**Prajwal Koirala**

Certified Multi-Cloud Architect | AWS, GCP & Azure | Software Engineer Specializing in Scalable Systems and Infrastructure Optimization

430 Park Ave New York, NY 10022

845-682-0433

prajwalkoirala638@gmail.com

**Summary**

* I am a Certified Multi-Cloud Architect with extensive expertise spanning AWS, Google Cloud, and Microsoft Azure. For over a decade, I have been at the forefront of designing, deploying, and managing scalable cloud infrastructures that power mission-critical systems in industries including finance, healthcare, e-commerce, and technology.
* My career in software development and cloud engineering is driven by a passion for innovation and a commitment to excellence, transforming legacy systems into agile, modern environments that leverage automation and continuous integration.
* I have led diverse teams in implementing robust Infrastructure as Code (IaC) practices using tools such as Terraform and CloudFormation to ensure that cloud deployments are repeatable, secure, and cost-effective. My work emphasizes both technical precision and strategic foresight, adhering to stringent security and compliance standards like GDPR, HIPAA, and CCPA while integrating CI/CD pipelines and DevOps methodologies to reduce deployment times and optimize performance. This balanced approach enables organizations to achieve operational resilience and sustainable growth.
* My credentials include AWS Certified Solutions Architect – Professional, Google Cloud Professional Cloud Architect, Microsoft Azure Solutions Architect Expert, and HashiCorp Terraform Associate, reflecting a deep and proven ability to execute complex multi-cloud strategies. I actively contribute to open-source initiatives and mentor emerging professionals, fostering collaborative environments that drive continuous improvement and innovation.
* By aligning cutting-edge technical solutions with strategic business objectives, I ensure that every project delivers tangible, long-term value. Dedicated to staying ahead of emerging trends, I consistently explore new technologies to solve intricate challenges and pioneer transformative change.
* Whether modernizing existing systems or architecting new cloud-native solutions, I work closely with stakeholders to deliver outcomes that are secure, scalable, and forward-thinking. I welcome opportunities to collaborate on initiatives that redefine cloud architecture and drive digital transformation. My unwavering commitment to innovation, security, and efficiency is the cornerstone of every project I lead, ensuring that organizations remain agile and competitive in a rapidly evolving digital landscape.

**Experience**

**Mapping United Dec 2023 - Nov 2024**

**Lead Software Developer**

Drone-Based Parking Solutions

Outcome: Reduced parking search time by 35%, saving 5,000 hours annually.

Key Actions: Led the development of a cutting-edge AI/IoT drone system aimed at streamlining urban parking. This involved coordinating a team of 8 engineers and collaborating with urban planning committees to understand the nuances of city dynamics and parking habits.

Value Added: This initiative improved urban mobility by reducing the time spent searching for parking spots, making cities more efficient. The system also helped municipalities optimize the use of their resources, enhancing overall traffic flow and reducing carbon emissions from cars circling for parking.

Engineering: The real-time drone-based parking detection system was built using Python for backend development and OpenCV for computer vision capabilities. We leveraged the YOLO (You Only Look Once) object detection algorithm to accurately identify available parking spaces in real-time. This system combined the use of drones equipped with cameras to scan parking areas and report free spots back to a central system for processing, allowing for quick, data-driven decision-making. The drones were integrated with IoT devices to ensure seamless communication between hardware and software components.

Multi-Cloud Real-Time Data Processing

Outcome: Processed 500,000+ transactions per second with 20% lower latency.

Key Actions: Designed and implemented a robust multi-cloud architecture across AWS and Azure to ensure seamless, fault-tolerant real-time data processing. The architecture was carefully crafted to balance performance, scalability, and resilience, reducing transaction latency by 20%. Integrated monitoring tools like AWS CloudWatch and Azure Monitor to ensure continuous oversight of performance metrics.

Value Added: The multi-cloud architecture delivered highly scalable and reliable performance, allowing the system to handle vast amounts of transactions per second while maintaining low latency. The use of multiple cloud platforms also provided redundancy, improving system resilience and availability. Additionally, operational costs were reduced due to the efficient use of resources across both AWS and Azure platforms.

