**NDIKUM TERENCE**

**Baltimore, Maryland, USA**

**Email:** [ndikumfru@gmail.com](mailto:ndikumfru@gmail.com)

**LinkedIn:** http://linkedIn.com/in/archindikumfru

**Cell: +1(443)9854204**

**SUMMARY**

A skilled Cloud/DevOps/Infrastructure Engineer leading teams and managing projects with 9+ years of solid hands-on and extensive experience in managing IaaS, PaaS, SaaS, IaC, and on-prem resources, automating the build, deployment, and operations of mission-critical applications and microservices at scale across different infrastructures and scalable cloud-based hosting (deployment of highly available and fault-tolerant web-scale and enterprise software applications) on the multi-cloud platforms. Architected efficient systems using best design principles, patterns, and practices. Highly proficient in automating infrastructure provisioning and deployment, supporting, and optimizing critical applications on Linux/Windows OS while leveraging AWS and Open-Source technologies, configuring and managing CI/CD pipeline toolchains. Having a deep knowledge of computing optimization, networking, storage, security, databases, monitoring and alerting, automation, log management and data solutions, service integration, disaster recovery, and business and dashboard reporting.

**WORK RELATED EXPERIENCE**

**Senior DevOps/Cloud Engineer – Experian, Costa Mesa, CA 92626**

**Jan 2021 – Present**

* Worked in an agile environment as part of a cross-functional team managed projects, led teams in the CI/CD with a mastery of the Software Development Life Cycle (SDLC) and made significant contributions to architecture.
* Leveraged Amazon ECS to deploy, scale, and manage containers, and ECS clusters with Fargate and EC2 instances, task definitions, and services.
* Built CI/CD automation pipeline using CodePipeline with CodeCommit, CodeBuild CodeDeploy, and ECS integrated.
* Wrote buildspec.yml and appspec.yml files to automate and customize application build and deployment in CodePipeline.
* Managed Kubernetes projects using imperative commands on live objects, imperative object configuration of individual files, and declarative object configurations of directories of files across development, and production environments.
* Built an architecture using CodeCommit, CodeBuild, Lambda, and CloudWatch to automate the build and testing of code changes in pull-requests before merging for high-quality code.
* Configured CloudWatch Event and deployment trigger to monitor events and send SNS notifications during build and deployment in AWS CodePipeline.
* Deployed sidecar container pattern using ECS to send metrics and logs to CloudWatch and S3 bucket.
* Deploying and managing stateless, multi-tier enterprise and web applications in Kubernetes leveraging AWS services; EKS, EC2 instances, Elastic load Balancers, databases, etc.
* Built CI/CD automation pipeline and configured Jenkins integration with Git/GitHub, Maven, Ansible, Docker, Nexus, SonarQube, and Kubernetes.
* Defined quality gate in SonarQube for Static Code Analysis (SAST) for a clean, stable, and safe code.
* Implemented monitoring dashboard using New Relic with over 50 Restful APIs for real-time monitoring, troubleshooting, and maintaining maximum application uptime.
* Deployed ELK (Elasticsearch, Logstash, and Kibana) to monitor, and collect applications, network, and system metrics and logs, identify trends, spikes, and anomalies, set alarms, and take corrective actions.
* Deployed a high availability Kubernetes cluster with an external etcd database cluster, multiple control plane nodes, worker nodes, and HAProxy.
* Extensive use of Terraform and Ansible scripts to provision and configure EKS, VPC, Route53, EC2 instances, etc. on AWS and GCP Cloud.
* Automated the deployment, scaling, and optimization of applications in Kubernetes clusters using Kubernetes objects, HP Auto-scaler, Deployment, Cluster Autoscaler, sanity health checks, etc.
* Coordinated and collaborated with project teams providing an enabling environment through access management, service integration, and best practices for Git/GitHub, Docker, Terraform, and Kubernetes such as appropriate branching strategy, labeling convention and dockerfile instructions, etc.
* Wrote and modified Docker files, created and pushed docker images to repositories in Docker Hub etc.
* Installed, configured and initialized docker swarm architecture creating master and worker nodes.
* Used terraform to configure resources in AWS including EC2 instances, VPCs, ELB, S3 buckets, EFS, auto-scaling groups, databases, etc.
* Introduced and installed Dynatrace to visualize application monitoring through charts, graphs and alerts.
* Provisioned managed and self-managed Kubernetes clusters using EKS, KOps, and Kubeadm.
* Deployed ELB/NodePort/Ngnix Ingress service to manage and direct internal and external HTTP traffic in Kubernetes clusters.
* Collaborated regularly with project managers, development teams, and customers to understand the project requirement, release plans, and provided support.
* Wrote groovy scripts, yaml files, docker files, and Ansible playbooks for software deployments.
* Installed and configured Nexus repository as a backup for build artifacts in case of disaster.
* Wrote, modified, and maintained Ansible playbooks for the user, file, and package management using Ansible modules.
* Configured CI/CD pipeline auto triggers including Poll SCM, build periodically, GitHub web hooks, etc.
* Used Git/GitHub as a version control service to store and managed project files.
* Configured S3 remote backend for Terraform and locked state file with DynamoDB table for security.
* Assessed DevOps Metrics and KPIs and ensured processes, pipeline, and tooling are meeting goals.
* Used Jenkins Declarative pipelines script to deploy MySQL to Kubernetes Clusters in GCP.
* Actively managed, improved, and monitored cloud infrastructure on GCP, and AWS Cloud, including backups, patches, and scaling.
* Built and deployed Docker containers to break up monolithic applications into microservices, to benefit from improved performance, vertical, and horizontal scalability leveraging AWS ECS and EKS
* Configured, launched, and resized Elastic Computers in AWS and Google cloud based on workloads.
* Used Jenkins, plugins, and Maven to build Java application packages such as Jar, war and ear packages ready for deployment.
* Troubleshoot, identify, and fix network connectivity, firewall, running processes, system performance, and application issues.

