**Sandeep Jamalpur Kasinath**  
**sandeepjk.devops009@gmail.com  
(484)320-7121**

**Summary**

Associated with IT industry for **7+ years** working as **AWS Cloud Engineer,** **Cloud/DevOps Engineer, DevOps/AWS Engineer** and **Systems/VMWare Engineer** within **Linux/Unix/Windows** environments to provide solutions to the complex issues utilizing most of the resources such as **Virtualization**, **Integration** and **configuration Management** tools, **different Infrastructures** (**On-Premises Bare-Metal, privatized or Public Cloud**) for **Server Provisioning, Configuring, Monitoring, Disaster Recovery**.

* Experience in utilizing **Amazon AWS** cloud includes services like **EC2, S3, VPC, ELB, IAM, Cloud Watch, Elastic Beanstalk, Security Groups, EC2 Container Service (ECS), Lambda, Kinesis, Code Commit, Code Pipeline, code deploy, Dynamo DB, Auto Scaling.**
* Built out Infrastructure as a Service (IAAS) private cloud **OpenStack** and managed deployment of Microservices using **Kubernetes**, **Docker** and **etcd** to production environment. Performed server monitoring, Application monitoring, Capacity Planning and log monitoring using **Nagios**, **Cacti**, **Zabbix** and **Splunk**.
* Hands-on on **Ansible** and **Ansible Tower** as Configuration management tool, to automate repetitive tasks, quickly deploy critical applications, Environment configuration Files, Users, Mount points, Packages and proactively manage change and written **Ansible Playbooks using Yaml** for Maintaining **Roles, Inventory Files** and **Groups Variables.**
* Experience in Installing Chef Server Enterprise On-premise/workstation/bootstrapped the nodes using knife and developed **Chef Cookbooks, Recipes, Roles** and **Data Bags** to automate the services related to deployment.
* Utilized **Kubernetes** as a platform to provide a platform for automating the deployments, scaling and operation of application containers across a cluster of hosts. Worked closely with development teams and test engineers for **EC2 size optimization** and **Docker build Containers**.
* Container management using **Docker** by writing **Docker file** in **JSON** format and place the automated build in the **Docker Hub**. Managed deployments using **Kubernetes**, created local clusters and deployed application container.
* Converted existing **Terraform** modules that had version conflicts to utilize **Cloud formation** **templates** during deployments, worked with Terraform to create stacks in AWS, and updated the **Terraform** scripts based on the requirement on regular basis.
* Responsible for configuring Volume Group using **LVM,** Logical and Physical Partitions and Importing New Physical Volumes and Managed **file system, created swap, monitored, and extended volume groups.**
* Experienced in troubleshooting, automating various builds and deployments using **ANT, MAVEN, Perl, Ruby, MS Build,** and **Shell Scripts** and deploying to web and application servers like **WebSphere, WebLogic, and Tomcat.**
* Experience in Installing and setting up of **RAID hardware/software** for **data backup and storage**. Highly experienced in creating **new file systems**, **mounting file systems,** and **unmounting file systems**.
* Expertise in implementation and maintenance of **Apache Http, SMTP, DHCP, NFS, NIS, NIS+, LDAP, DNS, SAMBA, SQUID, Postfix, send mail, FTP, Remote Access, Security Management Security trouble shooting skills.**
* Experience in working with **Linux** including **Red Hat Enterprise Linux, CentOS, Ubuntu, Debian,** configuration, and administration **of Red Hat Virtual machines in VMware Environment.**
* Communicate changes, enhancements, and modifications- verbally or through written documentation using **confluence** to various groups in the organization to facilitate new or improved **business processes** and systems via **change log**.
* Understanding standards and practices in **Software Configuration Management (SCM)** in **Agile-SCRUM & Waterfall methodologies.** Implemented **Change Management** for tracking different clients and setup tracking using **JIRA/ ServiceNow**.