Engineering: I architected a serverless, microservices-based data pipeline leveraging AWS Lambda, Azure Functions, and Kafka. The system utilized AWS Lambda for on-demand computing resources, Azure Functions for event-driven processing, and Kafka for managing real-time data streams. The combination of these technologies ensured that the data pipeline could scale horizontally, process thousands of transactions per second, and maintain minimal latency, ultimately enabling the system to handle large-scale real-time data processing efficiently.

User Engagement Improvements

Outcome: Boosted user engagement by 15%, with 2.3M daily active users.

Key Actions: Collaborated closely with product managers and UX teams to design and deploy user-centric features aimed at increasing engagement. Leveraged analytics tools to continuously measure and optimize the effectiveness of features and user journeys.

Value Added: This initiative strengthened the company's digital presence, improved customer satisfaction, and fostered higher user retention by making the platform more intuitive and responsive to user needs.

Engineering: I implemented a robust data-driven optimization approach using A/B testing frameworks and automated event tracking systems. Tools like Google Analytics and Mixpanel were used for tracking user behaviors, allowing us to collect data on how users interacted with the platform and identify pain points. This data was crucial for optimizing features in real-time, enhancing the user experience, and driving engagement improvements.

Global Supply Chain Monitoring

Outcome: Enabled tracking across 200+ locations, cutting stock discrepancies by 25%.

Key Actions: Deployed a scalable solution built on AWS to automate reporting across the supply chain. Integrated sensors and real-time tracking systems to collect data from 200+ locations, improving visibility and accuracy across the entire network. The solution helped to reduce manual errors by automating routine processes, saving time and reducing inefficiencies.

Value Added: This real-time monitoring system helped businesses track inventory and shipments more effectively, improving stock accuracy and reducing discrepancies. The streamlined process also saved valuable resources and increased supply chain visibility, contributing to overall efficiency improvements.

Engineering: Developed an automated reporting system integrated with IoT devices and AWS cloud services. Python was used to implement data processing and reporting functionalities, while AWS services like IoT Core, Lambda, and S3 were utilized to collect, store, and process the tracking data. The use of serverless computing and cloud-based storage ensured scalability and resilience, allowing the system to handle large amounts of real-time data seamlessly.

Demand Forecasting with Azure ML

Outcome: Reduced inefficiencies by 15%, saving $2M annually.

Key Actions: Developed predictive models using Azure Machine Learning (ML), incorporating historical sales data and seasonal trends to accurately forecast demand. Worked closely with business teams to align the models with actual business operations, optimizing resource allocation and reducing inefficiencies.

Value Added: By improving forecasting accuracy, the system enabled businesses to better allocate resources, stock inventory, and optimize logistics. This helped save significant costs annually by avoiding overproduction, stockouts, and wastage.

Engineering: Designed time-series forecasting models using Azure AutoML, which helped in automatically identifying the best algorithms for the given data and minimizing manual intervention. Additionally, I built an ETL pipeline using AWS Glue and Redshift, which centralized and analyzed real-time inventory data, ensuring that the forecasting models were constantly updated with the latest information for better prediction accuracy.

Compliance with GDPR and CCPA

Outcome: Achieved full compliance, avoiding fines over $1M.

Key Actions: Led security audits to ensure that the company's systems and data processing workflows complied with GDPR and CCPA regulations. Worked closely with legal and IT teams to implement best practices for data protection, including encryption protocols and secure data handling measures.

Value Added: Achieving compliance with these regulations mitigated significant risks of legal penalties and helped the company build trust with customers and stakeholders. The robust security measures also protected sensitive data and reinforced the company's reputation as a responsible data handler.

Engineering: I implemented AES-256 encryption to secure sensitive data and utilized automated key rotation processes through AWS KMS and Azure Key Vault for maintaining encryption keys. These tools provided secure storage and management of encryption keys, ensuring that all sensitive information remained protected from unauthorized access. Additionally, the compliance processes were automated, reducing human error and ensuring that data protection measures remained in place consistently.

