Vikas B

Sr. DevOps Engineer

E-mail: vikas.0212@outlook.com

Phone: (732)-902 0212

A highly skilled and experienced DevOps Engineer with a proven track record of success in delivering complex, multi-cloud solutions. Possesses deep expertise in CI/CD, build and release management, source code management, configuration management, containerization, and orchestration. Demonstrated ability to design and deploy secure, scalable, and reliable multi-cloud environments, including networking, system administration, and development.

**Technical Skills:**

|  |  |
| --- | --- |
| Languages | Perl, Bash, PowerShell, Python, Go, Ruby, Node |
| SCM Tools | Subversion (SVN), GitHub, GitLab |
| Build Tools | Ant, Maven |
| CI / CD | Docker, Jenkins, Code-Build, Code-Commit, Artifactory, Kubernetes, Bit-Bucket, GitLab CI-CD, Tekton |
| Monitoring Tools | Splunk, Nagios, Cloud Watch, Prometheus, Grafana, Sumo Logic |
| Cloud Platforms | AWS, Azure and GCP(Associate) |
| Configuration Tools | Chef, Puppet, Ansible, Terraform, CloudFormation |
| Servers | WebLogic, WebSphere, JBOSS, Apache Tomcat |
| Platforms | UNIX, Linux, Microsoft Windows Vista/ XP/ 2000 |
| App-frameworks | Serverless, Containerized, Single Page Apps, Monolithic and Micro-services. |

**Certifications:**

* AZ-400: Designing and Implementing Microsoft DevOps Solutions
* Google Cloud Certified: Professional Cloud DevOps Engineer
* AWS: Certified Solutions Architect – Professional

**Professional Experience:**

**Microsoft**

**Sr. Software Engineer**

**Redmond, WA**

**June 2024 – Current**

**Responsibilities**

* Designed and implemented a complex Azure infrastructure using Azure Bicep, ensuring modularity, scalability, and maintainability of resources.
* Configured and deployed critical Azure services, including AKS (Azure Kubernetes Service), Azure Firewall, Azure Bastion, Private Endpoints, and Virtual Network Peering for secure and scalable architecture.
* Developed reusable Bicep modules for networking, security, storage, and monitoring, enabling seamless integration and environment-specific customizations using parameterized files.
* Automated validation and deployment processes using custom Bash scripts, ensuring infrastructure consistency and reducing manual effort.
* Established private connectivity through Private Endpoints and secured access using Azure Firewall and Azure Bastion, adhering to best practices in cloud security.
* Integrated Log Analytics, Azure Monitor, and Application Insights for real-time observability and performance monitoring of deployed resources.
* Successfully implemented modular deployment workflows, allowing for easy updates and maintenance without disrupting the entire infrastructure.
* Optimized deployment workflows for multi-environment use (e.g., development and production) by leveraging parameterized configurations.
* Led the migration of infrastructure as code (IaC) from Terraform to Azure Bicep, enhancing readability, modularity, and native integration with Azure services.
* Converted complex Terraform configurations into Bicep templates, simplifying the management and deployment of Azure resources in a multi-tier architecture.
* Utilized Bicep’s native support for Azure Resource Manager (ARM) templates to achieve better compatibility and seamless deployment workflows.
* Enhanced infrastructure management by leveraging Bicep’s modular approach, reducing code duplication and improving the reusability of resource definitions across different environments.
* Migrated Terraform state management to Bicep, ensuring a smooth transition and maintaining consistent infrastructure states during deployment cycles.
* Implemented parameterized Bicep files to allow environment-specific configurations, simplifying the management of development, staging, and production environments.
* Re-architected the deployment pipelines to use Azure Bicep in CI/CD processes, streamlining infrastructure updates and improving deployment reliability.
* Improved team productivity by leveraging Bicep’s declarative syntax and tooling, enabling faster onboarding of new team members and reducing the learning curve.
* Addressed migration challenges by performing extensive testing and validation of Bicep templates to ensure parity with the previously defined Terraform infrastructure.
* Integrated Azure Monitor, Log Analytics, and Application Insights for comprehensive monitoring and diagnostics, ensuring real-time observability of the infrastructure and applications.
* Deployed Prometheus and Grafana to provide custom metrics and dashboards for monitoring the performance and health of resources, especially in Kubernetes clusters.
* Used Azure Security Center and Azure Sentinel to enhance security posture management and threat detection across the migrated infrastructure.
* Documented the migration process, creating a comprehensive knowledge base to guide future migrations and infrastructure evolution within the organization.