**Education**

Bachelor’s in computer science from Jawaharlal Nehru Technological University – 2012

Master’s in Computer Information Science from Bellevue University- 2017

**Technical Skills**

|  |  |
| --- | --- |
| Operating Systems | RHEL/CentOS 5.x/6.x/7, Linux-Ubuntu/Fedora, Windows 7/8/XP/10, AWS-Linux CLI |
| Networking | VPC, Route-53, LDAP, DNS, FTP, DHCP, SSH |
| Databases | MySQL, MongoDB, Cassandra, PostgreSQL, SQL Server |
| Source Control | GIT, SVN, Bitbucket |
| Configuration Management | Ansible, Puppet, Chef |
| Virtualization/ Containerization | VMWARE, VSPHERE 5.5/6.0, VCENTER, Docker, Pivotal Cloud Foundry, Vagrant |
| Web/Application Server | IIS, Tomcat, Apache, Jboss, WebSphere Application Server, Web Logic |
| Build/Deploy/Automation Tools | ANT, Maven, Gradle, Jenkins, Bamboo, TeamCity, TFS, Perl, Python, Shell, bash, Groovy |
| Infrastructure | AWS, OpenStack |
| Software Methodologies | Agile-SCRUM, Waterfall |
| Bug tracking/Monitoring | JIRA, Confluence, Rally, Remedy, HP Quality Center, Bugzilla, Splunk, Nagios |

**Professional Experience**

**Client: US Bank, Charlotte NC. Dec 2019 – Present**

**Role: AWS Cloud Engineer**

* Implemented AWS solutions using **EC2, S3, RDS, EBS, Elastic Load Balancer, Auto scaling groups**, Optimized volumes and **EC2**instances.
* End to end deployment ownership for projects on **Amazon AWS** which includes scripting for automation, scalability, build promotions for staging to production
* Integrated **Amazon Cloud Watch**with **Amazon EC2**instances for monitoring the log files, store them and track metrics, Automated the dev, QA environments to auto shutdown during off hours and weekends leveraging the AWS Lambda and CloudWatch Event Rule.
* Experience with **Terraform**to build infrastructure as a code, Utilization of **Terraform** for automated infrastructure deployments (**AWS ELB, AWS EC2, AWS Security Groups, AWS VPC, AWS Key Pairs, AWS IAM)**
* Create fully automated CI/CD using **Jenkins, Docker, Kubernetes, and GitHub.**
* Implemented a production ready, load balanced, highly available, fault tolerant, auto scaling, Kubernetes AWS infrastructure and micro service container orchestration.
* Manage and replicate infrastructure and instance configurations using **Terraform** and **Packer**
* Performed **GIT**/**Bit Bucket**migration and managed branching strategies using **GIT**flow workflow
* Built Jenkins jobs to create **AWS**infrastructure from **GitHub**repos containing **Terraform**code, to help the Dev and QA to setup the environment to progress the development and testing at faster pace.
* Created various **Terraform scripts**to deploy **Kubernetes, Dev, QA, and Pre-prod**environments in **AWS**Cloud.
* Managed **Ansible Playbooks**with Ansible roles, used different modules to build custom Ansible **playbook** Service modules, File modules, Template modules, etc.
* Created inventory in **Ansible** for automating continuous deployment and wrote **Playbooks** using **YAML** scripting
* Involved in Development and Implementation of Continuous Integration (**CI**) and Continuous Deployment (**CD**) pipeline involving **Jenkins**, **Ansible**, **Terraform** and **Docker**containers to complete the automation from commit to deployment
* Worked on creation of multiple Amazon Machine Images (AMI) for multiple regions and different environments using Packer Scripts and automated in **Ansible playbooks**
* Built **Terr grunt** project to manage Terraform configuration file DRY while working with multiple terraforms modules
* Installed **Docker**Registry for local upload and download of **Docker**images and even from **Docker**hub
* Implemented build stage to build the micro service and push the Docker container image to the private **Docker**registry