**IpEngine.xyz Apr 2023 - Dec 2023**

**Senior Software Developer**

**IP Intelligence Platform Development**

**Outcome: Increased data accuracy by 20%.**

**Key Actions: I played a key role in developing the core features of the IP intelligence platform. These features focused on enabling real-time analysis of IP addresses, enhancing our data collection, and ensuring that the data collected was more accurate. This allowed us to generate real-time risk scores for IPs based on various factors, such as geographical location, usage patterns, and historical data.**

**Value Added: Improving data accuracy is critical for smarter decision-making. With better IP intelligence, businesses could make more informed decisions regarding user verification, fraud prevention, and targeted services. The platform’s enhanced data integrity also streamlined planning processes, especially for clients relying on IP data for analysis or security purposes.**

**Engineering: To handle the real-time nature of IP risk scoring and classification, I built scalable data pipelines using Python for data processing and Go for performance-critical tasks. These pipelines were designed to ingest, analyze, and classify IP data in real-time, while ensuring high availability and low latency.**

**Backend System Optimization**

**Outcome: Achieved 99.9% uptime.**

**Key Actions: I focused on optimizing both database and API performance. This included improving database indexing to speed up query processing and refining API efficiency to handle high volumes of requests. I also implemented redundancy and failover mechanisms to ensure that the system remained operational even in case of a failure, minimizing downtime.**

**Value Added: Achieving 99.9% uptime significantly enhanced the reliability of our system. Customers could trust that the platform would be available when they needed it, reducing interruptions in their workflows. The optimization also lowered long-term maintenance costs by reducing the need for manual intervention and ensuring a more self-sustaining infrastructure.**

**Engineering: To optimize backend services, I refactored the core services using Golang for its performance advantages. I also enhanced PostgreSQL performance by adding efficient indexing strategies. For caching, I introduced Redis to reduce the load on the database and improve API response times, particularly for frequently accessed data.**

**Multi-Cloud Deployment Strategy**

**Outcome: Delivered scalable, high-availability systems.**

**Key Actions: I architected a multi-cloud deployment strategy using AWS and GCP. This allowed us to leverage the strengths of both cloud platforms while ensuring high availability and scalability. We used Docker containers to package our applications, and Kubernetes for orchestration, enabling us to run applications consistently across both clouds without worrying about vendor lock-in.**

**Value Added: The multi-cloud architecture provided us with the flexibility to choose the best services and features from AWS and GCP, while also ensuring that the system could scale easily to handle growing demand. Additionally, the high-availability setup meant that services were always up, even if one cloud provider experienced issues.**

**Engineering: I implemented a CI/CD pipeline using Terraform to manage infrastructure as code, Helm to deploy applications on Kubernetes, and GitHub Actions for automating the build and deployment processes. This setup allowed for seamless deployment and scaling across both cloud platforms, reducing manual work and improving overall deployment efficiency.**

**Legacy System Migration to Cloud-Native**

**Outcome: Reduced deployment times by 30% and lowered overhead.**

**Key Actions: As part of the migration process, I led the effort to move from a monolithic architecture to a microservices-based, cloud-native model. The transition involved breaking down legacy applications into smaller, manageable services that could be deployed independently. We also moved the applications to cloud-native technologies to take advantage of scalability, cost savings, and easier maintenance.**

**Value Added: By re-architecting the systems, we lowered operational costs and improved the maintainability of the infrastructure. The new microservices approach allowed us to innovate more rapidly and deploy changes faster, significantly reducing deployment times.**

**Engineering: I was responsible for re-architecting the monolithic applications into containerized microservices using Kubernetes for orchestration. We also used gRPC to facilitate communication between microservices, ensuring low-latency and efficient data transfer between services.**

**Mentorship & Team Development**

**Outcome: Elevated code quality and team collaboration.**

**Key Actions: I mentored junior developers, helping them grow their skills and understanding of best practices in coding, testing, and deployment. I led code reviews, providing constructive feedback to ensure that the code met quality standards and maintained consistency across the team. I also worked to establish a culture of continuous improvement by encouraging the adoption of new tools, frameworks, and practices.**

**Value Added: This initiative fostered a culture of knowledge-sharing and continuous improvement. The increased collaboration led to better code quality and more efficient workflows. The team's ability to innovate and solve problems also improved significantly, resulting in faster delivery of features and better overall product quality.**

**Engineering: I helped establish engineering guidelines that covered key areas such as CI/CD pipelines, automated testing, and infrastructure as code. This ensured that the team had a solid foundation for building, testing, and deploying software reliably and consistently.**

**System Monitoring & Incident Response**

**Outcome: Reduced downtime by 25%.**

**Key Actions: To improve system reliability, I implemented proactive monitoring systems that could detect issues before they caused significant downtime. I integrated tools like Prometheus and Grafana for real-time metrics monitoring and visualization, along with AWS CloudWatch for broader infrastructure monitoring. Additionally, I developed automated incident response mechanisms that triggered alerts and initiated self-healing actions to address issues without manual intervention.**

