NARESH AGRAWAL

857-415-8953 | nareshagrawal316@gmail.com | www.nareshagrawal.com | www.linkedin.com/in/naresh-agrawal

EDUCATION

Northeastern University, Boston, USA **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, Golang, Python, Shell Scripting, SQL
Web Technologies	JavaScript, Node.JS, React, HTML, CSS, SCSS
Frameworks	Spring Boot, Spring MVC, Hibernate, RESTful APIs, Microservices
DevOps Tools	Kubernetes, Amazon Web Services (AWS), Azure, Helm Chart, KOPS, Docker, Ansible, Terraform, Packer, Jenkins, Circle CI, Argo CD, Flux CD, Apache Kafka, Prometheus, Grafana
Databases & Server	MySQL, PostgreSQL, MongoDB, Apache Tomcat, Nginx
Version Control & Tools	GitHub, Bitbucket (Stash), Postman, Apache JMeter, Splunk, JIRA, ServiceNow, Confluence
Operating Systems	Linux, macOS, Windows

WORK EXPERIENCE

Juniper Networks, Sunnyvale, USA May 2023 – Present
Site Reliability Engineer 3

- Performed **EKS** clusters upgrade with zero downtime using blue green deployment strategies, ensuring continuous availability and minimizing service disruption
- **Designed** and **architected** health check API to retrieve health/status of pods, improving deployment efficiency by 30% through automation **of CI/CD** pipelines
- Conduct performance tuning and capacity planning to optimize resource utilization and reduce cost
- Improved **lifecycle** of services from inception and design through deployment, operation, and refinement
- Working on **infrastructure** automation with **Terraform**, **Packer**, CI/CD, and configuring systems using **Ansible**
- Working with **Linux** operating systems internals, file systems, administration, and networking

BlackRock, Atlanta, USA Oct 2021 – Mar 2023
Site Reliability Engineer 2

- Worked on cloud **infrastructure cost optimization** and reduced infrastructure costs by 20%
- Reduced system **downtime** by 10% through proactive monitoring, alerting, and incident management
- Collaborated with peers on development of new automation tools and services with **proof-of-concept** presentations
- Built **CI/CD** pipelines and operations processes using **Jenkins** and **Azure pipelines**
- Developed **automated** monitoring and testing solution in **Go**, ensuring that all the policies and pipelines run successfully
- Reduce **configuration drift** across 100s of servers via **Ansible** playbook

SS&C Intralinks, Waltham, USA June 2020 – Jan 2021
Site Reliability Engineer Intern

- Created a 'Status Page', integrated with monitoring tool (**SaaS**) informing customer about downtime and incident
- Set up **monitoring**, **alerts**, handled overloads on server and **automated** tasks via **CI/CD** pipeline
- Defined **SLAs, SLOs and error budgets** for mission critical platforms
- Deployed code updates on **Kubernetes cluster**, 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

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
- Used **KOPS** for creating Kubernetes cluster on **AWS**, Apache **Kafka** for communication among microservices and Installed **Nginx** Ingress Controller to route traffic on **DNS** and Let's-Encrypt to issue SSL certificate for secure connection

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

- 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 server and backend with **Spring Boot** REST API, persisted data with AWS **RDS**, Integrated Google map API and handled user session with JWT