

Name: Avinash Kukutla
Sr Cloud/DevOps Engineer
Email: avinash.kukutla@outlook.com
Phone: 408-658-6923

PROFESSIONAL SUMMARY

- Around 8 years of extensive experience in “**Automating, configuring and deploying**” instances on cloud environments and experience in the areas of “DevOps, CI/CD Pipeline, Build and release management and Linux/Windows Administration”.
- Experience in **Software Development Life Cycle (SDLC)** including requirements **analysis, design specification, Building, Migration, Training, Deployment** and **Production Support**
- Extensive experience in **Linux/Unix** system Administration, System Builds, Server Builds, Installations, Upgrades, Patches, Migration, Trouble shooting on **RHEL 6.x/7.x**
- Experience of **Jenkins, Bamboo, VS Build Pro, Apache Ant, Apache Tomcat, Subversion, Git, Maven**
- Expertise in using build tools like **MAVEN** and **ANT** for the building of deployable artifacts such as war & ear from source code. Good experience in converting **build.xml** into **pom.xml** to build the applications using **MAVEN** and used tools like **SonarQube** to check the integrity of code and **Nexus** and **JFrog**
- Experience in managing the artifacts generated by **MAVEN** in the **Nexus repository**, Implemented and maintained **Continuous Integration (CI)** and **Continuous Deployment (CD)** Pipelines in **Jenkins** using the Tools like **Git, Bit Bucket, Maven, Cucumber, JUnit, Nexus Repository, Ansible, Docker, Kubernetes** within environments such as **DEV, QA, STAGING, and PROD.**
- Involved in the functional usage and deployment of applications in **WebLogic, WebSphere, and Apache Tomcat Server** and Web Containers like **Nginx** etc
- Experienced in developing **Ansible roles** and **Ansible Playbooks** for the server configuration and Deployment activities. Used **Puppet** as a configuration Management tool, Created Modules and Manifests in **Puppet** to automate various application
- Professional experience dealing with **Windows Azure IaaS**, migrating, and creating **AZURE VMs**, storage accounts, Cloud Services, VPN, Load Balancing, VHDs, storage pools, migrating on premise servers to **AZURE** and creating availability sets in **AZURE.**
- Experience in Configuring and Deploying instances on **AWS & AZURE DevOps**
- Experience in Designing, Architecting, and implementing scalable cloud-based web applications using **AWS** and **GCP**
- Set up a **GCP** Firewall rules 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
- Worked on **Chef/Puppet cookbooks/recipes** to automate **Infrastructure as a code** and used scripting languages for automation such as **Ruby, Shell, and Python**
- Worked with AWS Code Pipeline and Creating Cloud formation **JSON** templates which is converted to **Terraform** for infrastructure as code
- Created **Python** scripts to automate **AWS Services** which includes **Cloud front, Lambda, ELB, database security and application configuration** also developed the scripts to take backup of **EBS volumes** using **AWS Lambda, CloudWatch**
- Experience in Server infrastructure development on AWS Cloud, extensive usage of **Virtual Private Cloud (VPC), IAM, Cloud Formation, Cloud Front, EC2, RDS, EBS, ELB, Elastic IP, S3, Microsoft Azure, Route53, SNS, SQS, Cloud Trail**
- Setup and build AWS infrastructure various resources, **VPC EC2, S3, IAM, EBS, Security Group, Auto Scaling, RDS** in **Cloud Formation JSON** templates
- Installed, configured multiple operating systems onsite and provisioned similar instances on **AWS cloud** and set up data in **AWS** using **S3bucket** and configuring instance backups to **S3 bucket**
- Hands-on Experience working on Migrations from On-Premises to **AWS Cloud**, created cloud front for accelerated delivery of websites, APIs, and content by routing the traffic using **Route 53**
- Provisioning **Cassandra DB cluster** on **AWS EC2** and **ELB** with automated **Terraform** scripts. Defined **Terraform** modules such as Compute, Network, Operations, and Users to reuse in different environments
- Source code management is performed using **Git** from master repository and knowledge on Container management using **Docker** in Creating images
- Knowledge on **Docker** based container deployments to create self-environments for dev teams and containerization of environments delivery for releases
- Configured and administered Monitoring tools like **Nagios, Splunk, and AWS Cloud Watch** etc
- Excellent analytical, problem solving communication and interpersonal skills. Providing 24*7 support for various production related environments and troubleshooting

