

BHANU TEJA INTURI



+1 469-673-1328 | ibhanuteja455@gmail.com | LinkedIn: <https://www.linkedin.com/in/bhanu-teja-inturi-a5478021b/>

Professional Summary

With **9 years of IT experience**, including 6 years in **cloud automation** on **AWS** and **Azure** and 3 years in **system administration**, Proven expertise in designing and implementing end-to-end **DevOps pipelines**, leveraging tools such as **Jenkins, GitHub, GitLab CI/CD, Ansible, Docker, Kubernetes**, and **Terraform**. I've developed **CI/CD pipelines** for creating **Docker images** and **deploying** them to **Kubernetes**. My expertise includes managing **Kubernetes clusters** with **Helm charts, RBAC**, and **resource management strategies**. Experienced in **automating cloud infrastructure** on **AWS (EC2, S3, RDS, CloudWatch)** and **Azure (VMs, Blob Storage, SQL Database, Azure Monitor)**. Strong scripting skills with **Python, PowerShell, and Bash**, enabling automation of operational tasks, **configuration management**, and **system provisioning**. Developed **Ansible playbooks** and reusable automation scripts for **infrastructure setup, security hardening**, and **cloud-native deployments**. Configured **Kafka, Prometheus, Grafana**, and **Splunk** for **application observability, real-time monitoring**, and **log analysis**. Experienced in implementing **infrastructure as code (IaC)** practices and delivering secure, scalable solutions in **Agile and DevSecOps environments**.

Education

- Bachelor of Technology in Electronic and Communication Engineering from **JNTUK UNIVERSITY** – 2017
- Master's in computer and information systems from **Christian Brothers University** – 2023

Certifications

- AWS Certified Solutions Architect Associate.**
- Terraform Associate.**
- CKA CERTIFIED KUBERNETES ADMINISTRATOR.**

Skills

SDLC Methodologies	Waterfall, Agile/Scrum
Operating Systems	Windows 11/10/2003/2008/2012, RHEL/CentOS, Ubuntu, Solaris, AIX, HP-UX
Cloud/IaaS/SaaS/PaaS	Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, VMware.
SCM, Build, CI/CD Tools	Git, SVN, Ant, Maven, Gradle, Jenkins, Gitlab, GitHub actions, Bamboo, TeamCity, Harness
Monitoring & Alert Tools	Datadog, PagerDuty, Prometheus, Grafana, Splunk, Nagios, and ELK.
Containerization/ Orchestration	Docker, Kubernetes, EKS, AKS, OpenShift, and Vagrant
Database Servers	Teradata, Oracle, MS SQL Server, and MySQL.
Ticketing Tools	Atlassian JIRA, Redmine, Confluence, and ServiceNow.
Scripting Languages	PowerShell, Shell, Groovy, Python, and SQL
Web Servers	Apache Tomcat, Apache HTTP Server, Nginx, and IIS
Infrastructure as Code	Terraform, Ansible, Vagrant, and Docker

Work History

Client: Meijer, Grand Rapids, MI

FEB2023– Ongoing

Role: DevOps Engineer/ Cloud Engineer

Responsibilities:

