A picture containing text, sign

Description automatically generatedA white sign with black text

Description automatically generated with low confidence**ASHISH**

DevOps/Cloud Engineer

Phone – 714 653 8183



**Professional Summary**

Over 8 years of IT Experience as **System Administrator, Cloud Engineer, Python Developer, DevOps Engineering, Build and Release Management** and Production Support of various applications using **Python** on **Red Hat Enterprise Linux, CentOS, Ubuntu** and have skilled experience on Windows Server on various hardware platforms and **AWS/Azure/GCP** cloud services.

* Experience in Design, development, implementation, deployment, testing, and maintenance using Software Development Life Cycle (SDLC) methodologies like **Agile** and **SCRUM**.
* Strong experience working in Big data, Bigdata Integration and Analytics technologies, including **Apache Spark**, **Hadoop,** **Kafka** & S**torm**.
* Design and Developed**Web Services, RESTful API**for Mobile Apps using **Python Django-REST** and**Flask Frameworks** on **Nginx.**
* Experience working on several **Docker** components like Docker Engine, Docker-Hub, Docker-Compose and Docker Registry.
* Implemented **Terraform** modules for the deployment of applications across multiple cloud providers
* 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**.
* Working Experience on **Azure Cloud Services, Azure Storage, SQL Azure** and in different PaaS Solutions with Web, and worker Roles and Azure Web Apps.
* Implementing & maintaining branching and build/release strategies utilizing GIT, Bit-bucket, **Azure repos**. Experience in deploying resources using **Azure Resource Management (ARM)** templates.
* Designed distributed private cloud system solutions using **Kubernetes** (Docker) on CoreOS. Automated various infrastructure activities like Continuous Deployment, Application Server setup, Stack monitoring using **Ansible playbooks**, and has Integrated Ansible with **Jenkins**.
* Experienced in Working with **AWS** cloud platform and its features which includes **EC2, VPC, AMI, IAM, EBS, SNS, RDS, Cloud Watch, Cloud Trail, CloudFront, Cloud Formation, IAM, Beanstalk, Elasticsearch, Kibana, ECS, EKS, EMR, DynamoDB, Autoscaling, S3, and Route53**.
* Knowledge about setting up **Python** **REST API** Framework using **Django**.
* Experience in working with **Python** ORM Libraries, including **Django** ORM.
* Well versed in the Design and development of presentation layer for web applications using **HTML**, **CSS** and J**avaScript**.
* Good Knowledge in implementation of Python best Practices (**PEP-8**).
* Proficient in using editors **Eclipse**, **PyCharm**, **Notepad++** and **Sublime Text** while developing different applications
* Proficient with **SQL** databases like **Oracle,** **MySQL,** **DB2** and expertise in **NoSQ**L database like **MongoDB**.
* Create clusters in **Google Cloud** and manage the clusters using **Kubernetes**. Using Jenkins to deploy code to Google Cloud, create new namespaces, creating docker images and pushing them to **container registry** of **Google Cloud**.
* Build and configure a virtual data center in the **Google cloud platform**to support Enterprise Data Warehouse hosting including **Virtual Private Cloud (VPC)**, Public and Private Subnets, Security Groups, Route Tables, **Google Cloud Load Balancing.**
* Proficient in working with version control systems – **GIT**
* Worked on various operating systems like **Windows**, **UNIX** and **LINUX**.
* Excellent experience with Python development under Windows, UNIX and Linux OS (**Ubuntu**, **RedHat** Linux)

