**Sundeep**

**Email id: sandeep16devops@gmail.com**

**Ph#: 949-484-6615**

**Title: Sr DevOps/Cloud Engineer GCP/AWS**

**Profiles Summary:**

* A Cloud enthusiastic team player having around 7**+ years of** Experience in IT industry as a **DevOps Engineer** with proven expertise in **Automation, Build/Release Engineering** and **Software development** involving cloud computing platforms like **Amazon Web Services (AWS), Azure** and **Google Cloud (GCP).**
* Expertise in Architecting and Implementing **Azure Service** Offering, such as **Azure cloud services, Azure storage, IIS, Azure Active Directory (AD), Azure Resource Manager (ARM), Azure Storage, Azure, Blob Storage, Azure VMs, SQL Database, Azure Functions, Azure Service Fabric, Azure Monitor, and Azure Service Bus**.
* Hands on experience on Backup and restore **Azure services** and in Design and configure **Azure Virtual Networks (VNets), subnets, Azure network settings, DHCP address blocks, DNS settings, security policies and routing. Azure cloud services, Blob storage, Active directory, Azure Service Bus, Cosmos DB.**
* Expertise in Azure **Scalability** and Azure **Availability** - Build VMs availability sets using the **Azure** portal to provide resiliency for **IaaS** based solution and **Virtual Machine Scale Sets (VMSS)** using **Azure Resource Manager (ARM)** to manage **network traffic.**
* Experience in Designing, Architecting and implementing scalable cloud-based web applications using **AWS** and **GCP**.
* Set up a **GCP** Firewall rules in order 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.
* Experience in migrating on-premises applications to **Azure** and configured **VNETs** and **subnets** as per the project requirement also performed **PowerShell scripting** to do **Patching,** **Imaging,** and **Deployments in** **Azure.**
* Knowledge of **Azure Site Recovery** and **Azure Backup** Installed and Configured the Azure Backup agent and virtual machine backup, Enabled **Azure Virtual machine** backup from the **Vault** and configured the **Azure Site Recovery (ASR).**
* Implemented a CI/CD pipeline using **Azure DevOps (VSTS, TFS)** in both cloud and on-premises with **GIT, MS Build, Docker, Maven** along with **Jenkins's** plugins.
* Experience in writing **Infrastructure as a code**(IaC) in **Terraform, Azure resource management, AWS Cloud formation.** Created reusable **Terraform modules** in both Azure and AWS cloud environments.
* Experience in Infrastructure Development and Operations involving **AWS Cloud Services, EC2, EBS, VPC, RDS, SES, ELB, Auto scaling, CloudFront, Cloud Formation, Elastic Cache, API Gateway, Route 53, Cloud Watch, SNS.**
* Expertise in building **CI/CD** on AWS environment using **AWS Code Commit**, **Code Build**, **Code Deploy** and **Code Pipeline** and experience in using **AWS CloudFormation, API Gateway,** and **AWS Lambda** in automation and securing the infrastructure on **AWS**.
* Experience in changing over existing **AWS** infrastructure to **Serverless** architecture **(AWS Lambda, AWS Kinesis)** through the creation of a **Serverless** Architecture using **AWS Lambda**, **API gateway**, **Route 53, S3 buckets**.
* Experience in **Kubernetes** to deploy **scale, load balance,** and **manage Docker containers** with multiple names spaced versions using **Helm charts**.
* Experience in managed Kubernetes services such as **Google Kubernetes engine (GKE), Azure Kubernetes and AWS Elastic container service.**
* Hands-on in using **OpenShift** for container orchestration with **Kubernetes,** container storage, automation, to enhance container platform multi-tenancy. Experience with **OpenShift, Kubernetes** architecture and design, troubleshooting issues and multi-regional deployment models and patterns for large-scale applications.