**Client: The Vanguard Group, Pennsylvania May 2019 – Dec2019**

**Role: Cloud/DevOps & Release Engineer**

* Hands-on experience with **Amazon Web services (AWS)** and implemented solutions using **EC2, S3, and RDS** in **cloud formation Json templates**, EBS, Elastic Load Balancer, and Auto Scaling Groups.
* Developed and managed **cloud VM’s** with **AWS EC2** **command line clients and management console.** Created **AWS Route53** to route traffic between different regions and alarms and notifications for EC2 instances using **Cloud Watch.**
* Build new **CI pipeline, Testing and deployment automation** with **Docker, Jenkins** and **Ansible.** Integrating **SonarQube** in the CI pipeline to analyze code quality and obtain combined code coverage reports after performing static and dynamic analysis.
* Used **Elasticsearch, Logstash and Kibana** (ELK stack) for centralized logging and analytics in the continuous delivery pipeline to store logs and metrics into **S3** bucket using lambda function.
* Worked with Application Performance Monitoring tools like **AppDynamics** for monitoring JAVA, .NET and PHP applications along with **NewRelic** for monitoring performance like browser and for tracking issues in SQL statements.
* Created and maintained various DevOps related tools for the team such as **provisioning scripts**, deployment tools and staged virtual environments using **Docker** and **Vagrant**.
* Managed **Nexus** and **Maven** repositories to **download artifacts** during the build and wrote **Ansible Playbooks**, which is entry point for **Server Provisioning,** where the automation identified through the tasks using **YAML format**.
* Experience using **Ansible Tower**, which provides an **easy-to-use dashboard**, and **role-based access control,** so that it is easier to allow individual teams access to use **Ansible** for their **deployments.**
* Managed **Microservices** using Docker to quickly spin up into reproducible dev environment and scaling them and orchestration using **Amazon EC2 container service** (ECS) and deploy it to an **Amazon EC2 instance.**
* Created **Docker compose** file using **YAML** for deploying **MEAN stack** (MongoDB, Express, AngularJS, Node JS) applications onto Docker containers before deploying them into production environment.
* Created **Docker Images** using a **docker file**, worked on **Docker Container Snapshots**, removing images and managing **Docker** Volumes deployed using **Kubernetes**.
* Used **Terraform** for managing the infrastructure through the terminal sessions and executing scripts in creating alarms and notifications for EC2 instances using **AWS Cloud Watch**. Wrote **Ansible Playbooks** for various applications and deploying them in AWS using **Terraform**.
* Used **bash** and **python** with **Boto3** to supplement automation provided by **Ansible** and **Terraform** for tasks such as encrypting EBS volumes backing AMI’s and scheduling lambda functions for routine AWS tasks.
* Deployed **java applications** into web **application servers** like **Apache Tomcat** servers to host the applications. Involved in configuring **Nginx** as a **web server.**
* Schedule and perform process audits and recommend process improvements. Continuously work towards optimization of process and workflow to improve turnaround and resolution time.

