Ashik Khulal

Phone: **979-977-0037**

Email: akhulal015@gmail.com

AREAS OF EXPERTISE

- Highly motivated DevOps engineer with overall **5+** years of extensive experience in Software Development Life Cycle (SDLC), Software Configuration Management (SCM), Change Management (CM), Build Automation and Release Management (RM). Involving cloud computing platforms like **Amazon Web Services (AWS),** and **Microsoft Azure** as well as some knowledge on **Google Cloud Platform (GCP).**
- Implemented Continuous Integration and **Continuous Delivery** pipeline using tools such as **Git**, **Jenkins**, **Azure DevOps Services**, **Ansible**, **Chef**, **Puppet**, **Docker**, **Kubernetes**, and **Terraform**.
- Worked on installation of **Docker** using Docker toolbox. Worked on creation of custom container images, tagging, and pushing the images and writing **Docker files** to manage **Docker images**.
- Used **Kubernetes** to orchestrate the deployment, scaling, and management of **Docker Containers**.
- Created and managed **Docker files** for creating necessary base images used in automation of build and deployment environments.
- Extensively worked in configuring **Amazon EC2, S3, RDS, Lambda, ELB, EBS, SNS, Route53, Glacier**, and different administrations in the AWS family.
- Experienced in dealing with **Microsoft Azure IaaS** - **Virtual Networks**, **Virtual Machines**, **Azure Functions**, **Cloud Services**, **Resource Groups**, **Blob Storage**, **Express Route**, **VPN**, **F5 Load Balancing**, **Application Gateways**, **Auto-Scaling**, and **Traffic Manager**.
- Experienced in various **SDLC** project phases: Requirement/System gathering, Requirement/System Analysis, Functional Specification, Business Logic's, Design/ Layered Architecture, Test plans, Coding, Code review, Testing, Performance tuning, Documentation, Implementation and Maintenance.
- Experienced in designing, architecting, and implementing scalable cloud-based web applications using **AWS** and **Microsoft Azure**.
- Expertise in deploying **Java**/ **J2EE** and **.Net** applications in **Web Servers** and **Application Servers**.
- Experienced in developing and maintaining build, deployment scripts for test, staging, and production environment using **MSBuild**, **Maven** and **Shell** scripts.
- Installation and configuration of **Apache Tomcat** / **Nginx / WebSphere** on Linux and Windows machines.
- Working knowledge of source code and configuration management solutions such as **Git** and **Bitbucket**.
- Experience with automation configuration management tools like **Chef**, **Puppet**, and **Ansible**.
- Knowledge on building and installing servers through **Azure Resource Manager** Templates (ARM) or **Azure Portal**.
- Worked with **Terraform** key features such as **Infrastructure as code** (IaC), Execution plans, Resource Graphs, Change Automation and extensively used Auto scaling launch configuration templates for launching **Azure VMs** and **Amazon EC2** instances while deploying **Microservices**.
- Worked on **Terraform** for automating VPCs, ELBs, security groups, SQS queues, S3 buckets, and continuing to replace the rest of our infrastructure. Using **terraform** as a tool, managed different infrastructure resources **Cloud**, **VMware**, **Bare Metal Servers**, and **Docker containers**.
- Experienced in **Ansible** automation including writing playbooks, customizing modules for test driven developments, implementing, and designing **AWS EC2**/ **Azure VMs** by **Ansible** roles to ensure deployment of web applications.

REFERENCES AVAILABLE UPON REQUEST.

