Kasevaddi Muthyalappa
Email: kasevaddi.95@gmail.com
Phone: 9177703828

Linkedin: https://www.linkedin.com/in/kasevaddi-m-087376247/
Github: https://github.com/kasevaddi

Course and assignment completion certificates: provided by IBM
https://courses.cognitiveclass.ai/certificates/5aa461984fc24eeebaccd219d9b630f5
https://courses.cognitiveclass.ai/certificates/d3f96984977444c99cd0415581f1bf5c
https://courses.cognitiveclass.ai/certificates/c87b7ffa831d44048a9bdd25e4e3193d

SUMMARY:

- ➔ Overall **3.5** years of IT experience as a Build and Release/DevOps Engineer, specializing in automation, code integration, deployment, and environment-to-environment release management.
- ➔ Experienced in working on **Devops/Agile** operations process and tools area (Code review, Build & Release automation, Environment, Service, **Incident** and **Change Management).**
- ➔ **Design, build, and maintain Continuous Integration**/Continuous Deployment (CI/CD) pipelines using tools like **Jenkins,Teamcity, Octopus and Azure DevOps**.
- ➔ Ensure automated **build, test, and deployment** processes for **faster releases.Experience** in working on source controller tools like Subversion **GIT.**
- ➔ Automate provisioning and configuration of environments **(Dev, QA, UAT, Prod**).Set up and manage monitoring tools like Prometheus, **ELK Stack, New Relic, and CloudWatch**.
- ➔ Used tools like **Ansible, Chef** to configure and manage servers.Maintain consistency across environments and manage system drift.Automate OS-level and application
- ➔ configurations (Node.js) using **Ansible playbooks and roles.**
- ➔ Create and manage **Chef cookbooks/recipes** to automate complex setups across servers.Upload configurations to **Chef Server**, which nodes fetch using the chef-client ideal for large infrastructures**.**
- ➔ Using **Terraform** Define infrastructure (servers, networks, databases) using HCL (HashiCorp Configuration Language) for **AWS.**Provision scalable, fault-tolerant compute environments using **EC2** with Auto Scaling Groups and **ELB.**

- ➔ Write **Dockerfile** and **docker-compose.yaml** to build a multi-container setup for local testing or microservices.Use **Docker Hub** or private registries to store and manage versioned container images efficiently.
- ➔ Create and manage pipelines using **Freestyle Jobs**, **Declarative/Multibranch Pipelines**, to automate **build, test, and deployment.** Configured plugins (Git, Maven, Docker, SonarQube, Kubernetes, etc.) to extend Jenkins functionality and integrate with external tools.
- ➔ Integrate with Git and configure **triggers** (e.g., on commit, schedule, branch) to automate builds using **Teamcity,** Create reusable parameterized build configurations to streamline pipeline setup across **QA DEV and prod env.**
- ➔ Exposed to all aspects of software development lifecycle (SDLC) such as Analysis, **Planning, and Developing, Testing, and Implementing** Post-production analysis of the projects.
- ➔ Developed **shell** scripts for software build conduct and management support.Use **if, else, and elif** to conditionally execute blocks based on values or system states.
- ➔ Monitor application performance, database queries, and external service calls in real-time.Monitor host metrics like **CPU, memory, disk.** Use **Kibana** for visualizing logs, building dashboards, and analyzing errors or performance trends
- ➔ Orchestrate application deployments across **Dev, QA, UAT**, and Prod environments using deployment pipelines using **Octopus** automate pipeline tool.
- ➔ Strong Experience on source controller concepts like **Branches, Merges and Tags**.
- ➔ Worked with Engineers, **DEV and QA and other teams** to ensure automated test efforts are tightly integrated with the build system and in fixing the error while doing the deployment and building (Agile Projects).
- ➔ Used **Yocto Build** customized embedded Linux **images** tailored to specific hardware using recipes and layers.Manage build processes using **BitBake** to control package compilation, configuration, and dependencies.

➔ Generate cross-compiled **toolchains, kernel**, root filesystem using a menu-driven config.Integrate **Buildroot into Jenkins**/GitLab for automatic generation of embedded firmware images.

