**Yamini V**

yamini.vadde@gmail.com

**+1 949 414 9802**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROFESSIONAL SUMMARY:**

* Having **9+ years** of experience in all the areas of Software development life cycle which are Included but not limited to automation, Implementation of **CI/CD**, System Administration, Cloud computing, Software Configuration Management **(SCM)**, Containerization, orchestration, Scripting for various environments.
* Setup and built **AWS infrastructure**. Performed data migration from on premises environments into **AWS**.
* Strong experience of data migration (to public cloud) practices and technologies.
* In-depth understanding of the principles and best practices of Software Configuration Management **(SCM)** in **Agile**, **SCRUM**, Waterfall methodologies.
* Working with varying business units, understanding their release requirements, resolving their release issues, and aligning them to appropriate release vehicles as per needs.
* Experience in setting up **CI/CD** using different **DevOps tools**.
* Experience in AWS services such as **EC2, ELB, Auto-Scaling, EC2 Container Service, S3, IAM, VPC, RDS, DynamoDB, Cloud Watch, Lambda, Glacier, SNS, SQS, Cloud Formation**.
* Experience on high-availability, fault tolerance, and auto-scaling in **AWS Cloud Formation**. Configured **AWS IAM** and Security Group in Public and Private Subnets in **VPC**. Created AWS Route53 to route traffic between different regions.
* Written Lambda functions to implement tasks like, taking **EBS** volume snapshots in regular intervals and copying the files in between the **S3** buckets only when the actions are triggered.
* Experience on Terraform a tool for building, changing, and versioning infrastructure safely and efficiently. Terraform key features such as Infrastructure as code, Execution plans, Resource Graphs, Change Automation.
* Expert in designing and distribution of Data across all the nodes and **Clusters** on different availability zones in AWS **Redshift** and experience in automating the infrastructure using **Terraform** in **AWS** console.
* Experience in building new application and deployment through Ansible and managing in production environment. Created **Ansible** Playbooks to Manage Configurations of **AWS Nodes**.
* Excellent experience on container management using **Docker** in creating images. Worked on Docker container snapshots, attaching to a running container removing, image managing, directory structures.
* Experience with containerization technology using **Docker** or evolving container management platforms like Kubernetes and using **Kubernetes** to manage dozens of **microservices**.
* Expertise includes bash and **Python scripting** with focus on DevOps tools, **CI/CD** Architecture, and hands-on Engineering.
* Implemented Continuous Integration with **Jenkins** integrated Jenkins with **Maven, SonarQube, Nexus** and **Ansible**.
* Designed and implemented Software configuration management **(SCM)** using **GIT**.
* Experience in implementing **Bitbucket** cluster. Experience in branching, tagging, and maintaining the version across the environments using **SCM** tools like **GIT/bitbucket**.
* Expertise in using build tools like **MAVEN** or the building of deployable Artifacts.
* Experience with **Oracle**, **MongoDB**, and **MS SQL** **Server DB** Tasks.

**PROFESSIONAL SKILLS:**

|  |  |
| --- | --- |
| **Cloud Platforms** | GCP, AWS, Terraform |
| **Configuration Management Tools**  | Ansible, Puppet  |
| **CI /CD Tools**  | Jenkins |
| **Build Tools** | Maven |
| **Containerization Tools****Programming Languages** | Docker, KubernetesGO, Python, Java |
| **Version Control Tools** | Git/GitHub, SVN  |
| **Logging & Monitoring Tools** | Nagios, CloudWatch, ELK stack |
| **Databases** | Oracle DB, DB2, MySQL, MongoDB, RDS |
| **Application/Web Servers** | Apache Tomcat, NGINX  |
| **Operating Systems** | UNIX/Linux, Windows, RHEL, CentOS, Ubuntu |
| **Virtualization Platforms** | Oracle VirtualBox, VMware Workstation, VMware. |
| **Bug Tracking Tools** | JIRA |
| **Scripting Languages** |  Python, YAML, Ruby, Shell, PowerShell, JSON, Groovy |

**ACADEMIC QUALIFICATIONS:**

* Bachelor of Engineering in **Computer Science and Engineering** from Anna University, Chennai.

**PROFESSIONAL EXPERIENCE:**

**Client: GitHub Jan 2022 – Till date**

**Location: SFO, CA (Remote)**

**Role: Infrastructure Engineer**

**Responsibilities:**

* Worked as Patch PRP (Primary Responsible Person). The Patch PRP Responsibilities are to identify the issues or bugs that need attention. Address merge conflicts. Merge backports.
* After the necessary backports have been merged, synchronize the release branches and After CI has verified the synchronization, merge the pull request.
* Check for unmerged patch-manifests PRs. Identify the next version and tag and Build .gm image for next version.
* Run initial sanity tests of available builds. Start generating the release notes, Release notes should help customers decide how urgent it is to upgrade use a combination of tracking issues, labels and metadata to generate the release notes.
* When all pull requests have been triaged, generate a gist of all patch release notes, then publish all the release notes to the respective repository.
* Generate a test issue for the testing team. They are responsible testing each patch version to be released.
* Once testing issues are clear, release all the Gold Master builds to Azure Marketplace.
* Create the release email and send it to customers to notify of the patch release.
* Tag all projects using shared Git repos Once tagging completes, verify the S3 and CloudFront checksums.
* Make the images publicly accessible.
* Cleanup the old builds, tags, and issues using Terraform scripts.

