

**Supriya S**

**Contact: (919)-725-8122**

**Email:** **priyas.official21@gmail.com**

**LinkedIn Id:** [**Linkedin Profile**](http://www.linkedin.com/in/supriya-sl)

A passionate engineering professional with around 7 years of comprehensive IT experience in Azure and AWS Cloud services along with Build and Release Engineering, Infrastructure provisioning, DevOps engineering with in-depth knowledge of various Automation tools, System Administration

* Experience in administrating of **IAAS & PAAS** VM’s and Web roles on **Microsoft Azure (Classic)**, Resource Manager and troubleshooting issues on **Azure VM’s**. Supported technologies, such as **Active Directory Federation Services (ADFS), Azure Active Directory (AD), and PowerShell.**
* Hands on experience in cloud services, **IaaS**, worker/web role, service bus, **Azure Blob**, Table storages, queue and **API Management**. Configured **NSGs** for two-tier ad three-tier applications.
* Experience with designing and deployment of applications using almost all **AWS** stack (Including **EC2, S3, Route53, ELB, EBS, ECS, VPC, RDS, DynamoDB, SNS, SQS, IAM, KMS, Lambda, Kinesis**) and focusing on high-availability, fault tolerance, **auto-scaling** in AWS Cloud Formation, deployment services and security practices (**IAM, CloudWatch, CloudTrail**).
* Experience in **Google Cloud**components/services, establishing security rules within GCP and GCP client libraries
* Expertise in writing **terraform** scripts from scratch and building highly automated infrastructure using automation frameworks such as **CloudFormation** and **Terraform**.
* Expertise with **Jenkins** **pipelines** to drive all microservices builds to the **Docker** **registry** and then deployed to **Kubernetes**, Created **Pods** and managed using **Kubernetes**.
* Worked in **DevOps** group running **Jenkins** in a **Docker** container with **EC2** slaves in Amazon AWS Cloud configuration. Also gain familiarity with surrounding technologies such as **Mesos** (**Mesosphere**) and **Kubernetes**.
* Production experience in large environments using configuration management tools like **Chef**, **Ansible** and **Puppet** supporting **Chef** Environment with 500+ servers and involved in developing **manifests**. Developed Chef **Cookbooks** to monitor systems configuration.
* Experience in developing and implementing **infrastructure as code**, automated provisioning and configuring using **Chef**, **Puppet** and **Ansible**.
* Expertise in implementing DevOps culture through **CI/CD** tools like **Source code management**, **Jenkins**, **Code Deploy, Code Pipeline, Maven,** and configuration management tools like **Chef, Puppet & Ansible**.
* Experience in building end to end **CI/CD** Pipelines in Jenkins to retrieve code, compile applications, perform tests and push build artifacts to Nexus and **UDeploy** to orchestrated changes across servers and components.
* Expertise in using build tools like **ANT, MAVEN and Gradle** for building of deployable artifacts such as war & ear from source code.
* Experience in **Nexus** and **Jfrog** Artifactory Repository managers for **Maven** builds and managed the artifacts generated by **MAVEN** in the Nexus repository.
* Experience with monitoring tools like **Cloud Watch, Nagios, ELK and Splunk**.
* Having experience in developing a knowledge pipeline using Kafka to store data
* Utilized **MySQL, MongoDB**, **DynamoDB** and **Elasticache** to perform essential database administration.
* Well-versed in Install, configure, and troubleshoot enterprise-wide Apache web servers and **Tomcat** application containers in a large **Red** **Hat**, Linux/CentOS environment.
* Proficient with **Shell, Python, Ruby, Power Shell, JSON, YAML, and Groovy** scripting languages.
* Strong ability to troubleshoot any issues generated while building, deploying and in production support.
* Experienced in **software Development Life Cycle (SDLC)**, Agile Methodologies, Waterfall processes, Project Management and project release.

