**Naga Undavalli**

Cloud/Devops Engineer

**Professional Summary:**

Over 10 years of professional experience as Cloud/DevOps Engineer - Automation, Building, Deploying, Managing, and Releasing of code from one environment to other environment.

* Experience in Amazon Web Services (AWS) cloud which includes services like EC2, S3, VPC, ELB, EBS, Glacier, RDS, Aurora, CloudFront, CloudWatch, Security Groups, Lambda, Code Commit, Code Pipeline, Code Deploy, DynamoDB, Autoscaling, Route53, RedShift, CloudFormation, CloudTrail, OpsWorks, Kinesis, IAM, SQS, SNS, SES.
* Experience in Cloud Computing technologies including Infrastructure as a Service, Platform as a Service, and Software as a Service provider (IaaS, PaaS, and SaaS).
* Experience in writing CloudFormation templates in YAML and JSON formats to build the AWS services with the paradigm of Infrastructure as a code.
* Experience in provisioning highly available EC2 Instances by using Terraform and cloud formation and wrote new plugins to support new functionality in Terraform.
* Experience in Terraform for creating stacks of VPCs, ELBs, Security groups, SQS queues, S3 buckets in AWS and updated the Terraform Scripts based on the requirement on regular basis.
* Expertise in creating Docker containers and building Docker images and pushed those images to Docker registry and Deploying and maintaining Micro services using Docker.
* Experience in Configuring the provider with Terraform which is used to interact with resources supported by Kubernetes to create several services such as Config Map, Namespace, Volume, Auto scaler.
* Experience in working on several Docker components like Docker engine, Docker Hub, Docker Swarm and Docker registry. Docker Swarm provides clustering functionality for Docker containers.
* Experience in Configuring Chef server, Chef Workstation, bootstrapping various enterprise nodes and automated the cloud deployments using Chef, Ruby and AWS Cloud Formation Templates.
* Experience in Designing, Installing and Implementing Ansible configuration management system and writing Playbooks for Ansible using YAML for maintaining roles and deploying applications.
* Experience in Ansible and Ansible Tower as a Configuration management tool, to automate repetitive tasks, quickly deploy critical applications and Environment Configuration Files.
* Expertise in Deploying servers using Puppet, and Puppet DB for configuration management to existing infrastructure and Implemented Puppet 3.8 manifests and Modules to deploy the builds for Dev, QA and Production.
* Experience in working with EC2 Container Service plugin in Jenkins which automates the Jenkins Master Slave configuration by creating temporary slaves.
* Knowledge in understanding the principles and best practices of Software Configuration Management (SCM) in Agile, Scrum, and Waterfall methodologies.
* Extensive experience in using Maven, Gradle and ANT as build tools for building deployable artifacts (jar, war & ear) from source code.
* Expertise in using Webhooks for integrating with Continuous Integration tools like Jenkins, TeamCity, Bamboo and ANT, Maven and Gradle for generating builds. Designed quality profiles and certain standards set by installing Quality Gates in SONARQUBE.
* Experience in setting up of end-to-end environment by defining DNS records, Load Balancer VIP's, Apache Proxies and backend Tomcat/WebLogic with registering authentication SiteMinder services.
* Supported Deployments into PROD, Pre-Prod environments with multiple Application server technologies like WebLogic, Jboss, Glassfish and Apache Tomcat.
* Experience in developing UI using JSP, HTML, CSS, JavaScript and NoSQL databases like Cassandra and MongoDB.
* Experience in Designing the application using HTML5, AngularJS, CSS, Ng-Grid, Bootstrap, Web-API, responsive web-design for mobile access.
* Served the ELK (Elastic search, Log stash, Kibana) stack community with use cases and Logstash plugin and actively participated in blogs and QA.
* Experience in Monitoring tools like Nagios, Splunk, AppDynamics and task scheduling tools like Cronjob.
* Experience in implementing use of Nagios and keynote for monitoring and analysing network loads on machines by enforcing custom Nagios monitoring, notifications, dashboard to exhibit various metrics using Shell Scripting.
* Knowledge on involving in setting up of JIRA as defect tracking system and configured various workflows.