**Environment**: AWS, Azure, Ruby, Shell, Python GitHub Actions, Retcon, Janky, Docker, Kubernetes, MySQL, Ngnix, Terraform

**Client: Five9 Nov 2019 – Dec 2021**

**Location: Nashville, TN**

**Role: Sr. DevOps/Cloud Engineer**

**Responsibilities:**

* Involved in designing and deploying multitude applications utilizing most of the AWS stack (Including EC2, Route53, S3, RDS, IAM) focusing on high-availability, fault tolerance, and auto-Scaling in AWS Cloud Formation.
* Strong knowledge of data migration (to public cloud) practices and technologies.
* Designed AWS Cloud Formation templates to create custom sized VPC, subnets to ensure successful deployment of web applications and database templates.
* Worked with Ansible for automating VPCs, ELBs, security groups, SQS, S3 buckets.
* Strong experience utilizing Jenkins for enterprise scale infrastructure configuration and application deployments, checking out code from Git and used maven to build war/jar artifacts.
* Implemented AWS solutions like EC2, S3, IAM, EBS, Elastic Load Balancer (ELB), Security Group, Auto Scaling, and RDS in Cloud Formation JSON templates.
* Optimized volumes and EC2 instances and created multi–Availability Zone VPC instances. Maintained Route 53, VPC, RDB, SES and SNS services in AWS cloud.
* Created snapshots and Amazon machine images (AMI) of the instances for backup and creating clone instance and can create complex Identity and Access Management (IAM) policies for delegated administration within AWS.
* Building / Maintaining Docker container clusters managed by Kubernetes, Linux, Bash, GIT, Docker, using Kubespray. Utilized Kubernetes and Docker for the runtime environment of the CI/CD system to build, test deploy.
* Implemented rapid-provisioning and life-cycle management for RedHat Linux using Amazon EC2, Ansible and custom Python scripts.
* Created Python scripts to totally automate AWS services which includes web servers, ELB, Cloud Front distribution, database, EC2 and database security groups, S3 bucket and application configuration, this script creates stacks, single servers, or joins web servers to stacks.
* Proficient in writing AWS Cloud formation templates to create custom sized VPC, subnets, NAT, EC2 instances, ELB's and security groups.
* Managed Kubernetes charts using Helm. Created reproducible builds of the Kubernetes applications, managed Kubernetes manifest files and Managed releases of Helm packages.
* Using GO, developed a microservice for reading large volume of data(millions) from PostgreSQL database.
* Managed AWS EC2 instances utilizing Auto Scaling, Elastic Load Balancing and Glacier for our QA and UAT environments as well as infrastructure servers for GIT and Ansible.
* Used IAM to create new accounts, roles and groups and polices and developed critical modules like generating Amazon resource numbers and integration points with S3, Dynamo DB, RDS, Lambda and SQS Queue.
* Worked on setting up lifecycle management policies to take snapshot backups of EBS volumes for all servers moved to AWS from Datacenter and S3 life cycle policies to back up the data from AWS S3 to AWS Glacier.
* Worked on Multiple AWS instances, set the security groups, Elastic Load Balancer and AMIs, auto scaling to design cost effective, fault tolerant and highly available systems.
* Hands on experience on Kubernetes for container management that runs Docker containerized applications in a cluster of EC2 instances in Linux Environment.
* Support the deployment and operation of Experian’s strategic global PaaS platform, based on OpenShift/Kubernetes.
* Created monitors, alarms, and notifications for EC2 hosts using Cloud watch.
* Used Nagios for both log monitoring as well as application monitoring.
* Experience with web deployment technology specifically Linux/Nginx/Apache/Tomcat.
* Experience in Kubernetes to deploy scale, load balance and manage Docker containers with multiple names spaced versions and good understanding of Open Shift Platform in managing Docker Containers and Kubernetes Clusters.

**Environment:** GIT, AWS, Jenkins, Ansible, RHEL, Terraform, LINIX, Python, Shell Scripts, Artifactory, Kubernetes, PostgreSQL, ELK, Docker, JIRA, CFT.