- Wrote Ansible **YAML** scripts that can store the **credentials** for various sandboxes and secured them on the remote servers**.**
- Automated using scripting languages such as **PowerShell, Bash, Python** and **YAML.**
- Expertise in using build tools like **Maven**, **Apache ANT**, and **MSBuild** for the building of deployable artifacts such as jar and NuGet package to the artifact repository like **JFrog** and **Nexus**.
- Automated deployment using **Ansible** to provision **Amazon AWS** Instances to enable continuous deployments.
- Implemented build per branch as part of CI process in Jenkins to run **SonarQube code coverage** and run unit and integration tests to improve the pipeline efficiency.
- Automated weekly releases with **Maven** scripting for compiling Java codes, debugging and placing builds into repository.
- Created and integrated **Git webhooks** with **Jenkins** to automatically trigger the build automation process.
- Experienced in using **Tomcat, JBOSS, Nginx** and **WebSphere** application servers for deployment.
- Expertise in writing **Jenkins'** pipeline file on Groovy, **YAML** configuration file for **Azure Pipelines** and **Ansible**, **Chef**'s cookbooks and recipes and knowledge on **Puppet**'s modules and manifests.
- Experienced in using monitoring tools such as **Nagios** and **Prometheus** in addition to **ELK** stack for log monitoring and application monitoring.

EDUCATION

Texas A&M University – College Station, TX
B. Sc. in Computer Science

TECHNICAL SUMMARY

Languages	C/C++, .Net, Java, PHP, Python, Shell, Ruby, YAML, and Go.
Source Control/Versioning	Git, GitHub, GitLab, Atlassian Bitbucket, and SVN.
Continuous Integration Tools	Jenkins, Atlassian Bamboo, TeamCity.
Artefact Repository Management Tools	Nexus, and JFrog.
Build Tools	Maven, MSBuild, Apache ANT, and Gradle.
Configuration Management Tools	Ansible, Puppet, and Chef.
Application/ Web Servers	Apache Tomcat, Nginx, JBOSS WebLogic, and WebSphere.
Operating Systems	Linux, RHEL, Ubuntu, Windows, VMware, and Virtual Box.
Databases	DynamoDB, Oracle 11g, MySQL, Postgres SQL, and Cosmos DB.
Monitoring Tools	Nagios, Prometheus, Azure Application Insight, and AWS CloudWatch.
Containerization/ Orchestration	Docker, and Kubernetes.
IaC Tools	Terraform.
Cloud Technologies	Amazon Web Services, Microsoft Azure, and GCP.
Bug Tracking Tools	JIRA, Confluence and Version One.

REFERENCES AVAILABLE UPON REQUEST.

PROFESSIONAL EXPERIENCE

Louisiana Pacific Cooperation, College Station, TX

Aug 2019 – Present

Azure DevOps Engineer

Responsibilities:

- Gained experience in dealing with **Azure IaaS** - **Virtual Networks**, **Virtual Machines**, **Cloud Services**, **Resource Groups**, **Express Route**, **VPN**, **Load Balancing**, **Application Gateways**, **Autoscaler**, and **Traffic Manager**.
- Experienced in configuring VM's availability sets using **Azure Portal** to provide resiliency for IaaS based solution and scale sets using **Azure Resource Manager** to manage network traffic.
- Partnering as an administrator on **Microsoft Azure** and part of DevOps Team for internal projects automation and build configuration management. Involved in configurating virtual machines, storage accounts, and resource groups.
- Building and installing servers through **Terraform** configuration files. Experienced in writing **Terraform Files** to setup automation for **Azure Resources Manager** provisioning using Rest APIs.
- Deployed and optimized multi-tier .Net application to **Azure Pipelines** for Continuous Integration and Continuous Delivery jobs to build, test, and deploy NuGet Packages to the **Azure Artifact Repository**. Utilized **SonarQube** for code analysis and code review.
- Experience in accessing **Azure Resource Manager** for provisioning resources using **Azure Portal**, **Azure PowerShell**, **Azure CLI** and **Rest APIs**.
- Developing build and deployment processes for Pre-production and Production environment.
- Developed **YAML** Scripts for **Azure Pipeline** automation purposes.
- Deployed **Kubernetes** cluster with **Azure Container Service (ACS)** from **Azure CLI** and used **Kubernetes** and **Docker** for the runtime environment of the **CI/CD** system to build, test, and deploy**.**
- Responsible for implementing containerized based applications on **Azure Kubernetes** by using **Azure Kubernetes service** (AKS), **Kubernetes Cluster**, which are responsible for cluster management, Virtual Network to deploy agent nodes, **Ingress API Gateway**, **MySQL** Databases and **Cosmos DB** for stateless storage of external data, and setup reverse proxy **Nginx** in the cluster.
- Evaluated **Kubernetes** for **Docker** container orchestration. 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 deployment and Managed releases of **Helm** packages.
- Participated with developers to identify necessary Azure resources and automate their provisioning.
- Integrated **Docker** container orchestration framework using **Kubernetes** by creating pods, config Maps, deployments.
- Created **Docker Images** using a **Docker file**. Worked on **Docker** container snapshots, removing images and managing Docker Volumes and experienced with **Docker container service**.
- Setup **Docker** to automate container deployment through **Azure Pipeline** and dealt with **Docker Hub** for making **Docker Images** and taking care of various images essentially for middleware establishments and space setups.
- Managed **Azure Container Registry** to store private **Docker Images**, which are deployed and **Azure Pipelines** to build, test, and deployments. **Azure Monitor** to collect metrics and logs. Configured **Monitor** to track performance and maintain security and used **Prometheus** tool to collect application metrics**.**

