**Meenakshi Sundaram**

**Email:** **minasundaram@gmail.com**

**Phone:** (984) 212-2896

**SNAPSHOT:**

DevOps Engineer with 8 years of experience in IT industry. Strong experience in migrating the applications to the cloud (**AWS**, **AZURE**, and **GOOGLE**). Designed and implemented CI/CD pipelines to reengineer the application development. With in-depth understanding of the Software Development Lifecycle, cloud application challenges, can mentor other engineers in setting up CI/CD pipelines, to develop robust and highly responsive applications. Experience in mentoring a team to migrate applications, which will leverage cloud infrastructure.

**PROFESSIONAL SUMMARY:**

* As a DevOps Engineer, I have worked on Automating, Configuring and Deploying instances on **AWS**, **AZURE**, **GCP.**
* Expertise in **Java J2EE** technologies like **Struts**, **Spring**, **JSP**, **EJB**, **JDBC**, **Servlets**, **MVC**, **JSON**, **Subversion**, **Clear** **Case**.
* Experience in developing dynamic web pages using **HTML**, **CSS**, **JavaScript**, **ES6**, **Angular** **JS** & **React JS.**
* Experience in using **Microsoft Azure** including **Azure CLI,** **Azure Management**, **Azure Portal**, **Azure PowerShell**, **Azure Management PowerShell Cmdlets, Red Gate Cloud Services.**
* Experience in **GCP** services such as **App Engine**, **Compute Engine**, **Kubernetes Engine**, **VM Instance**, **Firewall rules**, **Snapshots**, **Instance Templates**, **Health checks**, **Clusters**, **Storage**, **Workloads**, **Monitoring**, **VPC Network**.
* Expertise in **Amazon AWS Cloud** Administration which includes services like **EC2, S3, EBS, VPC, ELB, AMI, SNS, RDS, IAM, Auto scaling, Cloud Front, Cloud Watch, Cloud Trail, OPS Work, Security Groups,** **DynamoDB**, **Certificate Manager**, **Lambda**, **Glacier**, **SQS**, **EMR**, **Storage** **Gateway**.
* Experience in working with **Kubernetes** to automate deployment, scaling and management of web Containerized applications and Integrated **Kubernetes** with Registry, networking, storage, security and telemetry to provide comprehensive infrastructure and orchestrate containers.
* Experience in configuring **Docker File** with different artifacts to make an image and using **Ansible Playbooks** deployed those **Docker Images** to different servers for running the applications on containers.
* Experience in working on **Jenkins** by configuring and maintaining in Continuous Integration (CI) and for End-to-End automation for all build and deployments. Involved in writing **Groovy Scripts** for building **CI/CD** pipeline with **Jenkins file**, **terraform** scripts and **Cloud Formation** Templates Experience on **Maven**, **Ant** to create artifacts **(War, Jar, Tar, Ear)** from source code and deploy them in Nexus central repository for internal deployments.
* Managed the Version Control System **GIT/BitBucket/SVN** to record the various code changes.
* Expertise in detecting network outages and protocol failures by using **Nagios** monitoring system and experienced in configuring other **monitoring** tools like **Splunk**, **Dynatrice**, **CloudWatch** and **ELK** stack (Elastic Search, Logstash, Kibana) **AppDynamics.**
* Experienced in Installing and Querying **RDBMS** such as **Aurora**, **PostgreSQL, Oracle, MY SQL,** **SQL Server, MongoDB** and **Cassandra** for data Integrity.
* Experience in **System Builds**, **System Administration, Server builds, Installs, Upgrades, Patches, Migration, Troubleshooting, Security, Backup, Disaster Recovery, Performance Monitoring and Fine-tuning on UNIX** Red Hat Linux Systems.
* Expertise in automating the complete application using **Python**, **Bash** **Scripts**, **Perl** **Scripts, Windows Power Shell** on cloud environment.
* Expertise on **Linux** **administration** like installing, configuring, tuning and upgrading. With moderate knowledge on shell scripting.
* Experience in Managing/Tracking the defects status by using **JIRA** tool and Planning & resolving the issues as per SLA.