**Apple**

**Site Reliability Engineer**

**Austin, TX**

**February 2023 – May 2024**

**Responsibilities**

* Hands-on experience in migrating workloads to **AWS**, ensuring a seamless transition and minimal downtime.
* Worked on maintaining **EKS** clusters, ensuring performance, monitoring, and scaling as needed.
* Hands-on experience in using **Kustomize** for managing multi-environment deployments in **Kubernetes** and maintaining the deployment manifests for different environments.
* Experience in building enhanced CI/CD using **Tekton**, **Kubernetes**, and **Git** and writing custom tasks for **Tekton** pipelines and pipeline files.
* Worked on POC for migrating the existing **Kubernetes** CI/CD pipelines from **Tekton** to **Argo** **CD** and integrated the **Argo CD** for continuous delivery.
* Worked on POC and successfully migrated from **Nginx** Ingress controller to **Emissary** **Ingress** to streamline traffic management and API gateway functionalities for **Kubernetes** based applications.
* I have worked on creating customized **Kubernetes** pod templates with Groovy in Jenkins pipelines for the production environment.
* Worked on creating the **Cloud** **formation** **templates** (**CFT**) for creating custom roles and delegating permissions to them.
* Created **CFT** to automate the deployments in **AWS** **EKS** **Kubernetes** clusters.
* Deployed **Prometheus** and **Grafana** to monitor the **Kubernetes** clusters and, configured alerts, Integrated **Splunk** for pod health management.
* Migrated **EKS** self-managed node group to **EKS**-managed node group using custom AMIs using Node launch templates.
* Worked on diagnosing the cluster, troubleshooting the issues in the pod logs, and finding and fixing the root causes.
* Implemented **Ansible** playbooks and roles for automated configuration management, deployment, and orchestration of infrastructure, ensuring consistency and efficiency in operations.
* Worked on creating custom assume roles in **Kubernetes** and integrating them with **AWS** IAM, for fine-grained access control and enhanced security.
* Worked on creating **IAM** roles and infrastructure deployment using cloud formation templates (**CFT**).
* Worked on POC for seamlessly upgrading the **Kubernetes** cluster from 1.22 to 1.24 and was able to complete the transition.
* Worked on creating automation, CI, deployment, and delivery automation using **Groovy**, **JSON**, **YAML**, and **bash**.
* Worked on configuring the **Tekton** CI/CD pipeline to trigger changes in SRC, build the tagged **Docker** image of the application and deploy the changes to the **Kubernetes** cluster to keep the cluster updated.
* Created custom roles and role bindings in **Kubernetes** using **RBAC** to define fine-grained access control. Integrated with **LDAP** for user authentication, ensuring fine grained access management to the **Kubernetes** cluster.
* Maintained Security compliance by performing the regular code quality analysis (**CQA**) and configuring the **RBAC** for the **Kubernetes** clusters, regularly updating the images, and developing the Incident response plan.
* Worked on configuring the **mTLS** between deployments for encrypted authentication and updating the deployment files to mount the appropriate secrets as volumes.
* Worked on configuring **CoreDNS** to forward requests to the ingress service, leveraging **CoreDNS** for more flexibility, which replaced **kube-DNS** for **DNS** resolution in the Kubernetes cluster.
* Worked on integrating **Splunk** with a **Kubernetes** cluster to collect pod data for resource utilization and cluster health. Configured **Splunk** to publish alerts to **Slack** based on the collected data.
* Hands-on experience in troubleshooting and resolving issues, enforcing security and access controls, planning for disaster recovery optimizing performance and managing deployments, and planning for capacity and version upgrades.