QUALIFICATION

• Level	• Subject	• College / University / Year
• MBA	Computer Application	Jawaharlal Nehru Technological University,Anantapur/2019

SKILLS:

Operating Systems	Linux (CENTOS, UBUNTU), Windows (7,8,10)
Application Server	Apache Tomcat, WebLogic, IIS
SCM Tools	BITBUCKET, GIT,SVN, Jira
Build Tools	MAVEN, Cmake, Yocto, Buildroot
Continuous Integration Tool	Jenkins, Teamcity, Sailpoint
Web Server	Apache2.4
Languages	Python, C++
Scripting language	Shell and python and yaml
Cloud Platform	AWS, GCP, AZURE, AZURE DEVOPS
Others	Chef, Ansible, Docker, kubernetes,ELK, Newrelic, sonarqube, Terraform and Octopus, Grafana, Prometheus

PROFESSIONAL EXPERIENCE:

PROJECT: 1

➔ **Payroll : SEMACONNECT SYSTEMS INDIA PVT LTD**
➔ **Client : PERSISTENT SYSTEMS**

Project Profile:

- **Project Name** : **EV charging stations**
- Team Size **:** 2
- Duration **: Jan 2022 to May 2023**
- Client : Persistent
- Environment : DevOps, Shell scripting, Yocto,Cmake, Jenkins, Apache-Tomcat, Apache-2.4
- Operating System : Ubuntu
- **Role** **: Build Automation and integration Engineer**

Roles & Responsibilities:

- **Build** and **deploy** code of different environments. or different key api keyservers. Create multibranch pipelines to build **CMAKE, YOCTO for every 3 days have to create the tags/Labels for Release the serieses as per developers.**
- Automate the build tool like YOCTO, CAMKE using JENKINS. Tag must be matched with Tag pattern otherwise jenkins should ignore this, in this case i used **REGEX.**
- After creating the tag have to build and signin encrypt the artifact in api keyservers. After signin download the **artifacts** from the api key servers.
- Responsible for release management for different client production environments by doing **continuous integration and continuous deployments** of the software application.
- Automate the manual tasks within the Release Management using **Jenkins** & **Yocto** Automation tool.
- Creating automated build and release environment using continuous Integration Tool
- **Jenkins.**
- Creating tags, branches & setting up projects using version control like **GIT.** Creating new users and giving the proper **permission** to access the repository based on the project.
- Having experience installing packages, files etc in client nodes through yml playbooks. Coordinating with developers and testers **GIT** Related **issues,** Automate the build process using **Jenkins**.
- Creating Multi Branch code pipelines to use, build and release and deploy the artifacts.
- send the mail notifications to certain developers using SMTP if builds come success or fail.
- Developing and maintaining build files by using **Jenkins.** Responsible for creating the new projects & new build setup using **Jenkins.** FOLLOWUP and verify the **Backup** regularly (Backup maintenance).
- Creating and maintaining docker images & containers using containerization tool Docker. Creating images by using docker file system. If the build or Deployment fails, a Roll-Back plan is followed to make the previous changes reflect.
- Code Deployments using the **automation** through **Jenkins** tool and configuring the variables required to support the application. Creating and maintaining the documentation for all the new configuration and environment setups.

- Develop **Python** scripts to automate CI/CD pipelines, infrastructure provisioning, release processes, and environment setups.
- Integrate **Python** scripts with tools like **Jenkins**, **TeamCity**, and **Azure DevOps.**
- Write scripts for pre/post build tasks, versioning, artifact management, and log collection
- Use Python to interact with APIs from **New Relic**, **Prometheus**, **Grafana**, **Datadog**, or **Nagios** for monitoring setup or auto-remediation scripts.
- Automated the provisioning of **networking, compute, storage, and security components** across clouds using Terraform.
- Integrated Terraform with **CI/CD pipelines (Jenkins, GitLab CI, Azure DevOps)** for automated infrastructure deployment and testing.
- Implemented **Terraform workspaces** and variable strategies to manage environment-specific configurations (dev, QA, prod).
- Implemented **Jenkins backup and recovery** using the *Thin Backup* plugin to safeguard configurations and job data in case of system failures.
- **Migrated Jenkins home directory** from the default location /var/lib/jenkins to a custom path /disk1/jenkins to optimize storage and improve performance
- Provided knowledge transfer (KT) to developers regarding **Jenkins** setup and operational processes.
- Assisted in identifying specific **tag names** created by developers across repositories to ensure traceability and version control compliance.
- Used **Python** to automate repetitive system tasks such as **backups, log rotation**, and **system updates**.
- Developed and maintained **cron job scripts** for scheduled system operations and DevOps tasks.
- Created **Python-based custom scripts** for CI/CD processes including **code integration, build, testing, and deployment**.
- Built Python scripts to manage **configuration files, environment variables**, and to **automatically push changes** to target systems.