TECHNICAL SKILLS

- **Cloud Platform:** Azure, Amazon Web Services, GCP
- **Operating Systems:** Red hat Linux 4/5/6/7, Windows servers 2003, 2008, 2008 R2, 2012, Windows 7/8/10.
- **Languages/Technologies:** Groovy, Python, HTML, XML, CSS, SQL, PL-SQL, PHP, JavaScript, Servlets, Bootstrap, Shell Script, Ruby
- **Databases:** MongoDB, Oracle, SQL SERVER, My SQL, MS Sql.
- **CI/CD:** Jenkins, Azure DevOps, Circle CI
- **Configuration Management tools:** Chef, Ansible, Puppet
- **Version Control tool & Build tools:** Git, GitHub, Maven, ANT,
- **Network Protocols:** TCP/IP, UDP, DHCP, HTTP, VPN, DNS, NTP, FTP, SSH, and Telnet
- **Application Server:** Web Logic 9.x, 10.x, Apache Tomcat 5.x/7.x, JBoss 4.22. GA Web Sphere 6.x/7.x/8.x, Microsoft Active Directory 2000/2003/2008/2008 R2, DHCP, DNS, WINS
- **Virtualization:** VSphere 5.0/4.0/ 4.1, ESX/ESXi server 5.0/4.1/4.0/3.5/3.0 , V Sphere Client, VCenter, VMotion, Storage VMotion, VMware Update Manager, VMware Tools, VMware Converter, DRS, HA
- **Infrastructure As a Code (IAC):** Terraform, CloudFormation, ARM templates
- **Containerization & Orchestration :** Docker, Kubernetes, OpenShift

PROFESSIONAL EXPERIENCE

Client: SunNet Solutions Corporation, Houston, TX

Jan 20 - present

Role: Sr Cloud/DevOps Engineer

- Designing, planning and implementation for existing On-Premises applications to **Azure Cloud**, Create Cache Memory on **Windows Azure** to improve the performance of data transfer between SQL Azure and WCF services, created **Azure key-vault** in order to store all the credentials for **SQL databases** and for API keys
- Created Continuous Integration /Continuous Deployment Pipelines in **Azure DevOps** environments by providing their dependencies and tasks. Configured and deployed Azure Automation Scripts utilizing **Azure stack** Services and Utilities focusing on Automation
- Created **Azure** Automation Assets, **Graphical runbook**, **PowerShell** runbook that will **automate** specific tasks, deployed **Azure AD** Connect, configuring **Active Directory Federation Service (AD FS)** authentication flow, **ADFS** installation using **Azure AD Connect**, and involved in administrative tasks that include **Build, Design, Deploy** of **Azure environment**
- Deployed **Kubernetes clusters** on top of **Azure** using KOPS, Build **Jenkins pipelines** to drive all micro services builds out to the Docker registry and then deployed to Kubernetes, Created Pods and managed local deployments in Kubernetes creating local clusters and deploying application containers
- Created several pods using Master and Minion architecture of **Kubernetes** and developed **microservice** on boarding tools leveraging Python allowing for easy creation and maintenance of build jobs and Kubernetes deploy and services
- Building/Maintaining **Docker container clusters** managed by **Kubernetes, Linux, Bash, GIT, Docker** on **Azure** and used **Kubernetes** and **Docker** for the runtime environment of the CI/CD system to build, test deploy
- Deploying and maintaining multi-container applications through **Docker**. Worked on **Docker** container snapshots, attaching to a running container, removing images, managing directory structures, and managing containers
- Experience working on several **Docker** components like **Docker Engine, Hub, machine, compose and Docker registry**.
- Experience working on **Docker** Hub, creating **Docker** images, and handling multiple images primarily for middleware installations and domain configurations
- Worked on creation of **Docker** containers and **Docker** consoles for managing the application life cycle, worked on setting up the automated build on **Docker HUB**
- Worked with **Terraform** Templates to automate the Azure virtual machines using **terraform modules** and deployed virtual machine scale sets in production environment
- Written Templates for Azure Infrastructure as code using **Terraform** to build staging and production environments. Integrated **Azure Log Analytics** with Azure VMs for monitoring the log files, store them and track metrics and used Terraform as a tool, Managed different infrastructure resources Cloud, VMware, and Docker containers
- Wrote **Ansible Playbooks** using **Python modules** to manage configurations of **Open stack** Nodes and Test Playbooks on **AZURE**

