jdk3410
United States

jdk3410@jdk3410.com
https://jdk3410.com

SKILLS

Expert: AWS/GCP/Azure Compute, VMWare, Powershell, Bash, Apache, MySQL, MSSQL, Bigfix, Terraform
Proficient: Python, Jenkins, Docker, Kubernetes, Ansible, Cloudformation, Helm, GitHub Actions, git, Prometheus

EXPERIENCE

Principal DevOps Engineer

January 2021 - Present

- Automated migration of monitoring platform using configuration management software and scripting, saving support staff 100s of hours of manual work
- Wrote Python Lambda function to check for dev EC2 instances with >6 hours uptime and power them off saving 60% on monthly AWS spend
- Using AWS Cloudwatch and Lambda, created a Lambda function in Python which checks for a keyword on a website and sends an alert via SNS if the keyword is missing resulting in dramatically fewer outage incidents and increased observability
- Migrated dev EC2 instances into Docker containers to reduce administrative overhead and lower AWS spend by 50%
- Installed and configured Ansible, wrote playbooks to install LAMP stack, configure updates and firewall rules in line with industry best practices, decreasing need for oversight by administrators by 90%
- Developed Infrastructure as Code framework to quickly spin up dev environments into GCP for testing which diminished testing time by 95%
- Wrote automation library of 30 scripts in Bash and Powershell for server provisioning tasks
- Coached team in Agile methodologies and fostered blameless DevOps culture, resulting in a 25% improvement in project success rate and a 10% increase in team morale

DevOps Engineer

January 2018 - January 2021

- Developed and implemented a CI/CD pipeline, reducing deployment time by 50% and increasing team productivity by 25%
- Developed and maintained automation scripts for deployment, monitoring, and maintenance, reducing manual effort by 60% and improving overall system reliability by 30%
- Implemented and managed automated monitoring processes for customer monitoring portal, leading to a 75% reduction in resolution time for outage events
- Wrote Bigfix fixlets in Actionscript, powershell, and bash to do various tasks during provision and post-provisioning, such as online disk resize, install and configure Apache, configure NTP, which lessened provisioning time by 30% and curtailed need for post-provision configuration by 95%
- Automated VMWare template provisioning via Powershell and PowerCLI. Automated Redhat, Ubuntu, CentOS, and Windows template creation
- Participated in secure coding workshops, Attended All Day DevOps annually, participated in code reviews, ushering in a period of zero security incidents on our team

Lead Systems Administrator
August 2005 - January 2018

- Orchestrated a team of 10 Systems Administrators to manage daily operations, overseeing a vast array of 6500+ Linux, FreeBSD, and Windows dedicated servers and firewall devices, resulting in a 40% improvement in team productivity and a 15% reduction in production issues
- Managed the implementation of security policies and procedures, resulting in a 50% reduction in security incidents and a 30% increase in system security
- Developed Python automation scripts to streamline Cisco Firewall deployment, resulting in a 50% reduction in provisioning time
- Spearheaded the complete replacement of a legacy monitoring system, leading to a remarkable 99% enhancement in outage response effectiveness
- Evaluated various firewall platforms meticulously, focusing on speed, reliability, and security aspects
- Implemented virtualization technologies to migrate critical internal infrastructure onto Ubuntu/KVM and ESXi hosts, achieving an impressive 80% improvement in outage resolution time

PROJECTS

- Resume: Demonstrates a React Resume / AWS CI/CD setup where entire infrastructure is deployed without user intervention https://github.com/jdk3410/resume
- Wrote Terraform file to spin up EC2 instances and install Kubernetes using kubeadm with number of worker nodes specified in file. .tf stored in Github, update of file triggers terraform apply via Github actions
- Using GitHub Actions and Terraform, wrote a project that allows a spec file with instance name and webserver content to committed to a directory on a GitHub repo. Once committed, GitHub actions parses the file and creates the relevant variables from the input and runs Terraform which provisions an EC2 instance with the desired name and webserver content. Other than committing the spec file, no interaction is needed from the user to provision an EC2 webserver with a simple webpage https://github.com/jdk3410/ghtfdeploy