**TECHNICAL SKILLS:**

|  |  |
| --- | --- |
| **Cloud Platforms**  | AWS, Azure, Google Cloud Platform (GCP), PCF.  |
| **Continuous Integration Tools**  | Jenkins, Bamboo, TeamCity.  |
| **Java Technologies** | JDBC, Servlets, JSP, JSTL, Spring MVC, EJB, Spring boot. |
| **Continuous Deployment Tools**  | Docker, Kubernetes Clusters.  |
| **Configuration Management Tools**  | Ansible, Puppet and Chef.  |
| **Source Control Management Tools**  | GIT, Bitbucket and SVN.  |
| **Build Tools**  | Maven, ANT and Gradle.  |
| **Tracking Tools**  | ServiceNow, JIRA.  |
| **Artifact Repositories**  | Nexus and Artifactory.  |
| **Logging & Monitoring Tools**  | Nagios, Splunk, ELK Stack, CloudWatch.  |
| **APM Tools**  | Dynatrace, AppDynamics, New Relic.  |
| **Web/Application Servers**  | Apache, Nginx, JBOSS, Apache Tomcat and WebLogic.  |
| **Database**  | Aurora, Dynamo DB, MongoDB, Oracle, SQL, MySQL.  |
| **Operating Systems**  | Windows, Linux/Unix and MAC OS.  |
| **Scripting languages**  | Shell, Python, SQL, XML, HTML, CSS3, Ruby, JSON, YAML.  |
| **Cloud Migration,** **Infrastructure Spin-up Tools**  | Terraform, CloudFormation and ARM Templates.  |

**Certifications:**

AWS Certified Developer – Associate.