* Proficiency in using **Docker Hub, Docker Engine, Docker images, Docker Weave, Docker Compose, Docker Swarm,** and **Docker Registry** and used **containerization** to make applications platform when moved into different environments.
* Proficient in creating **Docker images** using **Docker File,** worked on **Docker container** snapshots, removing images, and managing **Docker volumes** and implemented **Docker** automation solution for **CI/CD** model.
* Extensive experience in installing, configuring and administering **Jenkins CI** tool on Linux machines and used **Jenkins Pipelines** to drive all Microservices builds out to the **Docker Registry** and then deployed to **Kubernetes,** Created Pods and managed using Kubernetes.
* Involvement in developing **APIs** using **Kubernetes** to manage and specify the copies of the containers to run the actual servers in the cloud environment. **Scheduled**, **deployed** and managed **container** replicas onto a **node cluster** using **Kubernetes** and deployed **Docker** containers through **Kubernetes** to manage the **microservices** using its **Nodes, Pods, ConfigMaps, Selectors, Services.**
* Expertise in configuring the **monitoring** and **alerting tools** according to the requirement like **Prometheus** and **Grafana,** setting up **alerts** and **deployed multiple dashboards** for individual applications in **Kubernetes.**
* Experience in **Configuration management tools** such as **Chef, Ansible,** and **Puppet.** Wrote Chef recipes and cookbooks in **Ruby** scripting.
* Hands-on experience in bootstrapping the nodes using **knife** and automated by **testing Chef Recipes, Cookbooks with test-kitchen and chef spec.** Refactored Chef and **Ops Works** in **AWS cloud** environment.
* Experienced in **Ansible** Tower, which provides an easy-to-use dashboard and role-based access control and in developing **Ansible** playbooks for managing the application/OS configuration files in GIT hub, integrating with Jenkins, and Verifying with Jenkins plugins, deploying the application in Linux environment.
* Experience in **Ansible** setup, managing hosts file, Using **YAML** linter, authoring various playbooks and custom modules with Ansible and Ansible Playbooks to automate in AWS services like Launching **EC2,** Provisioning **IAM,** Configuring **VPC,** **EBS,** Monitoring using **Cloud Watch** and **Cloud Trail.**
* Well versed in configuring an automated tool **Puppet** that included installation and configuration of **Puppet** and written **manifests** and modules for a **puppet** to do tasks like application **deployment**, **configuration** and create **filesystems**.
* Expert in Setting up Continuous Integration (CI) by configuring Build, Code, Deploy and test automation Jobs in **Jenkins** for different applications and in creation and deployment using Code ship to automate branch & project creation in Git using (Groovy language) in Jenkins's file and automating (using Ansible).
* Professional in **deploying** and configuring **Elasticsearch, Logstash, Kibana** (**ELK**) and AWS **Kinesis** for **log analytics** and skilled in **monitoring** servers using **Nagios**, **Splunk**, **AWS CloudWatch, Azure Monitor** and **ELK**.
* Skilled with **Python, Bash/Shell, PowerShell, Ruby**, **Perl**, **PowerShell**, **JSON**, **YAML**, **Groovy**. Developed **Shell** and **Python** Scripts used to automate day to day administrative tasks and automation of the **build** and **release process**.
* A highly motivated, energetic individual, a team player with excellent communication and inter-personal skills.
* Strong verbal and written Communications Skills with the ability to effectively distill critical details into actionable highlights.

