

PROFESSIONAL SUMMARY

Experienced in designing and deploying dynamically scalable and highly available services in Cloud. Proficient in designing solutions, development, support and deployment of cloud services across Amazon Web Services (AWS) and Microsoft Azure. Have put my best efforts to point out my skills, should help you leveraging the missing gaps of your recruitment.

SKILLS

Cloud Services :

Amazon Web Services (AWS)

EC2, VPC, Elastic Bean Stalk, Lambda, API Gateway, S3, EBS, RDS, Cloud Front, Load Balancer, SQS, Dynamo DB, Route 53, IAM, Kinesis, Aurora DB, SNS, Secret Manager, Step Function, Glue, Athena, Cognito, Auto Scaling, CICD Pipeline, Glacier, WAF, Shield

Microsoft Azure

Virtual Machine, Virtual Network, Load balancers, Traffic Manager, Logic Apps, Azure Functions, Active Directory, Backup and Site Recovery, Azure Storage, Azure CDN, Autoscaling, Cosmos DB, SQL DB, Azure Monitoring, App Service, Key Vault, Azure DevOps, Data Lake, Event Hub, RBAC, Azure Migrate, Application Gateway

Kubernetes : Deployment, POD, Services, Namespace, Volumes, Stateful Sets, Ingress, HA, HELM Charts

CI / CD / CM Tools : GitHub Actions, Jenkins, Azure DevOps, AWS CI/CD, JIRA, Confluence, Boards

AI : Chat - GPT - 3.5, 4, GitHub Co-Pilot

Networking : TCP-IP, Subnets, VPN, DNS Routing, telnet

Methodology : Agile, Software Design, Enterprise Architecture, Software Engineering, RDBMS, Data Modelling, Water Fall, OOPs

DevOps Tools : GitHub, Postman, Docker, Container, Kubernetes, Ansible, Terraform, ELK, CLI, Slack, Chef, Visual Studio Code, Jenkins

Technologies : Rest API, XML, yaml, JSON, Serverless, JD Edwards, HUBBLE, Service Now, UNIX, SQL,

Languages : Python, Shell, HTML, YAML

Database : Microsoft SQL Server, SQL, IBM DB2, Postgres SQL, Cloud Database, Vector Database

Operating System : Windows 7 - 8 - 10, Linux / Ununtu, MacOS, DOS, RHEL

Web Servers : IIS, Apache, Windows Server 2012, 2016, Nginx, Tomcat

WORK EXPERIENCE

Allianz Global Investors | EU | 2022-2023



- Implemented dynamic auto-scaling for Azure Virtual Machines, configuring rules based on metrics such as CPU utilization or network traffic to automatically scale in and out, optimizing resource utilization and cost efficiency.
- Implemented fault tolerance measures, such as deploying virtual machines across multiple fault domains and updating domains, to ensure resiliency against hardware failures and maintenance events.
- Implemented Azure Traffic Manager for global distribution of applications based on DNS routing.



- Successfully designed and implemented *Azure Virtual Networks (VNETs)* for diverse production environments, considering factors such as scalability, security, and high availability.
- Integrated *Azure Monitor* to collect and analyze performance along with *Azure Log Analytics* to centralize and analyze logs for better visibility into system behavior.
- Integrated *Azure Application Gateway* to optimize and secure web application delivery, incorporating features like SSL termination, WAF (Web Application Firewall), and advanced level 7 load balancing for enhanced performance.
- Implemented high availability and fault tolerance strategies using *Azure VM Scale Sets* for scalable and reliable solutions.
- Configured *Azure Blob Storage* for automated backups of on-premises databases. Utilised various types of storage (file, table, queue) for processing various types of workloads.
- Implemented *Azure Recovery Services* for a robust disaster recovery solution.
- Designed and configured *Azure Load Balancer* to distribute incoming network traffic across multiple virtual machines, ensuring even distribution of workloads and improving application availability and responsiveness.
- Implemented *Azure CDN* to distribute content (images, videos, scripts) globally, reducing latency and improving user experience. Leveraged network of edge servers to deliver content from locations closer to end-users.
- Improved website and application performance by caching static content at strategically located *CDN* edge nodes.
- Implemented and fine-tuned *Network Security Groups (NSG)* to control inbound and outbound traffic, enforcing granular security policies at the subnet and virtual machine levels.
- Implemented *RBAC* in *Azure AD*, defining roles and permissions to grant users the appropriate level of access to Azure resources, aligning with the principle of least privilege.
- Implemented *Single Sign-On* and *Multi-Factor Authentication* using *Azure AD*, enhancing user experience by allowing them to access multiple applications with a single set of credentials, also enforcing extra layer of protection against unauthorized access utilising *MFA*.
- Implemented *Azure Functions* with various trigger types, including HTTP triggers for webhooks, timer-based triggers for scheduled tasks, and bindings for seamless integration with Azure services like *Azure Storage* and *Azure Cosmos DB*. Thus implementing server-less computing.
- Implemented and fine-tuned consistency models in *Azure Cosmos DB*, choosing between strong, bounded staleness, session, and eventual consistency based on application requirements.
- Implemented and managed *Azure Data Lake Storage* for scalable and secure storage of large volumes of structured and unstructured data in the cloud.
- Maintaining *Dremio CI-CD* pipeline through *GitHub Actions*, Debugging and fixing various changes to CI and CD pipelines
- Provisioning various deployments and release of application version through pipeline using *GitHub Actions*
- Maintaining *Kubernetes Cluster* for Application hosted using *Azure AKS*
- Debugging and fixing various issues with pod failures / statefull set / OOM issues in *K8's*
- Integrated *DataDog* Monitoring to monitor / update and create new alerts for Application
- Implemented CI pipeline in *Azure DevOps*, to automate build, checkout for version control system, docker images, *Azure Repos* and configured CD pipeline for automated deployment of applications to various environments.
- Implementation / Monitoring / Debugging Scheduled pipeline and Event-Triggered Pipelines to automate routine tasks, builds and periodic deployments while build or response to specific events, like code commits or pull requests.