**Client: FIS May 2018 – Oct 2019**

**Location: Atlanta, GA**

**Role: AWS DevOps Engineer**

**Responsibilities:**

* Used Terraform and Ansible, migrate legacy and monolithic systems to Amazon Web Services. Used Terraform scripts to configure AWS resources.
* Wrote Ansible playbooks from scratch in YAML. Installing, setting up & Troubleshooting Ansible, created and automated platform environment setup.
* Set up CI/CD pipelines for Microservices and integrated tools such as Maven, Bitbucket, SonarQube, Nexus, Docker for providing immediate feedback to DEV teams after code check-in.
* Set up Git repositories and SSH Keys in Bitbucket for Agile teams.
* Helped teams to configure Webhooks in Bitbucket to trigger automated builds in Jenkins.
* Setting up Jenkins’s master, adding the necessary plugins and adding more slaves to support scalability and agility.
* Set up CI/CD pipelines for Microservices and integrated tools such as Maven, Bitbucket, SonarQube, Nexus, Docker for providing immediate feedback to DEV teams after code check-in.
* Created nightly builds with integration to code quality tools such as SonarQube.
* Created quality gates in SonarQube dashboard and enforced in the pipelines to fail the builds when conditions not met.
* Automated infrastructure provisioning on AWS using Terraform and Ansible.
* Wrote Ansible playbooks from scratch in YAML. Installing, setting up & Troubleshooting Ansible, created and automated platform environment setup on AWS cloud.
* Effectively monitored the infrastructure using tools like Splunk, CloudWatch and Nagios.
* Converted java projects into Maven projects by creating POM file and ensured all the dependencies are built.
* Written Bash, Python Scripts and Ansible playbooks for infrastructure Automation and Deployment process.
* Installed, Configured, managed, and upgraded MongoDB cluster to latest version. Written Python scripts for Mongo monitoring and back up and written database queries.
* Provisioned load balancer, auto-scaling group and launch configuration for micro services using Ansible.
* Building/Maintaining Docker container clusters managed by Kubernetes, Linux, Bash, GIT, Docker, using Kubespray. Utilized Kubernetes and Docker for the runtime environment of the CI/CD system to build, test deploy.
* Created and managed local deployments in Kubernetes, creating local cluster and deploying application containers. Performed Kubernetes upgrades test on local containers before moving to Production clusters.
* Managed Kubernetes charts using Helm. Created reproducible builds of the Kubernetes applications, managed Kubernetes manifest files and Managed releases of Helm packages.
* Trained staff on Kubernetes cluster administration and managing Kubernetes upgrades.
* Experience in Kubernetes to deploy scale, load balance and manage Docker containers with multiple names spaced versions and good understanding of Open Shift Platform in managing Docker Containers and Kubernetes Clusters.

**Environment:** Apache, Tomcat, Docker, Terraform, Aws, Kubernetes, Git, SVN, Splunk, Ansible, Jenkins, AWS, MYSQL

**Client: OCLC Oct 2015 – Apr 2018**

**Location: Dublin, OH**

**Role: DevOps Engineer**

**Responsibilities:**

* As a DevOps Engineer worked in AWS environment utilized cloud compute services EC2, ELB, storage services S3, VPC, RDS, Trusted Advisor, Code Deploy and Cloud Watch.
* Configured S3 versioning and lifecycle policies to and backup files and archive files in Glacier.
* Configured AWS Identity and Access Management (IAM) Groups and Users for improved login authentication.
* Developing a Continuous Delivery (CD) with Docker, Jenkins, GIT and AWS pre-built images.
* Virtualized the servers using the Docker for the test environments and development environment.
* Uploaded Artifacts in CI pipeline to the Nexus remote repository.
* Container management using Docker by writing Docker Files and set up the automated build on Docker HUB.
* Writing Ansible playbooks, replacing the dependency on Chef Cookbooks and Chef Recipes to automate infrastructure as a code.
* Automate new VM setup for Web & Application Servers using Ansible.
* Responsible for CI and CD using Jenkins, Maven and Ansible.
* Creating Branching and Merging as per the schedule.
* Writing Ansible playbooks, replacing the dependency on Ansible playbooks and Ansible roles to automate infrastructure as a code.
* Installing and configuring Jenkins, installing plug-ins, configuring security, creating a master and slaves for implementation of multiple parallel builds.
* Developed Bash and python Scripts for the manual deploying the code to the different environments and updating the team after the completion of the build successfully.
* Installation and configuration of Nginx as a Load Balancer.
* Created the AWS VPC network for the Installed Instances and configured the Security Groups and Elastic IPs accordingly.
* Written multiple scripts in Ruby and Bash Shell for different migration tasks in the company on to AWS.
* Installing the Load Balancers and VPC with Public and private subnets.
* Configured AMI images with the help of AWS CLI.
* Coordinated with developers, Business Analyst and Managers to make sure that code is deployed in the Production environment.

**Environment:** CI/CD, AWS, Jenkins, Chef, AppDynamics, Puppet, Selenium, Java, Apache, Python, Nginx, MySQL, SVN, Nagios, Ruby, Oracle.

**Client: Centro, India Nov 2011 - Aug 2014**

**Role: Linux System Engineer**

**Responsibilities:**

* Set up and administer user and groups accounts, setting permissions for users.
* Build and installed multiple Linux machines.
* System and Application troubleshooting and other related issues.
* Provided management of system process in areas like boot process, startup, and system shutdown.
* Install, configure, maintain, and administer Linux / UNIX operating systems and components. Diagnosed and resolved.
* Problems associated with DNS, DHCP, VPN, NFS, and Apache.
* Planned and executed packages and updates installations necessary for optimal system performance.
* Diagnosed and resolved problems associated with DNS, DHCP, VPN, NFS, and Apache.

**Environment:** Linux / UNIX, DNS, VPS, NFS, Apache.