**Technical Skills:**

|  |  |
| --- | --- |
| **Cloud Environment** | Amazon Web Services, Azure, Google Cloud Platform |
| **Source Code Management** | Subversion, GIT, SVN, Bitbucket, and CVS |
| **CI/CD Tools** | Jenkins, Bamboo |
| **DevOps Tools** | Docker, Jira, Kubernetes, Open Shift, Terraform |
| **Configuration Management** | Ansible, Chef, Puppet |
| **Build Tools** | Ant, Maven, Gradle |
| **Monitoring Tools** | Nagios, Splunk, CloudWatch, New Relic, Elasticsearch, Logstash, Kibana, Grafana, DataDog |
| **Scripting Languages** | Shell, Python, YAML, RUBY, JSON |
| **Databases** | SQL Server, Oracle, MySQL |
| **Networking/Protocol** | TCP/IP, NIS, NFS, DNS, DHCP, WAN, SMTP, LAN, FTP/TFTP |
| **Web/Application Servers** | Apache Tomcat, Web Logic, Oracle, Application Server, Ngnix |
| **Operating Systems**  | Redhat-5/6/7, Ubuntu 14/16, CentOS 5/6/7, Windows |

**Professional Experience:**

**Client: Cigna,** **Bloomfield, CT Mar 2020 to Present**

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

**Responsibilities:**

* Dealt with Azure IaaS **- Virtual Networks, Virtual Machines, Resource Groups, Express Route, Traffic Manager, VPN, Load Balancing, Application Gateways, Auto-Scaling**. Exposed Virtual machines and cloud services in the **VNets** (Virtual Networks) to the Internet using **Azure Load Balancer**.
* Involved in **managing Azure Cloud Service roles (Web Roles, Worker Roles), Azure SQL, Azure Storage, Azure AD Licenses**. Provided high availability for **IaaS** **VM's** and PaaS instances for access from other services in the **VNet** with Azure load balancers and implemented Azure classic and **Azure ARM** for the deployments
* Created and managed **Azure AD tenants,** applications with **Azure AD** and integrated on-premises Windows AD with Azure AD.
* Created subscription, Storage Account and tables using Windows PowerShell using automation in Azure.
* Used Azure cloud services, Azure storage, Azure active directory, Azure Service Bus. Led implementation of **Azure Active Directory** for single sign on, authentication, authorization and Azure Role-based Access Control (RBAC).
* Implemented Azure Key Vault to store Secrets, Password and critical information and configuring it for Application usage.
* Worked on GCP services like compute engine, load balancing, storage, cloud SQL, stack driver monitoring and cloud deployment manager.
* Implemented **­­­­­­­­** to allow or deny traffic to and from the VM’s instances based on specified configuration and used GCP cloud CDN (content delivery network) to deliver content from GCP cache locations drastically improving user experience and latency.
* Maintained image for **GCP** workspace including **product deployments, patches, and hotfixes** and also worked on migration of the services from **GCP** to **Azure**
* Used Terraform to provision Infrastructure for cloud with **Infrastructure as a code** (IaaC)**. Also, c**reated reusable **Terraform modules**in Azure environment.
* Worked with **Terraform Templates** to automate the Azure Infrastructure as a Service (IaaS) virtual machines using Terraform modules and deploy the virtual machine scale sets in a production environment and so implemented Terraform Enterprise to provision infrastructure across Azure workloads and Kubernetes clusters.
* Used Terraform to map more complex dependencies and identify network issue. Used Terraform for server provisioning and made use of this for CI/CD pipeline building using AzureDevOps.
* Built AzureDevOps built and release jobs to create GCP infrastructure from AzureDevOps repos containing terraform code
* Implemented **Azure Kubernetes** service to deploy a managed Kubernetes cluster in Azure and create an AKS cluster in the Azure portal, with the Azure CLI.
* Utilized **Kubernetes** to orchestrate the deployment, scaling and management of **Docker** Containers.
* Used **Helm charts** to package and deploy common applications in Kubernetes.
* Setting up **Kubernetes** (k8s) Clusters for running microservices and pushed microservices into production with Kubernetes backed Infrastructure. Developed automation of **Kubernetes clusters** via playbooks in **Ansible.**
* Performed code commit, revert, checkout and merge operations on GIT and pushed sourced code to GitHub.
* Defined dependencies and plugins in Maven pom.xml for various activities and integrated Maven with GIT to manage and deploy project related tags.
* Used Ansible to deploy applications on all servers through SSH. Gathered required configurations of all servers and maintained/updated the playbooks using Ansible. Automated various **infrastructure activities** like Continuous Deployment and Application Server setup
* Used **Ansible** Tower, which provides an easy-to-use dashboard and role-based access control, so that it’s easier to allow individual teams access to use **Ansible** for their deployments
* Defined and implemented monitoring processes for uptime, performance. Used **ELK** for monitoring purposes for LOG analytics and Application monitoring and integrated this with Azure for security purpose we use search guard to protect the logs.
* Worked with ARM templates and scripts using Azure PowerShell during Automation and Build Process. Sync Azure SQL Database with on-premises Databases using ARM Portal, SSMS.
* Maintained and Debugged Code to meet the required standards by using SonarQube and used Selenium for testing purposes. Also, monitored system activities to optimize performance and ensure security of systems.

