|  |  |  |
| --- | --- | --- |
|  | **Deepak** |  |
| **Kumar** |
| **Associate DevOps Engineer** | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Professional Summary** |  | He is an enthusiastic **Associate DevOps Engineer** with a strong foundation in infrastructure as code, continuous integration/continuous **deployment (CI/CD**), and cloud-native technologies. He possesses deep knowledge of containerization, orchestration, and monitoring tools. He recently worked on **automating the deployment pipeline for a Campus Hires (ATS) application**, implementing infrastructure as code using **Terraform**, implementing configuration management as code using **Ansible** and setting up CI/CD pipelines with **Jenkins, Bitbucket, GitHub Action, Azure DevOps and GitLab CI**, and managing cloud resources on **AWS** & **Azure**.  . He also developed monitoring and alerting solutions using **Prometheus Grafana, Elasticsearch Logstash Kibana (ELK)** and ensured high availability and scalability of the application through Kubernetes orchestration. He is skilled in **scripting and automation**, with a keen ability to streamline development processes and improve operational efficiency. He has the expertise in **Data Structure and Algorithm too.** |

|  |  |  |
| --- | --- | --- |
| **Education** |  | Bachelor of Technology, Computer Science & Engineering, Aliah University, Kolkata, India |

|  |  |  |
| --- | --- | --- |
| **Training and Certifications** |  | Got Certifications from Google for securing Global 351st Rank in Coding Round E, Got 10 Badges from NagarroU in DevOps |

|  |  |  |
| --- | --- | --- |
| **Skill Set** |  | * Areas of competencies include CI/CD pipelines, containerization, orchestration, cloud infrastructure management, monitoring and logging solutions, and infrastructure as code * Expertise in automating deployments, optimizing infrastructure performance, scripting and automation with tools like Ansible, Terraform, and Kubernetes, as well as cloud platforms such as AWS and Azure |

|  |  |
| --- | --- |
| **Programming Languages** | * C++, C, C#, Python, JavaScript, HCL KQL, SQL |
| **Web Technologies** | * NodeJS, Express, Flask, Angular, Asp.Net, HTML5, XML, Web API, React. Spring Boot, Junit, XUnit |
| **APIs and middleware** | * REST, Web Services, JSON, Hash Crop APIs, SOAP, RabbitMQ, Eureka, Ocelot, Kubernetes, Ansible, Helm, Docker |
| **Database** | * SQL Server, MongoDB, PostgreSQL, Redis |
| **Tools and platforms** | * Jenkins, Terraform, Ansible, GitLab, GitHub Action, Bitbucket, Jira, Kubernetes, Docker, Artifactory, AWS, Azure, Container Orchestration, Azure DevOps, Helm, Virtual Machine, Linux, Ubuntu, WSL, GitOps, ELK, Prometheus and Grafana, XUnit, JUnit, Nginx, Apache server, Tomcat server |

|  |  |  |
| --- | --- | --- |
| **Recent Projects** |  |  |

|  |  |  |
| --- | --- | --- |
| **Project name** |  | **A Microservice-based E-Commerce Solution** |
| Client |  | Nagarro |
| Duration |  | Jan 2024-Fab-2024 |
| Description |  | Developed A 'Seamless Retail' project—a cutting-edge E-Commerce app designed to transform online shopping. Developed in just two months, it uses M**icroservices** to scale, adapt, and seamlessly integrate various retail services. |
| Technology Stack |  | React Js, .Net, Eureka, Ocelot, SonarQube, Artifactory, Docker, Container Orchestrations, Jenkins, RabbitMQ, NLog, Exceptional Handling |
| Responsibilities |  | * Developed new features and maintained existing functionalities using React JS and .Net within the 'Seamless Retail' project, ensuring seamless integration and scalability with Microservices architecture * Led the migration efforts to upgrade features, leveraging technologies like Eureka, Ocelot, and Docker for streamlined performance and compatibility * Designed intuitive user interfaces (UI) that effectively integrated services such as RabbitMQ and Jenkins, optimizing performance and enhancing user experience * Spearheaded build generation across multiple environments using Jenkins, and performed administrative tasks like code merging and peer code reviews using SonarQube and Artifactory * Proactively identified and resolved issues discovered during Jenkins testing, ensuring smooth deployment and functionality across all stages of development, thereby maintaining high standards of quality assurance * Developed comprehensive unit test cases for components and services, utilizing best practices to maintain high code quality and reliability throughout the project lifecycle |

