

JESSICA CLAIRE

100 Montgomery St. 10th Floor
(555) 432-1000 - resumesample@example.com

SUMMARY

Self-motivated individual and Enthusiastic Senior DevOps Engineer. Passionate for technology and I have a proven track record of thriving in fast-paced environments. I am always searching for the latest tools and methodologies. Being able to work confidently by myself has allowed me to become proficient in a wide variety of technologies whilst my strong interpersonal skills have enabled me to excel in teams also. The tools and technologies I use most include Jenkins, Ansible, Kubernetes, Terraform, Docker, Linux, AWS, and Azure.

SKILLS

Cloud Services

AWS, Microsoft Azure, Google Cloud Platform

Infrastructure as Code (IaC)
Terraform, AWS Cloud Formation.

Operating Systems

Red Hat, CentOS & SUSE, Ubuntu, Solaris, DEBAIN, Windows.

Scripting/Programming Languages

Python, Perl, Shell, Groovy, Bash

Version Control Tools

GIT, GitHub, Bit Bucket.

Build Tools

Apache Maven, Gradle, AWS Code Build

Artifact Storage

CORE COMPETENCIES

- Experienced in Build/Release management using different cloud services of **AWS** (Amazon web service), and **Microsoft Azure** as major infrastructure.
- Demonstrated expertise in designing and providing high-quality products and feasible solutions with strong focus on quality and agility, as well as encouraging and automating deployments on **AWS** to drive overall business growth.
- Expertise in **DevOps**, Configuration Management, Cloud Infrastructure, End-to End Automation which includes Azure, Amazon Web Services, Jenkins, Maven, GitHub, Ant, SonarQube, Ansible, Chef, Tomcat, and Linux.
- Expertise in creating DevOps strategy in a mix environment of **Linux** (RHEL, CENTOS) servers and Windows along with (**AWS**) Amazon Web Services. Built and Maintained Docker container clusters managed by **Kubernetes**, **Linux**, **Bash**, **GIT**, and Docker on **AWS**.
- Experienced in installing and maintaining different application and web servers like **Web Sphere**, **Web logic**, **Apache Tomcat**, **Nginx**, **WebLogic** and **JBoss**.
- Worked on Azure resources such as **AKS**, **ACS**, **VPN**, **Azure VM**, **VMNet**, **VMSS**, **Azure AD**, **Azure Data Services**, **Azure Storage**, **Azure Databases (SQL & CosmosDB)**, **Azure Backup**, Azure Site Recovery, Azure Batch, API Management, Azure App Services.
- Extensively involved in writing **Infrastructure as a code (IaC)** in **Terraform**, **Azure resource management (ARM)**, **AWS Cloud Formation (CFT)**, and created reusable **Terraform modules** for both Azure and AWS cloud environments.
- Experience in **Kubernetes** to deploy scale, load balance, and manage Docker containers with multiple names spaced versions using Helm charts and managed Kubernetes services and **AWS** Kubernetes services.
- Developed Shell and Python Scripts used to automate administrative tasks and automation of the build and release process.
- Experience in created monitors, alarms & notifications for **EC2** hosts using **CloudWatch**. Implemented Kubernetes to deploy scale, **load balance**, scale and manage **Docker** containers with multiple names spaced versions.
- Extensively worked on **Jenkins** for continuous integration and for End-to-End automation for all build and deployments by writing Groovy scripts.
- Knowledge of **databases** like MySQL, MSSQL, PostgreSQL, Cassandra, Oracle, IBM DB2 and NoSQL.
- Created AWS EBS volumes for storing application files for use with AWS EC2 instances whenever they are mounted to them and installed Pivotal Cloud Foundry (PCF) on EC2 to manage containers created by PCF.
- Experience in working on **Docker Hub**, creating **Docker-images**, and handling multiple images primarily for middleware installations and domain configuration.
- Good Knowledge of Implementing Configuration Management Tools **Puppet**, **Chef** and **Ansible** and creating **Infrastructure as Code (IaC)**.