- Managing Content Delivery Network (CDN) on edge locations through *CloudFront* to improve performance. Setup Cloud Front *OAI* (Origin Access Identity) to allow access to S3 through only Cloud Front Distribution.
- Setup *Elastic Bean Stalk* for application (Node.js / Java) setup, congratulations and deployment. Blue / Green for application. Platform upgrade for Linux 2
- Infra + Application deployment to multiple locations (Ohio, Frankfurt) Build and Deploy *Lambda*.
- Create and manage buckets in S3 to store db, logs backup, images for CDN servers, application downstream data.
- Managing AMI / Snapshots / Volume / Capacity Upgrade - Downgrade AWS Resources (CPU, Memory, Timeout, EBS), Vertical and Horizontal scaling
- Restful API use for GET, PUT, POST for application data modification hosted in AWS. *POSTMAN* for testing API calls, altering and adding application data in json.
- Setup Custom attributes for SAML App Clients for Read / Write Access in *Cognito*
- Continuous Integration for automated software development life cycle through Jenkins. Reduced Deployment time cycle to reach code to production faster.
- Performed *serverless* deployment of code through pipeline and sls deploy.
- Used *Maven* as Build (Compile, review, testing, packing JAR) for code that was moved to testing in Selenium. Continuous testing was performed to reach feedback to developers faster.
- Setup of version Control tool, *GIT* for source code repository. Performed git clone to copy repository, used token for CLI activities through *gitbash*.
- Performed actions git branch, merge, git status, commit, pull, push, add, git config, init, log, checkout. Additionally worked on *GitHub* for compare, fork, review. *GitHub Co-Pilot* tester integrated with *Visual Studio*.
- Configured system configuration for applications like java setup environment variables, path. Setup aws configure for accessing various regions. Maven configuration for env M2_HOME
- Writing *Terraform* IAC for *EC2*, *VPC*, *Lambda*, *Load Balancers*, Managing infrastructure automation using terraform for deploying across different regions in AWS
- Developed *Anisble* Plays to configure and maintain software components of existing infrastructure.
- Integrated *Kubernetes* with Network, storage and security to provide comprehensive infrastructure and orchestrated container across multiple hosts
- Creating *IAM* Group, policies for users and application. Working with *Active Directory*
- Implemented CI workflows in *GitHub Actions* to automatically build and validate code changes upon pull / push request to branches, ensuring early detection of issues
- Managed environments and defined *variables* within GitHub Actions workflow allowing for flexible CI/CD solutions adaptable to different stages and environments
- Utilized GitHub Actions *Secrets* to securely manage and access sensitive information during CI / CD ensuring data protection and compliance.
- Enabled Manual *Workflow Dispatch* in GitHub Actions, allowing users and developers to trigger workflows manually for controlled deployment and testing.
- Configured GitHub Actions *workflows* for various deployment strategies, including Blue-Green or canary to minimize downtime for production deployment.
- Utilized *Self-Hosted Runners* in GitHub Actions to run workflow on custom infrastructure providing flexibility over execution environment
- Embraced pipeline as code by defining CI/CD Pipeline using *YAML* syntax