**TECHNICAL SKILLS:**

|  |  |
| --- | --- |
| **Programming Languages** | Python, C, Shell Script, SQL, Perl. |
| **Cloud Technologies** | Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Open stack, Open Shift |
| **Databases** | Amazon Aurora, RDS, Redshift, DynamoDB, PostgreSQL |
| **Web Servers** | Apache, Nginx, Tomcat, Web Logic, WebSphere, LAMP. |
| **Containerization/Virtualization** | Docker, Kubernetes, ECS, VMware vSphere, PCF, EKS. |
| **Ticketing/Testing Tools** | Service Now, Asana, Jira, Salesforce, Postman, Junit, Mockito, Selenium. |
| **Operating Systems** | Windows, Sun Solaris2.x, RHEL, UNIX, LINUX, Mac OS, Centos, Ubuntu. |
| **Network Protocols** | TCP/IP, Subnets, FTP, SMTP, UDP, ICMP, HTTP/HTTPS. |
| **Repository Management Tools** | Nexus, J-Frog, Artifactory, Docker Hub, Amazon ECR |
| **Monitoring Tools** | Cloud Watch, Cloud Trail, Nagios, Splunk, ELK, AppDynamics, Dynatrace, Datadog. |
| **Build/Automation Tools** | Ansible, Chef, Puppet, Salt Stack, Ant, Maven, Gradle, Jenkins, Bamboo, Octopus, U-Deploy. |
| **Provisioning Tools** | Terraform, Cloud Formation, ARM Templates |
| **Version Control Tools** | GIT, GITHUB, Bit-Bucket, Azure Repos, Code Commit |
| **Web Technologies** | HTML, DHTML, XHTML, XML, J-Query |
| **Project Management** | Agile, Scrum |
| **Framework** | Django, Flask |
| **IDE Tools** | Android Studio, Microsoft Visual Studio, Sublime Text, Cloud9. |

**DevOps Engineer  Client: ReinforceAi**

**Responsibilities: Jan 2020 – Present**

* Migrated application (**Cross Account**) includes migration of code repository (**code** **commit),** **DynamoDB**, **Sage-maker**, **Cognito**, and **S3** artifacts.
* Deployed and launched the frontend application by integrated with CloudFront distribution and the **Alexa skill** using **lambda** and **Raspberry ISO** image in the production environment.
* Designed and Deployed **IoT things** in the **IoT Core** using **Cloud9** as the IDE. Based on the attributes, integrated **SNS** with the things for the real-time updates.
* Established Intercommunication between things using **IoT Greengrass** for local processing of telemetry data.
* Wrote automation scripts in Python for Extracting Data from **JSON** and **XML** files.
* Created **CI/CD** pipelines and setup auto-trigger, auto build and auto-deployment with the CI/CD tool like Bamboo.
* Used **ETL** to develop jobs for extracting, cleaning, transforming and loading data into data warehouses.
* Resolved update, merge and password authentication issues in **Jenkins** and **JIRA.**
* Installed, Configured, Managed Monitoring Tools such as **Splunk**, **Nagios** and **CloudWatch** for Resource Monitoring/Network Monitoring/Log Trace Monitoring.
* Implemented Docker -**maven**-plugin in and Maven POM.XML to build Docker images for all microservices and later used Docker files to build the Docker images from the java, jar files.
* Automated the frontends platform into highly scalable, consistent, repeatable infrastructure using a high degree of automation using **Chef**, **Jenkins**, and **Cloud Formation.**
* Real-time streaming the data using Spark with **SQS**. Responsible for handling Streaming data from web server console logs.
* Written bash and python scripts, integrating **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.
* Created Azure Automation Assets, Graphical runbook, **PowerShell** runbook that will automate specific tasks, deployed **Azure-AD Connect**, configuring Active-Directory Federation Service (ADFS) authentication flow, ADFS installation using Azure AD Connect, and involved in administrative tasks that include Build, Design, Deploy of Azure environment.
* Implemented a CI/CD pipeline, 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.
* 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**plugins.
* Experience in designing, configuring, and deploying solutions on Microsoft Azure using **ARM** Templates, Azure PowerShell Module and, Azure CLI focusing on high - availability and auto-scaling.
* Working Experience on **Azure Databricks** cloud to organizing the data into notebooks and making it easy to visualize data using dashboards.
* Deploying windows **Kubernetes (K8s)** cluster with **Azure Container Service (ACS)** from **Azure CLI** and Utilized **Kubernetes** and **Docker** for the runtime environment of the **CI/CD** system to **build**, **test** and Octopus Deploy.
* Written Terraform modules for automating VPC's and **Azure VM** Instances, modules for creating **VPC** and **VPN** connection from Data Center to the production environment and cross-account VPC peering.
* Created **Azure VM** instances and implemented large multi-node Hadoop clusters in AWS cloud from scratch using automated scripts such as Terraform.