**Value Added: By reducing downtime by 25%, system availability and performance were improved, leading to better user experience and higher customer satisfaction. Proactive monitoring also allowed us to quickly detect and fix issues, minimizing disruption.**

**Engineering: I developed custom scripts using Terraform and Ansible to automate the monitoring and incident response processes. These scripts helped to automate the setup of monitoring dashboards and alerting systems, while also handling basic self-healing actions like restarting services or scaling infrastructure based on load.**

**Collaboration with Product & Design**

**Outcome: Delivered user-centric features that improved customer satisfaction.**

**Key Actions: I worked closely with product managers and design teams to ensure that new features were aligned with customer needs and business goals. This collaboration helped create user-centric features that enhanced the overall experience, ensuring that each feature added value to the users and supported the company’s objectives.**

**Value Added: Cross-functional collaboration led to more thoughtful, intuitive features that better met customer needs. By integrating feedback from multiple teams, we were able to deliver a product that was both functional and user-friendly, which in turn improved customer satisfaction and retention.**

**Engineering: I implemented feature flags and A/B testing to ensure that new features were tested in real-world conditions before being fully deployed. We also used API versioning to maintain backward compatibility while iterating on features, allowing for a smoother rollout of changes without disrupting existing users.**

**Performance Optimization**

**Outcome: Increased system throughput by 30%.**

**Key Actions: To boost system performance, I focused on eliminating backend bottlenecks and optimizing both database queries and API calls. I implemented caching strategies using Redis to reduce load times for frequently accessed data and optimized database queries to speed up data retrieval.**

**Value Added: This resulted in a more responsive system, reducing latency and allowing the platform to handle more users and higher traffic volumes. The performance improvements also laid the foundation for future scalability, ensuring that the system could grow alongside demand.**

**Engineering: I used PostgreSQL’s EXPLAIN ANALYZE command to identify slow queries and then optimized them for better performance. I also integrated Redis caching to reduce load on the database and speed up response times for commonly queried data, enabling the system to handle a higher throughput efficiently.**

**Cloud Native Computing Foundation Nov 2020 - Mar 2023**

**Lead Cloud Engineer**

**Multi-Cloud Architecture Design**

**Outcome: Optimized cost efficiency and scalability for clients.**

**Key Actions: I designed multi-cloud architectures leveraging AWS, GCP, and Azure. The architecture used AWS Lambda, Azure Functions, and Google Kubernetes Engine (GKE) to create highly available, fault-tolerant solutions. The use of these technologies allowed for seamless workloads across different cloud platforms, reducing vendor lock-in and providing clients with the flexibility to choose the best services based on cost, performance, and geographic proximity.**

**Value Added: By designing multi-cloud systems, clients were able to take advantage of the unique strengths of each cloud provider, optimize their cloud service usage, and lower costs. This setup improved agility, as clients could shift workloads between clouds depending on the requirements or changes in pricing, thus enabling better overall flexibility.**

**Engineering: I built cloud-agnostic systems using infrastructure automation and orchestration to ensure smooth operation across all three cloud environments. This approach minimized downtime during deployments, ensured optimal cloud resource use, and allowed the architecture to scale dynamically with demand.**

**Legacy System Migration to Cloud**

**Outcome: Reduced infrastructure costs by 30%.**

**Key Actions: I led the migration of legacy systems to cloud-native solutions by re-architecting them into microservices, leveraging containerization, and adopting serverless architectures. The migration process involved breaking down monolithic applications into smaller, independent services that could scale autonomously and be deployed more easily. This allowed for more efficient resource use and faster feature delivery.**

**Value Added: The migration reduced overhead costs associated with maintaining legacy systems and improved the scalability and flexibility of the infrastructure. With a cloud-native setup, the systems could quickly adapt to changes in demand, provide faster feature releases, and lower maintenance costs.**

**Engineering: I re-architected the existing monolithic systems into microservices using Docker for containerization and Kubernetes for orchestrating those containers. For serverless compute, I used AWS Lambda to enable on-demand processing, allowing us to scale with minimal infrastructure overhead while ensuring seamless operations.**