**DevOps/Cloud Engineer – Kaiser Permanente, Oakland, CA 94612**

**Jan 2019 – Jan 2021**

* Built, managed and maintained complex automated CI/CD software delivery pipeline using interconnected tools that enhance speed, efficiency, and quality software.
* leveraged ansible roles to deploy Apache and Tomcat to a target group of servers.
* Installed and configured Ansible server from scratch, set up inventory file, and passwordless login.
* Worked on application delivery toolchain, installed and managed Jenkins plugins, maintained pipelines, and deployed applications to dev, stage, and production environments.
* Automated infrastructure, deployments and system-wide configuration management using orchestration scripts (Groovy scripts, Ansible Playbooks, Terraform HCL, Kubernetes Manifest files, Docker files, Helm Charts, Bash Shell Scripts, and python scripts).
* Used KOps software to deploy full production-grade, highly available, and scalable Kubernetes clusters.
* Configured and managed Persistent Docker Volumes, and EBS volumes using the Rexray plugin.
* Established metrics and Key Performance Indicators (KPI) to evaluate and improve processes, systems, and application performance and high availability.
* Installed packages including Maven, Jenkins, SonarQube, java, and dependencies on Linux Servers using Bash shell scripts.
* Leveraged Prometheus, New Relic, and, Nagios for real-time application monitoring, collect metrics and logs, establish Key performance Indicators, set threshold values for alarms, and dashboard data visualization.
* Used Terraform best practices (Workspaces for different environments, modules for building blocks,

and Terraform import for existing resources).