EXPERIENCE

- 11/2020 to Current **DevOps Engineer**
Radancy – Burbank, IL
- Extensive experience in designing and implementation of **Continuous Integration**, **Continuous Delivery**, **Continuous Deployment** through **AWS** and **Jenkins**.
 - Experience in Amazon Web Services (**AWS**) platform and its features including **IAM**, **EC2**, **EBS**, **RDS**, **Cloud Watch**, **Cloud Formation**, **Autoscaling**, **Cloud Front**, **S3**, **SQS**, **SNS**, and **Route53**.
 - Designed, configured, and managed public/private cloud infrastructures utilizing Amazon Web Services (**AWS**) including **Virtual Private Cloud (VPC)**, Public and Private Subnets, **Security Groups (SG)**, Route Tables, **Elastic Load Balancer (ELB)**, Network ACL's and NAT Gateways.
 - Worked on implementing a 'Serverless' architecture using **API Gateway**, **Lambda** and **Dynamo DB** and deployed **AWS Lambda code** from Amazon S3 buckets. Created a Lambda Deployment function and configured it to receive events from your S3 bucket.
 - Migrated few Cloud formation templates to **Terraform** by using various **AWS** modules and Secured the **Terraform** state file in **S3** buckets.
 - Designed and implemented security standards in cloud and on premise as per CIS (Centre for Internet Security) benchmarking for hardening multiple AWS accounts through automation scripts using Terraform.
 - Extensively involved with Infrastructure as Code (IaC), execution plans, resource graphs and change automations using Terraform. Also, managed AWS infrastructure as code using Terraform.
 - Performed application migration to **Docker** and **containers** for various projects and established the use of **Blue/Green** Deployments, **Load Balancers** and **AWS EC2** spot instances.
 - Responsible for analyzing various cross-functional, multi-platform applications systems enforcing **Python** best practices and provide guidance in making long term architectural design decisions.
 - Used Git **Webhooks** and Poll SCM with Jenkins to automate **Jenkins** tasks and responsible for writing **Groovy** scripts for **Jenkins Pipelines**.
 - Created and maintained the configuration of the Spring Application Framework (**IOC**) and implemented business logic using **EJB**.
 - Automated Build artifacts (JAR, WAR & EAR) using continuous integration tools and created pom.xml in **Apache Maven** and run the builds using Continuous Integration tool **Jenkins**.
 - Created Maven POMs to automate the build process for the new projects and integrated them with third party tools like **SonarQube**, **JFrog**.
 - Performed various types of testing like Unit Testing, Performance Testing, Integration Testing, Sanity Checks, Cross Browser, Ad-Hoc Testing, Load Testing and UAT.
 - Using **Bash** and **Python** with **Boto3** to supplement automation provided by **Terraform** for tasks such as encrypting **EBS** volumes backing AMIs and scheduling **Lambda** functions for routine AWS tasks.
 - Built scalable **Docker** infrastructure for Microservices utilizing **Elastic Container Service (ECS)** by creating **Task definitions** JSON file.
 - Experience with container-based deployments using **Docker** and working with **Docker images**, **Docker Hub** and **Docker Registries**, installation and configuring **Kubernetes** and clustering them.
 - Used **Ansible** and **Ansible Tower** as Configuration management tool to automate repetitive tasks, quickly deploying critical applications and proactively managing changes.
 - Created alarms and trigger points in **AWS CloudWatch** based on thresholds and monitored the server's CPU Utilization, performance, and disk usage. Utilized **AWS CloudWatch** services to monitor the environment for performance and operational metrics during load testing.
 - Managed **Kubernetes** charts using Helm. Created reproducible builds of the **Kubernetes** applications, managed Kubernetes manifest files, also managed releases of **Helm packages**.
 - Implemented a production ready, load balanced, highly available, and fault tolerant **Kubernetes infrastructure**.
 - Experience in working with **GIT** to store the code and integrated it with Ansible Tower to deploy the **Ansible Playbooks** written in **YAML**.
 - Implemented Performance testing using **Apache JMeter** and created a dashboard using **Grafana** along with **Prometheus** to view the results and analyze the infrastructure and service monitoring with required alerts for clean workspace.
 - Used **Bash** and **Python**, to supplement automation provided by **Ansible**, **terraform** for tasks such as encrypting **EBS** volumes backing AMIs, scheduling **Lambda** functions for routine AWS tasks.
 - Involved in setting up **JIRA** as a defect tracking system and configured various workflows, customizations, and plugins for JIRA bug/issue tracker and used **JIRA** for bug tracking, issue tracking, and project management.
- 09/2019 to 10/2020 **DevOps Engineer / SRE**
Compulink Management Center, Inc – Long Beach, CA
- Migrated Services from On-premises Kubernetes to Azure **Cloud Environments**. Collaborated with application teams to maintain high-quality deployment.
 - Implemented **Azure** services such as **Azure Active Directory (AD)**, **Azure cloud services**, **IIS**, **Azure storage**, **Azure Blob Storage**, **Resource Manager (ARM)**, **SQL Database**, **Azure VM**, **Azure Functions**, **Azure Service Bus**, **Azure Service Fabric**.