**Technical Skills:**

|  |  |
| --- | --- |
| **DevOps Tools** | Chef, Ansible, Terraform, Puppet, Docker, Kubernetes, SonarQube, Jenkins, AWS, Azure. |
| **Public & Private Cloud Technologies** | Microsoft Azure, Amazon Web Services (AWS), Google Cloud Platform (GCP), OpenStack |
| **Web Servers** | Apache Tomcat, Nginx, WebSphere, WebLogic, JBoss, Samba, SQL Server. |
| **CI/CD Tools** | Jenkins, Bamboo, GitLab CI, Travis CI, Hudson, VSTS, TFS. |
| **Build & Testing Tools** | Maven, Ant, Gradle, Selenium, JUnit, NUnit, xUnit |
| **Version Controls** | Git, Subversion, GitHub, Bitbucket. |
| **IDE Tools** | MS Visual Studio, NetBeans, Eclipse, IntelliJ IDEA, PyCharm, Oracle SQL Developer |
| **Web Technologies** | HTML5, CSS3, Bootstrap, JSON, jQuery, JavaScript, PHP, XML |
| **Performing/Monitoring & Bug Tracking Tools** | ELK, Nagios, CloudWatch, Azure Monitor, New Relic, Splunk, Grafana, Prometheus, Confluence, Jira. VMware ESXI, Vagrant |
| **Scripting/Programming** | Java, Python, Shell, Bash, Ruby, Groovy, PowerShell, YAML, |
| **Servers/ Monitoring** | Apache Tomcat, Jboss, WebLogic, Ngnix, Splunk, Nagios. |
| **Data Bases** | SQL Server, Oracle, Mongo DB, Oracle DB |
| **Operating Systems** | RHEL, CentOS, Ubuntu, Fedora, Debian, Solaris, Windows, MacOS |