- Designed and configured AWS services: EC2, S3, ELB, ECS, RDS, VPC, Route53, CloudWatch, Lambda, and IAM.
- Set up secure VPC environments with public/private subnets, NACLs, route tables, and security groups for traffic control.
- Automated infrastructure deployments with Terraform for AWS resources like EC2, VPC, ELB, and IAM.
- Deployed Kubernetes clusters using Terraform scripts and AWS EKS for containerized applications.
- Configured Docker containers, built serverless computing engines with AWS Fargate, and managed repositories in ECR.
- Used CloudWatch for event monitoring and SNS for alert notifications. Created Lambda jobs to integrate CloudWatch logs into ELK.
- Managed application rollouts using Kubernetes rolling updates and adjusted pod definitions for deployment.
- Created automation pipelines in AWS CodePipeline for Dockerized applications using CloudFormation, CodeBuild, and CodeDeploy.
- Worked with product and development teams to understand business requirements for the Java application and defined technical specifications.
- Developed a Java application using Spring Boot to create RESTful web services with CRUD operations connected to a database (e.g., MySQL or PostgreSQL).
- Dockerized the Java application by creating a Docker file and pushing the container image to Amazon ECR.
- Set up an EKS cluster using Terraform to manage Kubernetes workloads and deploy Helm charts for standard services like Ingress, Prometheus, and Grafana.
- Set up Jenkins pipelines for continuous integration (build and test) and continuous delivery (deploy to Kubernetes), integrating with GitHub for source code management. Configured automated testing for the Java application during the CI phase in Jenkins.
- Implemented GitOps with Argo CD to synchronize Kubernetes manifests from GitHub to the EKS cluster during the deployment phase.
- Built a pipeline that pulls code from GitHub, builds the Java application using Maven/Gradle, and runs unit and integration tests.
- Used Helm charts to deploy the Dockerized Java application to EKS, integrating with Argo CD for automatic synchronization.
- Configured Kubernetes Horizontal Pod Autoscaler (HPA) to scale the application based on CPU or memory usage and set up AWS Auto Scaling for EKS worker nodes.
- Integrated Prometheus to monitor application metrics (e.g., response time, error rate) and set up Grafana dashboards to visualize performance.
- Set up Fluentd or Logstash to collect logs and store them in Elasticsearch, with Kibana for log visualization and troubleshooting.
- Automated infrastructure provisioning with Terraform scripts for all AWS resources (VPC, subnets, EKS, ECR). Wrote Ansible or shell scripts to manage AWS credentials and Kubernetes configurations.
- Designed, deployed, and maintained highly available data store solutions using Cassandra, Oracle, and MongoDB, ensuring performance, scalability, and fault tolerance across distributed systems.
- Automated database provisioning, configuration, and scaling using Terraform and integrated into CI/CD pipelines with Jenkins, enabling faster and more consistent deployments.
- Implemented backup and disaster recovery strategies for mission-critical databases, minimizing RTO and RPO in enterprise-grade systems.
- Conducted capacity planning, performance tuning, and optimization for large-scale NoSQL and RDBMS environments, improving query latency and resource utilization.
- Implemented on-call rotation within the DevOps team for 24/7 application and CI/CD pipeline support.
- Tested the CI/CD pipeline, ensuring successful builds and deployments, and troubleshooting any issues. Tested the Kubernetes deployment with load-testing tools like Apache JMeter for performance validation.

Client: KPMG, Grand Rapids, MI

NOV2021 - JAN2023

Role: Cloud Platform Engineer

Responsibilities:

- Migrated enterprise applications to AWS with zero downtime using ELB, Auto Scaling, and Multi-AZ architecture for high availability.
- Designed cost-efficient serverless solutions with AWS Lambda and automated resource provisioning with Terraform.
- Configured CI/CD pipelines using Jenkins, GitHub Actions, and Kubernetes for streamlined deployments.
- Managed AWS EKS clusters for scalable workloads and optimized performance for up to 10,000 users.
- Secured AWS environments with IAM roles, VPC configurations, and AWS Shield for enhanced security.
- Developed microservices with Spring Boot, AWS SQS, and Kafka for real-time data processing.
- Implemented Infrastructure as Code (IaC) with reusable Terraform modules for collaborative deployments.
- Built real-time data pipelines with AWS Glue and Kinesis for scalable analytics. Reduced cloud costs by 25% using AWS Cost Explorer and Budgets to optimize resource usage.
- Integrated DevSecOps tools like SonarQube, AWS Inspector, and Aqua Security into CI/CD pipelines. Automated operational tasks with AWS Lambda, Systems Manager, and Ansible, reducing manual intervention.
- Configured monitoring solutions with Datadog, Splunk, and CloudWatch, creating custom dashboards and alerts.

- Mentored junior engineers on **cloud architecture, microservices, and DevOps** best practices.
- Streamlined **Node.js microservices** for resilience and **fault tolerance** in **high-demand environments**. Enhanced testing workflows for **Node.js applications**, enabling rapid and **error-free releases**.
- **Automated monitoring** configurations using **New Relic APIs** and **Terraform** for consistent deployments.

Client: Verizon, Gundi Chennai, India.

MAY2019-AUG-2021

Role: DevOps Engineer

Responsibilities:

- Experienced in **Azure Development**, working on **web applications, App Services, Azure Storage, Azure SQL Database, Virtual Machines, Azure AD, and Notification Hub**.
- Deployed **Infrastructure as Code** using **ARM Templates** via **Azure Portal** and **PowerShell** to automate **Azure VM** provisioning and managed **Azure Blob Storage, file blobs, and disks**. Integrated and **troubleshooted on-premises AD sync** with **Azure AD**.
- Utilized **Kubernetes** for **containerized application deployment, scaling, and developed microservices in Golang**.
- Built **CI/CD pipelines** using **Cloud Bees, Docker, and Kubernetes**, leveraging **Helm charts** for **deployments**.
- Implemented **secure Docker deployments** by integrating **image vulnerability scanning** into **CI/CD pipelines**.
- Designed scalable containerized solutions with **Kubernetes, Docker, and Helm**, improving deployment efficiency.
- Automated infrastructure provisioning using **Terraform** and modular practices.
- Managed **CI/CD pipelines** with tools like **Jenkins** and **Azure DevOps** and integrated **SonarQube** for code quality.
- Developed custom **Kubernetes operators and CRDs** for seamless system integration. **Migrated legacy applications to containerized architectures, reducing downtime and operational costs**.
- Deployed and maintained observability tools such as **Prometheus, Grafana, and ELK** for real-time monitoring.
- Automated workflows using **Python, Bash, and PowerShell** to streamline operations.
- Configured artifact repositories like **Artifactory** and **Nexus** for efficient dependency management. Ensured cloud-native security with **RBAC, pod security policies, and image scanning**. Designed **disaster recovery and backup strategies** to ensure **system reliability**.
- **Collaborated on compliance frameworks** to meet industry standards and regulations. Delivered training on **CI/CD, Kubernetes, and DevOps** best practices to **cross-functional teams**.
- Configured **Istio** for mutual **TLS, traffic policies, and fine-grained access control** between **microservices**.
- Enhanced **AKS** observability with **Prometheus, Grafana, and Fluent**, integrated with **Istio telemetry**.
- Migrated legacy workloads to **AKS**, improving **scalability and reducing costs**. Optimized **AKS clusters** with **auto-scaling** and workload resource quotas. Mentored teams on **AKS, Terraform, Istio, and Azure security** best practices.
- Administered **JIRA** and **Bamboo**, using **Ansible** for **deployment automation and infrastructure management**.
- Automated **Angular** and **React.js** builds and tests, **integrating Node.js and Docker** for **production deployments**.
- Managed system monitoring with **Datadog, Tomcat servers, and MySQL databases**, and provided end-user IT support.
- Created custom **Dynatrace extensions** to **monitor third-party applications** for expanded **observability**.

Client: DSquared, Hyderabad, India:

AUG 2016 -MAY 2019

Role: Linux Admin

Responsibilities:

- **Configured Red Hat Kickstart** for **automating server installations** and managed **Debian, Red Hat, CentOS, and SUSE servers across multiple data centers**.
- Installed and managed **Puppet** for **system automation**, writing **custom modules** and **upgrading Puppet Community and Enterprise versions**.
- Migrated **consumer data** between **production servers** using **Bash** and **Perl scripts**.
- **Installed and configured Nagios** for **network and disk monitoring**, and **automated Java application deployment** on **Linux servers** using **Python** and **Puppet**.
- Managed **version control** with **GitHub** and **CI/CD pipelines** with **Jenkins**, troubleshooting **build issues** and **server crashes**.
- Deployed and monitored **scalable infrastructure** on **AWS**, managing **configurations** using **Puppet**.
- **Configured password-less authentication** with **FUSE** and managed **DHCP, DNS, LDAP, NFS, and mail services**.
- **Monitored performance** using **Sar, iostat, and Munin**, and managed **LVM-based disk configurations**.
- Installed and configured **PostgreSQL databases, Apache Web Server** with **SSL**, and **automated tasks** using **shell** and **Python scripts**.
- Maintained **FTP servers**, monitored **infrastructure** using **Nagios, Cacti, Zabbix, and OSSEC**, and supported **server builds, patching, and performance tuning**.
- Administered **VERITAS Volume Manager, Clustering, and VMware environments**, managing over **1,800 VMs** across **three data centers**.
- Configured and supported **VMware vSphere** networking, storage, and **vCenter Operations Manager**.
- Developed **Python scripts** for **building automation** and **Django web applications**, and integrated **Amazon API** for **instance operations**.
- Deployed and maintained **Red Hat Enterprise Linux 6.x/7.x** and **CentOS servers**, including **package installations** and **patch management**.

Key Achievements

- Reduced **deployment time by 40%** through **CI/CD automation**.
- Optimized **AWS costs by 25%** via **resource monitoring** and **auto-scaling**.
- Enhanced **Kubernetes security** by implementing **RBAC** and **pod security policies**.
- Migrated **monolithic applications** to **microservices**, improving **scalability** and **resilience**.
- Established **GitOps practices** with **Argo CD**, enabling automated **Kubernetes deployments**.