**Automated Deployment Pipelines**

**Outcome: Reduced deployment times by 40% and sped up release cycles.**

**Key Actions: I automated the CI/CD pipelines using tools like Terraform and AWS CloudFormation, creating an infrastructure as code (IaC) setup that allowed for rapid, consistent releases. These pipelines were designed with integrated testing and security checks to ensure code quality and compliance before deployment.**

**Value Added: By automating deployment processes, the development cycle was accelerated, allowing for more frequent releases and faster delivery of features. This also reduced manual intervention, increasing developer productivity and lowering the risk of human error in deployments.**

**Engineering: I developed fully automated pipelines that handled everything from infrastructure provisioning with Terraform to environment setup with CloudFormation. This ensured that deployments were not only faster but also more reliable and repeatable, improving overall workflow efficiency.**

**Cloud Cost Optimization**

**Outcome: Saved clients 25% on annual cloud costs.**

**Key Actions: I worked closely with clients to optimize their cloud usage by analyzing resource consumption with AWS Cost Explorer and Azure Cost Management. We implemented right-sizing strategies for instances, adjusted storage costs, and optimized resource allocation to ensure that clients only paid for what they actually needed. Additionally, I enabled auto-scaling to ensure that resources were allocated dynamically based on usage, reducing unnecessary costs.**

**Value Added: The cost optimization efforts saved clients a significant portion of their annual cloud expenses without sacrificing performance. By aligning cloud resources more closely with demand, clients achieved better budget efficiency and avoided over-provisioning.**

**Engineering: I implemented auto-scaling and continuous resource monitoring using cloud-native tools to ensure that the infrastructure adapted to changing demands. This was critical in maintaining optimal cloud costs while preserving the required performance levels for clients.**

**Disaster Recovery and Availability**

**Outcome: Achieved 99.9% uptime and improved continuity.**

**Key Actions: I designed and implemented highly available, fault-tolerant systems using multi-region and multi-AZ architectures, ensuring that the infrastructure could recover quickly from any disruptions. Additionally, I integrated cloud-native backup solutions and automated failover mechanisms to guarantee that the system remained operational, even during outages.**

**Value Added: The architecture improved the overall reliability and availability of services, reducing downtime and minimizing data loss in case of failure. This provided clients with peace of mind, knowing that their systems would remain up and running even during unforeseen incidents.**

**Engineering: I designed systems that leveraged multi-region deployments and distributed cloud resources to provide continuous availability. Automated failover and backup strategies were implemented using native cloud tools, allowing the systems to recover swiftly and ensure that clients’ services were always available.**

**Containerization and Kubernetes Adoption**

**Outcome: Modernized deployments and improved resource management.**

**Key Actions: I championed the adoption of containerization and Kubernetes for managing microservices. The use of Docker allowed for consistent packaging of applications across different environments, and Kubernetes provided orchestration for scaling, managing, and maintaining containers efficiently. This allowed for faster and more reliable deployments and centralized management of resources.**

**Value Added: By containerizing applications and managing them through Kubernetes, we improved system reliability and reduced the complexity of deployment workflows. Kubernetes also allowed us to scale applications seamlessly, adjusting resources dynamically to meet demand and optimizing the use of infrastructure.**

**Engineering: I worked with Helm for Kubernetes to streamline deployment management, allowing us to define, install, and upgrade applications efficiently. Kubernetes provided the orchestration needed to manage containerized applications, while Docker ensured consistency across development, testing, and production environments, enabling smooth scaling and faster time to market for new features.**

**Nvidia Jul 2017 - Sep 2020**

**Cloud Operations Specialist**

**Cloud Operations for SaaS Applications**

**Outcome: 99.9% uptime.**

**Key Actions: Managed SaaS applications hosted on AWS and Azure using auto-scaling and load balancing strategies to handle traffic spikes. Auto-scaling dynamically adjusted resources based on real-time demand, ensuring that the system could handle peak loads without performance degradation. Implemented load balancing to distribute incoming traffic evenly across servers, maintaining high availability during periods of heavy traffic.**

**Value Added: These measures ensured continuous high availability, minimized disruptions, and optimized the user experience. By maintaining stable performance even during peak traffic, customer satisfaction and retention were significantly improved.**

**Engineering: Utilized AWS Auto Scaling, Azure Scale Sets, Elastic Load Balancing (ELB), and Azure Load Balancer to manage scaling and distribute traffic.**