using **Python**

- Implemented multi-tier application provisioning in **OpenStack cloud**, integrating it with **Ansible**, Integrated **GIT** into **Jenkins** to automate the code checkout the process, configured plugins for integration tools to the **version control tools**
- Administered and Engineered **Jenkins** for managing weekly **Build, Test, deploy chain** and **GIT** with Migrating the applications across environments such **DEV/Test/Prod** Branching Model for weekly releases. Created a fully Automated Build and Deployment Platform and coordinating code builds promotions and orchestrated deployments using **Jenkins/Bamboo**, and **GitHub**
- Build out server automation with Continuous Integration – Continuous Deployment tools like **Jenkins/Maven** using tools like **Git, Bit Bucket, Maven, Cucumber, JUnit, Nexus Repository, Ansible, Docker, Docker central, Kubernetes** and **Oracle DB** for deployment and build management system. Defined **branching, tagging** and **merge** strategies for all applications in **GIT**
- Analyzed the **ANT** build projects for conversion and converting to **Maven** build projects, installation and configuration of **Jenkins** to support Java builds and Jenkins Plugins to automate continuous builds and publishing **Docker images** to **Nexus Repository**
- Worked on **google cloud platform (GCP) services** like compute engine, cloud load balancing, cloud storage, cloud SQL, stack driver monitoring and cloud deployment manager
- Setup **GCP** Firewall rules 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
- Configured and monitored infrastructure using **Nagios, Splunk, ELK stack (Elasticsearch, Logstash, Kibana)**

Client: Charter Communications St. Louis, MO

Mar 18 - Jan 20

Role: Site Reliability Engineer

- Hands on experience in **Amazon Web Services (AWS)** provisioning and good knowledge of **AWS cloud services** such as **EC2, auto-scaling** and **VPC** to build secure, highly scalable, and flexible systems that handled expected and unexpected load bursts
- Designing and implementation of public and private facing servers on **AWS Cloud**, Configured and managed various **AWS Services** including **EC2, RDS, VPC, S3, Glacier, Cloud Watch, Cloud Front, and Route 53** etc. Created users, roles and groups using **Identity Access Management (IAM)** and attached the required policies
- Worked with **AWS Code Pipeline**, Configured and Managed **EC2, Elastic Beanstalk, RDS, S3 Buckets, VPC, Security Groups, ELB, Auto scaling, Elastic Cache, Glacier, SNS, SQS, SES, CloudWatch, ECS, CloudTrail, CloudFormation**
- 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. Migrating On-Premises Infrastructure to **AWS Cloud** focusing on high-availability, fault tolerance, and auto scaling
- Experience with container-based deployments using **Docker**, working with **Docker** images, Docker HUB and Docker registries
- Setting up the build and deployment automation for **Terraform** scripts using **Jenkins** and Provisioned the highly available EC2 Instances using **Terraform** and **Ansible Playbooks**
- Wrote playbooks for **Ansible** and deploying applications using **Ansible**. Worked with **AWS Cloud Formation Templates, terraform** along with **Ansible** to render templates with Orchestration templates in OpenStack Environment
- Manage the **build activities** of multiple teams, delivering multiple **releases** of services and applications. Coordinate with the **Development, Database Administration, QA, and IT Operations teams** to ensure there are no resource conflicts
- Build, manage, and continuously improved the build infrastructure for the **development, staging and production software environments** and interacted with client teams to understand client deployment requests
- Created **JIRA issues** to prioritize and act on what is important and stayed up to date with what is going on around the project
- Continuously look to improve the scalability of the **build, release** and **deployment** processes and operation. Coordinate and support project branches merging process with **source code control**
- Evaluated **Ansible** framework and tools to automate the cloud deployment and operations. Created and implemented chef **cookbooks** for deployment and used Chef Recipes to create a Deployment directly into **Amazon EC2 instances**
- Creating CloudFormation **JSON** templates to create custom **VPC**. Migrated a production infrastructure into AWS utilizing Code Deploy, Code Commit, Knowledge of Amazon ECS Clusters and able to manage containers in it
- Used Ansible to Setup **ELK stack (Elasticsearch, Log stash, Kibana)** and troubleshoot the build issues with ELK and work towards the solution. Utilized **Ansible** and **AWS lambda, elastic cache** and **CloudWatch** logs to automate the creation of log aggregation pipeline with **Elasticsearch, Logstash, Kibana stack (ELK stack)** to send all our team's logs coming into CloudWatch, to process them and send them off to Elasticsearch