 - Created a new Azure **Active Directory (Azure AD ADFS)** application and service principal that can be used with the role-based access control access to **Azure** Stack resources.
 - Configured **Azure Container Registry (ACR)** to store Docker images and **Azure Kubernetes Service (AKS)** to deploy the spring applications and managed the Kubernetes cluster in Azure.
 - Implemented cluster services using Docker and **Azure Kubernetes Services (AKS)** to manage local deployments in **Kubernetes** by building a self-hosted Kubernetes cluster using Jenkins CICD pipeline.
 - Managed Kubernetes charts using **Helm** and created reproducible builds of the Kubernetes applications, templated **Kubernetes manifests**, and provide a set of configuration parameters to customize the deployment and managed releases of Helm packages.
 - Managed **Azure Kubernetes Services (AKS)** policies, providing access to different **Azure** resources and developing and improving the workflows that govern access.
 - Created **Terraform** templates for provisioning **virtual networks**, **VM Scale sets**, **Load balancers**, and NAT rules and used terraform graph to visualize execution plan using the graph command.
 - Terraform** was used along with **Packer** to create custom machine images, and once the infrastructure was provided the software dependencies were then installed using **Ansible**.
 - Wrote **Terraform** Reusable module for Infrastructure provisioning of various resources like V-net, v-net peering, subnets, API-Management service, cosmos-DB, storage accounts.
 - Integrated **Ansible** plugins with **Jenkins** to provide automation, continuous integration & Continuous Deployment (CI/CD) through Jenkins and wrote **Ansible Playbooks** to automate Ansible target nodes using **YAML** scripting.
 - Created CI/CD Pipelines in **Azure DevOps** environments by providing their dependencies and tasks. Also implemented and managed continuous delivery systems and methodologies on Azure and created end to end automation with Continuous Integration Procedures using **Jenkins** and automated by integrating **Maven build** plugins with **Jenkins**.
 - Implemented **Jenkins** Workflow and Plugins for repeatable deployments of multi-tier applications, artifacts, and services to Docker.
 - Created a custom **golden image** using **hashi corp packer** based on the base image for installing the required software for the teams and deployed the VMs using terraform modules.
 - Worked with Azure Monitoring tools such as **Azure Network Watcher**, **Azure Service Health**, and **Azure Log Analytics** to diagnose and reduce service degradation. Used **Kibana** and **Datadog** Monitor Tool to troubleshoot services, root cause analysis, and security solutions.
 - Involved in setting up Azure Monitoring tools like **log analytics** and **Azure Application Insights** for all the resource of Access level logs, resource level logs of entire subscription, and store the logs in event hub.
 - Deployed Site-Site VPN and VNets using PowerShell and deployed Virtual Machines (VMs) into various Subnets and enabled communication between various Subnets.
 - Worked with version control tools such as **GIT** and **Bitbucket**. Comprehensive knowledge of source controller concepts including branches, tags, conflicts, merges, and Branching Strategies.
 - Designed **Agile** custom Templates for an Organization within the **Azure DevOps** for multiple Projects, like users **Onboarding** with different levels of access into the organizations and projects.
 - For identifying and updating project defects and for tracking user stories and backlogs we have used the **Rally** tool.
- 04/2018 to 08/2019 **AWS Developer / DevOps Engineer**
Paysafe Group – City, STATE
- Implemented fault-tolerant, elasticity, and exceptionally accessible AWS Cloud Infrastructure automated with different resources, **EC2**, **IAM**, **VPC**, **ELB**, **Route53**, **S3**, **Glacier**, **Cloud Watch**, **Cloud Trails**, **RDS**, **SNS**, **Auto Scaling Group** for many applications using code reusable AWS modules.
 - Implemented **Security groups** for inbound/outbound access, **network ACLs** for controlling traffic through subnets, **Internet Gateways**, **NAT instances**, and **Route Tables** to direct the network traffic and to ensure secure zones for organizations in AWS public cloud.
 - Performed **Elastic Load Balancing** to distribute incoming traffic among multiple AWS EC2 instances for scaling up and scaling down depending on income traffic.
 - Created **IAM Roles** and **Policies** to provide specific permissions to the AWS Services depending on the application and the environment in which the application is deployed (**Development**, **Testing**, and **Production**).
 - Rewritten **AWS CloudFormation** templates to convert them into **Terraform templates** by writing the **reusable Terraform modules**.
 - Created Re-usable **Nested stacks**, which were used in the **Cloud formation** template for the creation of multiple resources in AWS. Created multiple cloud formation **Stack sets** to deploy the templates across multiple AWS accounts and regions in single operations.
 - Provided high availability for **IaaS VMs** and **PaaS role instances** for access from other services in the V Net with Azure Internal Load Balancer.