**TECHNICAL SKILLS**

|  |  |
| --- | --- |
| **Monitoring Cloud services** | AWS, GCP |
| **Operating System** | Linux, Centos, Redhat, windows, Ubuntu |
| **CI/CD Tools** | Jenkins, Hudson and Bamboo, Nexus, JFrog, Artifactory and SonarQube |
| **Scripting language** | Shell, Perl, Python, Bash Scripts, JSON and YAML |
| **Containerization Tools** | Docker, Packer |
| **Build Tools** | Maven, ANT, MS Build and Gradle |
| **App Servers** | JBOSS, WebLogic and Web Sphere |
| **Methodologies** | Agile, V-Model, Waterfall and UML |
| **SCM Tools** | Clear Case, GIT, Stash and Bit-Bucket |
| **Bug Tracking Tools** | Jira, Fisheye, Crucible, Rally, and Remedy |
| **Web Servers** | Apache, Apache Tomcat and Nginx, WebSphere, Jboss |
| **Orchestration Tools** | Kubernetes, Docker Swarm |

**Work Experience:**

**Client: TikTok, Mountain View, CA March 2023 – Present**

**Role: Sr Cloud/DevOps Engineer**

**Responsibilities**:

* As an DevOps Engineer, worked with application teams to build a platform for deploying their applications.
* Created several Kubernetes clusters depend upon the user’s requirement such as Microservice clusters, Stateful clusters, Stateless clusters and GPU clusters.
* Created monitors, alarms and notifications for EC2 hosts using Cloud Watch and monitored system performance, managed Disk Space LVM (Logical Volume Manger) and performed system Backup and Recovery.
* Built S3 Buckets and managed policies for S3 buckets and used S3 Bucket and Glacier for storage and backup on AWS and created Snapshots and Amazon Machine Images (AMI's) in EC2 instance for Snapshots and creating clone instances.
* Initiated Microservices application through Containerd and Kubernetes Cluster formation for scalability of the application, and creation of Images.
* Extensively used Google Kubernetes Engine (GKE) to manage microservices. GKE’s automatic node scaling, rolling updates, and integration with GCP services streamlined our CI/CD pipelines.
* Configured GCPIdentity and Access Management (IAM) Groups and Users for improved login authentication. Also handled federated identity access using IAM to enable access to our GCP account.
* Implemented Continuous Delivery pipelines using Harness to automate deployment processes across multiple environments, reducing manual intervention and deployment time by 70%.
* Implemented AWS solutions using EC2, S3, Redshift, Lambda, RDS, EBS, Elastic Load Balancer, Auto Scaling Groups, SNS, Optimized volumes and Cloud Formation templates.
* Used Elastic Container Service (ECS) to support Docker containers to easily run applications on a managed cluster of Amazon EC2 instances and used Terraform in AWS Virtual Private Cloud to automatically setup and modify settings by interfacing with control layer.
* implemented Identity and Access Management (IAM) policies to ensure users and services had the least privilege required. We used service accounts for secure automation of deployments and API access.
* Used Cloud Monitoring, Logging, and Trace to monitor system health, troubleshoot issues, and improve app performance across the infrastructure.
* Data in GCP was always encrypted at rest and in transit by default. I worked on ensuring compliance with GDPR and ISO 27001, and conducted regular audits using Cloud Security Command Center.
* GCP’s multi-region and zonal infrastructure allowed us to deploy resilient applications by distributing services across zones, ensuring high availability and disaster recovery.
* We built isolated network environments using Virtual Private Cloud (VPC). Features like firewall rules, custom subnets, and Shared VPC helped us maintain tight control over internal and external traffic.
* Build scripts on AWS cloud for scheduling EC2 Auto Scaling load balancer with python SDK.
* Created Clusters using Kubernetes, kubectl and worked on creating many Pods, Replication controllers, Services deployments, Labels, Health checks and ingress by writing YAML files.
* Developed Amazon Elastic Container Registry for integrating with Amazon ECS and the Docker CLI, for development and production workflows and worked on creation of various subscriptions and topics using SNS and SQS based services and automated the complete deployment environment on GCP.
