SHASHIDHAR KASA
Sr. Cloud DevOps Engineer/ SRE
E-mail: shashidhark3839@gmail.com
Mob: 913-703-6693

PROFESSIONAL SUMMARY:

Experienced IT professional with over 11 years of comprehensive experience, including 6+ years as a DevOps Engineer and 4 years as a Linux Administrator. Adept at driving end-to-end SDLC initiatives with a focus on automation, cloud infrastructure, CI/CD pipeline design, and configuration management. Possesses strong hands-on expertise in deploying and managing scalable cloud solutions on AWS and Azure, containerization using Docker, orchestration with Kubernetes, and implementing infrastructure as code using Terraform and ARM templates. Passionate about optimizing release cycles, enhancing system reliability, and fostering DevOps culture across cross-functional teams.

- Demonstrated proficiency in utilizing a comprehensive suite of AWS cloud services, including autoscaling, leveraging AWS Storage, deploying, and managing EC2 instances, configuring ELB and EBS, establishing and securing VPCs, implementing infrastructure using CloudFormation, managing security groups and access control lists (ACL), utilizing Route53, RDS, S3, IAM, and monitoring with CloudWatch.
- Created, developed, and tested various application environments by skillfully provisioning Kubernetes clusters on AWS using Docker, Ansible, and Terraform, ensuring seamless execution of projects.
- Orchestrated the virtualization of servers on AWS, Linux, and Windows platforms, leveraging Docker to build and deploy containers using Docker Engine, Docker-Hub, and Docker Machine, resulting in highly scalable microservices-oriented environments.
- Managed and optimized hosting plans for Azure Infrastructure, facilitating efficient deployment of workloads on Azure Virtual Machines (VMs), while proficiently utilizing other Azure Cloud Services including storage, web apps, active directory, application insights, and logic apps.
- Spearheaded the implementation of end-to-end continuous integration (CI)/continuous deployment (CD) pipelines for Azure Cloud Services, utilizing Ansible to streamline the development and deployment processes.
- Demonstrated proficiency in Jenkins, encompassing plugin management, security, performance optimization, analytics, scaling strategies, and successful integration of code analysis and test phases, resulting in efficient CI/CD pipelines.
- Demonstrated proficiency in the principles and best practices of Software Configuration Management (SCM) in Agile, Scrum, and Waterfall methodologies. Utilized Atlassian tools such as JIRA, Kanban, and Bitbucket to facilitate defect management, team collaboration, source code management, and enable seamless continuous integration and deployment practices.
- Collaborated with development teams to establish best practices for using Harness, resulting in increased adoption and improved developer productivity.
- Led the development and maintenance of robust CI/CD pipelines, utilizing Jenkins and AWS cloud-native services, including CodeCommit, CodeBuild, CodePipeline, and CodeDeploy.
- Designed and implemented end-to-end Continuous Integration (CI) solutions across multiple environments, enabling agile, automated, and repeatable development processes with multiple deployments per day. Ensured seamless support for Azure Kubernetes Services (AKS).
- Managed provisioning and management of Azure Infrastructure as a Service (IaaS), facilitating efficient deployment of VMs, Virtual Networks, Web Apps, and Microsoft SQL Server. Utilized ARM Templates and harnessed the power of Azure DevOps CI/CD pipelines.
- Collaborated with cross-functional teams to optimize CI/CD processes, leveraging the strengths of Jenkins, Azure Pipelines, GitHub Actions, and GitLab Pipelines for efficient software product delivery.
- Implemented and optimized monitoring and alerting tools like Prometheus and Grafana, enabling proactive issue detection and providing comprehensive dashboards for applications in Kubernetes.
- Configured VNET Peering using Terraform Modules and fine-tuned NSGs for two-tier and three-tier applications, ensuring secure network traffic and facilitating resource connectivity across Virtual Networks.

- Implemented centralized log aggregation with the ELK stack, consolidating logs from diverse sources into a unified location for streamlined troubleshooting and proactive incident management.
- Demonstrated expertise in branching, tagging, version control, and maintaining various Version Controlling tools like Git, Subversion (SVN), TFS, and SCM client tools like GitLab, GitHub, Bitbucket.
- Utilized Python, Ruby, and Shell Scripts for configuration management and maintaining enterprise applications.
- Proficiently used JIRA for ticketing, defect tracking system, and configured various workflows, customizations, and integration with Jenkins, GitHub, Maven.
- Installed, configured, and managed monitoring tools such as Splunk and Nagios for resource monitoring, network monitoring, and log trace monitoring.