REFERENCES AVAILABLE UPON REQUEST.

- Created **Nginx** configuration for the application to work as a Multitenant app.
- Working with **Ansible** tower to manage Web Applications, config files, data Base, commands, user mount points, packages and for running **playbooks** stream in real-time and amazed to see the status of every running job without any further reloads.
- Worked with development team and key stake holders to create plan for monitoring **Azure Resources**.
- Utilized configuration management tool **Chef** and created **Chef Cookbooks** using recipes to automate system operations for better compliance.
- Implemented **Chef Recipes** for Ansible deployments on build on internal data center servers to make it more secure.
- Deployed and configured **Elasticsearch**, **Logstash** and **Kibana** (**ELK**) for log analytics, full text search, application monitoring in integration with **Prometheus** and **Azure Functions** for **Application Insights.**
- Used **Azure Pipelines** to drive all micro services builds out to the **Docker Registry** and then deployed to **Kubernetes** pods, and managed them using **Kubernetes**.
- Container management using **Docker** by writing **Docker files** and set up the automated build on **Docker HUB** and installed and configured **Kubernetes**.
- Used **Jenkins and pipelines** to drive all microservices builds out to the **Docker registry** and then deployed to **Kubernetes**, Created Pods and managed using Kubernetes**.**
- Integrated **Git** into **Azure Pipeline** to automate the code check-out process.
- Developed automation and deployment utilities using **Power Shell** and **Python**.
- Implemented relational databases, **NOSQL** Database Management systems like **MySQL**, **MSSQL**, **Cosmos DB**.
- Utilizing **Jira** as ticket tracking and workflow tool.
- Successfully managing concurrent deliverables on time and with quality in fast-paced environment and under deadlines.

Environment: Microsoft Azure (Virtual Machines, Autoscale, Express Route, Traffic Manager, Application Insight, Load Balancer, Virtual Network, Blob Storage, Functions), Git, Azure Pipeline, Azure Artifacts, SonarQube, Chef, Ansible, Nginx, Terraform, Kubernetes, Power Shell, Docker, Docker Hub, Helm, Prometheus, ELK, SQL, NoSQL, Cosmos DB, MSSQL, and Jira.

Fiserv, Alpharetta, GA

Sept 2018 – Aug 2019

DevOps/ AWS Cloud Engineer

Responsibilities:

- Utilized AWS services like **EC2, VPC, Auto scaling, S3, EBS, ELB, CF, LAMBDA, IAM, SNS, SQS, DynamoDB, Elastic Bean Stalk,** and **CloudWatch** services to build highly available, scalable, and fault tolerance applications.
- Performed **Auto Scaling, Elastic Load Balance (ELB)** using **AMIs** and utilized **EBS** to store persistent data and mitigate failure by using snapshots.
- Implemented a **CI/CD pipeline** using **Jenkins, Maven, Ansible, Chef, Docker, Kubernetes** in **Linux (RHEL/ Ubuntu)** environment.
- Assisted in managing **IAM Policies**, providing access to different AWS resources, design and refine the workflows used to grant access.