|  |  |  |
| --- | --- | --- |
| **Project Name** |  | **Deployed Java-Based Application on Kubernetes Cluster Using On-Premises Infrastructure** |
| Client |  | Nagarro |
| Duration |  | May 2024-June 2024 |
| Description |  | In this project, **I successfully deployed a Java-based application on a Kubernetes cluster** using on-premises infrastructure. Utilizing **Helm charts** for streamlined deployment, I orchestrated the setup and configuration of Kubernetes using **VirtualBox** and managed the entire process through a **Jenkins pipeline**. Additionally, **I implemented Prometheus and Grafana for monitoring and visualization** of metrics across the Kubernetes cluster, and set up the **ELK (Elasticsearch, Logstash, Kibana) stack** for centralized logging and analysis of application logs. These monitoring and logging solutions were **automated through Jenkins pipelines**, ensuring **consistency and reliability** in deployment and configuration.  To ensure seamless operation of the Kubernetes cluster, **I had to enable communication between the master node and worker nodes**. This involved configuring network settings, such as establishing proper **firewall rules** and ensuring **network connectivity**, to facilitate communication and data exchange across all nodes within the cluster.  Moreover, **I integrated Artifactory into the CI/CD pipeline** to upload the artifact (such as a WAR file) generated from the Java-based application. This artifact was stored in Artifactory and then pulled from Artifactory during the build process to create a **Docker image**. The Docker image was subsequently passed as an argument in the **Helm command for deployment**, ensuring a seamless and automated deployment workflow |
| Technology Stack |  | Spring Boot, Helm Chart, Kubernetes, Kubeadm, Kubectl, Kubelet, ELK, Prometheus-Grafana, Ansible, Docker, Eclipse, GitOps, GitLab, DNS, Virtualization, CNI, Maven |
| Responsibilities |  | * Integrating Artifactory into the CI/CD pipeline to upload artifacts (e.g., WAR files) generated by the Java-based application * Pulling artifacts from Artifactory to build Docker images and passing these images as arguments in Helm commands for deployment * Writing comprehensive unit tests to verify the functionality and reliability of all services within the application * Orchestrating the setup and configuration of Kubernetes on-premise infrastructure using VirtualBox * Deploying Prometheus and Grafana using Helm charts to monitor and visualize metrics across the Kubernetes cluster * Setting up the ELK stack for centralized logging and analysis of application logs * Managing the entire deployment process of the Java-based application and associated stacks through Jenkins pipelines, ensuring seamless integration and continuous delivery |

|  |  |  |
| --- | --- | --- |
| **Project name** |  | **Deployed Java-Based Application on Azure Cloud with Terraform and Ansible via Azure DevOps** |
| Client |  | Nagarro |
| Duration |  | May2024-May 2024 |
| Description |  | In this project, I successfully deployed a **Java-based application** **on Azure Cloud using Azure App Service**. The deployment process involved building the Java application**, running unit tests**, **performing code analysis with** **SonarQube**, and **storing the artifact (WAR file) in Artifactory**. Subsequently, I built a Docker image from the **artifact stored in Artifactory** and deployed it to **Azure Container Registry** using **Azure VM** with a **self-hosted agent** installed |
| Technology Stack |  | Azure DevOps, Azure Repo, App Service, ACR, Docker, Azure VM, Java, Maven, Artifactory, SonarQube, Azure Marketplace, Self-Hosted agent, Git |
| Responsibilities |  | * Led the end-to-end deployment of a Java-based application on Azure Cloud using Azure App Service, ensuring seamless integration and functionality * Managed the Azure DevOps pipelines for continuous integration and delivery (CI/CD), orchestrating the entire deployment process from code repository to production deployment * Built and configured Azure infrastructure using Terraform to provision resources necessary for application deployment, including Azure App Service and Azure Container Registry (ACR) * Implemented Ansible playbooks for configuration management of Azure VM and Docker environments, ensuring consistency and reliability in application deployment * Executed build processes for the Java application using Maven, ensuring code quality through comprehensive unit testing and integration testing * Conducted code analysis and quality checks using SonarQube, maintaining high standards of code cleanliness and performance optimization * Managed artifacts (WAR files) generated from the Java application build, storing them in Artifactory for artifact management and version control * Orchestrated the creation of Docker images from artifacts stored in Artifactory, optimizing deployment efficiency and scalability * Utilized Azure VM with a self-hosted agent installed from Azure Marketplace for Docker image deployment to Azure Container Registry, ensuring secure and efficient containerized application deployment |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Project name** |  | **Book Borrow System** |
| Client |  | Nagarro |
| Duration |  | Nov 2023-Dec 2024 |
| Description |  | The Project was a web app that provides Book borrow services. It has features like: Borrowing a book, returning borrowing books, rating the book, User dashboard with complete access, and inspection control. |
| Technology Stack |  | React Js, .Net, ORM, Identity Framework, SQL Server |
| Responsibilities |  | * Handled all the CRUD operation on data in backend of the application. * Designed and developed backend functionalities such as user authentication, user authorization, Book management etc. * Created and updated functionalities in the frontend React application. * Created attractive UI for clean and hassle-free user experience |