TECHNICAL SKILLS:

Cloud Technologies	AWS, MS Azure, Knowledge on GCP
AWS Services	IAM, EC2, S3, ELB, Autoscaling, CloudFormation, EBS, DynamoDB, RDS, EKS, ECR CloudWatch, CloudTrail, VPC, Route53, Security Groups, Lambda, Glacier, SNS, Load Balancer.
System Administration	Red Hat Linux, VMWare Administration.
CICD and CM tools	Jenkins, Azure Pipelines, GitHub Actions, GitLab Pipelines, Chef, puppet, Ansible.
Containerization and Orchestration	Docker, Kubernetes, Helm, Open-Shift
Build tools	Ant & Maven, Gradle, SonarQube, JUnit.
Monitoring tools	Nagios, Grafana, Prometheus, Datadog, Splunk & Zabbix.
Bug Tracking	JIRA, Remedy & ServiceNow.
Infrastructure	Terraform, Cloud Formation, ARM
Web Servers	Tomcat, JBoss, Web logic & Web Sphere.
Scripting Language	Shell, Bash, Yaml, Groovy, Python.
Source Control	GIT, GitHub, SVN, GitLab, Bitbucket.
Database	MySQL, PostgreSQL, Microsoft SQL, MongoDB & Oracle.
Programming	C++, java

WORK EXPERIENCE:

Client: DXC Technology, Ashburn, VA. (Nov 2023 to Present)

Role: AWS Cloud Engineer/SRE.

Responsibilities

- Involved in designing and deploying a multitude of applications utilizing AWS stacks including EC2, Route53, S3, RDS, Dynamo DB, SNS, SQS, LAMBDA, Redshift, focusing on high-availability, fault tolerance, and auto-scaling in AWS cloud formation.
- Supports dozens of AWS implementations; including Amazon EC2 (IaaS) and all Amazon RDS (DBaaS) offerings-AWS services include provisioning, implementation, migration, heterogeneous conversions and ongoing administration and monitoring support for SQL Server, Oracle, MySQL, MariaDB, PostgreSQL and Amazon Aurora.
- Migrating an infrastructure into an Amazon Web Services utilizing AWS Cloud formation, Code Deploy, EBS and launched the environment in VPC.
- Designed and implemented end-to-end CI/CD processes with Harness, enabling automated testing and deployment across multiple environments.
- AWS CloudFormation templates were created to establish custom sized VPCs, subnets, and NAT instances to ensure the successful deployment of Web applications and database templates.
- Employed automation, utilizing Terraform scripting practices and CloudFormation templates, to ensure repeatable deployments and for automating the AMI image deployment.

- Familiarity with Redshift security best practices, including IAM policies, VPCs, encryption, and audit logging.
- Creating and designing the Terraform templates to create customized VPC and NAT subnets for the deployment of Web application and database templates.
- Used Terraform modules to deploy EC2 instances from AWS SMS produced AMIs into the client's infrastructure, as well as to automate the entire process by connecting Terraform with Jenkins.
- Implemented AWS Code Pipeline and created cloud formation (JSON) templates in Terraform for infrastructure as code.
- Worked with Lambdas integrated with Elastic Search and Kibana to store certain log metrics and monitor cluster health in UI.
- Conducted training sessions for cross-functional teams on utilizing Harness effectively, resulting in improved team collaboration
- Created and deployed RESTful APIs using API Gateway and Lambda functions using Cloud Formation templates and AWS Serverless Application Model.
- Employed Apache Kafka on EC2 to provide a scalable solution for ingesting streaming data.
- Implemented a Docker delivery pipeline allowing for CI/CD of Java tomcat web service containers.
- Deployed application which is containerized using Docker onto a Kubernetes cluster which is managed by Amazon Elastic Container Service for Kubernetes (EKS).
- Worked on Deployment Automation of all micro-services to pull image from Private Docker registry and deploy to Kubernetes Cluster.
- Created Clusters using Kubernetes kubectl utility and worked on creating many pods, replication controllers, services, deployments, labels, health checks and ingress by writing YAML files.
- Worked on rolling updates using the deployments feature in Kubernetes and implemented BLUE GREEN deployment to maintain zero downtime to deploy process in Tomcat, and Nginx using Python and Shell Scripts to automate log rotation of logs from web servers and automate administration tasks.
- Used Ingress Resources in Kubernetes to support a high-level abstraction that allows simple host or URL or HTTP-based routing and used it to expose the applications.
- Authored Ansible Playbooks with YAML scripting and Puppet Manifests with DSL to provision Nginx, Apache Spark, Apache Web servers, and Tomcat servers.
- Build pipelines using Jenkins and configured Jenkins with the Amazon EC2 plugin that allows Jenkins to start slaves on EC2 on-demand and kill them as soon as they become idle.
- Worked on using GIT webhooks in Jenkins to trigger code and automate the CI/CD build process.
- Worked on build tasks using Maven, Ant, Gradle and GNU Make files and worked with development team to migrate Ant scripts to Maven.
- Performed tuning and troubleshooting Java and JS application by performing thread and heap dump analysis and utilizing profiling and monitoring tools like Dynatrace and Google Dev tools. Also, provided log monitoring solutions like Splunk and ELK stack.
- Utilized Python, jQuery, and Java etc. to design server applications and client interfaces.