**Professional Experience:**

**Client:** **BOFA, Pennington, NJ. Feb 2020 – Feb 2021**

**Role: Sr DevOps/Cloud Engineer GCP/AWS**

**Responsibilities:**

* Created recommendations on how to duplicate a subset of on-premise machines to the **Azure Infrastructure as a Service** (IAAS) offering which will be used for disaster recovery. This analysis included the specifics to synchronize on-premise data with SQL Server and SharePoint instances hosted in VMs.
* Worked on Azure Site Recovery and Azure Backup- Deployed Instances on Azure environments and in Data centers and migrating to Azure using **Azure Site Recovery** and collecting data from all Azure Resources using Log Analytics and analyzed the data to resolve issues.
* Worked on google cloud platform (GCP) services like compute engine, cloud load balancing, cloud storage, cloud SQL, stack driver monitoring and cloud deployment manager.
* Setup GCP Firewall rules 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.
* Configured Azure Multi-Factor Authentication (MFA) as a part of Azure AD Premium to securely authenticate users and worked on creating custom Azure templates for quick deployments and advanced **PowerShell** scripting. Deployed Azure SQL DB with GEO Replication, Azure SQL DB Sync to standby database in another region & Fail over configuration.
* Worked on **Serverless services**, created and configured **HTTP** Triggers in the **Azure Functions** with application insights for **monitoring** and **performing** load testing on the applications using the **Visual Studio Team Services** (**VSTS)** alsocalled as **Azure DevOps Services**.
* Created **Azure** Automation Assets, **Graphical runbook**, **PowerShell runbook** that will **automate** specific tasks, deployed **Azure AD** Connect, configuring **Active Directory Federation Service (AD FS)** authentication flow, **ADFS** installation using **Azure AD Connect**, and involved in administrative tasks that include **Build**, **Design**, **Deploy** of **Azure environment**.
* Configure **Continuous Integration** from source control, setting up build definition within **Visual Studio Team Services (VSTS)** and configure **continuous delivery** to automate the deployment of **ASP.NET MVC** applications to **Azure web apps** and managed **Azure Active Directory**, **Office 365** and applied upgrades on a regular basis.
* Implemented a **CI/CD** pipeline with **Docker**, **Jenkins** (**TFS** Plugin installed), **Team Foundation Server (TFS)**, **GitHub** and **Azure Container Service**, whenever a new **TFS/GitHub branch** gets started, **Jenkins**, our **Continuous Integration (CI)** server, automatically attempts to build a new **Docker container** from it.
* Worked with **Terraform** Templates to automate the Azure Iaas virtual machines using terraform modules and deployed virtual machine scale sets in production environment.
* Managed **Azure** Infrastructure **Azure Web Roles**, **Worker Roles**, **VM Role, Azure SQL**, **Azure Storage**, **Azure AD** Licenses, **Virtual Machine Backup** and Recover from a Recovery Services **Vault** using **Azure PowerShell** and **Azure Portal**.
* Managed AWS infrastructure as code (IaaS) using Terraform. Expertise in writing new python scripts in order to support new functionality in Terraform. Provisioned the highly available EC2 Instances using Terraform and cloud formation and Setting up the build and deployment automation for Terraform scripts using Jenkins
* Designed AWS Cloud Formation templates to create custom sized VPC, to set up IAM policies for users, subnets, NAT to ensure successful deployment of Web applications, database templates and Security groups
* Managed Docker orchestration and Docker containerization using Kubernetes. Used Kubernetes to orchestrate the deployment, scaling, and management of Docker Containers
* Created and deployed Kubernetes pod definitions, tags, labels, multi-pod container replication. Managed multiple Kubernetes pod containers scaling, and auto-scaling.
* Deployed pods using Replication Controllers by interacting with Kubernetes API server defining through declarative YAML files.
* Implementation of new tools such as Kubernetes with Docker to assist with auto-scaling and continuous integration (CI) and Upload a Docker image to the registry so the service is deployable through Kubernetes. Use the Kubernetes dashboard to monitor and manage the services.
* Using **Ansible** created multiple **playbooks** for machine creations and **SQL server, cluster server** and my SQL installations.
* Used **Ansible** to Setup/teardown of **ELK** stack (Elasticsearch**, Log stash, Kibana)** and troubleshoot the build issues with **ELK** and work towards the solution.
* Written **Ansible handlers** with multiple tasks to trigger multiple handlers and to decouple handlers from their names, making it easier to share handlers among Playbooks and Roles.
* Managed **Kubernetes** charts using Helm, Created reproducible builds of the Kubernetes applications, managed Kubernetes manifest files and Managed releases of Helm packages.
* Implemented Docker -maven-plugin in **Maven** pom.xml files to build Docker images for all microservices and later used Docker file to build the Docker images from the **Java** jar files.
* Designed, wrote and maintained systems in **Python scripting** for administering **GIT**, by using **Jenkins** as a full cycle **continuous delivery** tool involving package creation, distribution, and deployment onto **Tomcat application** servers via shell scripts embedded into **Jenkins** jobs.
* Maintained **Artifacts** in binary **repositories** using **JFrog** Artifactory and pushed **new Artifacts** by configuring the **Jenkins** project **Jenkins Artifactory plugin**.
* Involved in **Jira** as defect **tracking** system and configure various workflows, customizations, and **plugins** for **Jira bug/issue tracker** integrated **Jenkins** with **Jira**, **GitHub**.
* Worked with **Nagios** for **Azure Active Directory** & **LDAP** and Data consolidation for **LDAP users.** **Monitored** system performance using **Nagios**, maintained **Nagios** servers and added new **services & servers.**
* Built and **managed** a highly available **monitoring infrastructure** to monitor different **application servers** like **JBoss, Apache Tomcat** and its components using **Nagios.**

