ANKIT SINGH RATHI

Senior DevOps Engineer | AWS Solutions Architect Professional Certified | ankitrathi.info | CKA | CKAD | Ex Walmart SE

📞 +971-585569492 @ codeaprendiz.@gmail.com 🔗 www.ankitrathi.info 📍 Dubai

EXPERIENCE

DevOps Engineer
01/2020 - Ongoing

Tradeling.com
Dubai, UAE

- Set up Infrastructure on **self-managed kubernetes** cluster from scratch on AWS using **ansible**, **terraform** and poseidon/typhoon. Generated k8s manifests using **php**.
- Migrate all backend APIs to **AWS Edge Optimized API Gateway** and **EKS**. Also used AWS **WAF**, API **Throttling**, API **Caching**, Network LB and **VPC Link**
- Caching APIs at **AWS Cloudfront** with dynamic origins thereby reducing the homepage response time by **75%** (105ms to 25ms)**.** Also used custom **CORS** and caching policies.
- Reduced the **avg** website **load time** in China by **80%** (2500ms to 500ms) by testing with **Zenlayer Global Accelerator, Huawei CloudConnect, Alibaba Global Accelrator, Geolocation Records** and reduced the cost of solution from 3500$ to 70$ i.e. **98%**
- Closed around 70 plus vulnerabilities reported in ASV, Network VAPT, Web Application PT to make the infrastructure **PCI compliant Level 1**
- Set up **Monitoring** for Tradeling Infrastructure from scratch using **ELK stack, metricbeat, journalbeat** and **ansible** which collects data from 50+ servers across 3 environments
- Set up **CI-CD** on **Github Actions** and **Docker Jenkins** using ansible running **15000** jobs daily. Maintained jenkins jobs using groovy scripts
- Created Monitoring **Dashboards** in **Kibana** for API Performance, MongoDB, NATS, MySQL, Kubernetes, Hosts , System Monitoring and Docker.
- Daily Automated Releases using **python** script (**2000+ lines of code**) sending **automated email change log** of **50+ repositories** to respective teams. Attend **stand-ups** and **sprint planning** sessions.
- Maintaining Self Managed Databases including **MySQL**, **MongoDB** using docker-compose
- **Daily backups** of MongoDB, MySQL using ansible being uploaded to AWS S3 buckets
- 24/7 automated website monitoring using **Prometheus,** alertmanager, blackbox exporter with automated slack alerts.

DevOps Engineer
11/2018 - 01/2020

Accelya Group
Dubai, UAE

- Setting up Infrastructure of the new customers for Development, UAT and Production Environments on **Weblogic** and **Tomcat** Platforms.
- **Migration** of the build-jobs of **66 customers** to separate the 'Builds & Deployments' phases.
- Build **JIRA Dashboard** for the team. This was extensively used by all teams to log tickets and follow the DevOps Approach

Software Engineer
01/2017 - 10/2018

Walmart Labs
Bangalore, India

- Set up Infrastructure for **microservices** on **OneOps** with **DR** enabled involving **1000s** of servers across two data centers.
- Created DevOps Automation **Dashboard** using **Django, Python** to automate **splunk** alert actions.

CERTIFICATIONS

PROJECTS

API Caching At AWS Cloudfront

By caching the GET APIs at AWS Cloudfront we were able to **reduce** the **response** time by **75% (105ms to 25ms)**

- Created **dynamic origins** with **custom caching** and **custom CORS** policy
- Were able to achieve **single digit millisecond latency** for some GET APIs endpoints

SKILLS

Infrastructure As Code Tools
Ansible · Terraform · Cloud Formation

Cloud Providers
AWS · GCP · Alibaba · Huawei

Container And Orchestration
Docker · DockerSwarm · Kubernetes · ECS · Fargate

Languages
python · go · shell · nodejs · php

Version Control Systems
Git · SVN · Code Commit

Ingress Controllers
Traefik · Istio · Kong · Nginx

Monitoring
Grafana · Prometheus · Elastic Search · Kibana · Logstash · Splunk · Datadog · Journalbeat · Metricbeat · Alertmanager · Blackbox Exporter · Cloudwatch