**Professional Experience:**

**Client: Motifworks, Baltimore, Maryland June 2019- Present**

**Role: Cloud/DevOps Engineer**

**Responsibilities:**

* Involved in various aspects and phases of architecting, designing, and implementing solutions in IT infrastructure with emphasis on **Azure** cloud and hybrid solutions.
* Worked in the migration from **GCP** to **Azure** as there are many built-in features and services like built-in disaster recovery as services (**DRaaS**) and backup and has strong advantage in its network of data centers, while Google uses the third-party service and not widely distributed as Azure.
* Contribute to the support forums (specific to **Azure** Networking, **Azure** Virtual Machines, **Azure** Active Directory, Azure Storage) for Microsoft Developers Network.
* Migrated on-premises instances to Azure Cloud using **ARM** subscription with Azure site Recovery and hands-on experience with building and installing servers through **ARM** **Templates**.
* Provided support on configuring **load balancers** like NLB, ARR for multi node environments.
* Working on Azure for highly available customer facing B2B and B2C applications.
* Involved in Setting up a **GCP** Firewall rules in order to allow or deny traffic to and from the **VM's** instances based on specified configuration and used **GCP** cloud **CDN** (content delivery network) to deliver content from **GCP** cache locations drastically improving user experience and latency.
* Experience in migrating the Legacy application into **GCP** platform and managing the GCP services such as Compute engine, cloud storage, BigQuery, VPC, Stack Driver, Load Balancing and IAM.
* Worked on **Google** Cloud components, Google container builders and GCP client libraries and Cloud SDK’s.
* Managing **Google** infrastructure and automation with **CLI** and **API**. Working on Inbound and Outbound services with automation of **Chef**. Deploying multiple resources simultaneously, using Deployment Manager Templates in Google.
* Deployed **Azure** **IaaS** virtual machines (VMs) and Cloud services (**PaaS** role instances) into secure VNets and subnets.
* Acted as build and release engineer, deployed the services by **VSTS** (Azure DevOps) pipeline. Created and Maintained pipelines to manage the IAC for all the applications.
* Involved on migrating SQL Server databases to SQL Azure Database using **SQL Azure Migration Wizard** and used **Python API** to upload agent logs into Azure blob storage and in migration from on-premises to Azure Cloud and created custom images for VM's and Configured **Azure** Backup Service for taking backup of Azure VM and data of on premise to Azure.
* Implemented **Hashicorp** **Terraform** to create, change and improve production infrastructure and maintained versioning infrastructure safely and efficiently.
* Created **Terraform** templates for provisioning virtual networks, subnets, VM Scale sets, Load balancers and **NAT** rules and used **Terraform** graph to visualize execution plan using the graph command.
* Extended the generic process by attaching the **Jenkins Jobs, Web Hook** to all the current **Java** and **Scala**-based projects in **GitHub.**
* Developed Build and Deployment Scripts using build tools **MS Build, Maven** in Jenkins to migrate from one environment to another environment and utilized **Nexus**, **Maven**, **SonarQube**.
* Written **Jenkins Pipeline Groovy Scripts** for Continuous integration and build workflows and used Jenkins for uploading **Artifacts** into **Nexus Repository**.
* Implemented **Jenkins** and built pipelines to drive all microservice builds out to the **Docker** registry and then deployed to **Kubernetes**.
* Written **Chef cookbooks** and recipes to Provision several pre-prod environments consisting of **Cassandra DB** installations, web logic domain creations, and several proprietary middleware installations.
* Involved in writing **Chef cookbooks** for setup and configuration of tools including **Jenkins** and **Artifactory** on remote servers using **REST API** protocol and Created playbooks for deployments and bug fixes with **Ansible**.
* Implemented container-based deployments using **Docker** and designed **Docker** images, Hub and registries and installed **Docker** **Registry** for local upload and download of **Docker** **images** and even from **Docker hub**.
* Implemented **Docker** for convenient environment setup for development and testing. Worked on Docker container snapshots, attaching to a running container, removing images, managing directory structures and managing containers.
* Deployed microservices based applications on **Azure** **Kubernetes** by using **AKS**, Ingress API Gateway, MySQL databases and Cosmo DB for stateless storage of external data, and setup reverse proxy **Nginx servers.**
* Worked on setting up a two-step load balancer setup by configuring a load balancer such as **NGINX** in front of the **Kubernetes Cluster**. Used **Nginx** to configure “back end” for each Kubernetes service to proxy traffic to individual pods.
* Managed **Kubernetes** using **Helm charts**. Created reproducible builds of the **Kubernetes applications**, templatize Kubernetes manifests, provide a set of configuration parameters to customize the deployment and Managed releases of **Helm packages**.
* Deployed and configured **Prometheus** to monitor **Kubernetes** nodes with node-exporter, monitor Kubernetes API.
* Worked on **Dynatrace** tool for monitoring all the **Kubernetes** clusters for efficient cluster visibility, proactive monitoring and triggering action.
* Automated configuration of infrastructure and deployment of web applications on webservers **Apache/httpd** and Java on application servers like **Tomcat** by using **Ansible** **playbooks**
* Implemented REST services by **Golang** with microservices architecture and developed new RESTful API services that work as middleware between applications and third-party APIs using **Golang**.
* Involved in setting up JIRA as defect tracking system and configured various workflows, customizations and plugins for the **JIRA bug/issue tracker.**
* Implemented relational Databases, **NOSQL Database** Management systems like **MySQL, MSSQL, Oracle, PostgreSQL**, **Cassandra**, CouchDB and **Mongo DB**.
* Worked with **PowerShell scripts** to automate the Azure cloud system creation including end-to-end infrastructure, VMs, storage, firewall rules.
* Used **Python** libraries such as paramiko, pycrypto, XML parser and logging libraries to develop automatic storage, networking deployment tool on scale-out environment with **Linux.**

**Environment:** Microsoft Windows Azure, Azure AD, Azure Networking, Azure SQL, Azure Storage, Kubernetes, Virtual Machines, Jenkins, Ansible, Docker, Python, Power Shell, MongoDB.