·

**DevOps Engineer. Client: Wells Fargo**

**Responsibilities: Feb 2019 – December 2019**

* Build and configure a virtual data-center in the **Google cloud platform**to support Enterprise Data Warehouse hosting, including **Virtual Private Cloud (VPC)**, Public and Private Subnets, Security Groups, Route Tables, **Google Cloud Load Balancing.**
* Create clusters in **Google Cloud** and manage the clusters using **Kubernetes**(k8s). Used Jenkins to deploy code to Google Cloud. Created new namespaces, creating Docker images and pushing them to **the container registry** of **Google Cloud**.
* Configure, monitor, and automate Google Cloud Services and be involved in deploying the content cloud platform using **Google compute engine, Google storage buckets**.
* 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.
* Involved in maintaining the user accounts (**IAM**), **RDS**, **S3**, **Route 53**, AWS Lambda, VPC, RDS, **Dynamo DB**, SES, SQS and SNS EMR services in **AWS cloud**.
* Leveraged AWS cloud services such as **EMR**, auto-scaling and **VPC (Virtual Private Cloud)** to build secure, highly scalable, and flexible systems that handled expected and unexpected load bursts and can quickly evolve during development iterations.
* Worked on Auto-scaling **Cloud Watch**(monitoring), **AWS Elastic Beanstalk**(app deployments), **AWS Lambda**, **Amazon S3**(storage) and **Amazon EBS**(persistent disk storage) and **Dynamo DB**.
* Built **S3**bucketsand managed policies for S3 buckets and used **S3 bucket**and **Glacier**for storage and backup on **AWS**.
* Configured **Kinesis Shards**for optimal throughput **in Kinesis Streams** for **Spark Streaming** Applications on **AWS**.
* 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** automation and securing the infrastructure on **AWS**.
* Created an **AWS RDS** **Aurora DB** cluster and connected it to the database through an Amazon RDS **Aurora DB** Instance using the **Amazon RDS** Console.
* Utilized **Jenkins**master/slave architecture to distribute builds on nodes, and trigger **Jenkins**job to build the artifacts using **Maven**, deployed the **Terraform**templates to create the stack.
* Implemented a **CI**/**CD** pipeline with **Docker, EKS,** **Jenkins,** and **GitHub** by virtualizing the servers using Docker for the Dev and Test environments by achieving needs through configuring automation using Containerization.
* Implemented **AWS** **Code Pipeline**and Created **Cloud formation** **JSON**templates in **Terraform**for infrastructure as code.
* Extensively involved in **infrastructure as code**, execution plans, resource graph and change automation using **Terraform.**
* Experience on infrastructure as code with **Hashicorp Terraform** tool by operating AWS infrastructure through Terraform and **Terraform Enterprise** (TFE) by writing Terraform scripts, modules etc.
* Converted existing **terraform modules** that had version conflicts to utilize cloud formation during **Terraform**deployments to enable more control or missing capabilities.
* Used **Python** to automate provision by **Terraform**for tasks such as encrypting **EBS**volumes backing **AMIs,**and scheduling Lambda functions for routine **AWS**tasks and stored result in **Dynamo DB**.
* Worked with container-based deployments using **Docker**, working with **Docker** images, **Docker** hub and **Docker** registries, installation and configuring **Kubernetes** and clustering them.
* 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.
* Managed **Kubernetes** charts using **Helm**, Created reproducible builds of the **Kubernetes** applications, managed Kubernetes manifest files and Managed releases of Helm packages.
* In-depth Knowledge and hands-on experience of **Kubernetes**, including setting up a **Kubernetes cluster**, managing clusters, creating and managing pods, deployments, application management, rolling updates, exposing using services & routes, persistent volumes and persistent volume claims.
* Automated the configuration management and deployments using **Ansible** playbooks. Furthermore, **Yaml**for resource declaration, and creating roles and updating Playbooks to provision servers using **Ansible**.
* Installing, Configured, and management in **Ansible** Centralized Server and creating the playbooks to support various middleware application servers and configuring the **Ansible** tower as a configuration management tool to automate repetitive tasks.
* Created AWS Infrastructure monitoring through **Datadog** and application performance monitoring through **App Dynamics.**
* **Pager-duty** integrations with **App dynamics, Splunk, Wave-front, Cloud watch**, and slack through API's and define escalation policies and schedules.
* Extensive programming experience in **Python, PowerShell,**and **Bash**scripting. Wrote bash and Shell scripts for auto-launch and web server configurations.
* Designed workflows in **Atlassian JIRA** to deal with issues and maintained all the user stories for tracking as per Agile style.

