RESUME

Name : Bainaboyina Adinarayana

Mobile : 9133054143

Email : adinarayanainfo12@gmail.com

Professional Summary:

* Having around 5 years of experience in DevOps, GCP, AWS,and Site Reliability Engineering.
* Experience in various tools like Git, GitHub, Bitbucket, Maven, Jenkins, Docker, Kubernetes, Helm, Terraform, SonarQube, Artifactory, and Nexus.
* Worked on GCP resources like - Google Compute Engine, App Engine (GAE), Google Kubernetes Engine, Google Cloud Function, VPC, Cloud CDN, Cloud Load Balancing, Cloud Interconnect, Cloud DNS, NAT, VPN, Private service connect, Cloud SQL, Cloud Storage, Cloud Monitoring, cloud logging, OS Patch Management, Identity and Access Management, Deployment Manager, Billing, Instance Groups, Instance Templates, Snapshots, Health Checks, KMS
* I worked on AWS services like AWS IAM, VPC, EC2, EKS, ECR, RDS, S3, Lambda, ELB, Auto Scaling, Route 53, Cloud Front, WAF, Certificate Manager, AWS Inspector, Cloud Watch, Cloud Trail,SNS.
* Experience in creating and managing computer, networking, and storage services on Azure and AWS using Terraform.
* I am experienced in creating various dashboards, metrics, alarms, and notifications for servers using **AWS Cloud Watch**, **Grafana, and** **Prometheus**.
* Implemented a **CI/CD pipeline** involving **GITHUB**, **Jenkins**, **Maven**, **Docker, Kubernetes, and Helm** for complete automation from commit to deployment.
* Managed GitHub repositories and permissions, including branching and tagging.
* Strong experience in system administration, installation, upgrading patches, configuration, troubleshooting, security, backup, disaster recovery, and performance monitoring on Red Hat Linux servers.
* I worked on the automation of deployment of all the microservices to pull images from the private Docker registry’s and deploy them to the Kubernetes cluster using HELM.
* Responsible for setting up and maintaining the Kubernetes clusters where ArgoCD will be deployed.
* Uses Python for infrastructure automation, configuration management, and deployment scripts.
* Create develop and test environments of different applications by provisioning Kubernetes clusters on AWS using Docker and Terraform.
* Experience in writing Kubernetes manifest files for Deployment, service, configmaps, secrets and ingress.
* Familiar with writing Docker Files, building Docker Images, and running the Docker Containers.
* Experience in end-to-end building and deploying the application in Dev environment to Production.
* Expertise in Deploying, Maintaining, Managing, and Troubleshooting of Production Systems.
* Setup and maintained Development, QA, Pre-production, and Production environments.

TECHNICAL SKILLS:

* Cloud Platforms:  AWS, GCP
* Operating Systems:  Red Hat Enterprise Linux, CentOS, Windows Server.
* Virtualization Tools:  Docker, Kubernetes, VMware.
* IAC Tools: Terraform, Cloud formation, ARM.
* Version Control Tools:  GIT, GITHUB, Bitbucket
* Web/Application Servers:  Apache Tomcat, JBOSS, IIS
* Automation Tools:  Jenkins
* Build Tools:  Maven, Helm
* Scripting:  Shell, YAML
* Ticketing tools: ServiceNow, PagerDuty, Jira Service Desk
* Monitoring Services: Cloud Watch, Datadog, Stack Driver
* Configration Management Tools: Ansible, Chef.

PROFESSIONAL EXPERIENCE:

Working as a Cloud and DevOps Engineer at Brillio from Aug 2019 to Till Date.