Environment: AWS EC2, S3, RDS, IAM, AMI, Redshift, Lambda, VPC, Chef, Java, Git, Jenkins, Terraform, Python, Linux, Bash, Groovy, Subversion, Rest API, Ant, Maven, Nexus, U-deploy, SQL, Cloud Formation, Golang, OpenShift, Subversion, Selenium, UNIX, SVN, Docker, Jira, Python, Ruby, Shell Scripts, Tomcat, Ansible.

Client: CVS Health, Richardson, TX. (Oct 2022 to Oct 2023)
Role: Azure Cloud Engineer.
Responsibilities

- Designed and configured Azure Virtual Networks (VNets), subnets, Azure network settings, DHCP address blocks, DNS settings, and Security policies.
- Led implementation of Azure Active Directory for single sign-on and Authentication for Web Applications.
- Also configured Azure Role-based Access Control (RBAC) to segregate duties within our team and grant only the amount of access to users that they need to perform their jobs based on Roles defined.

- Developed event-driven architecture using Apache Kafka, ensuring real-time data processing and simplifying data flows between distributed systems.
- Created and deployed VMs on the Microsoft cloud service Azure, created and managed the virtual networks to connect all the servers and designed ARM templates/Terraform for Azure platform.
- Configured three types of blobs, block blobs, page blobs, and append blobs in Azure for storing a large amount of unstructured object data such as text or binary data, that can be accessed via HTTP or HTTPS and enabling data redundancy and Lifecycle Rules and Events.
- Worked on Managing the Private Cloud Environment using Ansible and Enhanced the automation to assist, repeat and consist of configuration management using Ansible based YAML scripts.
- Setup custom Domains and configure network security group (NSG) rules to specify ingress/egress traffic restrictions.
- Worked on Terraform to create the various services like AKS, ACR, VNET, VM. etc as infrastructure as code in various environments as per the project need.
- Monitored and cared for the Splunk infrastructure, guaranteeing optimal performance and availability.
- Oversaw the on boarding of new data, parsed diverse log sources, and guaranteed data accuracy on Splunk.
- Created inventory in Ansible for automating CD & developed Ansible playbooks and Roles using YAML scripting.
- Used ELK stacking to monitor the logs for detailed analysis, worked on dash boarding using Elastic, Logstash & Kibana (ELK), & setup real time logging & analytics for CD pipelines & applications.
- Worked on creation of Docker images on top of micro services and deployed on Azure Kubernetes services.
- Worked on Kubernetes cluster creation and creation of Deployments, services, RBAC and Ingress.
- Migration of on-premises data (Oracle/ SQL Server/ MongoDB) to Azure SQL/Cosmos DB using Azure Data Factory.
- Experience in Azure infrastructure management (Azure Web Roles, Worker Roles, SQL Azure, Azure Storage, Azure AD Licenses) using Terraform and managed Azure Infrastructure through Blueprints and Landing Zone.
- Utilizing Azure Stack (Compute, Web &Mobile, Blobs, ADF, Resource Groups, Azure Data Lake, Azure Data Factory, Azure SQL, App Services, and CosmosDB) and services for configuring and deploying Azure Automation Scripts for multiple applications.
- Deployed multiple micro services into Azure Kubernetes by Docker zing them and using Jenkins and Azure DevOps.
- Migrated to Build forge projects to Azure DevOps with all the work items, source codes, build and release pipelines by using custom PowerShell tool.
- Organizing, prioritizing, and monitoring task and milestone progress using GitLab's issue tracking and project management tools ensured that the project was in line with the company's goals.
- Integrated Docker container orchestration framework using Kubernetes by creating pods, and deployments.
- Competent in leveraging infrastructure-as-code and scripting to automate infrastructure activities.
- Developed Kubernetes clusters were managed and maintained to ensure the high availability and dependability of containerized applications.
- Deep understanding of infrastructure as code (IaC), automation, and scripting with Ansible, Terraform, and Puppet.
- Strong knowledge of version control systems, such as SVN and Git.
- Worked on using a GIT branching strategy that included developing branches, feature branches, staging branches, and master.