* Managed applications deployment with zero downtime by using suitable strategies such as Blue/Green, Canary, and rolling updates.
* Installed web/application servers (Nginx, Apache, JBoss/WildFly, and Tomcat) on Linux OS using bash shell scripts.
* Carried out post-incident analysis, identify root causes- and implemented possible remediation measures.
* Used cron to schedule and run routine tasks such as database backup.
* Used Helm charts to deploy Prometheus, Grafana, Elastic Stack, and Kibana in Kubernetes clusters.
* Created Jenkins shared library, and configured and use the library in pipeline scripts.
* Used Docker to build images and containerized applications, and Kubernetes to orchestrate and scale containers.
* Configured Bastion host to securely connect to EC2 instances running in our private subnet on AWS.
* Created and mounted logical EBS volumes on EC2 instances, and EFS to sync data across multiple EC2 instances.
* Set up Apache and Nginx webservers with AWS ELB and installed SSL Certificates using ACM to encrypt data in transit and redirect all HTTP traffic to HTTPS for security.
* Strong experience in enhancing security configuration in Linux/ Windows Servers creating and managing files, users, groups, and permissions.
* Identified and troubleshoot network-related issues regarding TCP/IP, DHCP, and DNS.
* Created lunch templates based on customized AMI and use them to configure auto-scaling groups.
* Analysed cloud infrastructures and recommended improvements for performance gains and cost efficiency.
* Managed system processes by running various shell commands (ps, start, stop, kill, top, df, netstat, etc.).
* Configured file systems using LVM and also added Swap space where necessary.
* Created and managed file systems, mount points, and RAID volumes in Unix/Linux.

**Cloud Engineer/Solutions Architect – Minsante, Yaoundé, Cameroon**

**Jan 2015 – Jan 2019**

* Leveraged Amazon CloudWatch as a monitoring and observability tool to collect granularitymetric**,** events, and logs across EC2 instances, containers, and serverless applications.
* Participated in architecting, and implementing DevOps cloud-native platform including Docker, and Ansible.
* Architected, and participated in building application delivery pipeline, and stateless microservice application deployments.
* Supported, and implemented tools for integration, automated testing, and build and release management.
* Leveraged AWS IAM to create and manage users, groups and roles granting least privilege permissions to access resources.
* Architected scalable web applications with static websites and dynamic REST API to deliver a consistent customer experience.
* Implemented a solution to stream data using Lambda from CloudWatch Logs to Amazon Elasticsearch through a subscription filter for analytics.
* Implemented multi-tier architecture with public-facing ELB, application, and database tiers.
* Leveraged Auto-Scaling and multi-availability ones for high availability and self-healing components.
* Configured Virtual Private Cloud with public and private subnets to host public and private resources respectively.
* Used AWS Organization to consolidate and simplify management of multiple AWS accounts using OUs with consolidated billing, and centralized policy control using SCPs.
* Optimized cost through a mix of Reserved, Spot, and On-Demand EC2 instances, S3 storage classes, and lifecycle policies, reducing cost by 22%.
* Launched different EC2 instance types and EBS volumes suitable for different use cases based on target workloads.
* Experienced strategizing disaster recovery plans with best RPO and RTO as needed using technics such as data replication, fail-over routing, snapshotting, baked AMI etc.
* Created S3 buckets and storage classes to manage and store objects including files, logs, images, and multi-media content based on access patterns to minimize storage cost.
* Used Amazon SQS as a middleware to decouple, scale and integrate services, avoiding application disruptions due to high traffic volume.
* Reviewed system utilization metrics and Key Performance Indicators (KPI) to utilize computing resources efficiently to meet system application requirements.
* Configured CloudTrail to log all account API calls and deliver logs from multiple accounts to a central S3 bucket location for auditing and management.
* Leveraged Amazon SNS in an event-driven pub/sub model to enable communications between distributed systems and microservices.
* Set up AWS Config rules to track and evaluate resource configuration compliance status such as restricted SSH traffic, encrypted http traffic, and access key rotation, and remediate non-compliance resources.
* Configured SSM Patch Manager and implemented custom patch baseline to a target group of managed EC2 instances.
* Implemented a solution to stream logs and events in real-time from CloudWatch to S3 bucket using Amazon Kinesis Firehose.
* Implemented Runbooks and Playbooks for quick response and incident remediation reducing MTTD and MTTR by 23%.
* Configured cross-account access using Security Token Service (STS) to enable access between accounts and facilitate resource sharing.
* Leveraged Amazon Route 53 routing policies to enable low latency networking, configured active-active and active-passive failover for a resilient architecture.
* Set up Ansible server from scratch and wrote Ansible playbooks to manage users, files, packages, security configuration, and patch management.
* Configured CloudFront distribution to speed up delivery of static content globally with low latency from an S3 bucket configured as a website.
* Leveraged AWS KMS to encrypt data at rest in S3 bucket and EBS volumes for data integrity and confidentiality.
* Used R53 to create record types and routing policies to route internet traffic to backend resources.
* Implemented event-driven automation with EventBridge to respond to systems-wide resource state changes.
* Configured Amazon EBS snapshot lifecycle policy to regularly backup critical application data to S3 bucket.
* Maintained and patched systems across multiple accounts and regions using AWS Systems Manager Automation.
* Analysed, and made recommendations in the areas of cost optimization, performance, security fault tolerance and service limits.
* Configured ELB and Auto Scaling groups with scaling policies to dynamically adjust the capacity of compute resources.
* Configured NAT gateway to enable private resources in a VPC to be updated via the public internet.