**Disaster Recovery Solutions**

**Outcome: RTOs under 10 minutes.**

**Key Actions: Designed and implemented disaster recovery solutions that included cross-region replication, automated failover, and real-time backup strategies to reduce recovery time objectives (RTO). In the event of a failure, traffic was automatically rerouted to a secondary region, and data was restored from real-time backups.**

**Value Added: The ability to recover in under 10 minutes minimized downtime, ensuring continuity of services for clients. This improved system resilience and enabled seamless operation during emergencies.**

**Engineering: Used AWS Route 53 for DNS failover, AWS S3 for backup storage, Azure Site Recovery for failover, and Azure Backup for real-time data protection.**

**Automated Cloud Resource Provisioning**

**Outcome: Reduced manual workload by 40%.**

**Key Actions: Automated the provisioning of cloud resources using AWS CLI, PowerShell, and Google Cloud SDK. I created reusable scripts to automate the setup of resources like EC2 instances, VPCs, and storage, reducing human error and the time spent on manual configuration.**

**Value Added: This automation increased operational efficiency by reducing manual tasks, minimizing configuration errors, and allowing teams to focus on higher-value work.**

**Engineering: Created scripts to provision AWS EC2 instances, S3 buckets, and Google Cloud Compute Engine instances using the AWS CLI, PowerShell, and Google Cloud SDK.**

**Custom Monitoring Dashboards**

**Outcome: Real-time system insights.**

**Key Actions: Developed custom monitoring dashboards using CloudWatch, Azure Monitor, and Google Cloud Operations Suite to provide real-time insights into the health of applications and infrastructure. These dashboards monitored key metrics such as CPU usage, memory utilization, and network throughput.**

**Value Added: The custom dashboards allowed teams to detect performance degradation or system issues proactively, which helped with faster resolution and improved decision-making.**

**Engineering: Integrated CloudWatch for AWS, Azure Monitor for Azure, and Google Cloud Operations Suite to pull real-time performance metrics into custom dashboards.**

**Cloud Infrastructure Cost Optimization**

**Outcome: Reduced cloud spending by 20%.**

**Key Actions: Optimized cloud infrastructure costs by analyzing usage patterns, rightsizing instances, and implementing reserved instances. This included using auto-scaling policies to ensure that only the required resources were in use at any given time.**

**Value Added: The optimization efforts reduced unnecessary resource allocation, lowering cloud costs by 20%, while ensuring that the systems maintained the necessary performance and scalability.**

**Engineering: Leveraged AWS Cost Explorer, Azure Cost Management, and auto-scaling capabilities in AWS EC2 and Google Cloud Compute Engine to adjust instance sizes based on traffic and usage.**

**Automated Scaling Policies**

**Outcome: Efficient handling of fluctuating workloads.**

**Key Actions: Developed automated scaling policies for AWS EC2, Azure VMs, and Google Cloud Compute Engine instances. These policies dynamically scaled resources up or down based on traffic demand, ensuring cost-efficiency while maintaining optimal performance.**

**Value Added: The automated scaling policies helped reduce costs during low-traffic periods, while ensuring that performance remained optimal during traffic surges, leading to more efficient resource utilization.**

**Engineering: Implemented auto-scaling using AWS Auto Scaling, Azure VM Scale Sets, and Google Cloud Instance Groups.**

**Security Best Practices Implementation**

**Outcome: Enhanced security and reduced vulnerabilities.**

**Key Actions: Implemented multi-factor authentication (MFA), role-based access control (RBAC), and encryption for data in transit and at rest. These security measures ensured that sensitive data was protected and access to cloud resources was securely managed.**

**Value Added: By improving security practices, the risk of breaches was minimized, customer trust was increased, and compliance with regulations like GDPR and CCPA was maintained.**

**Engineering: Configured AWS IAM roles and policies, Azure RBAC, and used AWS KMS and Azure Key Vault for data encryption, and enforced MFA for critical resources.**

**CI/CD Pipeline Management**

**Outcome: Reduced deployment time by 30%.**

**Key Actions: Managed and streamlined CI/CD pipelines using Jenkins, Azure DevOps, and GitLab CI, automating testing, integration, and deployment processes. The automation included setting up validation checks to ensure consistent and secure deployments across multiple environments.**