**Environment:** Azure, Tomcat Apache, Terraform, Dynamo DB, Elastic Search Kibana, Git, Linux, AzureDevOps, Maven, Ansible, SonarQube, Docker, Kubernetes**.**

**Client: Aetna, Cary, NC Sept 2019 to Feb 2020**

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

**Responsibilities:**

* Involved in designing and deploying multitude applications utilizing almost all **AWS stack** (Including EC2, S3, AMI, Route53, RDS, SNS, SQS, IAM) focusing on high-availability, fault tolerance, and Auto-Scaling
* Worked on Multiple **AWS instances**, set the **security groups, ELB**s and **AMI**s, Auto scaling to design cost effective, fault tolerant and highly available systems.
* Handled **migration of on-premises** applications to the cloud and created resources in the cloud to enable this. Used **ELB** and **Auto-Scaling** policies for scalability, elasticity, and availability.
* Configured an AWS Virtual Private Cloud (VPC), NACL, and Database Subnet Group for isolation of resources within the Amazon RDS and Aurora DB clusters.
* Created IAM policies within AWS and configured IAM users, roles, policies to grant access of AWS resources to users.
* Managed different infrastructure resources, like physical machines, and even Docker containers using Terraform.
* Created **Docker images** using a Docker file, worked on Docker container snapshots, removing images and managing Docker volumes. Achieved containerization of Web application using Docker.
* Worked on Opensource development tools like **Docker Containers** and installation of Docker using YUM, RPM package management service and used **Docker Swarm** to manage Docker clusters using swarm manager.
* Developed Production environment of different applications on **AWS** by provisioning Kubernetes clusters on **EC2** instances using **Kubernetes** Operations (KOPS) a cluster management tool to spin up a highly available production cluster.
* Created and deployed **Kubernetes** pod definitions, tags, labels, multi-pod container replication and managed multiple **Kubernetes** pod containers scaling, and auto-scaling using KOPS
* Worked with **Kubernetes framework** using Ansible with AWS ELB, Istio, Grafana by ensuring autoscaling on encrypted EBS volumes
* Configured and maintained Jenkins to implement the CI process. Built Jenkins jobs to create AWS infrastructure from SCM repos containing Terraform code.
* Configured **Git,** build scripts using **ANT** and **MAVEN** build tools with Jenkins and schedule jobs using POLL SCM option and integrated to automate the code checkout process.
* Used Ansible for deployment configuration automation on production servers. Created **Ansible playbooks**, which are the entry point for Ansible provisioning, where the automation defined through tasks using YAML format and run Ansible Scripts to depending on provision to servers.is.
* Created **Custom Ansible modules** for finding facts about Cloud watch alarms and taking actions to pause/unpause those alarms during deployments. Docker, Ansible and Mesos, which reduced considerably deployment risks.
* Worked with automation of OS builds and application installations through Ansible
* Automated few of the applications and servers using Python and monitoring of these applications and servers using Nagios, also worked with plug-ins in Nagios for monitoring resources.
* Created scripts in Python, which integrated with Amazon API to control instance operations.
* Used Kafka for asynchronous **communication** between microservices to avoid issues in relational databases
* Designed REST API’s using Golang and worked with developers’ team on frameworks
* Worked on NoSQL, Dynamo DB to process large data documents.