REFERENCES AVAILABLE UPON REQUEST.

- Created the **AWS VPC** network to spin off **AWS EC2 Instances** and configured the **Security Groups** and **Elastic IPs** accordingly.
- Used **Security Groups**, **Network ACLs**, **Internet Gateways** and **Route Tables** to ensure a secure zone for organization in AWS cloud.
- Created **NAT gateways** and instances to allow communication from the private instances to the internet through bastion hosts.
- Created **AWS Route 53** to route traffic between different regions.
- Created **AWS S3 buckets**, performed folder management in each bucket, managed cloud trail logs and objects within each bucket.
- Utilized AWS serverless function of **Lambda** using the triggers for light-weight functions.
- Used **Terraform** for server provisioning and automating infrastructure.
- Wrote Terraform configuration files for **AWS** infrastructure as a code using **Terraform** to build staging and production environments.
- Built, changed and versioned infrastructure codes on AWS provider with **Terraform**.
- Utilized Shell Scripting (**Bash**) and **Python** as well as third party APIs to automate processes.
- Used **Jenkins file** and **Git Webhooks** to automate the build trigger process whenever any event occurred.
- Created **Master-Slave** configuration using existing **Linux machines** and **EC2 instances** to implement multiple parallel builds through a build farm.
- Used **Maven** and **ANT** as a build tool to validate, compile, test, install and deploy **Java** applications to package them into jar files.
- Associated in implementing **SonarQube** code quality checks. Integrated it in **Jenkins** for code quality analysis and review.
- Installed and configured **Nexus repository** manager for sharing artifacts between internal teams and created proxy Nexus repositories to speed up the build process.
- Responsible for creating and managing **Docker files** for creating necessary base images used for automation of build and deployment environments.
- Virtualized the AWS servers using the **Docker**, created the Docker files and version controlled them.
- Utilized configuration management tools like **Chef** and **Ansible** for more secure, better compliance binding and to manage and update the servers.
- Automated the tasks using **Ansible playbooks**, **Shell Scripting** and **Python**. Provisioned and patched servers extensively using **Ansible**.
- Gained experience with setup, configuration, and maintenance of **ELK** stack (Elasticsearch, Logstash and Kibana).
- Utilized **Chef** to deploy and configure **Elasticsearch**, **Logstash** and **Kibana** (**ELK**) for log analytics, full text search, application monitoring in integration with **Nagios**, and **AWS CloudWatch.**
- Used **Kubernetes** to orchestrate the deployment, scaling, and management of Docker Containers.
- Built and maintained **Docker** container clusters using by **Kubernetes**.
- Utilized **Kubernetes** and **Docker** for the runtime environment of the CI/CD system to build, test and deployment jobs.
- Utilized **Docker, Kubernetes**, **Terraform** and leveraging multiple cloud platforms in AWS, both public and private, to deliver a ubiquitous and consistent global platform for continuously deploying applications.
- Gained experience in integration, deployment, and automation of Apache **Tomcat** application server.
- Worked on **DynamoDB**, **NoSQL** database using boto3, which included creation of tables, items and automating the tasks.
- Integrated **AWS DynamoDB** using **AWS Lambda** to store the values the items and backup the **DynamoDB** streams as well as utilized **Amazon Cognito** to authenticate users' access into the application.

REFERENCES AVAILABLE UPON REQUEST.

- Used **Jira** as ticket tracking and workflow tool.
- Successfully managed concurrent deliverables on time and with quality in fast-paced environment and under deadlines.

Environment: Amazon Web Services (EC2, S3, RDS, Beanstalk, Elastic Load Balancer, CF, SQS, SNS, CloudWatch, Route53, Auto Scaling, DynamoDB, Cognito, Lambda, Internet Gateway, Network ACLs, Elastic Ips, Security Groups, IAM Policies) Chef, Ansible, Git, Docker, Kubernetes, Terraform, Python, Java, Shell Scripting, Jira, Jenkins, Maven, Nexus, Apache Tomcat, Nagios, and SonarQube.