**DevOps Engineer Client: Merck**

**Responsibilities: January 2017 – Nov 2018**

* Developed security policies and processes. Developed views and templates with **Python** and **Django's** view controller and templating language to create a user-friendly Website interface to migrate to AWS.
* 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.
* 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 PowerShellscripting. 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.
* Implemented a CI/CD pipeline, 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.
* 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**plugins.
* 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.
* Architect & implement medium to large scale **BI** solutions on Azure using Azure Data Platform services **(Azure Data Lake,** Data Factory, Data Lake Analytics, Stream Analytics, Azure SQL DW, HDInsight/**Databricks**, NoSQL DB)
* Deployed the Azure VM’s, **Azure Storage (Blob, Files**) using **ARM** Templates and PowerShell scripts.
* Experience in designing, configuring, and deploying solutions on Microsoft Azure using **ARM** Templates, Azure PowerShell Module and, Azure CLI focusing on high - availability and auto-scaling.
* Working Experience on **Azure Databricks** cloud to organizing the data into notebooks and making it easy to visualize data using dashboards.
* Deploying windows **Kubernetes (K8s)** cluster with **Azure Container Service (ACS)** from **Azure CLI** and Utilized **Kubernetes** and **Docker** for the runtime environment of the **CI/CD** system to **build**, **test** and Octopus Deploy.
* 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.
* Developed **Ansible** playbooks for **Splunk** in cloud environments with auto scaling for task force initiatives requiring big data analysis.
* Created **Docker**slave Nodes using custom Docker images and pulled them to **ACR**and worked on all major components of Docker such as **Docker Daemon**, **Docker Hub**, **Docker Images**, **Docker registry**.

**System Operation Engineer Client: CA Technologies**

**Responsibilities: June 2015 – December 2016**

* Developed and designed an **API** (RESTful Web Service). Used the **Python** language to develop web-based data retrieval systems.
* Involved in doing **AGILE** (SCRUM) practices and planning of sprint, attending daily agile (SCRUM) meetings and **SPRINT** retrospective meetings to produce quality deliverables within time.
* Worked on Restful web services, which enforced a stateless client-server and support **JSON** few changes from **SOAP** to **RESTFUL** Technology. Involved in the detailed analysis based on the requirement documents.
* Designed and maintained databases using Python and developed Python based **API** (RESTful Web Service) using **Flask**, **SQL-Alchemy** and **PostgreSQL**.
* Created complex dynamic **HTML** UI using **jQuery**. Automated Regression analysis for determining fund returns based on index returns (**Python/Excel**). Worked on development of **SQ**L and stored procedures, trigger and function on **MYSQL**.
* Developed views and templates with Python and **Django**'s view controller and templating language to create a user-friendly website interface.
* Involved in writing **SQL** queries implementing functions, triggers, cursors, object types, sequences, indexes etc. and Developed and tested many features for dashboard using Python, **Bootstrap**, **CSS**.
* Performed advanced procedures like text analytics and processing, using the in-memory computing capabilities of **Spark** using.
* Responsible for delivering datasets from **Snowflake** to One Lake Data Warehouse and building **CI/CD** pipeline using Jenkins and **AWS lambda** and Importing data from **DynamoDB** to **Redshift** Batches using Amazon Batch using **TWS** scheduler.
* Created and managed all hosted or local repositories through Source Tree's simple interface of GIT client, collaborated with GIT command lines and Stash.
* Developed and reviewed **SQL queries** using joins clauses (inner, left, right) in **Tableau** Desktop to validate static and dynamic data for data validation.
* Designed and developed components using **Python** with **Django** framework. Implemented code in **Python** to retrieve and manipulate data.
* Involved in developing the enterprise social network application using Python, Twisted, and **Cassandra** and responsible for setting up Python **REST API** framework and spring framework using **Django**.
* Maintained and developed complex **SQL** queries, stored procedures, views, functions and reports that meet customer requirements using Microsoft **SQL** **Server 2008 R2**.

**Python Developer Protel Infoserve Pvt. Ltd**

**Responsibilities: Feb 2014 – May 2015**

* Understanding the client requirement and developing the code using **Python** **scripting**.
* Developing the code by using **Python modules**.
* Involved in developing in Login, Register information module.
* Involved in Design and implementation of project.
* Involved in creating page templates required for the project.
* Interacting with team members on technical programming and solving production support issues (such as bug fixes, **queries** etc.).
* Prepared documentation for the generated reports. Developed the customer complaints application using **Django** Framework, which includes and Python code.
* Profiled python code for optimization and memory management.
* Implemented locking mechanisms using **multi-threading** functionality.
* Involved in the development of main modules like **CSV** import, bulk content upload.
* Writing unit test cases wherever required on the application.
* Performing code optimization.