PROJECTS

Re-Architecting Backend APIs with AWS API Gateway

We improved the **security** and **performance** of all the backend APIs by using Edge Optimized AWS **REST API gateway.**

- Applied AWS **WAF** at AWS API Gateway to improve the security of APIs.
- Also used API **throttling,** API keys and custom domain names.
- Used **Edge Optimized** REST API gateway to improve performance by using caching features
- Other AWS Services used : **Network Load Balancer**, **EKS** and **VPC Links**

Architect Cloud Infrastructure for China - Multi Cloud

We improved the website performance in China by **reducing** the avg website **load time by 80%** (2500ms to 500ms). This project was multi-cloud and was implemented in 3 phases.

- We initially used **Alibaba Global Accelerator** and **geolocation** records. Cost of the project was around **$3500** per month. We purchased dedicated bandwidth from HongKong to India.
- We then used **Huawei Cloud Connect**, **SNAT**, **DNAT**, NLB, geolocation records. The cost was around **$1500** per month.
- Finally we used **Zenlayer Global Accelerator** with geolocation records. This had the least cost of **$100** per month. This was based on data-transfer per month.

Migrating Self Hosted Services to AWS Managed Services

- MySQL was migrated to AWS RDS MySQL
- MongoDB was migrated to MongoDB Atlas
- Elasticsearch was migrated AWS ElasticSearch
- Self managed kubernetes cluster using kops was migrated to EKS

Monitoring And Logging Setup Using ELK Stack

Tradeling Monitoring and Logging Setup from scratch to set up dashboards

- **ELK** i.e. Elastic Search, Logstash and Kibana was used. All were set up on individual instances using **docker-compose** and **ansible playbooks**
- **Journalbeat** was used to set up logging for kubernetes. **Metricbeat** was used for getting metrics of kubernetes, instances, mongodb, mysql and NATs.
- **Dashboards** set up for Kubernetes, System, Host, MongoDB, MySQL, NATs, Docker monitoring. **URL monitoring** and **Alerts** on slacks using Prometheus, Alertmanager, Blackbox-Exporter
- **Prometheus**, **Alertmanager**, **Blackbox-exporter** set up done using ansible, docker-compose

Self Managed kubernetes infrastructure using opensource Typhoon

- Created using AWS, Terraform and OpenSource poseidon/typhoon repository.Used **reserved** and **spot** Instances to **save cost**
- The cluster was **self healing**. Whenever spot instance was terminated, **lamda** function would trigger a **webhook** in Jenkins which would run **ansible playbook task** to **drain** the corresponding kubernetes cluster node.

Automated Releases to Staging and Production

- **Python** script with **2000+** lines of code and triggered via **ansible-task** to automate release to staging environment which was in turn triggered through Jenkins
- **Automated Change Log-Email** sent to the dedicated audience after the release
- **Live Updates** of the release on **Slack Channel** showing status of builds and deployments

Infrastructure setup using Terraform, Ansible

- Setting up 3 environments - **Prod**, Stage, Dev
- Each environment consisting of **VPCs**, **EC2**, **Route53** records, **Route** tables, **Security** Groups, **Bastion** hosts, **NAT** Gateways etc
- Planning regular **upgrades** of self managed services like **MongoDB, MySQL, NATs**

Payment Card Industry Data Security Standards (PCI-DSS) Compliance - L1

Tradeling Infrastructure was made **PCI-DSS complaint L1** by closing ~70 vulnerabilities across different scans : **ASV**, **Network VAPT** and Web Application **Penetration** Testing.

- Used AWS **KMS** to encrypt card holders data before it was saved to the database.
- Provided necessary documentation and closed all the gaps that were high-lighted.

ankitrathi.info | DevOps Essentials

Creating a website for DevOps Essentials

- Terraform: *https://github.com/codeaprendiz/terraform-kitchen*
- Kubernetes: *https://github.com/codeaprendiz/kubernetes-kitchen*
- Ansible: *https://github.com/codeaprendiz/ansible-kitchen*
- DevOps-Essentials : *https://github.com/codeaprendiz/devops-essentials*
- Linux - https://ankitrathi.info/devops-essentials/linux-commands/linux-commands-index.html