* Integrated Harness with CI tools like Jenkins and GitLab to streamline end-to-end DevOps workflows and ensure reliable artifact delivery to production.
* Designed AWS CloudFormation Templates to create custom sized VPC, Subnets, NAT to ensure successful deployment of Web applications and database templates which integrated with Amazon API to control instance operations.
* Implemented Packer based scripts for continuous integration with the Jenkins server and deployed those scripts on to the Amazon EC2 instances and customized AMI’s based on already existing AWS EC2 instances by using create image functionality, hence using this Snapshot for disaster recovery.
* Involved in Dynatrace project and installed one-agent on one of the microservice clusters and created dashboards in Dynatrace such as CPU, Memory utilization, Pods status, Node status etc.
* Worked on GPU machines to fix the MPS sidecar issues and also worked with OCI to upgrade NVIDIA Driver and Cuda Version.
* Configured a Google cloud Virtual Private Cloud (VPC) and Database Subnet Group for isolation of resources and created several Cyber Security compliance processes for the organization.
* Configured Harness pipelines for canary and blue-green deployments, improving deployment safety and minimizing downtime during releases.
* Worked as a POC for the core TikTok applications AML and ADS and closely worked with them to understand the pain points that they are facing and assisted them in fixing the issues and finding the RCA for the issues that they encountered.
* Worked with System team and IAAS team for requesting machines such as Bare metals and virtual machines and worked with them to build an INIT script to install the various packages on the machines.
* Worked with several teams to fix the LVM machines which have mount issues and automated the script to convert CPU machines to LVM machines.
* Implemented RBAC policies in Harness to ensure secure access control and compliance with enterprise governance standard
* Managed service configurations and deployment strategies within Harness, ensuring consistency across development, staging, and production environments.
* Did several automations which will help us our day-to-day SRE operations which would be helpful to fix the oncalls quickly.
* Debugged complex software issues in a high-pressure environment, leading to a 30% increase in system efficiency and reliability.
* Automated data pipelines cleaning processes using Python, saving approximately 15 hours of manual work per week.
* Collaborated with a cross-functional team and debugged critical system errors, resulting in a 25% reduction in downtime.
* Provided KT sessions to NOC team to initial triage the oncalls and closely worked with them to setup a Dashboards for monitoring the clusters and assisting them when they have any questions over oncalls.
* Worked closely with E-commerce team to make sure they have got enough quota and cluster requirements to deploy their Applications on our TCE platform.
* Automated the tasks using Ansible Playbooks and migrating the servers with the required configuration changes and testing and deploying the machines using Ansible Commands.
* Led migration efforts from legacy deployment scripts to Harness pipelines, significantly reducing deployment errors and standardizing DevOps practices across teams.
* Integrated Jenkins CI/CD with GIT version control and implemented continuous build based on check-in for applications and created GitHub Webhooks to setup triggers for commit, push, merge and pull request events to drive all builds from SCM Repository and then deployed to Kubernetes, created pods using Kubernetes.
* Participated on a 24/7 oncalls rotation on a monthly basis and creating docs with the issues how to fix them and sharing the docs to team’s knowledge.
* Utilized Harness feature flags to control the rollout of new features and perform A/B testing, improving user experience and minimizing risk.
* Provided KT sessions to the team on the projects that currently working on and on the tools how to leverage the tools for providing better support.
* Worked with several teams across different regions and providing support depends upon the user’s requirement.
* Setup Virtualization technologies VMWare, Virtual box, Vagrant for creating virtual machines and provisioning environments.