Panasonic Avionics Co, Lake Forest, Il

Mar 2017 – Aug 2018

DevOps Engineer Jr.

Responsibilities:

- Leveraged AWS cloud services such as **EC2**, **Auto-Scaling** and **VPC** to build secure, highly scalable, and flexible systems that handled expected and unexpected load bursts.
- Gained experience on implementing Continuous Integration pipelines using **Git**, **Maven**, **Jenkins**, and **Puppet**.
- Launched **Amazon EC2** Instances using **Linux** AMIs (Amazon Linux/ RHEL) and configuring launched instances with **JBOSS** application server for implementing Java applications. Installed updates and other applications on **AWS EC2** through **AWS Portal**.
- Gained knowledge on **AWS VPC, Security Groups, Elastic IPs** etc. to spin off **AWS EC2** accordingly.
- Utilized **Security Groups**, **Network ACLs**, **Internet Gateways** and **Route Tables** to ensure a secure zone for organization in AWS cloud. Created **NAT gateways** and instances to allow communication from the private instances to the internet through bastion hosts.
- Used **Elastic Load balancing** to setup **Auto Scaling** policies to scale in or out the instances.
- Acquired experience in creating **AWS S3 Buckets**, performed folder management in each bucket, managed cloud trail logs and objects within each bucket.
- Installed, configured, administered **Jenkins** Continuous Integration tool on **Linux** Machines.
- Assisted in configuring the **Jenkins** pipeline for build automation, testing and deployment.
- Contributed to branching, tagging, merging, and release activities on **Git** as a version control tool.
- Partnered in build automation using **Maven** as build tools in Jenkins to move from one environment to other environments.
- Participated in automating weekly releases with **Maven** scripting for compiling **Java** code, unit testing and placing Builds into **JFrog** repository.
- Installed and configured the **JFrog** repository manager for sharing the artifacts within the company.
- Engaged in utilizing **Puppet** to automate configuration management and to manage web applications, config files, databases, commands, users mount points, and packages.
- Used Shell Scripting (Bash) for automation purposes.

Environment: AWS Services (EC2, LB, Auto Scaling, VPC, Elastic IP, Security Groups, S3) Shell Script, Git, Jenkins, Puppet, JFrog, Amazon Linux, RHEL, Maven, and Apache Tomcat.

REFERENCES AVAILABLE UPON REQUEST.

Disney, Orlando, FL

Feb 2016 – Feb 2017

Build/Release Engineer

Responsibilities:

- Worked with Software Development and Quality Engineer to refine and improve the development process.
- Associated with other developers to create knowledge repository, setup, and maintenance of development environments.
- Utilized **SVN** as a source code management tool and for version control tool.
- Observed the migration of **SVN** to **Git**.
- Gained working experience in **Agile/ Scrum development**.
- Analyzed requirements for enhancements of build processes.
- Experienced working with development teams providing deployment services from initial development through production deployment.
- Gained experience in managing and improving the integrated build pipeline in **Jenkins**, including all development, testing, acceptance, and staging environment.
- Partnered in build and deployment of **Java**/ **J2EE** using **Maven** to a **WebSphere** web application server in an **Agile** Continuous Integration environment and automation process.
- Developed **Shell Scripts** for automation of the build and release processes.
- Participated in unit testing using **JUnit** and code quality tests as a part of build process.
- Developed build and deployment processes for Pre-production environments.
- Participated in all daily stand-ups, sprint planning meetings, sprint review meetings, sprint retrospectives and backlog refinement meetings.
- Developed, maintained, and distributed release notes for each scheduled release.
- Got exposure to **Version One** for project management.

Environment: Agile/ Scrum, Java/ J2EE, Shell Scripting, GIT, SVN, Maven, Jenkins, Junit and Version One.

REFERENCES AVAILABLE UPON REQUEST.