hudson, Jenkins, SonarQube, ANT scripts, Java, J2EE, Oracle, GitHub, Atlassian, TFS, Apache, Tomcat server, MySQL.