**Client: American Express, Phoenix, AZ June 2020 to Jan 2023**

**Role: Cloud/DevOps Engineer**

**Responsibilities:**

* Leveraged AWS services such as EC2, ELB, Auto-Scaling, EC2 Container Service, S3, IAM, VPC, RDS, DynamoDB, Certificate Manager, Cloud Trail, Cloud Watch, Lambda, Elastic Cache, Glacier, SNS, SQS, Cloud Formation, Cloud Front, EMR, AWS Workspaces, Elastic File System, Storage Gateway.
* Design and develop cloud-native applications using AWS services such as ECS, EKS, and Fargate.
* Deployed Application which is containerized using Docker on to a Kubernetes cluster which is managed by the Amazon Elastic container service for Kubernetes (EKS)
* Managed High-Availability, Fault Tolerance, and Auto-scaling in AWS CloudFormation. And configured AWS IAM roles and Security Group in Public and Private Subnets in VPC and created AWS Route53 to route traffic between different regions.
* Architected and implemented containerized microservices architecture using Docker and Kubernetes on AWS EKS.
* Managed the AWS VPC network for the Launched Instances and configured the Security Groups and Elastic IP's accordingly. Worked with Cloud Trail, Cloud Passage, Check Marx, Qualys Scan tools for AWS security and scanning.
* Configuration Automation using Ansible and Docker Containers and Implemented and designed AWS virtual servers by Ansible roles to deployment of web applications. Automation of various administrative tasks on multiple servers using Ansible. Demonstrated on Ansible along with Ansible Tower can be used to automate different software development processes all over the team organization.
* Used Ansible with AWS to reduce costs for the department and eliminate unwarranted resources.
* And Managed AWS infrastructure and automation with CLI and API and working on Inbound and Outbound services with automation Ansible.
* Worked on Hydra Cloud OpenShift 4.11 Version supporting applications that deployed on Hydra Cloud.
* Used ServiceMesh and Istio for the application’s traffic routing and supported Hydra cloud Infrastructure and automated the Hydra cloud Infrastructure using Ansible Playbooks.
* Worked with Redhat team for setting up the Hydra Infrastructure and managed the infrastructure with Capacity planning with more reliability.
* Worked on Terraform for managing the infrastructure through the terminal sessions and executing the Scripts and creating Alarms and notifications for EC2 instances using Cloud Watch.
* Created custom workflows using Harness' YAML-based templates, promoting reusability and maintainability of deployment configurations.
* Developed Architecture, Designing and operationalization of Red Hat OpenShift environment. application containerization management with OpenShift.
* Administering and supporting the OpenShift Container Platform ecosystem for the organization in both public cloud and on premise and experienced in Container management using OpenShift and Kubernetes.
* Utilized Convolution Neural Networks to implement a Machine Learning image recognition component
* Used Multiple Machine Learning algorithms, including random forest and boosted tree, SVM, SGD, neural network, and deep learning using TensorFlow.
* Integrated Jenkins with various DevOps tools such as Nexus, SonarQube and used CI/CD system of Jenkins on Kubernetes container environment, utilizing Kubernetes and Docker for the runtime environment for the CI/CD system to build and test and deploy.
* Developed Production environment of different applications on AWS by provisioning Kubernetes clusters on EC2 instances using Kubernetes Operations (KOPS), a cluster management tool to spin up a highly available production cluster.
* Created, managed and performed container-based deployments using Docker images containing middleware (Apache Tomcat) and Applications together and evaluated Kubernetes for Docker container orchestration.
* Involved in customer interaction programs to discuss the Application Health’s, Application build and the issues that Application team is facing.
* Integrated Harness with cloud platforms (AWS, GCP, Azure) for seamless deployments to EKS, ECS, GKE, and AKS clusters.
* Managed Docker orchestration using Kubernetes to orchestrate the Deployment, Scaling, and management of Docker Containers.
* Created and deployed Kubernetes pod definitions, tags, labels, multi-pod container replication and managed multiple Kubernetes pod containers scaling, and auto-scaling using KOPS.
* Deployed pods using Replication Controllers by interacting with Kubernetes API server defining through declarative YAML files.
* Setup Kubelet, Kubernetes Master/Worker nodes as well as API Server and scheduler to orchestrate the deployment of instance in real-time.
* Monitored deployment health using Harness’ in-built verification steps and integrated observability tools (like Prometheus, Datadog, and New Relic.
* Created customized Docker Images and push them to Google Compute engine, worked on Docker and deployed and maintaining Micro Services in Dev and QA, implemented Jenkins slaves as Docker containers auto scalability.
* Assisted users on day-to-day Tickets that were created in ServiceNow.