**Environment:** AWS, Azure, PCF, Office 365, Terraform, Maven, GCP, Jenkins, Ansible, Azure ARM, Azure AD, Azure Site Recovery, Kubernetes, Python, Ruby, XML, Shell Scripting, PowerShell, Nexus, JFrog Artifactory, Jenkins, Git, Jira, GitHub, Ansible, Docker, Windows Server, TFS, VSTS, LDAP, Nagios.

**Client: Pimco, Newport Beach, CA. Dec 2018 – Jan 2020**

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

**Responsibilities:**

* Provisioned and administered **EC2** instances and configuring **EBS, Simple Storage(S3)** cross region replication, **Elastic Load Balancer, configure Auto scaling, setting up CloudWatch alarms, Virtual Private Cloud (VPC)**, mapping with multi-AZ VPC instances and **RDS**, based on architecture.
* Worked on Amazon EC2 setting up instances, **virtual private cloud** (VPCs), and security groups and created AWS Route53 to route traffic between different regions and used BOTO3 and Fabric for launching and deploying instances in AWS.
* Configured **Amazon S3, Elastic Load Balancing, IAM** and **Security Groups** in Public and Private Subnets in VPC, created storage cached and storage volume gateways to store data and other services in the AWS.
* Architected and configured a virtual data center in the AWS cloud to support Enterprise Data Warehouse hosting including Virtual Private Cloud (VPC), Public and Private Subnets, Security Groups and Route Tables.
* Used Security Groups, Network ACLs, Internet Gateways, NAT instances and Route tables to ensure a secure zone for organizations in AWS public cloud.
* Worked on migration services like AWS **Server Migration Service** (SMS) to migrate on-premises workloads to AWS in easier and faster way using Rehost "lift and shift" methodology and AWS **Database Migration Service** (DMS), AWS Snowball to transfer large amounts of data and Amazon S3 Transfer Acceleration.
* Worked in an **Agile** development team to deliver an end-to-end continuous integration/continuous delivery product in an open-source environment using tools like Puppet & Chef.
* Written **Terraform** scripts to automate **AWS services** which include **ELB**, CloudFront distribution, RDS, **EC2**, database security groups, **Route 53**, VPC, Subnets, Security Groups, and **S3** Bucket and converted existing AWS infrastructure to **AWS Lambda** deployed via **Terraform** and **AWS CloudFormation**.
* Implemented **AWS Elastic Container Service (ECS)** scheduler to automate application deployment in the **cloud** using **Docker** Automation techniques.
* Implemented **Docker** - **maven**-plugin in **Maven pom.xml** files to build **Docker images** for all **microservices** and later used **Docker File** to build the **Docker** images from the **Java jar files** also Created **Docker** images using a **Docker File**, worked on **Docker container** snapshots, **removing** images and managing **Docker volumes.**
* Wrote **Python** scripts using the **Boto3** library to automatically sign up the instances in **AWS EC2** and **OpsWorks stacks** and integrated with **Auto Scaling** with configured **AMI’s.**
* Used **Git** version control to manage the source code and integrating **Git** with **Jenkins** to support build **automation** and integrated with **Jira** to **monitor** the commits.
* Worked on **AWS costs** by writing the **Ansible Playbooks** for auto **Start/Stop** of AWS resources at the time of the day by triggering it from **Jenkins** and knowledge on **RHEL** on **Ansible Playbooks,** **Modules** and **Roles.**
* Designed various **Jenkins** jobs to continuously integrate the processes and executed **CI/CD** pipeline using **Jenkins,** **Ansible Playbooks** and **Ansible Tower.**
* Deployed and configured **Chef** server including bootstrapping of **Chef client** nodes for provisioning and experience in developing **Chef recipes** through **Knife** command-line tool, customized Ruby scripts to create Chef cookbooks to manage systems configuration on **Chef Server** to initiate **Chef Nodes**.
* AWS Cloud management using **Chef Automation** and Automated the cloud deployments using chef, python and AWS Cloud Formation Templates.
* Used **Jenkins** and pipelines to drive all micro services builds out to the **Docker-registry** and then deployed to Kubernetes, Created Pods and managed using **Kubernetes.**
* Set up and managed **ELK (Elasticsearch, Logstash & Kibana)** Stack to collect, search and analyze **log files** across servers, log monitoring and created geo-mapping visualizations using Kibana in integration with **AWS CloudWatch** and **Lambda**. Evaluated system logs using ELK software stack.

**Environment:** AWS, Terraform, Chef, Ansible, Docker, Jenkins, Git, Jira, Jenkins, Kubernetes, Maven, Nagios, ELK, Java, SonarQube, Shell, Bash, Python, DynamoDB, Cassandra.

**Client: Chewy.com, Dania Beach, FL. Jan 2016 – Oct 2018**

**Role: DevOps/Build/Release Engineer**

**Responsibilities:**

* Participated in the release cycle of the product which involved environments like development QA UAT and Production.
* Documented all the process of **CI/CD Pipeline** to make sure that all the steps are completed successfully and for feature reference of Dev teams.
* Installed and configured an automated tool **Puppet** that included the installation and configuration of the **Puppet** master, agent nodes and an admin control workstation.
* Deployed **Puppet, Puppet** Dashboard, and **Puppet DB** for configuration management to existing infrastructure.
* Created all the **Puppet boxes** by using the kickstart file system and involved in patching work for stopping and starting the clients.
* Created **Cloud Formation** template stacks to automate for building new VPCs using **JSON** files.
* Created quality gates in **SonarQube** dashboard and enforced in the pipelines to fail the builds when conditions not met.
* Build Scripts using **MAVEN and ANT** build tools in Jenkins to move from one environment to other environments.
* Created, expanded, and configured automation in **Maven/Ant** to automatically build, package, and deploy Java applications to multiple development and testing environments.
* Installed and Configured the **Nexus repository** manager for sharing the artifacts within the company.
* Integrated **Docker** container orchestration framework using **Kubernetes** by creating pods, config Maps and deployments.
* Used **Nagios** as a monitoring tool to identify and resolve infrastructure problems before they affect critical processes and worked on **Nagios** event handlers in case of an automatic restart of failed applications and services.
* Responsible for tagging and maintain code on version control **GIT** and Creating branches and tags on **GIT** repository and provided branches access permission to dev team.
* Worked on integrating **GIT** into the continuous Integration (CI) environment along with **Jenkins.**
* Created branches **in GIT** implementing parallel development process.
* Written **Shell scripts** to apply the Integration label to all the files which needs manual labelling of files.
* Build and release software baselines, code merges, branch and label creation and interfaced between development and infrastructure.
* Implementing a Continuous Delivery framework using **Jenkins** pipelines.
* Installed **Jenkins on a Linux machine** and created a master and slave configuration to implement multiple parallel builds through a build farm.
* Prepared the Release plan and Coordination activities from Release Management.
* Active participant in scrum meetings, reporting the progress and maintain effective communication with each team member and manager.