**Value Added: Reduced deployment times by automating manual steps, resulting in faster release cycles and higher-quality code delivered to production with fewer errors.**

**Engineering: Integrated Jenkins for automation, Azure DevOps for build and release pipelines, and GitLab CI for source code management and CI pipeline automation.**

**Cloud-Native Monitoring Solutions**

**Outcome: Improved response times.**

**Key Actions: Deployed cloud-native monitoring solutions that integrated proactive alerting mechanisms. These alerts notified teams of potential disruptions, enabling a quick response to resolve issues before they escalated.**

**Value Added: Real-time alerting significantly reduced response times and minimized downtime by resolving issues before they affected end-users.**

**Engineering: Used AWS CloudWatch, Azure Monitor, and Google Cloud Operations Suite to implement monitoring and alerting systems for proactive issue resolution.**

**ComplexOrganizations Apr 2012 - Jun 2017**

**Software Developer**

**Cloud Operations for SaaS Applications**

**Outcome: 99.9% uptime.**

**Key Actions: Managed SaaS applications hosted on AWS and Azure using auto-scaling and load balancing strategies to handle traffic spikes. Auto-scaling dynamically adjusted resources based on real-time demand, ensuring that the system could handle peak loads without performance degradation. Implemented load balancing to distribute incoming traffic evenly across servers, maintaining high availability during periods of heavy traffic.**

**Value Added: These measures ensured continuous high availability, minimized disruptions, and optimized the user experience. By maintaining stable performance even during peak traffic, customer satisfaction and retention were significantly improved.**

**Engineering: Utilized AWS Auto Scaling, Azure Scale Sets, Elastic Load Balancing (ELB), and Azure Load Balancer to manage scaling and distribute traffic.**

**Disaster Recovery Solutions**

**Outcome: RTOs under 10 minutes.**

**Key Actions: Designed and implemented disaster recovery solutions that included cross-region replication, automated failover, and real-time backup strategies to reduce recovery time objectives (RTO). In the event of a failure, traffic was automatically rerouted to a secondary region, and data was restored from real-time backups.**

**Value Added: The ability to recover in under 10 minutes minimized downtime, ensuring continuity of services for clients. This improved system resilience and enabled seamless operation during emergencies.**

**Engineering: Used AWS Route 53 for DNS failover, AWS S3 for backup storage, Azure Site Recovery for failover, and Azure Backup for real-time data protection.**

**Automated Cloud Resource Provisioning**

**Outcome: Reduced manual workload by 40%.**

**Key Actions: Automated the provisioning of cloud resources using AWS CLI, PowerShell, and Google Cloud SDK. I created reusable scripts to automate the setup of resources like EC2 instances, VPCs, and storage, reducing human error and the time spent on manual configuration.**

**Value Added: This automation increased operational efficiency by reducing manual tasks, minimizing configuration errors, and allowing teams to focus on higher-value work.**

**Engineering: Created scripts to provision AWS EC2 instances, S3 buckets, and Google Cloud Compute Engine instances using the AWS CLI, PowerShell, and Google Cloud SDK.**

**Custom Monitoring Dashboards**

**Outcome: Real-time system insights.**

**Key Actions: Developed custom monitoring dashboards using CloudWatch, Azure Monitor, and Google Cloud Operations Suite to provide real-time insights into the health of applications and infrastructure. These dashboards monitored key metrics such as CPU usage, memory utilization, and network throughput.**

**Value Added: The custom dashboards allowed teams to detect performance degradation or system issues proactively, which helped with faster resolution and improved decision-making.**

**Engineering: Integrated CloudWatch for AWS, Azure Monitor for Azure, and Google Cloud Operations Suite to pull real-time performance metrics into custom dashboards.**

**Cloud Infrastructure Cost Optimization**

**Outcome: Reduced cloud spending by 20%.**

**Key Actions: Optimized cloud infrastructure costs by analyzing usage patterns, rightsizing instances, and implementing reserved instances. This included using auto-scaling policies to ensure that only the required resources were in use at any given time.**

**Value Added: The optimization efforts reduced unnecessary resource allocation, lowering cloud costs by 20%, while ensuring that the systems maintained the necessary performance and scalability.**

