

**Ashutosh Teknur**

**Professional Summary:**

**1.5 Years of Industry experience in the field of DevOps (Software Engineer).**

**Summary:**

* Kubernetes to orchestrate the deployment, scaling and management of Docker Containers.
* Ingress rule created for accessing service with the application load balancer.
* Managing container replicas onto a node using Kubernetes and experienced in creating Kubernetes clusters work with Helm charts running on the same cluster resources.
* Knowledge with Helm charts to manage and release of helm packages.
* Terraform used to automate the provisioning of infrastructure.
* Good hands-on knowledge of Linux Administration Centos7, Ubuntu
* Expertise in creating Jenkins jobs.
* Version Control/Source code management using Git, GitHub.
* Knowledge of Linux and Scripting languages (Bash, PowerShell, Python).
* Good understanding of Infrastructure as Code (Programmable Infrastructure), and how you can achieve that by using Ansible
* Good hands-on knowledge of Source Code Management (Version Control System) tools like Git, GitLab.
* Experience with automation/ integration tools like Jenkins.
* Creation of the Azure DevOps pipeline for CI/CD.
* Created Azure PowerShell scripts to support deployments and automate the azure infrastructure.
* Created and configured the resource groups, virtual machines and storage accounts on Azure.
* Experience with Azure PaaS, Storage, IaaS, Resource Groups, Cosmos DB, Stream Analytics, DPS, IOT and other services.
* Experience with the Windows Azure cloud platform (IaaS and PaaS) and Azure CLI.
* Knowledge of major cloud service providers, like AWS, GCP and Azure etc.
* Knowledge on Azure ARM-templates
* Knowledge of software containerization platforms like Docker and container orchestration tools like Kubernetes.

**Key Technical Skills:**

|  |  |
| --- | --- |
| Continuous Integration/Delivery | Jenkins |
| Scripting Language | Shell Script, Python, PowerShell |
| Configuration Management | Ansible |
| Container Management | Docker, Kubernetes |
| Container Orchestration Tools | Kubernetes |
| Source Control | GitHub, GitLab |
| Cloud Platforms | AWS, Google Cloud, Azure |
| Artifactory Management | JFrog |
| Code Review | Gerrit |

**Education:**

|  |  |  |  |
| --- | --- | --- | --- |
| Degree | Board / University | Score | Year of passing |
| B.E (Computer Science) | Pune University | 62% | 2018 |

**Professional Experience:**

**Project: 1**

|  |  |
| --- | --- |
| Project Name | EdgeCore |
| Project Description | Infrastructure automation using ansible:   * GitLab * Jenkins * Gerrit * JFrog |
| Responsibility | Created CI/CD using below tools:  GitLab for the repository  Jenkins for the CI  Gerrit for the Code review  JFrog for Artifactory  AWS Cloud:   * EC2 for the servers * IAM Role for the RBAC (Role Base Access Control) * Creating the EC2 server and managing the servers * Attaching Elastic IP’s to EC2 servers * Attaching Volumes to EC2 servers * Attaching Security groups to servers, * Application Load balancer to divide traffic between servers * Amazon simple storage service(S3) to access data remotely * Lambda function and Cloud Formation for execution and automation of services   Gerrit was to review the code and if reviewed it will store the updated code in GitLab.  All repositories where stored in GitLab, if any update in the repository it will trigger the build.  Jenkins will that trigger build and store the artifact in JFrog - JFrog  Created Ansible playbook for automating the infrastructure -Automatic Installation of (Jenkins, GitLab, Gerrit, JFrog) tools using ansible playbook and the creation of multiple users and multiple groups. |
| Cloud | Amazon Cloud |
| Duration | 6 Months |

**Project: 2**

|  |  |
| --- | --- |
| Project Name | Chatbot using Amazon Lex (POC) |
| Project Description | Chatbot using Amazon Lex:   The bot is integrated with slack. So remotely we can start and stop the ec2 Instances without going to the Amazon portal. |
| Responsibility | Amazon Lex service was used to create a customize chatbot, Amazon Lex was integrated with Lambda function, Lambda function where functions to start, stop, list all running instances, get IP of instances, Amazon Lex was integrated with the slack so remotely send request using laptop or mobile phones using NLP.  AWS Cloud   Services:   * EC2 * Lambda function * Amazon Lex * Iam Role   Slack. |
| Cloud | Amazon Cloud |
| Duration | 2 Months |

**Project: 3**

|  |  |
| --- | --- |
| Project Name | Buckman |
| Project Description | The project was based on .NET, Azure and Platform Automation framework. |
| Responsibility | Created an automation framework that creates various resources on Azure and Configures them.   Resources that will be created by the Platform Automation framework:   • DSP   • IOT Hubs   • Stream Analytics   • Cosmos DB   • Web Apps   • Function Apps For CI/CD using Azure DevOps,   Azure Key vault for storing credentials,   PowerShell ARM Templates for Automation. DevOps CI/CD pipeline, Azure DevOps, Continuous Integration/ Continuous Development, Azure Monitoring and Diagnostics |
| Cloud | Azure |
| Duration | 2 Months |

**Project: 4**

|  |  |
| --- | --- |
| Project Name | DigitalSky |
| Project Description | AWS code commit, code deploy and code build is used for CI/CD and Kubernetes for deployment of application. Application was deployed using helm charts |
| Responsibility | AWS Code Commit and code build/deploy used for CI/CD. Post build application were deployed in Kubernetes ENV.  Kubernetes cluster was deployed in EKS (Elastic Kubernetes Services). Kubernetes Cluster was monitored using Prometheus and Grafana Dashboard. ELK (Elasticsearch Logstash and Kibana) for aggregating logs for faster troubleshooting. EC2 instances and load balancers used for computing. Custom Dashboard created for monitoring  Zeebe was used for workflows and deployment was done using helm charts. Kubernetes orchestration, deployment of tools and troubleshooting was the main responsibility helm chart used for packaging application,  Managing ALB and mapping with the application for accessing publicly, Domain mapped to the ALB so we can access service using domain. |
| Cloud | Amazon Cloud |
| Duration | 3 Months |