PROJECT: 2

- **Payroll : AMANTYA TECHNOLOGIES**
- **Client : ZEISS**

Project Profile:

- **Project Name** : **Zeiss Medical devices**
- Team Size **:** 5
- Duration **: May 2023 to Aug 2024**
- Client : zeiss

➔ Environment : DevOps, Shell scripting, Yocto,buildroot, Azure Apache-Tomcat, Apache-2.4

➔ operating system : ubuntu

➔ **Role : Build and Release Engineer**

Roles & Responsibilities:

- Designed and managed cloud infrastructure using **AWS services such as EC2, S3, VPC, IAM, ELB**, and CloudWatch for high availability and scalability.
- Implemented Auto Scaling Groups and Elastic **Load Balancers** (ALB/NLB) to ensure performance and cost efficiency.
- Created and maintained **Terraform** modules to provision and manage AWS resources efficiently.
- Containerized applications using **Dockerfiles** and optimized images for reduced build time and footprint.
- Managed container lifecycle and orchestrated **multi-container** environments using Docker Compose.
- Deployed and managed containerized workloads on **Kubernetes** (EKS/AKS) clusters.
- Created and optimized Helm charts and **Kubernetes** manifests for microservice deployments.
- Automated infrastructure provisioning and application deployment using **Ansible** playbooks and roles.
- Managed server configurations, **patches**, and software installations across multiple environments.
- Configured build pipelines in **TeamCity** for continuous integration of Java and Node.js applications.
- Integrated **SonarQube, Git, and Artifactory** with TeamCity for quality checks and artifact management.
- Designed and implemented **CI/CD pipelines** in Jenkins for automated build, test, and deployment processes.
- Integrated Jenkins with **Git, Maven, Docker, Terraform, Ansible**, and Kubernetes for end-to-end automation.
- Implemented version-controlled IaC workflows integrated with **CI/CD pipelines.**
- Automated **multi-environment** infrastructure setup (Dev, QA, Prod) for consistent configuration across deployments.
- Integration with third-party tools Utilizing tools like **Jenkins, Terraform, Ansible**, etc., in conjunction with Azure services.
- Use python Automate Git operations: branch creation, **merging, tagging,** repository cloning, and commit analysis.
- Use Python to interact with APIs from **New Relic**, **Prometheus, Grafana**, for monitoring setup or auto-remediation scripts
- Applied **Terraform plan, apply, and destroy commands** with change reviews to ensure controlled infrastructure modifications.
- Automate security tasks such as vulnerability scanning, compliance checks,

and secret rotation using **Python scripts**.

- ◆ Use Python scripts to build **Docker images,** manage containers, and interact with Kubernetes clusters.
- ◆ Used **Ansible** to define and enforce configuration states across environments (Dev, QA, Prod).
- ◆ Maintain idempotent **playbooks** to ensure consistent and repeatable deployments.
- ◆ Write YAML-based **playbooks** to manage infrastructure declaratively.
- ◆ Integrate **Ansible with CI/CD** tools like Jenkins, GitLab CI, and monitoring tools like Prometheus
- ◆ Automate builds for Java-based applications using **Maven** lifecycle phases (compile, test, package, install, deploy).