**ACHIEVEMENTS & IMPACT**

* Member of a team that Successfully managed 200 running applications with 120 deployments a month.
* Built processes and systems that witnessed significant improvement in Key Performance Indicators (KPI) with a 26% increase in weekly deployments, a failure rate dropped by 30%, and Mean Time To Recovery (MTTR) dropped by 26%.
* Led cost optimization plans by carefully selecting purchase options for virtual machines, storage solutions and classes, and life cycle policies which reduced cost by 22%.
* Successfully earned 100% issue-resolution scores by delivering excellent service to internal/external customers, while serving as the main escalation point for troubleshooting advanced network/systems issues.

**EDUCATION & CERTIFICATION**

* **Bachelor of Science Degree University of Buea, Cameroon**
* **AWS Certified Solutions Architect Pearson VUE**
* **DevOps Master Program Edureka (E-Learning)**
* **AWS Solutions Architect Udemy (E-Learning)**

**TECHNICAL SKILLS:**

Architecture, Technical Documentation, Customer Service, Leadership, System Administration, GCP, Microsoft Azure, Technical Support, SDLC, Disaster Recovery, **Compute** (EC2, Lambda), **Storage** (S3, EBS, EFS), **Networking** (VPC, Elastic Load Balancers, CloudFront, Route 53, IGW, NAT Gateway, API Gateway), **Version Control** (Git), **Repositories** (GitHub, Bitbucket, GitLab, Nexus, JFrog), **Atlassian Tools (Jira, Confluence), Monitoring & Logging** (Prometheus, Nagios, Elasticsearch, Logstash, Kibana, Splunk, Dynatrace, New Relic, CloudWatch, Athena), **Data Streaming** (Amazon Kinesis), **Kubernetes (**Amazon EKS, Rancher, KOps, Kubeadm)**,** **IaC** ­­­(Terraform, CloudFormation), **Programming** **Languages** (Yaml, HTML, java, JavaScript, Python), **Shell** **Scripting** (Bash, Powershell), **Data** **Migration** (MGN), **Configuration Mgt (**Ansible, AWS SSM, Chef), **Servers (**Apache, Nginx, JBoss/WildFly, Tomcat), **Database** (DynamoDB, MongoDB, MySQL, PostgresSQL, Oracle, Microsoft SQL Server)**, CI/CD Platform (**Jenkins, Maven, CodeCommit, CodeBuild, CodeDeploy, CodePipeline, Fargate, Elastic Beanstalk ), **Containerization** (Amazon ECS, Docker, Docker Hub, Amazon ECR), **Code Quality** (SonarQube), **Agile Methodology (**Scrum**), Operating System (**Linux, Windows, Unix, Solaris), **K8s Package Manager** (Helm ), **Internet Protocol Suite (**TCP/IP), **Cache** (Redis and Memcached)**, Security** (Security Group, Network ACL, IAM, MFA, KMS, CloudTrail, Parameter Store, Secret Manager, SCP, Amazon Inspector, GuardDuty, WAF, Shield), **Capacity Optimization** (Auto Scaling)**, Service Integration** ( Amazon SNS, SQS, EventBridge, etc.), Compliance (AWS Config, Trusted Advisor), Data Streaming (Kinesis), Governance (AWS Organization).