**Microsoft**

**Sr. Azure DevOps Engineer**

**Redmond, WA**

**May 2021 – January 2023**

**Responsibilities**

* Hands-on experience in planning, designing, and developing strategies for deploying **JavaScript** and **Python** applications onto open-source package managers npm, PyPI, ACR, and **Docker** Hub.
* Worked on writing **Docker** files, developing **Docker** images, and releasing them to **Azure** Container Repository (**ACR**) and **Docker** hub.
* Experience in developing new CI/CD pipelines from scratch to automate the build, testing, and deployment stages of applications.
* Worked on migrating CI/CD pipelines from Travis CI to **Azure** DevOps (**ADO**) and **Azure** DevOps (**ADO**) to **GitHub** **Actions**.
* Worked on migrating the CI/CD pipelines from one org to another in **Azure** DevOps (ADO) and worked on GitHub issues involving source code modifications.
* Worked on configuring the **Azure** file share and integrating them with **REST** APIs and created documentation for mounting the file share on the developer's local machine.
* Used **Azure** **Kubernetes** Service (AKS) to deploy a managed **Kubernetes** cluster in **Azure** and created an **AKS** cluster in the **Azure** portal, with the **Azure** CLI using the **ARM** template.
* Worked on creating infrastructure automation scripts in PowerShell and integrating them with **Azure** **Blob** **storage** and other **Azure** services.
* Worked on debugging and diagnosing the issues in the source code and releasing the fixed versions to various application platforms.
* Manually installed the **Linux** servers, configured remote connections on them, added them as manually hosted agents in the **Azure** DevOps from scratch, and configured CI/CD pipelines to run on them.
* Worked on diagnosing the device connectivity issues for **IoT** devices and configuring them for new machines.
* Worked on creating an **MQTT** connection using **Azure** IoT hub for interacting with the manually hosted devices on the **Azure** DevOps.
* Worked on troubleshooting using the **Azure** IoT SDKs for **C#**, **Java**, **Node.js** and integrating IoT devices with **Azure** **IoT** Hub.
* Hands-on source code debugging for identifying, isolating, and resolving bugs in packages using **Typescript** and **JavaScript**.
* Administered and authored **Azure** Device Twins to enable to Remote configuration and status reporting to monitor the health of the IoT devices.
* Worked on writing rules for **Azure** Autoscaling to limit the compute resources and **Azure** Load Balancer (**ALB**)in the **AKS** to scale the infrastructure and optimize the costs.
* Configure the **Azure** Service Bus and Web jobs for passing the messages between **REST** and **WEB** **API** to decouple them to ensure load balancing, scalability, safely routing data, and control across service and application boundaries.
* Managed **Azure** Container Registry (**ACR**) to store private **Docker** images, which are deployed, and Configured **Azure** Monitor to collect metrics and logs to track performance and maintain security.
* Configured **BGP** (Border gateway protocol) routes to enable **ExpressRoute** connections between the on-premises data centers and **Azure** cloud. Designed Network Security Groups (**NSGs**) to control inbound and outbound access to network interfaces (**NIC**s), **VM**s, and **subnets**.
* Worked on onboarding new projects to **Azure** DevOps and worked on configuring the manually hosted agents.
* Worked on configuring the **Azure** **Monitor** for the telemetry collection and facilitating application insights.
* Worked with **Wireshark** to capture the outputs of the IoT devices for testing the ThreadX module in the **Azure** RTOS.
* Maintained and administrated **Git** **Hub**, created branches, performed tagged deployments in GIT Hub, and implemented and maintained the branching and release strategies.