- Configured **POM** file to manage Maven plugins such as Maven build plugins and Maven Reporting plugins and resolve **Maven** dependencies required for the project
- 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
- Installed, configured, and administrated all **LINUX** servers on **Amazon Web Services (AWS)**, includes the design and selection of relevant hardware to Support the installation/upgrades of **Red Hat, CentOS** operating systems
- Developed automated deployment scripts using **python** to deploy war files, properties file and database changes to development server or **QA server** and **Staging/Production server**. Developed the automated **build** and **deployment** using **ant** and **Jenkins** using **tomcat**

Client: Sophos, Burlington, MA

Aug 16 - Mar 18

Role: DevOps Engineer

- Experience in **Amazon Web Services (AWS)** provisioning and good knowledge of AWS services like **IAM, EC2, Elastic Load-balancers, Elastic Container Service, S3, Elastic Beanstalk, Cloud Front, Elastic File system, VPC, Route53, Cloud Watch, Cloud Trail, Cloud Formation**
- Defined AWS Security Groups which acted as virtual firewalls that controlled the traffic allowed to reach one or more AWS EC2 instances
- Resources like Amazon RDS, **Kubernetes** cluster which hosts production servers are hosted in private subnets in a **VPC**, Security to infrastructure is achieved through **NACLs**, VPC flow logs, Nat gateways, Bastion Hosts, MFA, LDAP implementation
- Working on Multiple AWS instances, set the security groups, Elastic Load Balancer and AMIs, Auto scaling to design cost effective, fault tolerant and highly available systems
- Used **AWS Cloud Watch** as a monitoring tool to check the health status of the virtual instances
- Plan, deploy, monitor, and maintain Amazon **AWS cloud infrastructure** consisting of multiple EC2 nodes and Vm's as required in the environment
- Worked on creating the **Docker containers** and **Docker consoles** for managing the application life cycle.
- Automating Server Builds using **Chef**, Installed and configured **Splunk** on Linux servers for log analysis, Installed Altiris agent for patching on Windows servers
- Designed DevOps workflow for multiple applications by orchestrating Test, Build, Release and Deploy phases through various CI/CD pipelines using **Git, Jenkins, Docker, Chef & Cloud formation tools**
- Managed and Maintained code repository in **Git**, improved practices of branching and code merge for custom needs of development team
- Automation of new job creation by implementing and managing **Jenkins** DSL. Setup Jenkins master/slave architecture to use Jenkins as pipeline as service
- Hands on experience with Jenkins **CI/CD** pipeline involves Git repository, Maven build and post build steps like notification and continuous delivery in the **Dev, Test** and **UAT** environment
- Created jobs for builds and deployments, installed several plug-ins in Jenkins to support multiple tools required for the implementation of projects
- Configuring Jenkins pipeline for auto deployment through continuous integration (Jar build, **AMI** creation, Integration test, contract test, performance test, deploy in Test environment and then into Production)
- Deployed and configured Elastic search, Fluentd and Kibana (**EFK**) for log analytics, full text search, application monitoring in integration with AWS Lambda and X-Ray
- Configured the logs using Prometheus locally and Forwarding the logs to **Splunk** using **Fluentd**

Client: BNSF Railway, Fort Worth, TX

Feb 16 - Aug 16

Role: Build & Release Engineer

- Developed build and supported the **Software Release Management** procedures and deployment processes
- Responsible for design and maintenance of the **Subversion/CVS** Repositories, views, and the access control strategies. Performed all necessary day-to-day **Subversion/CVS** support for different projects. Created the branches in Subversion and Perforce to implement the parallel development process