 - Identifying opportunities to improve infrastructure that effectively and efficiently utilizes the Microsoft Azure Windows server 2008/2012/R2, Microsoft SQL Server, Microsoft Visual Studio, Windows PowerShell, Cloud infrastructure.
 - Designed **Network Security Groups (NSGs)** to control inbound and outbound access to **network interfaces (NICs)**, VMs, and subnets.
 - Written **Terraform** templates to provision AWS dynamic Infrastructure as code to build staging and production environments. Managed different infrastructure resources, like VMs, Docker containers.
 - Created and written **Chef Cookbooks** with additional requested packages and prompted **Jenkins** to pull the code from **GitHub** and deployed them in various non-production environments.
 - Wrote **Chef Cookbooks** to automate the installation and update process of **Apache Tomcat**, **MySQL**, and **Java** in the horizontal clusters by using CI Jenkins pipelining. Responsible for version control of Chef Cookbooks, testing of Cookbooks using Test Kitchen, and running recipes on nodes managed by on-premises **Chef Server**.
 - Integrated **Maven**, **Ant**, and **GitHub** in **Jenkins** to build and deploy artifacts to **Nexus Artifactory**.
 - Worked in handling **multiple Docker images** primarily for middleware installation and domain configurations.
 - Achieved continuous delivery goal on the highly scalable environment using Docker coupled with load-balancing tool **Nginx**.
 - Managed major architectural changes from a single server large software system to a distributed system with **Kubernetes orchestration**.
 - Configured and maintained **Jenkins Master-Slave** setup by enabling **password-less SSH login** between server and nodes to manage and distribute the building workload evenly across different nodes.
 - Configured central and distributed repository and assisted developers with branching, **labeling/naming** conventions using **Subversion (SVN)** and **Git source control**.
 - Involved in creating **Dynatrace** and **SPLUNK** dashboards for business defined KPIs (Key Performance Indicator) and performance dashboards for applications performance monitoring using application insights and **Dynatrace SAAS**.
- 03/2016 to 01/2018 **Build and Release Engineer**
Kantar – City, STATE
- Experienced with Java build/automaton tools such as **ANT**, Jenkins, and Maven. Involved in conversion of the source code from the SVN to GIT. Worked closely with other teams, which was migrating from SVN to GIT. Helped them in the CM Planning and Build/Deploy processes.
 - Implemented **Shell**, **Perl** and **Python scripts** for release and build automation. Manipulated and automated scripts to suit the requirements.
 - Responsible for automated Scheduled Builds/Emergency Builds and Release using **ANT scripts** for Enterprise application.
 - Created and Maintained **Subversion Branching, Merging, and tagging** across each production releases. Assist developers in integrating their code with the mainstream.
 - Deployed and maintained infrastructure using **Ansible playbooks**.
 - Worked on Configuration Management Tool Chef & created **Chef Cookbooks** using recipes to automate system operations. Server automation with Chef and used tools like Maven for deployment and build management.
 - Created featured develop release branches in **STASH (GIT)** for different application to support releases and CI builds.
 - Enabled Continuous Delivery through Deployment into several environments of Development, Test and Production using **Maven** and **SonarQube**.
 - Created a script to generate jar files for the change-set related to a particular **JIRA ticket**, which was then uploaded automatically to the FTP server.
 - Provided end-user support, performed baseline build, merges, software release management, and other **SCM activities**.
 - Setting up the new build environment for various products in **Linux environment** implemented the Release practice and responsible for pushing builds into DEV/ QA / SIT / UAT.
 - Implemented new **JIRA workflows** for the QA / SIT / UAT. Worked on Splitting JIRA server's configuration. Managed Jira users and created workflows, issue types, fields in production.
 - Developed and implemented **Software Release Management** strategies for various applications according to the agile process.
 - Performed installing **Red Hat Linux** using kickstart and applying security polices for hardening the server based on the company policies.
 - Creating, cloning Linux Virtual Machines, templates using **VMware ESX 4.x, 5.x.**, Performing capacity analysis, monitoring, and controlling disk space usage on systems.
 - Monitoring **system activities** and fine-tuning system parameters and configurations to optimize performance and ensuring security of systems.
 - Designing **Firewall rules** for new servers to enable communication with application, Setting up network environments using **TCP/IP, NIS, NFS, DNS, VSFTP and DHCP**
 - Setting up user and group login ID's, printing parameters, network configuration, password, resolving permissions and access related issues via **ACL**.

EDUCATION AND TRAINING

Master: Management of Technology
Fairfield University

Bachelor of Technology: Computer Science
Aurora College of Engineering