**Client: United Airlines, Chicago, IL. July 2019 to March 2020**

**Role: Senior Cloud Engineer**

**Responsibilities:**

* Configured OpenShift Cluster on both AWS Cloud and On-prem cluster with four Environments Eng, Dev, QA and Prod.
* Patched the OpenShift servers by Evacuating the Pods on the Nodes after that from that Nodes connect to the Spacewalk server where all the repos would be stored and patch the OpenShift servers.
* Developed Kubernetes with Rancher on AWS EKS and bare metal with oracle.
* Added Nodes to the existing OpenShift Cluster either Master or Infra or Worker node by using the Scaleup Manifest file.
* Implemented Prune Jobs to the OpenShift Cluster for Images, Projects and Resource Builds.
* Used EFK stack for OpenShift-logging, Elasticsearch, Fluentd and Kibana, and used to monitor the logs for the Applications that are deployed on the OpenShift Cluster.
* Implemented Quotas to the Projects in OpenShift cluster like Small, Medium, Large and X-large Quotas for the Resource limits and the Pod’s limits in that project.
* Used Prometheus for OpenShift-Monitoring, Alert-Manager for triggering alerts to the Big-Panda and Grafana for running data analytics, pulling up metrics and monitoring the apps with the help of customizable dashboards.
* Configured Datadog as a Daemon Set for Monitoring the OpenShift Metrics and checking the PV and PVC utilization.
* Wrote Yaml files for the PV and PVC and implemented Persistent Volume to the cluster and Persistent Volume Claim to the Apps.
* Configured the Logging-Curator for clearing the space in the Elasticsearch Persistent Volume and changing the Curator Default logs as requested by the App users.
* Resetting the Kibana User’s Index whenever the user fails to login to Kibana and unable to view the logs.
* Utilized OpenShift Cluster as a platform for automating the Deployments, Scaling and Operation of Application containers across a Cluster of hosts and worked closely with Application teams.
* Worked on creating Pods, Replication controllers, Services deployments, Labels, Health checks and ingress by writing YAML files.
* Used both CLI and Web UI in OpenShift-cluster and Deployed Images through Source-to-image(S2I).
* Worked on end-to-end setup of the Artifactory pro as a Docker Container with a secure private Docker Registry and local Docker repositories for storing the built Docker Images.
* Used Amazon ECR for hosting images in a highly available and scalable architecture and allowing to deploy containers for applications Integration with AWS Identity and Access
* Created AWS Multi-Factor Authentication (MFA) for instance RDP/SSH logon, worked with teams to lock down security groups and created IAM roles so AWS resource can securely interact with other AWS
* Utilized AWS CLI to automate backups of ephemeral data-stores to S3 Buckets, EBS and created AMIs for mission critical production servers as backups.
* Automated the tasks using Ansible Playbooks and migrating the servers with the required configuration changes and testing and deploying the machines using Ansible Commands.
* Enhanced the automation to assist, repeat and consist of configuration management using Ansible based YAML scripts and worked on deployment automation of all the Microservices to pull image from the private Docker Registry and deploy to OpenShift Cluster using Ansible.
* Used ServiceNow for ticketing tool and created the Production change request and the Storage request for the Servers.
* Worked with the storage team and created the NFS Storage for the OpenShift Servers for storing the Pod’s data on On-prem Servers.
* As an OpenShift Cluster Admin, Used LDAP server for providing access to the Application users.
* Developed Cron Jobs for the Logging-Curator and for the Master Backups.
* Built Jenkins pipelines to drive all micro-services builds out to the Docker Registry and then deployed to OpenShift cluster and created Pods and managed by using OpenShift.
* Created and managed Virtual Memory (swap spaces) and Filesystems, while also supporting data management through on-site and off-site storage and retrieval services.
* Configured and worked on scripts for DNS look up tests on net group, auto mounting and unmounting the shares at Linux end.
* Used GIT for creating the Local Repo, Cloning the Repo, adding, committing and pushing the changes in the local repo and creating and maintaining GIT Repositories also analysing and resolving conflicts related to merging of source code to GIT.