**Environment:** CI/CD, Jenkins, Maven, Puppet, Ant, Docker, Nexus, SonarQube, GIt, Shell, Bash, Python, Window and Linux.

**Company: Benecode Solutions Pvt Ltd., India. Jan 2015 – Dec 2015**

**Role: Build/Release Engineer**

**Responsibilities:**

* 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.
* Used **Chef** as Configuration management tool, to automate repetitive tasks, quickly deploys critical applications, and proactively manages change.
* Worked on construction of **Ansible** manifest files to install tomcat on instances and to manage configuration files for multiple applications.
* Created **puppet** manifests, profiles and roles module to automate system operations.
* Worked on **AWS EC2, IAM, S3, EBS, Elastic Load balancer (ELB),** auto scaling group services.
* Used **Maven** as build tool for the development of build artifacts on the source code.
* Maintain and track inventory using **Jenkins** and set alerts when the servers are full and need attention.
* Used **GIT** as source code repositories, Maintained and Administered **GITHUB** Source Code Tool.
* Created and maintained the **Python** deployment scripts for Web Sphere web application.
* Wrote **Python/Power Shell scripts** for automating time-consuming specialty needs for external clients.
* Worked on **Docker Hub**, creating Docker images and handling multiple images primarily for middleware installations.
* Involved in configuring the Prometheus alerts from the **Kubernetes** cluster via email-notifications and web-hook mechanisms.
* Supporting large-scale web applications, we indexed database queries using **MY SQL server** by writing **SQL** queries. Responsible for the Plugin Management, User Management, Build/Deploy Pipeline Setup and End-End Job Setup of all the projects.
* Responsible for day to day Build and deployments in pre-production and production environments.

**Environment:** Chef, Ansible, Puppet, AWS (EC2, IAM, S3, EBS, Elastic Load balancer (ELB)), Maven, Jenkins, GIT, GITHUB, Python, Shell Script, Docker, Kubernetes, SQL, and Windows.

**Company: Value Labs, India Jan 2014 – Dec 2014**

**Role: Jr.Linux System Administrator (Intern)**

**Responsibilities:**

* I worked in team of five System Administrators and five System Analysts to perform daily tasks required for supporting an enterprise Hospital Network with over 1500 users, 150 servers, and 10 remote Clinics.
* Maintained 10 Linux Servers which provided security and services for other Windows machines on the network such as: DNS, NAT, Anti-Virus/Spam, and Firewall via iptables.
* Managed backups for Windows, Linux, PACS, and File Servers to EMC Clarion SAN device.
* Provisioned, audited, and administered server logs, backups, and Active Directory objects including remote forests.
* Researched and procured BlackBerry Enterprise Server. Assisted with migration from Juniper based core and wireless network to 100 Cisco based.
* Worked with application owners and developers to understand complex details about how their applications worked to ensure minimal downtime during migrations.
* Performed, documented, and automated the audit of lower-tier server environments, and helped to ensure consistency of configuration files going forward using Puppet based on many factors.
* Helped to automate deployment and testing processes using platforms such as Jenkins/Maven

**References**: Will be provided upon request.