**Engineering: Leveraged AWS Cost Explorer, Azure Cost Management, and auto-scaling capabilities in AWS EC2 and Google Cloud Compute Engine to adjust instance sizes based on traffic and usage.**

**Automated Scaling Policies**

**Outcome: Efficient handling of fluctuating workloads.**

**Key Actions: Developed automated scaling policies for AWS EC2, Azure VMs, and Google Cloud Compute Engine instances. These policies dynamically scaled resources up or down based on traffic demand, ensuring cost-efficiency while maintaining optimal performance.**

**Value Added: The automated scaling policies helped reduce costs during low-traffic periods, while ensuring that performance remained optimal during traffic surges, leading to more efficient resource utilization.**

**Engineering: Implemented auto-scaling using AWS Auto Scaling, Azure VM Scale Sets, and Google Cloud Instance Groups.**

**Security Best Practices Implementation**

**Outcome: Enhanced security and reduced vulnerabilities.**

**Key Actions: Implemented multi-factor authentication (MFA), role-based access control (RBAC), and encryption for data in transit and at rest. These security measures ensured that sensitive data was protected and access to cloud resources was securely managed.**

**Value Added: By improving security practices, the risk of breaches was minimized, customer trust was increased, and compliance with regulations like GDPR and CCPA was maintained.**

**Engineering: Configured AWS IAM roles and policies, Azure RBAC, and used AWS KMS and Azure Key Vault for data encryption, and enforced MFA for critical resources.**

**CI/CD Pipeline Management**

**Outcome: Reduced deployment time by 30%.**

**Key Actions: Managed and streamlined CI/CD pipelines using Jenkins, Azure DevOps, and GitLab CI, automating testing, integration, and deployment processes. The automation included setting up validation checks to ensure consistent and secure deployments across multiple environments.**

**Value Added: Reduced deployment times by automating manual steps, resulting in faster release cycles and higher-quality code delivered to production with fewer errors.**

**Engineering: Integrated Jenkins for automation, Azure DevOps for build and release pipelines, and GitLab CI for source code management and CI pipeline automation.**

**Cloud-Native Monitoring Solutions**

**Outcome: Improved response times.**

**Key Actions: Deployed cloud-native monitoring solutions that integrated proactive alerting mechanisms. These alerts notified teams of potential disruptions, enabling a quick response to resolve issues before they escalated.**

**Value Added: Real-time alerting significantly reduced response times and minimized downtime by resolving issues before they affected end-users.**

**Engineering: Used AWS CloudWatch, Azure Monitor, and Google Cloud Operations Suite to implement monitoring and alerting systems for proactive issue resolution.**

**Education**

**State University of New York College at Potsdam** **2015 - May 2019**

B.S. Computer Science

**Certifications**

**HackerRank Feb 2025**

Software Engineer Certificate

SQL (Basic) Certificate

SQL (Intermediate) Certificate

Software Engineer Intern Certificate

SQL (Advanced) Certificate

Rest API (Intermediate) Certificate

React (Basic) Certificate

R (Basic) Certificate

Python (Basic) Certificate

Node (Basic) Certificate

JavaScript (Basic) Certificate

Java (Basic) Certificate

Go (Basic) Certificate

CSS (Basic) Certificate

C# (Basic) Certificate

Angular (Basic) Certificate

**AWS Certified Cloud Practitioner Sep 2022**

**AWS Certified Developer – Associate Oct 2022**

**AWS Certified Solutions Architect – Associate Oct 2022**

**AWS Certified SysOps Administrator – Associate Oct 2022**

**AWS Certified Solutions Architect – Professional Oct 2022**

**AWS Certified DevOps Engineer – Professional Nov 2022**

**AWS Certified Advanced Networking – Specialty Nov 2022**

**AWS Certified Security – Specialty Nov 2022**

**AWS Certified Machine Learning – Specialty Nov 2022**

**AWS Certified Database – Specialty Dec 2022**

**AWS Certified Data Analytics – Specialty Dec 2022**

**AWS Certified SAP on AWS – Specialty Dec 2022**

**HashiCorp Certified: Terraform Associate (002) Dec 2022**

**GCP Professional Cloud Architect Jan 2023**

**GCP Professional Cloud Developer Jan 2023**

**GCP Professional Data Engineer Jan 2023**

**GCP Professional Cloud Security Engineer Jan 2023**

**GCP Professional Cloud Database Engineer Jan 2023**