- Used **Jenkins** innovatively to automate most of the build related tasks. Improved throughput and efficiency of build system by providing EO/managers rights to trigger required build. Configured and maintained **Jenkins** to implement the CI process and integrated the tool with Ant and Maven to schedule the builds
- Merged release branches to the trunk after the production release and resolved the conflicts rose during the merge in both Subversion and Perforce
- Implemented **GitLab** for version control of puppet modules and process documentation, used **JIRA** for change control & ticketing.
- Installed and administered **Artifactory** repository to deploy the artifacts generated by **Maven** and to store the dependent jars, which are used during the build
- Implemented **Puppet Version 3** for **Automation** in the Non-production environment and utilize Puppet for configuration management of hosted Instances within AWS
- Handled Code reviews and merging Pull requests, resolving merging issues during build and release by conducting meetings with developers and managers. Responsible for designing and deploying best **SCM** processes and procedures
- Worked on **ANT and MAVEN** Scripts to automate the build process. Implemented **Maven** builds to automate **JAR and WAR**
- Developed build and deployment scripts using **ANT** and **MAVEN** as build tools in Jenkins to move from one environment to other environments
- Worked closely with software developers and DevOps to debug software and system problems. Developed and maintained integration and release processes, create scripts for system administration and **AWS** using languages such as **BASH** and **Python**
- Responsible for Connectivity issues among various Servers and various software components as well as setting up network environments using **TCP/IP, NIS, NFS, DNS, SNMP** agents, **DHCP** and **Proxy**
- Deployed the applications to **Web sphere Application Server** and static content to **apache web servers**

Client: Infinite Computer Solutions, Bangalore, India

Oct 14 - Dec 15

Role: Linux Administrator

- Extensively involved development process using Struts framework for handling web requests using Action classes
- Designed and developed front-end pages using struts framework, **JSP, JavaScript, jQuery, AJAX and HTML**
- Worked extensively on **DOM** to create dynamic elements using **JavaScript**
- Designed Use Case diagrams, Sequence diagrams, Class diagrams, and ERD diagrams for the application
- Used **Spring MVC** (Model View Controller) to handle/intercept the user requests and used various controllers to delegate the request flow to the Backend tier of the application
- Used **Oracle SQL** Developer to create, debug and deploy application modules
- Analysis, design and development of Application based on **J2EE using Struts, Spring and Hibernate**
- Worked on the Web Services using **SOAP** and Restful web services
- Hibernate was used for **Object Relational mapping** with Oracle database
- Created stored procedures, functions, triggers using **PL/SQL** to interact with the Oracle database
- Log4j was used for logging mechanism
- Used **Ajax** for interactive web pages. The presentation layer front end was enhanced using AJAX technology, implemented single-screen refresh, instant validations (server based validation), and dynamic menu system
- Compiled Spring Bean configuration files. Used **XSLT** to transform and render the XML output retrieved by the AJAX queries
- Validating XML documents with Schema using **SAX** and **DOM** parser
- Wrote Shell scripts to automate **Jboss** deploy process
- Used **JUnit** as a unit testing framework to prepare unit test plans and used **WinCVS** for version control

Client: NCR Corporation India PVT Ltd – Hyderabad, India

May 2013 - Oct 14

Role: Linux Administrator

- Managed and administrated of all **UNIX** servers, includes **Linux** operating systems by applying relative patches and packages at regular maintenance periods using **Red Hat Satellite server, YUM, RPM** tools
- Installation and administration of **RHEL 6.x** and Configured kickstart server and updating patches to the servers using Red Hat Satellite server. Tune the Servers and configuring networks for optimum performance
- Planned and performed the upgrades to **Linux (RHEL 5x, 6x, SUSE 10, 11, CENTOS 5, 6)**, operating systems and hardware maintenance like changing memory modules, replacing disk drives

- Setting up network environments using **TCP/IP, NIS, NFS, DNS, SNMP agents, DHCP and Proxy**, user account maintenance, backup & recovery, Auto-mounting, License Management, Printer configuration
- Installed and configured monitoring tools Nagios for monitoring the network bandwidth and hard drive status. Monitoring CPU, memory, physical disk, Hardware and Software RAID, multipath, file systems, network using the tools **NAGIOS 4.0** monitoring Performing failover and integrity test on new servers before rolling out to production
- Deployment and Configuration of application server **Tomcat** deploying **Java** and **Web Application**
- Planned, scheduled and Implemented OS patches on **Linux** boxes as a part of proactive maintenance
- Used **Chef** for managing server application server such as **MySQL, Apache Tomcat** Installation, configuration, and customization of services **Send mail, Apache, FTP** servers to meet the user needs and requirements.
- Writing **Shell scripts** for automation of daily tasks, documenting the changes that happen in the environment and in each server, analyzing the **error logs**, analyzing the **User logs**
- Identify, troubleshoot, and resolve problems with the OS build failures
- Performing **kernel** and database configuration optimization such that it limits I/O resource utilization on disks

Education

Master of Science, Information Science/studies

Trine University, MI

Bachelor of Technology, Computer Science Engineering

Sana Engineering College, India