**Environment**: OpenShift, AWS, Kubernetes, CloudFormation, IAM, Docker, Ansible, Jenkins, Git, Elasticsearch, Fluentd, Kibana, Prometheus, Alert-Manager, Grafana, Datadog, ServiceNow, Linux, YAML

**Client: ATOS, Chicago, IL. July 2018 to May 2019**

**Role: DevOps Engineer.**

**Responsibilities:**

* Configured, Monitored and Automated GCP Services as well as involved in deploying the content cloud platform using Google Compute Engine, Google Storage Buckets and created Google storage buckets and maintained and utilized the policy management of these buckets and Glacier for storage and backup on Google cloud.
* Built custom data pipelines feed automation tool to read Stackdriver metrics onto on premise VALET dashboard such as Volume, Availability, Latency, Errors and Tickets and used Google API's to get the metrics and created the metrics.
* Created the Dataflow pipeline to continuously sending the logs from Stackdriver to GCS bucket
* Automated Project creation, Network Firewall and Compute Instance creation using Terraform.
* Managed IAM policies with active directory integration to manage security in GCP and AWS. Configured Hybrid Cloud setup on GCP using VPN with two different regions and used Google Cloud console to create and manage GCP and GKE workloads.
* written the Python script to send the Stackdriver logs using Cloudfuntion with integration of Pub/Sub and BigQuery and automated all the infrastructure workflows using Terraform.
* • Exported the Stackdriver logs to Pub/Sub and from the Pub/sub sending data to the GCS bucket and the Big Query.
* Created Composer environment and worked on the Airflow for scheduling the jobs by using the Dags and accomplished the POC which is Triggering the Dags by using the REST-API calls from On-Prem Unix Server.
* Used Data prep for converting the Raw data to the Redefined data and used Cloud Storage bucket for storing that data and from Cloud Storage exported the data to PostgreSQL and the Big Query.
* Created Identity-Aware-Proxy for O-Auth Authentication for triggering Dags from the On-prem server by using the Rest-Api calls and used Python Scripts for Dags for scheduling the jobs
* Installed the Syslogng server on the local machine and with the help of the Logstash exported the logs from pub/sub to the Syslogng server
* Writing new plugins in Nagios to monitor resources and working in implementation team to build and engineer servers on Ubuntu and RHEL Linux provisioning virtual servers on VMware and ESX servers using Cloud.
* Configured, supported and maintained all Network, Firewall, Load balancers, Operating systems in AWS EC2 and created detailed AWS Security groups which behave as virtual firewalls that controlled the traffic allowed reaching one or more AWS EC2 instances.
* Integrated Amazon Cloud Watch with Amazon EC2 instances for Monitoring the log files, store them and track metrics. Created AMI’s to implement automatic deployments of application components and bootstrapping AWS EC2 Instances by passing user data to download files from S3.
* Configured a Kubernetes Cluster on GKE and managed, production-ready environment for deploying containerized applications and deployed the Kubernetes dashboard to access the cluster via its web-based user interface.
* Created Clusters using Kubernetes, kubectl and worked on creating many Pods, Replication controllers, Services deployments, Labels, Health checks and ingress by writing YAML files.
* Developed and Test environments carrying different operating system platforms like Windows, Ubuntu, Red Hat Linux, Centos, Unix.
* Used JIRA for creating bugs tickets, storyboarding, pulling reports from dashboard, creating and planning Sprints

**Environment**: GCP, GCE, GKE, App Engine, Stackdriver, Pub/Sub, Cloud Function, BigQuery, Dataflow, Cloud Shell, VPC, AWS, EC2, Python, Cloud watch, Kubernetes, Windows, Linux, Jira.