Environment: Azure, AAD, Azure DevOps, Terraform, AKS, Networking, Docker, Ansible, Prometheus, Grafana, Bash, Python, Linux, Jira, Bitbucket, Apache Tomcat, ARM, Virtualization, CRON, YAML, JSON, Splunk.

Client: Visa, Miami, FL. (April 2020 to Sep 2022)

Role: Site Reliability Engineer.

Responsibilities

- Installing, setting up & Troubleshooting CHEF. Created and automated platform environment setup.
- Worked as administrator on web and application servers like WebLogic, Tomcat, Apache, and Nginx.
- Regular Build jobs are initiated using the Continuous Integration tool with Jenkins.
- Implemented a CD pipeline involving GIT -Jenkins - Ansible to complete the Automation from commit to Deployment.

- Automated JAVA builds using TFS by installing Team Foundation Build Extensions. Installed Team Explorer Everywhere plugin for Eclipse users to be able to connect to TFS server.
- Installed packages on remote Agent less nodes through Ansible playbooks with Ansible roles
- Used file modules in Ansible playbooks to copy / remove / modify the files on agent less remote servers from Ansible control server
- Worked on playbooks for Ansible in YAML scripting. Downloaded and managed Ansible roles from Ansible Galaxy to automate the infrastructure.
- Written Python scripts for deployment of Java applications on bare servers or Middleware tools.
- Written Python scripts to apply the Integration label to all the files which needs manual labelling of files.
- Used Kafka to collect Website activity and Stream processing.
- Experience in migration of consumer data from one production server to another production server over the network with the help of python scripts.
- Owner of Jenkins, GIT Hub Artifactory and all internal build systems for the core development team on an enterprise-level Java-based cloud orchestration/automation tool.
- Utilized Kubernetes and Docker for the runtime environment of the CI/CD system to build, test deploy.
- worked on Docker and Kubernetes on cloud providers, from helping developers build and containerize their application (CI/CD) to deploying either on public or private cloud.
- Virtualized the servers using the Docker for the test environments and dev-environment needs and verified those environments in Docker Containers for Sanity tests before upgrading to production environment.
- Used Flume, Kafka to aggregate log data into HDFS.
- Worked on creating Docker containers leveraging existing Linux Containers and AMI's in addition to creating Docker containers from scratch.
- Created the Docker file and built various containers using Docker engine and Docker Machine environments and set up the automated build on Docker HUB.
- Defined and created usable and customer friendly, intuitive interfaces to the JIRA tool in a fast-paced evolving environment.
- Planning to move from CLOUD to GCP
- Implemented new JIRA workflows for the QA teams and worked on Splitting JIRA server's configuration. Managed JIRA users and created workflows, issue types, fields in production.
- Worked closely with the team to review code for compatibility issues, resolve issues as they arise, and implement deployment processes and improvements on a continuous basis.
- Installed, administrated Monitoring and visualization tools like Zabbix, Kibana and monitor the systems/services health and to generate the reports, raise incidents when critical alerts triggered as monitoring solution.
- Gathered all the stakeholder approvals, necessary signoffs while acting as a release manager for two development teams. Installed and configured the DHCP server for giving IP leases to the production servers.
- Managed RedHat Linux user accounts, directories, groups, and file permissions and Installed packages using rpm and yum.
- Migrated applications to the PKS, GCP cloud.
- Applied Clustering Topology for meeting High Availability and Failover requirement for functionality and performance.

Environment: GIT, Ant, Maven, TFS 2010, VSTS, UNIX, SVN, Jira, Ansible, Docker, Kubernetes, Zabbix, Python, Shell Scripts, Tomcat, JBoss, WebLogic, WebSphere, Jenkins.