PROJECT: 3

- ➔ **Payroll : CONSIGN SPACE SOLUTION**
- ➔ **Client : INFOSYS**

Project Profile:

- ➔ **Project Name** : **XPO LOGISTICS**
- ➔ Team Size **:** 10
- ➔ Duration **: Sep 2024 to Mar 31 2025**
- ➔ Client : XPO,Infosys
- ➔ Environment : DevOps, TeamCity, GCP, Octopus, New Relic, SonarQube, DevOps Admin, DevOps Release,ELK
- ➔ Operating System : Linux, windows
- ➔ **Role : Devops Consultant Engineer**

Roles & Responsibilities:

- ◆ **Decommission** the legacy servers as per developers request and create new servers like window and linux servers.
- ◆ Plan and execute weekly **UAT and Production releases**. Ensure smooth Deployments using **DevOps Release Portal, TeamCity, and Octopus**.
- ◆ Coordinate with development and **QA teams for pre-release** and post-release activities.
- ◆ Troubleshoot and resolve release-related issues in different environments. Manage deployments to different environments (test, development, **production**) seamlessly.
- ◆ Integrate with tools like **TeamCity, Jenkins** to trigger deployments after successful builds.
- ◆ Monitor application performance and detect bottlenecks using **ELK** and New Relic. Manage and optimize **CI/CD p**ipelines in TeamCity for automated builds and deployments.
- ◆ Support and maintain **Dev, QA, UAT**, and Production environments. Investigate and resolve QA and Dev issues to ensure system

stability.

- Analyze logs and errors using ELK to debug issues across environments. Work with teams to identify and resolve performance bottlenecks using **New Relic**.
- Provide support during critical incidents, rollback scenarios, and post-release validation.
- Developed and maintained CI/CD pipelines using TeamCity and **Octopus Deploy** for consistent application deployment across Dev, QA, UAT, and Prod environments.
- Configured and maintained **New Relic** and ELK Stack for application and infrastructure monitoring.
- User Management Create, manage, and delete users and groups.
- Permission Assignment Grant least privilege access using roles or policies.
- **Authentication and Authorization** Enforce MFA, password policies, and access controls.
- Policy Management Create, review, and update IAM policies (JSON-based in AWS, Role-Based in Azure).
- Access Auditing Monitor access logs, review user actions, and conduct security audits.
- Created **azure devops** pipelines for build, push, test the **micro service**s and integrated the **azure c**ontainer registry with pipelines to store the docker images.
- Provide azure access (read, write, admin) to new and existing users using **Microsoft entra ID** for access to the Azure resources like VMs and azure pipelines.
- migrated complete github repos to azure pipeline along with the source code as well the pipelines.
- Setup the meetings with clients through **microsoft365** to discuss the release plans.
- Integrated the **jira** between the **CICD tools** and **github** to check the status of the deployment.
- Using **Identity Governance** and Administration provides certain access to project teams. Configure role-based access control (RBAC) and least privilege access.
- Define IAM strategy and architecture including JML workflows.
- write Python scripts to replace manual shell commands.
- Schedule and execute automation tasks via cron, Jenkins, or Airflow.
- Create branches, pull commits, generate release notes using Python.
- Use **Cookiecutter** to auto-generate scaffolding as part of **CI/CD pipelines**.

- Integrate**Cookiecutter** with tools like Jenkins, GitLab CI, or TeamCity to spin up new microservices or infra setups.
- Combine **Cookiecutter** with infrastructure-as-code (e.g., Terraform templates, Helm charts) for reproducible deployments.
- Package applications and their dependencies into **Docker** containers.
- Write efficient and secure **Dockerfiles** to build container images.
- Build, tag, and version **Docker images.**
- Store and manage images in registries like **Docker Hub**, Amazon ECR, or GitHub Container Registry.
- Collaborated with cloud architects and DevOps teams to define infrastructure standards and best practices using T**erraform.**
- Use **groovy and shell** to create and manage custom build scripts that compile code, manage dependencies, and generate build artifacts.
- Automate Git workflows (branching, merging, tagging, pull requests) using **bash** with libraries like gitpython.
- Write **bash** scripts to schedule and automate software releases across different environments (Dev, QA, UAT, Prod).

◆ DECLARATION:

I hereby declare that the above-mentioned information is true and correct to the best of my knowledge and belief.