**Client: Toyota, Ann Arbor, MI. February 2017 to May 2018**

**Role: DevOps Engineer**

**Responsibilities:**

* Setting up of CI/CD pipeline using continuous integration tools such as Cloud Bees Jenkins and automated the entire AWS EC2, VPC, S3, SNS, RedShift, EMR based infrastructure using Terraform, Chef, Python, Shell, Bash scripts and managing security groups on AWS and custom monitoring using CloudWatch.
* Created an AWS RDS Aurora DB cluster and connected to database through an Amazon RDS Aurora DB Instance using Amazon RDS Console and used BOTO 3 and Fabric for launching and deploying instances in AWS and configured Inbound or Outbound in AWS Security groups according to the requirements.
* Leveraged AWS S3 service as Build Artifact repository and created release-based Buckets to store various modules/branch-based Artifact storage.
* Used Terraform templates along with Packer to build images for application deployment in AWS.
* Created Kubernetes YAMLs using objects like Pods, Deployments, Services and ConfigMaps and created reproducible builds of the Kubernetes applications, managed Kubernetes manifest files and Helm packages.
* Used Docker to containerize custom web applications and deploy them on Ubuntu instance through SWARM Cluster and to automate the application deployment in cloud using Vagrant.
* Created a microservice environment on cloud by deploying services as a Docker container and used Amazon ECS as a container management service to run micro services on a managed cluster of EC2 instances.
* Implemented Docker Containers to create images of applications and dynamically provision slaves to Jenkins CI/CD pipelines and reduced build and deployment times by designing and implementing Docker workflow.
* Involved in writing Jenkins file by using Groovy Scripts for building CI/CD pipeline for automation of Shell Scripts.
* Configured Jenkins jobs to automate build create Artifacts and Execute unit tests as part of the build process. Also, the integrated build process with Sonar for Code Quality analysis.
* Used Knife and Chef Bootstrap processes and worked on Chef Server management console with proficient knowledge on all different components like Nodes and Workstations.
* Worked with Chef Enterprise Hosted as well as On-Premises, Installed Workstation, Bootstrapping the Nodes and used Chef Ohai to collect system configuration data, which is provided to the Chef-Client for use within the Cookbooks to determine the System State.
* Used Ruby scripting on Chef Automation for creating Cookbooks comprising all resources, templates, attributes and used Knife commands to manage Nodes.
* Used ANSIBLE role to create an ELK cluster for non-log purposes to search and analytics of product data and pricing data.
* Designed ELK (Elastic search, Logstash, Kibana) system to monitor and search enterprise alerts installed, configured and managed ELK Stack for Log management within EC2 / Elastic Load balancer for Elastic Search.
* Used AWS Beanstalk for deploying and scaling web applications and services developed with Java, PHP, Node.js, Python, Ruby, and Docker on familiar servers such as Apache, and IIS.
* Involved in setting up application servers like Tomcat, WebLogic across Linux platforms as well as wrote shell scripts, Bash, Perl, Python, Ruby scripting on Linux.
* Used JIRA for creating bugs tickets, storyboarding, pulling reports from dashboard, creating and planning Sprints.

Environment: AWS, Packer, Cloud Bees, Jenkins, Terraform, Kubernetes, Docker, Docker Swarm, Ansible, Chef, Python, Bash Scripts, Shell Scripts, YAML, Groovy Script, Git, Maven, ELK, Splunk, Nagios, Ubuntu, RHEL, Java, PHP, Ruby, Jira.