**Cardinal Health**

**Sr. DevOps Engineer**

**Twinsburg, OH**

**May 2019 – April 2021**

**Responsibilities**

* Experienced in writing **ARM** templates for creating infrastructure in **Azure** Cloud using **Azure** DevOps as a CI/CD tool and saving the templates in the resource control repository.
* Created a three-tier architecture in **Azure** Cloud using subnets, DNS, load balancer (**ALB**), and application gateway and created a web server and maintained databases on the private subnets.
* Deployed a Kubernetes cluster in Azure using Azure Kubernetes Service (**AKS**). Created an **AKS** cluster in the Azure portal, with the **Azure** CLI, and used template-driven deployment options such as Resource Manager Templates (**ARM**) and **Terraform**.
* Integrated Applications insights from **Azure** monitor with Visual Studio App center for monitoring the live applications.
* Manage **AWS** Environment via **Terraform** for Ops Code and IAM via CloudFormation, CloudFormation Drift Detection Automation via Lambda.
* Author **Terraform** (**TFE**) Sentinel Policies for Infrastructure components like Networking, Storage, and Account Baselines.
* Built **AWS** EC2 Image Factory via **Terraform**, **AWS** Inspector for scanning the Image, SSM Documents to harden any Vulnerabilities, **AWS** Lambda, and **AWS** SNS for Customized notification (status and inspector reports) to Infra and Security teams.
* Authored **Helm** Charts to package **Kubernetes** native applications written in **Python** and Java.
* Designed and deployed **Gitlab** CI files to deploy core Infrastructure (**Terraform** Custom Modules)
* Performed Code Quality Analysis Techniques like checking styles, finding bugs, NUnit, and JUnit, with the CI tool **Jenkins**. Implemented testing framework for infrastructure ops code (**Python**) using tox and pytest.
* Maintained version control setup of configuration files, automount maps, software repos, etc. using **Git** repositories.
* Configured the **Ansible** Tower by writing **Ansible** playbooks, so clients can run them easily by just a push button.

**Precise Software Solutions Inc**

**Sr. Cloud Engineer**

**Rockville, MD**

**April 2018 to May 2019**

**Responsibilities**

* Implemented app logging service using logging tool **Splunk** and deployed stacks using **AWS** cloud formation templates.
* Authored **Terraform** Modules to create custom sized **VPC**, subnets, and **NAT** to ensure successful deployment of Web applications and database templates.
* Hands-on experience in building cloud infrastructure from scratch using **EC2**, **ASG**, **ELB**, **Route53**, **DynamoDB** and **VPC**. Chef is used as the provisioning tool along with **Terraform**.
* Created **Terraform** templates that can be used as modules by passing the parameters.
* Implemented **Ansible** to manage all existing servers and automate the build/configuration of new servers. All server types were fully defined in **Ansible** so that the newly built server could be up and ready for production.
* Authored **Docker** files for app team references, worked on **Docker** container snapshots, removed images, and managed **Docker** volumes.
* Worked on the end-to-end setup of the Artifactory as a **Docker** container with a secure private **Docker** registry and local **Docker** repositories for storing the built **Docker** images.
* Created **Kubernetes** clusters using Kops replication controllers, services, deployments, labels, health checks, and ingress by writing Kube config files. Created as a **Terraform** module for team self-service.
* Managed **Kubernetes** charts using **Helm**. Created reproducible builds of the **Kubernetes** applications, managed **Kubernetes** manifest files, and managed releases of Helm packages.