Client: AbbVie, Chicago, IL. (Oct 2018 to Mar 2020)
Role: Cloud Engineer.
Responsibilities

- Responsible for the build and deployment of Java applications on different environments like Dev/QA & Production.
- Installed and configured Jenkins on a Linux machine, implementing a master and slave configuration to facilitate multiple parallel builds through a build farm.
- Leveraged Jenkins and Build Forge for continuous integration and deployment into Tomcat Application Server.

- Installed, configured, and administered Jenkins CI tool on Linux machines.
- Executed application installation and troubleshooting for Solaris, Red Hat Linux, and CentOS.
- Designed, administered, and troubleshooted database servers like MySQL 5.7, MySQL 5.6, MySQL 5.5.
- Implemented and managed the Nexus repository manager to streamline artifact sharing within the company.
- Developed and edited Shell Scripts to collect system performance information and account details.
- Designed and implemented efficient SCM processes and procedures while ensuring regular patching of Linux servers.
- Upgraded and maintained Splunk Indexers, Search heads, and forwarders.
- Collaborated closely with software developers and DevOps teams to debug software and system problems.
- Maintained and coordinated environment configuration, controls, code integrity, and resolution of code conflicts.
- Integrated release management with Jira for requirements management, utilized Bamboo for seamless build and deployment, and incorporated ServiceNow for efficient change management processes through relevant plugins.
- Configured XL Release templates encompassing pre-deployment, deployment, and post-deployment phases for streamlined application releases.
- Developed Puppet modules for streamlined configuration management of Tomcat/Apache/Splunk forwarder services in the distributed infrastructure.
- Administered Linux Servers (Red Hat – RHEL 3/4/5) encompassing diverse functions such as managing Apache/Tomcat servers, mail servers, MySQL databases, and firewalls in both development and production environments.
- Created monitors, alarms, and notifications for EC2 hosts using CloudWatch to ensure proactive monitoring and response.
- Employed advanced configuration management and automation tools like Puppet and Docker to streamline and automate infrastructure management.
- Oversaw the installation, configuration management, maintenance, and systems development of Red Hat Linux, Oracle Enterprise Linux, and UNIX Systems.
- Implemented a robust continuous delivery framework utilizing Jenkins, Puppet, Maven, and Nexus in a Linux environment.
- Managed user accounts, groups, directories, and file permissions on Red Hat Linux systems to ensure secure and efficient operations.
- Utilized JIRA custom workflows to track issues and utilized JBoss application server for streamlined application deployments.
- Maintained and updated entitlements for Red Hat Linux servers, ensuring compliance with the latest subscriptions.
- Implemented high availability solutions, load balancing, and clustering for critical Linux servers.
- Created and managed instances on OpenStack to enable flexible and scalable infrastructure deployments.

Environment: AWS, Tomcat Apache, Cloud Formation, Terraform, Elastic Search, Git, Linux, Jenkins, Maven, Chef, Ansible, SonarQube, Docker, Kubernetes, Windows servers, WebLogic, Webservers.

Client: TD Bank. Virginia. **(Feb 2014 – Sep 2018)**
Linux Administrator
Responsibilities

- Created repositories according to the structure required with branches, tags, and trunks.
- Created hooks and managed permissions on the branches for GIT.
- Responsible for designing and deploying best SCM processes and procedures.
- Installed and configured SSH server on Red hat/Cent OS Linux environments.
- Created and maintained several build definitions and publish profiles to handle automated builds in an Agile / SCRUM environment.
- Attended sprint planning sessions and daily sprint stand-up meetings. Configured application servers (JBOSS) to deploy the code.
- Used Apache Tomcat as application server for deploying the artifacts.
- Installed, administered, and configured Jenkins/Bamboo Continuous Integration tool.
- Used Ant, Maven to perform daily and weekly Software Builds.
- Helped developers and other project teams to set views and environments.

- Replicated the Jenkins build server to a test VM using Packer, Virtual Box, Vagrant, Chef, Perl brew and Server spec.
- Wrote shell scripts for automated installations to extract logs.
- Releasing code to testing regions or staging areas according to the schedule published.
- Used Jira as ticket tracking and workflow tool.

Environment: SVN, GIT, ANT, MAVEN, Chef, Windows/Linux, C#, Jenkins, JIRA AWS, JBOSS, Apache Tomcat, VM ware, Virtual Box, Perl Scripts, Shell Scripts, Unix/ Linux environment.