**Client: State Auto, Columbus, Ohio July 2018- May 2019**

**Role: AWS DevOps Engineer**

**Responsibilities:**

* Designed the data models to use in **AWS Lambda** applications which are aimed to do complex analysis creating analytical reports for **end-to-end traceability** and definition of Key Business elements from **Aurora.**
* Wrote **Lambda** functions in python for AWS Lambda and invoked **python** scripts for data transformations and analytics on large data sets in **EMR clusters** and **AWS Kinesis data streams**.
* Designed scalable and highly available Domain Name System (DNS) within **AWS Cloud** to route traffic to AWS Cloud from users connected using AWS Direct Connect for larger deployments.
* Utilized **AWS** Migration services to migrate on premise **MySQL** Databases to **MySQL** RDS. Used AWS Secrets Manager to safely store the Database Passwords. Created and configured an IPsec VPN tunnel between on premise and AWS using Open swan VPN.
* Provisioning of **EC2** instances, built via **Chef**, and integrated into local area offices in 4 time zones. **Amazon RDS**, **VPC construction**, **Security Group policies**, **IAM**, **Route 53**, **Cloud Formation**, **S3**, **Glacier** and **OpsWorks**.
* Experience with **Terraform** key features such as Infrastructure as code, Execution plans, Resource Graphs, Change Automation and extensively used Auto scaling for launching Cloud instances while deploying microservices and used Terraform to map more complex dependencies and identify network issue.
* Experience in Converting the existing **AWS** Infrastructure to serverless architecture with **AWS Lambda** and Kinesis deployed using **Terraform** templates.
* Installed Pivotal Cloud Foundry (**PCF**) on EC2 to manage the containers created by PCF. Used **Docker** to virtualize deployment containers and push the code to EC2 cloud using PCF.
* Used **Docker** for running different programs on single VM, Docker images includes setting the entry point and volumes, also ran Docker containers and worked on installing Docker and creation of Docker container images, tagging and pushing the images.
* Experience in writing **Docker** files and deploying applications on containers and deployed **Docker** containers by using images stored in repository using **Kubernetes**, both on premise and on AWS cloud.
* Deployed **Kubernetes** cluster on AWS using **Kops** with Multi-Master setup to increase the availability, deployed Consul clusters for service discovery, weaver to perform the subnetting between the **PODS**.
* Deployed Build artifacts using **Ansible** **Playbooks** into **Apache** instances which were integrated using Python and Shell scripts. Used **Ansible-Vault** to Encrypt and Decrypt the file and Deployed on Client servers.
* Worked with **Ansible** for Orchestration of Deployments for various servers and managed **Ansible Playbooks** with Ansible modules,implemented CD automation using Ansible, also involved in managing existing **Tomcat Webservers** and automated the configuration of new servers.
* Implemented **Ansible Tower** for scheduling playbooks and used GIT repository to store playbooks and updated multiple manifests and customized facts.
* Implemented Continuous Integration using **Jenkins** and wrote Jenkins installation script and **Jenkins Pipeline** for multiple projects.
* Documented process documents for **Jenkins**, **Git**, **Bit Bucket**, **CD Pipeline** and **Code** **deployment** guide for Developers, Testers and Production Management.
* Worked in migrating new projects to **Jenkins** and assist developers to refactoring **Maven** Pom.xml to use **JFrog** **Artifactory**.
* Integrated **AWS Elastic Container Service (ECS**) with load balancing and to distribute traffic across containers using application load balancers and network load balancers.
* Provided end-user training for all **GitHub** users to effectively use the tool and coordinate/assist developers with establishing and applying appropriate branching, labeling/naming conventions using **GIT source control**.
* Involved in writing **Maven scripts** for the configuration of the Java applications and executed the Maven builds locally to troubleshoot Java code issue and merging related issues.
* Created monitors, alarms and notifications for EC2 hosts using **Cloud Watch** and used **Nagios** for **monitoring** the services and network. Configured Nagios for new services for onsite severs and **AWS cloud** infrastructure.
* Build and release of Cloud based products containing **Linux** and **Windows** environments, using **Power Shell** and **Python Scripting**.
* Installed and Configured the **Nexus repository** manager for sharing the **artifacts** within the company. Maintained and deployed the **artifacts**, used **UDeploy** release build and produce products for testing and use in production.

**Environment:** AWS Services, Python Scripts, Chef, Ansible, Terraform, Docker, Kubernetes, Jenkins, Git, BitBucket, Maven, JFrog, Nexus, Nagios, UDeploy, Linux, Windows, Power Shell and Python Scripting.