**OpenText**

**DevOps Engineer**

**Austin, TX**

**September 2016 to April 2018**

**Responsibilities**

* Worked on Microsoft **Azure** (Public) Cloud to provide support to clients by creating Virtual Machines (**VM**) through **PowerShell** Script and **Azure** Portal. Created, and managed virtual Network End Points, Storage Accounts, and Affinity Group in **Azure** Portal.
* Experience in using Microsoft **Azure** including **Azure** CLI, **Azure** Management, **Azure** Portal, **Azure** **PowerShell**, Cloudmonix, and Red Gate Cloud Services.
* Configured **Azure** Virtual Machines by creating and managing storage resources, including disk striping for performance improvement, and utilized **Azure** **Backup** to configure and restore backups for Azure Virtual Machines. Additionally, managed storage accounts in the Azure Portal for storing data used by Azure services.
* Deployment of Cloud service including Jenkins and **Nexus** using **Terraform**.
* Worked on **Ansible** to manage the infrastructure such as Access policy and permissions, Load balancers, and Autoscaling policy.
* Written **Ansible** playbooks for various DB configurations to modularize and optimize product configuration and server provisioning using **Ansible** playbooks.
* Worked with **Ansible** to manage the containers and the environments around the containers.
* Implemented a continuous delivery pipeline involving **Jenkins**, and **Ansible** to complete the automation from commit to deployment.
* Container management using **Docker** by writing **Docker** files and setting up the automated build on **Docker** HUB.
* Virtualized the servers using the **Docker** for the test environments and dev-environments needs.

**Aspect Software**

**AWS Cloud Engineer**

**Phoenix, AZ**

**January 2015 to August 2016**

**Responsibilities**

* Implemented a Continuous Delivery framework using **Jenkins** and **Maven** in a **Linux** environment.
* Involved in Building and configuring **Red** **Hat** **Linux** servers remotely using Kick Start Servers.
* Deployment and Configuration onto the application server **Tomcat**.
* Experience in developing and deploying applications using **Maven**. Monitoring, maintaining, and upgrading application servers and My SQL databases along with configuring replication.
* Built servers using **AWS** by importing volumes, launching EC2 instances, configuring RDS (Relational Database Service), creating security groups, setting up auto-scaling, and configuring load balancers (**ELBs**) within the defined virtual private connection (**VPC**).
* Wrote **Python** and **Perl** scripts for automated deployments, especially in handling all the tasks.
* Maintained build-related scripts developed in shell for **Maven** builds. Created and modified build configuration files including POM.xml.
* Created **Ansible** playbooks for different environments for release and converted **Puppet** scripts into the **Ansible** Playbooks
* Worked with **Ansible** on various deployment configurations.
* Created multiple custom **Nagios** checks and event handlers to monitor and maintain servers.
* Used **Nagios** as a monitoring tool to identify and resolve infrastructure problems before they affect critical processes and worked on **Nagios** Event handlers in case of automatic restart of failed applications and services.

**Sonata Software Limited**

**Build and Release Engineer**

**Hyderabad, India**

**June 2014 to December 2014**

**Responsibilities:**

* Experience creating DevOps strategy in a various distribution of **Linux** and **Windows** alongside using DevOps tools suites like **Subversion** (**SVN**), **GIT**, **CVS**, **ANT**, **Maven**, **Jenkins**, **Chef**, **Nagios**, and **Splunk** in traditional environments, as well as cloud environments like **AWS**, Open stack.
* Administered Linux servers for several functions including managing **Apache**/**Tomcat** server, mail server, and MySQL databases in both development and production.
* Automated both .**Net** and **Java** Applications using the industry's best automation tool **Jenkins**.
* Author shell scripts (Bash), **Ruby**, **Python**, and **PowerShell** for setting up baselines, branching, merging, and automation processes across the environments using **SCM** tools like **GIT**, Subversion (**SVN**), Stash, and **TFS** on Linux and Windows platforms.
* Created a **Nagios** instance from scratch to include all servers in all environments.
* Implemented the setup for master-slave architecture to improve the Performance of **Jenkins**.
* Hands-on Experience in using configuration management tools like **Puppet**/ **Chef**.
* Deployed **Puppet** for configuration management to existing infrastructure.
* Coordinate/assist developers with establishing and applying appropriate branching, labeling/naming conventions using **GIT** source control.