**Environment:** AWS, Jenkins, Git, ANT, Maven, Docker, Kubernetes, Istio, Calico, Mesos, Ansible, Nagios, Terraform, WebLogic, Shell, Python, Ruby, PowerShell, Groovy, Perl.

**Client: Accenture, Hyderabad, India May 2017 to June 2019**

**Role: DevOps/Cloud Engineer**

**Responsibilities:**

* Implemented AWS solutions using EC2, S3, RDS, EBS, IAM, SNS, SQS, Route 53, Elastic Load Balancer, Cloud Watch, VPC, Cloud Formation, Cloud Front, Auto scaling groups. Also created Cloud Formation templates using AWS's Server less Application Model and deploy RESTful API's using API Gateway and Lambda.
* Launched LAMP stacks in multitier AWS instances in different subnets in Amazon VPC, and Security Groups to maintain high security. Worked with tools for log analysis and alert triggering and connecting them to different monitoring tools (ELK, AWS CloudWatch and SCOM)
* Implemented AWS (Amazon Web Services) including deploying new server instances through automation with Chef and Jenkins. Also, designed on-premise to off-premise CI/CD docker pipelines with ECS, Glue, Lambda, ELK, firehose and kinesis stream.
* Created **AWS** cloud formation templates to create custom-sized **VPC**, **subnets**, **EC2** instances, **ELB**, security groups. Worked on tagging standard for proper identification and ownership of **EC2** instances and other **AWS** Services like **Cloud** **Front**, **cloud** **watch**, **RDS**, **S3**, **Route53**, **SNS**, **SQS**, **Cloud** **Trail**.
* Implemented a 'server less' architecture using API Gateway, Lambda, and Dynamo DB and deployed AWS Lambda code from Amazon S3 buckets. Created a Lambda Deployment function, and configured it to receive events from your S3 bucket
* Created **S3** **buckets** maintained and utilized the policy management of **S3** buckets and Glacier for storage and backup on **AWS**.
* Automated application deployment in the cloud using Docker technology, Created and managed a Docker deployment pipeline for custom application images in the cloud using Jenkins.
* Implemented **Docker Swarm** applications and deployed spring boot applications.
* Installed and configured Continuous Integration tools such as Jenkins for build and deployment automation.
* Implemented **Continuous Delivery pipeline** with GIT Hub, Jenkins, Docker and AWS AMI's (Linux), whenever a new GIT Hub branch gets created Jenkins attempts to build a new Docker container from it, the Docker container has the AMI baked in and leverages Linux containers.
* Managed the Maven Repository using Nexus tool and used the same to share the snapshots and releases of internal projects.
* Developed build and Deployment Scripts using **ANT and Maven** as build tools in Jenkins to move from one environment to other environments. Integrated Junit tests in **ANT & Maven** and configured Jenkins to send the Unit test report to the development team.
* Responsible for design and maintenance of the **Subversion/GIT Repositories**, views, and the access control strategies.
* Utilized Configuration Management Tool Chef and also created and managed Chef **Cookbooks** using recipes to automate **system operations.** Involved in Configuring, monitoring and multi-platform servers by defining Chef Server from workstation to manage and configure **Chef Nodes**.
* Implemented CI process for Chef Cookbooks development workflow with help of **Test Kitchen**, **Foodcritic,** Robocop and Chef Spec. Implemented behavior-driven tests for Chef-managed infrastructures using Cucumber.
* Implemented environments, roles, and data bags in Chef for better environment management. Written Chef Cookbooks and recipes in Ruby to Provision several pre-prod environments consisting of Cassandra DB installations, WebLogic domain creations and several proprietary middleware installations.
* Worked on integrating **application logs** with **Splunk** and wrote several custom Splunk queries for monitoring and alerting.
* Worked on developing Restful webservices and Micro Services using Golang
* Implemented RESTful API services that work as a middleware between our application and third-party APIs that we will used using Golang.
* Worked with scrum and development teams in overseeing bug tracking, test prioritization, test automation and releases.