EDUCATIONAL QUALIFICATION:

* B-Tech in VIGNAN’S LARA INSTITUTE OF TECHNOLOGY & SCIENCE, VADLAMUDI Affiliated to JNTUK

PROJECT 2:

Company: Brillio

Client: Silicon Valley Bank

**Role: Google Cloud Engineer/SRE**

**Responsibilities:**

* Experience in designing, configuring, and implementing Google Cloud solutions .
* Design, implement, and maintain GCP resources, including, but not limited to, virtual machines, virtual networks, storage, and GCP data services.
* Proficient in deploying and managing applications on Google Cloud Platform (GCP).
* Knowledge of various cloud services, such as Compute Engine, App Engine, Kubernetes Engine, and Cloud Functions.
* Extensive experience in networking, virtual private clouds (VPC), load balancing, and security in GCP.
* Ability to architect highly available and scalable infrastructure solutions using GCP services.
* Experience in infrastructure as code using tools like Terraform or Deployment Manager to automate deployments.
* Expertise in monitoring, logging, and troubleshooting tools in GCP, such as Stack driver.
* Monitor and optimize GCP infrastructure for performance, cost efficiency, and security.
* Support deployment efforts for the delivery of project-based assignments including proof-of concept, analysis, design/architecture, deployment, and support.
* Ensure that GCP resources and services are compliant with industry standards and company policies.
* Implement and maintain security best practices for GCP, including identity and access management, firewall rules, and encryption.
* Create the file system with xfs from the EBS volumes and mount with Directories.
* Deployment of the Business processes by creating JAR, WAR and EAR files to Tomcat application server.
* Implement Chef for automated application deployments and release management.
* Automate the deployment of application updates and releases using Chef cookbooks.
* Collaborate with development teams to streamline the release process.
* Use Chef to automate the deployment and configuration of servers and applications.
* Collaborate with Chef developers to define infrastructure requirements and configurations.
* Responsible for setting up and maintaining the Kubernetes clusters where ArgoCD will be deployed.
* Collaborates with application development teams to integrate their applications with ArgoCD.
* Writes and maintains YAML manifests for applications and ensures they follow best practices.
* Sets up automated pipelines for testing, building, and deploying applications using ArgoCD.
* Monitors and troubleshoots deployment pipelines and ArgoCD operations.

PROJECT 1:

Company: Brillio

Client: nCino Bank Operating System

**Role: Aws cloud and Devops Engineer/SRE**

**Responsibilities:**

* Written Terraform scripts to automate AWS services, which include ELB, EKS, RDS, EC2, security groups, Route 53, VPC, subnets, and S3 buckets, are deployed via Jenkins.
* Utilized Jenkins master/slave architecture to distribute builds on nodes and trigger Jenkins jobs to build the artifacts using Maven, and deployed the Terraform templates to create the stack.
* Implemented a CI/CD pipeline with Docker, Jenkins, and GitHub by virtualizing the servers using Docker for the Dev and Test environments and achieving needs through configuring automation using containerization.
* Defined AWS Security Groups, which acted as virtual firewalls that controlled the traffic allowed to reach one or more AWS EC2 instances, and developed Terraform scripts to automate EC2 instances.
* Build S3 buckets, manage policies for S3 buckets, and use S3 buckets and Glacier for storage and backup on AWS.
* Proficient in creating Docker images using the Docker File
* I maintained artifacts in binary repositories using JFrog Artifactory and pushed new artifacts by configuring the Jenkins project Artifactory plugin.
* Launching AmazonEC2 Cloud Instances using Amazon Linux, RHEL, and Configuring Launched Instances with Respect to Specific Applications.
* I installed and configured Jenkins for continuous integration and release management and created branches and tags for each release and particular environments.
* Actively manage, improve, and monitor cloud infrastructure on AWS, EC2, S3, and RDS, including backups, patches, and scaling.
* Implemented the release practice and was responsible for pushing builds into the QA, UAT, pre-production, and production stages.
* Managed Kubernetes charts using Helm, Created reproducible builds of the Kubernetes applications, managed Kubernetes manifest files, and managed releases of Helm packages.
* Integrated Docker container orchestration framework using Kubernetes by creating pods, config maps, and deployments.
* Used MAVEN as a build tool for the development of build artifacts on the source code and deployed artifacts into the Jfrog (Jar and War file) repository, defined dependencies and plugins in Maven pom.xml for various activities, and integrated Maven with GIT to manage and deploy project-related tags.
* Implemented Docker -maven-plugin in Maven pom.xml files to build Docker images for all microservices and later used the Docker file to build the Docker images from the Java jar files.