**Client: Gartner, Stamford, Connecticut. August 2015 – January 2017**

**Role: AWS Engineer**

**Responsibilities:**

* Involved in designing and deploying multitude applications by utilizing almost all the AWS stacks including EC2, Route53, S3, RDS, Dynamo DB, SNS, SQS, IAM focusing on high-availability, Fault tolerance, and Auto-Scaling in AWS Cloud formation.
* Migrated Production Infrastructure into an Amazon Web Services utilizing AWS Cloud Formation, Code Deploy, EBS and Ops Works. And Deployed and migrated applications using AWS CI/CD tools like Code Pipeline, Code Commit.
* Setting up private networks and sub-networks using Virtual Private Cloud (VPC) and creating security groups to associate with the networks and set up and administer DNS system in AWS using Route53.
* Configured AWS Identity and Access Management (IAM) Groups and Users for improved login authentication. Also handled federated identity access using IAM to enable access to our AWS account.
* Used Elastic Container Service (ECS) to support Docker containers to easily run applications on a managed cluster of Amazon EC2 instances and used Terraform in AWS Virtual Private Cloud to automatically setup and modify settings by interfacing with control layer.
* Written Terraform templates and pushed them onto Chef for configuring EC2 Instances and solved Gateway time issue on ELB and moved all the logs to S3 Bucket by using Terraform.
* Converted existing Terraform modules that had version conflicts to utilize CloudFormation templates during deployments and to create Stacks in AWS, and updated these scripts based on the requirement on regular basis.
* Initiated Microservices application through Docker and Kubernetes Cluster formation for scalability of the application, and creation of Docker Images to upload or download in and out from Docker Hubs.
* Created Clusters using Kubernetes, kubectl and worked on creating many Pods, Replication controllers, Services deployments, Labels, Health checks and ingress by writing YAML files.
* Used Docker as a Container management for writing Docker file in JSON format and place the automated build in Docker Hub and managed deployments using Kubernetes and created local clusters and deployed application containers.
* Worked on infrastructure with Microservice models like Docker Containerization and collaborated with development support teams to set up a Continuous Delivery environment with the use of Docker.
* Installed and administered Artifactory repository to deploy the Artifacts generated by Maven and to store the dependent jars which are used during the Build.
* Worked with Jenkins for any automation buildings which are integrated with GIT as part of infrastructure automation under continuous integration (CI).
* Developed Version control of Chef Cookbooks, testing of Cookbooks using Test Kitchen and running recipes on Nodes managed by on premise of Chef Server.
* Deployed and configured Chef Server and Chef Solo including Bootstrapping of Chef Client Nodes for provisioned and created Roles, Cookbooks, Recipes, and Data Bags for Server configuration.
* As an ELK developer, I worked on all the internal tools. Designed, deployed and coordinated with different teams to enhance ELK platform and took ownership of new technologies
* Configured Apache webserver in the Linux AWS Cloud environment using Chef Automation and evaluated Chef Framework tools to automate cloud deployment and operations.
* Experienced in authoring pom.xml files, performing releases with the Maven release plugin in Java projects and managing Maven repositories.
* Developed Cron jobs and Shell Scripts and Python for automating administration tasks like file system management, process management, backup and restore.
* Developed Splunk Queries and dashboards targeted at understanding application performance and capacity analysis and worked on the setup of various reports and alerts in Nagios.
* Designed and administered databases for Oracle, MySQL to support various web programming tasks.

**Environment**: AWS, GKE, Terraform, Kubernetes, SWM, Docker, Git, ANT, BitBucket, Maven, Jenkins, Chef, Ruby, Nagios, Cacti, Zabbix, Splunk, Shell Scripts, Python, Nginx, Apache, JSON, Vagrant, WebLogic, Oracle, MySQL, Java.

**Client: Lowes Company Inc, Mooresville, North Carolina November 2014– July 2015**

**Role: DevOps Engineer**

**Responsibilities:**

* Worked on AWS CloudWatch for monitoring the application infrastructure and used AWS Email services for notifying.
* Configured, supported and maintained all Network, Firewall, Load balancers, Operating systems in AWS EC2 and created detailed AWS Security groups which behave as virtual firewalls that controlled the traffic allowed reaching one or more AWS EC2 instances.
* Migrating a production infrastructure into an Amazon Web Services utilizing AWS CloudFormation, Code Deploy, EBS and Ops Works.
* Created docker files for each microservice's and changed some of the Tomcat configuration file which are required to deploy Java based application to the Docker Container.
* Used Git for deployment scaling and load balancing to the application from dev through prod, easing code development and deployment pipeline by implementing Docker Containerization with multiple name spaces.
* Created, tested and deployed an End-to-End CI/CD pipeline for various applications using Jenkins as the main Integration server for Dev, QA, Staging, UAT and Prod environments.
* Created Puppet automation with multiple modules as per component like MySQL, Http collectors & Schema registry to install and configure EC2 instances to implement release schedules, created Rollout Plans, tracked the Project Milestones.
* Built and managed a highly available monitoring infrastructure to monitor different application servers and its components using Nagios, with Puppet Automation.
* Worked with Chef Ohai plugin to push jobs and exposure to Chef Supermarket to leverage the existing Cookbooks for quick automation of general deployment and Infrastructure tasks.
* Involved in developing JUnit Test Cases to validate the type of data in the XML Files. Used Log4J for logging and tracing messages.
* Implemented and maintained the branching of build/release strategies utilizing Clear Case.
* Environment: AWS, Docker, Chef, Puppet, Git, Maven, Ant, Jenkins, Java, Kafka, Zookeepers, MySQL, Nagios, XML, Log4J, Junit, Clear Case, Apache Tomcat, JDk, Spark

**EDUCATION**: Master of Science (Webster University, USA)