- Deployed Java and Node js applications to application server in Agile Continuous Integration (CI)
- Utilized Ansible as Configuration Management to deploy consistent infrastructure across multiple environment
- Implement and monitoring of production alerts of servers using CloudWatch, CloudTrail and Nagios
- Implemented micro services in load balanced, highly available and Fault Tolerant Kubernetes Infrastructure
- Implemented SOA Microservices with JSON HTTP REST API for recent activity feeds using AWS SQS
- Real time Analytics of streaming data using Kinesis. Sharding of data streams for high data rate. Using Kinesis Firehose to push data in S3
- Elastic Search, log stash and Kibana (*ELK*) for filtering our data from servers. Stored in Central place logstash. created reports and charts through Kibana.
- Managed maintained monitored various *EC2* instances running on various applications like CloudFront, Elastic Bean Stalk etc
- Creation of *VPC*, NAT Gateway, NACL, subnet, Peering, Security Group, ACL, DNS
- Implemented AWS *WAF* and *Shield* for CloudFront
- Updated code in *Lambda* that required modifications in *Python*
- Worked with *Route 53* to route traffic to server. DNS changes to domain, changing CNAME
- Used AWS *Glue* for ETL, transformation and data warehousing. Implemented Glue Crawlers. Joining data with two data sources creating nodes, filtering and using drop fields
- Worked with high availability and self tolerant relational database *Aurora DB* for creating MySQL and Postgre SQL
- Creating *Docker Containers* leveraging existing Linux Containers and AMI's
- Worked on Release Management and Deployment tools like AWS *Code Pipeline*, *Code Deploy*
- Created Dev and Test environment of different applications by provisioning *Kubernetes* Clusters on AWS using Docker, Ansible and Terraform.
- Managed source code repository in *GIT* of multiple applications in development and production
- Worked with AWS *Secret Manager* to store keys and values of various applications running on Cloud
- Worked with *Cloud Watch*, *Cloud Formation* and *CloudFront* to setup and manage cached content delivery. Implemented Load Balancers and Route 53 to achieve high availability and fault tolerance by reducing failover and latency options to edge locations

- Maintaining application deployed in *Docker*.
- Committing images and saving to *Docker Hub*
- Running Services inside container, docker run / docker exec, connecting to ports
- Performing various actions in docker - docker start, stop, compose
- Checking logs for application failures in Docker and containers, working on fixing them
- Creating Docker File for execution
- Maintained various Windows, Unix and RHEL servers, periodic server maintenance activities.
- Creating, replicating and publishing Various *HUBBLE* Reports
- Creating Custom Report from Corvu Reports implementing latest filters for real-time data



- Experience in implementing AWS *Lambda*, set concurrency to run servers without managing them and to trigger using S3 and SNS
- Worked with *Ansible* Deployments, written various Ansible Playbook with multiple roles, tasks, template, service management, group variables etc
- Deploying applications using *Jenkins* for continuous integration. Troubleshoot Build and release errors, worked with engineers on resolution
- Experience in working with various *Jenkins* Plug-ins like Maven, EC2 container service, selenium in master - slave configuration
- Establish and applying appropriate branching, labelling using *GIT* source control. Analyzed and resolved conflicts related to merging, committing or adding in source code.
- Performed various API testing in *POSTMAN*, used GET, PUT, POST for various testing of code.
- Involved in *Agile SCRUM* (stand-up, planning, demo, review, retrospective)
- Configured *Nagios* to monitor EC2 Linux instances for logging and monitoring.
- Worked with various transfer and network tools - telnet, PUTTY, winscp, ftp
- Troubleshoot network connectivity using PING, NETSTAT, IFCONFIG and TRACE ROUTE commands.
- Created and scheduled CRON jobs in Linux servers
- Worked with AWS Step Functions process for Lambda, SQS
- Replication of data from Source db to Vector db in HUBBLE
- Various layers of Support L1, L2, L3. Worked on Critical incidents in Production.
- Expertise in installation and configuration of web servers (Apache, Nginx, Tomcat)
- Process administration and management like Monitoring, start, stop, Kill various process
- Worked on AWS RDS Aurora db, connection, clustering of databases.
- Created public and internal load balancer in Azure
- Worked with AWS Glacier for archival of downstream data from S3
- Deployed Virtual Machines and Virtual Network in Azure
- Worked with Azure Traffic Manager
- Modifying package.json in VS Code for various application deployment
- Worked on AWS API Gateway for creating, monitoring and securing REST, HTTP & WebSocket API

RESEARCH AND INTERESTS

- Artificial Intelligence Modelling
- Quantum Computing
- Space, James Webb, Hubble
- Educator, Trainer on Cloud and DevOps

EDUCATION

Bachelor's in Computer Science (BCA)

2012 – 2015 | Burdwan University | BU

EXPERIENCE

Tata Consultancy Services - TCS

2015 - 2024 :: 8 Years

Cloud Engineer | Infrastructure | DevOps | AWS | Azure