**Client: Capgemini, Hyderabad, India March 2016- April 2018**

**Role: DevOps Engineer**

**Responsibilities:**

* Designing and deploying a multi-tier application utilizing almost all the main services of the **AWS stack (like EC2, S3, RDS, VPC, IAM, ELB, Cloud watch, Route 53, Lambda and Cloud Formation**) focused on **high - availability**, fault tolerance environment.
* Configured and managed **AWS VPC peering** connection between VPCs to route traffic between them by using **IPV4**, **IPV6** addresses to establish communication across multiple EC2 instances.
* Implemented **Chef Recipes** for Deployment on build on internal Data Centre Servers. Also re-used and modified same Chef Recipes to create a Deployment directly into **AmazonEC2** instances.
* Installed and Configured **Chef Server**, **Workstation**, Client servers and nodes. Written several **Recipes**, **Cookbooks** in Chef to automate the environment provisioning, Middleware Infrastructure Installations.
* Worked on **Docker** **Hub**, creating **Docker** **Images** and handling multiple images primarily for middleware installations and domain configurations.
* Configured the **Docker** containers and creating Docker files for various applications as well as created **Docker** images using a **Docker** file, worked on **Docker** container **snapshots**, removing images and managing **Docker volumes**.
* Research and implement new tools such as **Kubernetes** with **Docker** to assist with auto-scaling, continuous integration, rolling updates with no downtime.
* Integrated **GIT** with Jenkins using the **GitHub** plugin to automate the process of source code check-out by providing the URL and credentials of the GIT repository.
* Experience in integrating **JIRA** with **CI/CD** pipeline as defect tracking system and configured workflows to automate deployment and issue tracking.
* Hands on experience on various build tools like **Maven**, **ANT** and integrated these plugins with Continuous integration like **Jenkins.**
* Experienced in User Management and Plugin Management for Jenkins and Configured **Jenkins** for Continuous Integration. Automation of Build using Jenkins pipelines and develop workflow to drive projects to move from TeamCity to Jenkins.
* Developed **Ant** and **Maven** scripts to automate the compilation, deployment and testing of Web and J2EE applications to the above platforms.
* Configured **POM file** to manage **Maven** plugins such as Maven build plugins and Maven Reporting plugins and resolve Maven dependencies required for the project.

**Environment:** Compute, Kubernetes Engine, StackDriver, Elastic Search, AWS services, Chef, Docker, Kubernetes, GIT, Ant, Maven, TeamCity, Jenkins.

**Client: Cybermate Infotek limited, Hyderabad, INDIA Feb 2015-Feb 2016**

**Role: Java Developer**

**Responsibilities:**

* Involved in all phases of **Software Development Lifecycle** including Requirements, Collection, Analysis of the Customer Specifications, Development, Handling Change Request, Bug Fixing, Code Review and Customization of the Application.
* Developed the application using **J2EE** architecture specification and Design patterns.
* Responsible to enhance the UI using **HTML**, **Java Script**, **XML**, **JSP**, **CSS**, **React.js** as per the requirements and providing the client side using jQuery validations.
* Wrote Single Page Web Applications using **RESTFUL** web services, Ajax, Bootstrap, Angular Js.
* Experience in using **jQuery** and **Bootstrap** plugins for the WebPages.
* Worked on Proof of Concept for developing application using **Node** **Js** and **Mongo** **DB**.
* Used **REST** Client and **POSTMAN** to test the rest-based services and used **Junit** and **Mockito** to test the middleware services.
* Experience in CI and CD with **Jenkins** implemented automation of new projects build framework using Jenkins and Maven as build framework tools.
* Used **GIT** as version control system, to keep track of all the work & all changes, allowing several developers to collaborate and used **Maven** as build and configuration tool and **Jenkins** for continuous integration tool.
* Prepared Documentation for **Unit** **Test** Plan and Unit Test Results for the QA purpose. Participated in the release night shakeouts/overnight deliveries and QA Support.

**Environment:** J2EE, HTML, Java Script, XML, JSP, CSS, React.js, jQuery, POSTMAN, MongoDB, Junit, Mockito, GIT, Maven.

**Client: Gryphonsoft, Bangalore, India March 2013- Dec 2014**

**Role: Java Developer**

**Responsibilities:**

* Actively participated in development phase that followed **Waterfall methodology.**
* Used **Core** **Java**, reflection, Collection framework, multithreading, and Open-source integration extensively with **JSP, EJB’s, Servlet**.
* Developed application using **JAVA**, **J2EE** and **Spring Frameworks**.
* Created the tables using **SQL** and connectivity is done by **JDBC**.
* Responsible for coding **SQL** Statements and Stored procedures for back end communication using **JDBC**.
* Used **JBoss** application server to deploy the application.
* Utilized **Ant** to build the project and **CVS** for source version.
* Worked extensively on bug fixes on the server-side applications deployed on **Apache Tomcat** and made cosmetic changes on the UI side.

**Environment:** JAVA, J2EE, Spring Frameworks, SQL, JDBC, JBoss, Ant, CVS, Apache Tomcat.