**Environment:** AWS, Java/J2EE, ANT, Maven, GIT, Subversion, Jenkins, Docker, Chef, Jira, Cobertura, Apache, WebLogic, Oracle, PowerShell, Bash, Python, Perl.

**Client: CGS IT Technologies Ltd, Visakhapatnam, India July 2015 to April 2017**

**Role: Build and Release Engineer**

**Responsibilities:**

* Build, manage, and continuously improved the build infrastructure for global software development engineering teams including implementation of build scripts, continuous integration infrastructure and deployment tools.
* Installed and customized rational tools like ClearCase and ClearQuest, support and maintenance of these tools.
* Installed and configured automated tool Puppet that included the installation and configuration of the master and agent nodes. Created **Puppet manifests**, **classes and modules** to automate system operations. Used Puppet Dashboard and Puppet DB for configuration management to existing infrastructure.
* Automated deployment modules of IIS web applications, bindings and configuration settings using a combination of PowerShell scripts and Puppet.
* Worked on Artifactory repository to maintain artifacts and used as a local repository
* Configured **Bamboo**, **created Jobs** for automating build and deploy activities.
* Used ANT as a build tool on java projects for the development of build artifacts on the source code. Developed ANT Scripts to automate the build process.
* Designed and implemented Subversion, CVS metadata including elements, labels, attributes, triggers and hyperlinks.
* Implemented branching, merging and build/release strategies utilizing Subversion. Also, assisted Migrating code base from **CVS to Subversion** Client.
* Developed NIX and Perl Scripts for the purpose of manual deployment of the code to the different environments and email the team when the build is completed.
* Build **artifacts** (war and ear) and deployed into a WebLogic server by integrating the WLST scripts to Shell Scripts.
* Executed user administration and maintenance tasks including creating users, groups, reports and queries.
* Worked as a **system administrator** for the build and deployments process on the enterprise server.
* Integrated Junit, finding bugs, check style, code analysis and testing during the build. Integrated Eclipse with Selenium Server for doing the automated smoke testing of the application standalone.
* Participated in the release cycle of the product, which involves environments like Dev, QA, UAT and production.

**Environment:** Linux, UNIX, SVN, CVS, Tomcat, JBoss, Bamboo, Puppet, Clear Case, Subversion, JIRA, Shell, Perl, MS SQL Server, WebLogic, Eclipse, Remedy, Oracle.

**Client: Quant Systems, Visakhapatnam, India Aug 2013 to June 2015**

**Role: Linux Administrator**

**Responsibilities:**

* Installation, administration, configuration of RHEL 5/6 including **package management, patch management**, regular system administration tasks. Developed and supported the Red Hat Enterprise Linux based infrastructure in the cloud environment.
* Installed Red hat Linux on bare metal servers and virtual machines using kick start and hardening the servers.
* Creating, cloning Linux Virtual Machines, templates using VMware Virtual Client 3.5 and migrating servers between ESX hosts.
* Performed daily maintenance routines on Linux servers, **monitoring system access**, **managing file space** and **tuning the system** for optimum performance.
* Installing and configuring Apache and supporting them on Linux production servers.
* Resolved all UNIX systems problems as a part of on call rotation on a timely basis. Security, Backup, Disaster Recovery, Performance monitoring and Fine-tuning of UNIX systems on different OS levels.
* Participated in migrating environment from 4.x to latest 5.x version.
* Setup user and group login ID, printing parameters, network configuration, password, resolving permissions issues, user and group quota.
* Responsible for **change management**, **troubleshooting** and resolution of server issues for end user’s support.
* Written shell scripts for automated Back- ups and Cron Jobs. Responsible for maintenance of development tools and utilities and to maintain shell, Perl automation scripts.
* Monitored server and application performance & tuning via various stat commands (vmstat, nfsstat, iostatetc) and tuned I/O, memory, etc.
* Written Queries in RDBMS such as Oracle and MYSQL by using SQL for Data integrity.
* Troubleshooted critical hardware and software issues and other day-to-day user trouble tickets.

**Environment:** VERITAS NetBackup, VERITAS Volume Manager, Samba, NFS, NIS, LVM, Linux, Shell Programming, SQL, Active Directory, Json, Nunit, BEA Web logic servers.