**Client: Symplr – Houston, TX Mar 2017 – Apr 2019**

**Role: DevOps/AWS Engineer**

* Designing and deploying large number of applications utilizing most of the AWS stack includes not limited to **EC2, Route53, S3, RDS, Dynamo DB, SNS, SQS** and **IAM** focusing on **high-availability, fault tolerance** and **auto-scaling** in **AWS cloud formation**.
* Involved in working with **cloud/storage systems** along with few **SaaS applications**. Worked in deployment of c**ontent delivery networks (CDN)** and **system development in cloud environments**.
* Designed **AWS cloud formation templates** to create **custom sized VPC**, **Subnets**, **NAT** to ensure successful deployment of web-applications and database templates. Configuring and networking of virtual private cloud (**VPC)**, **written cloud formation templates** and **deployed AWS** resources using it.
* Expertise in configuring and deploying instances on Cloud environments and **On-Premise** Data centers. Experience in the areas of **DevOps**, **CI/CD** Pipeline, Build & release management, **AWS,** and Linux/Windows Administration.
* Storage on **AWS EBS, S3** and **Glacier** and automate sync data to Glacier. Database services on AWS like RDS, Dynamo DB, Elastic Transcoder, Cloud front, Elastic Beanstalk.
* Configured AWS Identity and Access Management (IAM) groups and users for improved login authentication. Designed **Network Security Groups (NSGs)** to control inbound and outbound access to **network interfaces (NICs), VMs and subnets**.
* Writing **Terraform templates**, Chef Cookbooks and push them onto Chef Server for configuring EC2 Instances and **Bamboo AWS Code Deploy plugin** to deploy applications. Configured/Integrated **Bamboo** with **Bitbucket** to pull codes, **ANT** to generate builds and push artifacts to AWS S3.
* Configured **Chef** to manage AWS environments and automate the build process for core AMIs used by all application deployments including Auto scaling, and incorporating **Chef** using **AWS OpsWorks** on the **JSON** Cloud Formation scripts for the configuration deployment onto the nodes.
* Created **EBS snapshots**, rebuilt new EBS volumes from snapshots to migrate applications, and storing these snapshots into S3 buckets for backup.
* Managing **GIT version control system** in creating branches, tags, merge, resolving conflicts. Developed CI/CD system with Jenkins on Docker for the run-time environment to build, test and deploy.
* Experience in working with several Docker components like **Docker Engine, Hub, Compose, and Docker Registry** for storing **Docker** **images and files**, running multiple containers in **staging and production environments.**
* Worked to setup **Jenkins** as a service inside the **Docker Swarm cluster** to reduce the failover downtime to minutes and to automate the Docker containers deployment without using configuration management tool.
* Used **AppDynamics** for administrative activities like user management, application management and monitoring **controller performance**.
* Worked on deployment automation of **all micro-services** to pull image from the private **docker registry** and deploy to **Docker Swarm cluster** using Ansible.
* Experience in working with Docker service rolling updates and involved in implementation of **BLUE GREEN DEPLOYMENT** to attain zero downtime.

**Client: Locuz Enterprise Solutions Ltd – Hyderabad, India Aug 2012 – Feb 2015**

**Role: Systems/VMWare Engineer**

* Experience in Installing, upgrading, configuring Red Hat Linux 3.x, 4.x, 5.x, 6.x, 7 using kickstart servers and interactive installation.
* Expertise in package management using **Red Hat RPM**, **YUM** used in several Linux distributions such as Red Hat Enterprise Linux and configuring different servers (FTP, WEB, NFS, SAMBA, Send mail & Autofs).
* Experience in installing and configuring **SELinux** using command line interface and setting up permissions to the **files and directories** according to the requirement.
* Managing systems **routine backup**, scheduling jobs like **disabling and enabling cron jobs**, enabling system logging, network logging of servers for maintenance, Performance tuning and testing using.
* Configuration and maintenance of **WebLogic 8.1** with **JDBC, JNMS, JNMI, SNMP** and **J2EE** services, deploying applications in (**WAR, JAR, and EAR**) in domain and clustered environments. Performed **WebLogic Server 8.x** administration tasks such as installing, configuring, monitoring and performance tuning.
* Worked extensively on large variety of integration technologies Mule soft, Apache Camel, Jboss Fuse, Fuse Fabric8 and creating Active MQ with different topologies in Enterprise Integration.
* Experience in installing, configuring, and maintaining application Server like **WebSphere** and Web Servers like **Web Logic**, **Apache HTTP** and **Tomcat** on UNIX and Linux.
* Experience in **installation**, **configuration**, **performance tuning**, security, backup, recovery and upgrade/patches of Linux and UNIX servers using **Crontab.**
* Installation and administration of **VMware vSphere ESXi 6.5** environment Evaluated and implemented new hardware and software solutions that benefited architecture and operations implemented **VMware SRM** and troubleshot performance issues.
* Worked on **VMware** for Virtual server builds, increasing memory, CPU, adding disks, Maintenance on **ESX hosts VM** and server/storage migrations.
* Installing and upgrading virtual machine hardware version using **VMware tools**. Provisioning virtual machines and Deploying virtual machines from Templates by cloning. Creating and deleting of **VM's snapshots.**
* Assigning **LUNs** to **ESX/ ESXi** servers and creating data stores. Adding virtual machine memory and vcpu.