NARESH AGRAWAL

857-415-8953 **|** agrawal.nar@northeastern.edu **|** www.linkedin.com/in/naresh-agrawal **|** github.com/nareshagrawal

EDUCATION

Northeastern University, Boston, USA   **Expected August 2021**
Master of Science in Information Systems
Shri G.S Institute of Technology and Science (SGSITS), Indore, India   **April 2018**
Bachelor of Engineering in Electronics & Telecommunication

TECHNICAL SKILLS

Programming languages	Java SE, Java EE (*J2EE*), Bash Scripting, SQL
Web Technologies	HTML, JSP, CSS, SCSS, Bootstrap, JavaScript, React
Frameworks	Spring Boot, Spring MVC, Hibernate, RESTful APIs, Microservices, Swing, JUnit, Log4J
DevOps Tools	Kubernetes, Amazon Web Services (AWS), Helm Chart, KOPS, Docker, Scripting, Apache Kafka, Ansible, Terraform, Jenkins, GitHub Actions, CircleCI, Prometheus.io, Maven, Git
Databases & Server	MySQL, AWS RDS, MongoDB, DynamoDB, Apache Tomcat, Nginx
Version Control, Tools & Operating Systems	GitHub, Bitbucket (Stash), Postman, Apache JMeter, Splunk, JIRA, ServiceNow, Confluence, Eclipse IDE, IntelliJ IDEA, Visual Studio Code, Linux, macOS, Windows

WORK EXPERIENCE

SS&C Intralinks, Waltham, USA   June 2020 – Jan 2021
Site Reliability Engineering Co-op (SRE)

- Created a 'Status Page', integrated with monitoring tool (SaaS) informing customer about downtime, incident, maintenance of the products and services, easily communicate real-time status to end-users
- Set up monitoring, alerts, handled overloads on server and automated tasks via CI/CD pipeline
- Deployed code updates, worked to roll environment forward and performed release engineering
- Troubleshoot and escalate bugs for Live server product, examining, investigating, and resolving problems to help smooth product performance and tracking progress through JIRA, ServiceNow and Git Repositories
- Created, maintained documentation of systems and processes for existing and new systems on Confluence

PROJECTS

Weather Alert API (*AWS, Kubernetes, Helm Chart, Docker, Jenkins, Ansible, Kafka, Prometheus, Java*)

- Built Microservices based REST Spring Boot API, support 10,000 requests per second with an uptime of 99.99%, deployed containerized applications on Kubernetes cluster using Helm charts, Ansible and Jenkins via CI/CD pipeline
- Designed Jenkins Pipeline to deploy application on each push to GitHub repository, build, push docker images to Docker Hub, and perform rolling update style deployments on the Kubernetes cluster
- Used KOPS for creating Kubernetes cluster on AWS, Apache Kafka for communication among microservices, implemented Cluster auto-scaling and Horizontal pod auto-scaling using Metrics Server to handle dynamic load
- Installed NGINX Ingress Controller to route traffic on DNS and Let's-Encrypt to issue SSL certificate for secure connection, EFK stack for logging and Prometheus for analyzing and monitoring, visualize it on Grafana dashboard

Uber (*AWS, Azure, EKS, AKS, Jenkins, Ansible, Terraform, Docker, React, Spring Boot, RDS, JWT*)

- Built Microservices based application, deployed containerized frontend app on AKS cluster and backend API on EKS cluster using Helm charts, and Jenkins via CI/CD pipeline
- Developed a dynamic, responsive frontend using JavaScript React library running behind NGINX and backend with Spring Boot REST API, persisted data with AWS RDS, Integrated Google map API and handled user session with JWT
- Designed CI/CD pipeline using Jenkins to build, push docker image to Docker Hub and perform rolling update style deployments on both AKS and EKS cluster
- Implemented Cluster auto-scaling and Horizontal pod auto-scaling to handle dynamic load, EFK stack for logging and Prometheus for analyzing and monitoring, visualize it on Grafana dashboard

Online Book Store (*AWS, CircleCI, Terraform, Spring Boot, Java, Hibernate, RDS*)

- Engineered Spring Boot app based on MVC architecture, created AWS resource stack using Terraform script and deployed application on EC2 via CI/CD pipeline using CircleCI
- Executed Lambda function to trigger SES when a message is published on SNS Topic and performed auto-scaling for EC2 instance using CloudWatch alarm, Incorporated S3 buckets, AWS RDS to store object metadata with SSL/TLS connection
- Configured Load Balancer and security groups to route traffic on DNS, attached SSL certificate for secure connection, LogGroup for logging all user and system actions and metrics for analyzing and monitoring

Job Lender (*Spring MVC, Hibernate, Java, MYSQL, JSP, Bootstrap, Log4J, JUnit*)

- Developed a Spring MVC and Hibernate application for making job search easier through a job board
- Incorporated Apache Commons Email API, functionality to download applicant profile as a PDF using iText API, utilizing Log4J for logging all user and system actions and JUnit for testing application before iterative deployment
- Implemented features job recommendations, searching, viewing applied jobs, withdrawing submitted job applications