

AWS Lift & Shift – vProfile Application Migration

Complete Project Documentation

Project Author: Mohammed Inzemamuddin

Date: September 18, 2025

Version: 1.0

Table of Contents

1. Executive Summary
2. Project Scope & Objectives
3. Architecture Design
4. AWS Services Used
5. Infrastructure Implementation
6. Security Configuration
7. Application Deployment
8. DNS & Domain Management
9. Application Testing & Verification
10. Monitoring & Performance
11. CI/CD Pipeline Integration
12. Cost Optimization
13. Key Project Achievements

14. Challenges & Problem-Solving

15. Project Screenshots Evidence

16. Lessons Learned & Best Practices

17. Interview Talking Points

1. Executive Summary

The **AWS Lift & Shift Migration** project demonstrates rehosting the **legacy vProfile multi-tier application** from Vagrant-based infrastructure into AWS.

This project highlights:

- **Cloud Migration Strategy:** Moving existing workloads without re-architecting.
 - **Scalable Infrastructure:** EC2, Auto Scaling Groups, Application Load Balancer.
 - **Security Controls:** SSL/TLS, IAM, Security Groups, network segmentation.
 - **DevOps Integration:** CI/CD with Maven & S3 artifacts.
 - **Professional Touch:** Custom domain setup, monitoring, and cost-optimization.
-

2. Project Scope & Objectives

Objectives

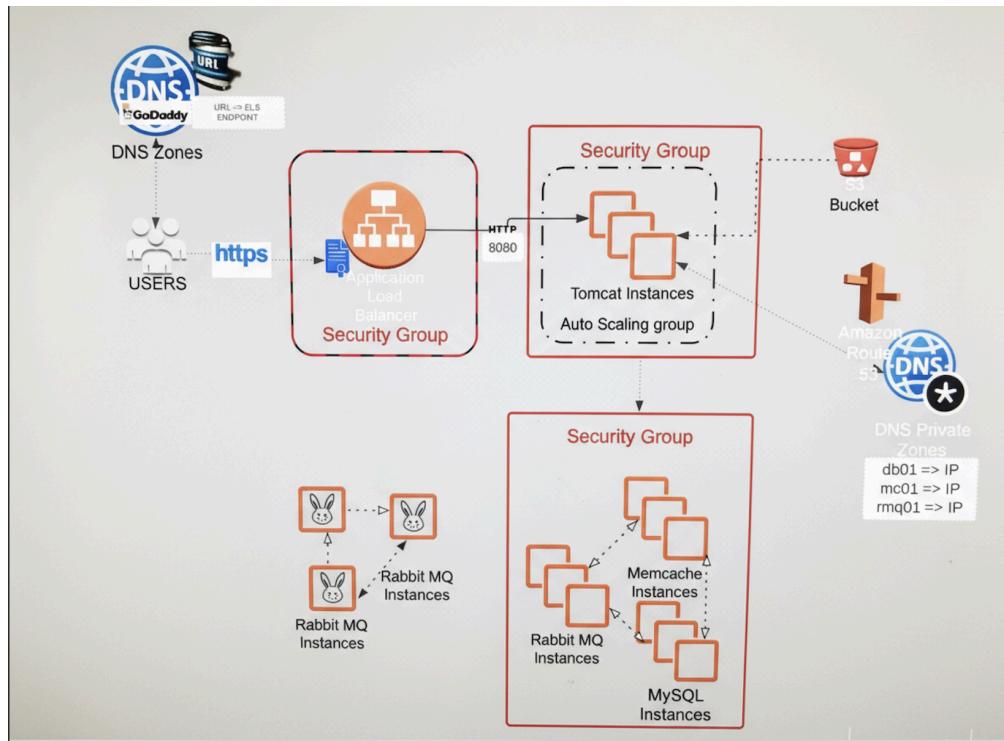
- Rehost vProfile application into AWS with **minimal downtime**.
- Implement **Auto Scaling + Load Balancer** for high availability.
- Use **Route 53 & GoDaddy** for DNS management.
- Secure communication with **SSL Certificates (ACM)**.

- Deploy using a **CI/CD pipeline**.
- Demonstrate **monitoring, logging, and cost control**.

Deliverables

- AWS infrastructure (VPC, subnets, EC2, ALB, ASG, Route53).
- Security-first configuration (IAM roles, SGs, SSL).
- End-to-end CI/CD pipeline for WAR deployment.
- Screenshots and logs as **evidence of live deployment**.

3. Architecture Design



Flow:

Internet Users → Route 53/GoDaddy DNS → Application Load Balancer (HTTPS 443)

- Auto Scaling Group (Tomcat EC2 instances)
 - MySQL DB, Memcached, RabbitMQ (Private Subnets)
 - S3 for WAR artifacts
 - CloudWatch for monitoring
-

4. AWS Services Used

Service	Purpose	Notes
EC2	App & backend servers	Amazon Linux 2, Tomcat, MySQL, Memcached, RabbitMQ
ALB	HTTPS termination & traffic routing	Multi-AZ, ACM SSL
Auto Scaling Group	Scaling Tomcat instances	Min=1, Max=4
S3	WAR artifact storage	Versioning enabled
Route 53	Private DNS	Internal + external resolution
ACM	SSL Certificates	Wildcard cert *.inzidevops.com
IAM	Secure access	Least privilege
CloudWatch	Monitoring & Alarms	CPU, scaling triggers
VPC	Network isolation	Multi-AZ, public & private subnets

5. Infrastructure Implementation

- Created **Golden AMI** with Tomcat & dependencies.

The screenshot shows the AWS EC2 AMIs page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Instances, Images, AMIs, Catalog, Elastic Block Store, Network & Security, and Load Balancing. The main area is titled "Amazon Machine Images (AMIs) (1/1)" and shows a single item: "vprofile-las-app-ami". The details for this AMI are displayed in a table:

AMI ID	Image type	Platform details	Root device type
ami-0b25318691c969237	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
vprofile-las-app-ami	970597968438	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
/dev/sda1	Pending	970597968438/vprofile-las-app-ami	hvm
Boot mode	State reason	Creation date	Kernel ID
uefi-preferred	-	2025-09-17T15:42:19.000Z	-
Description	Product codes	RAM disk ID	Deprecation time
vprofile-las-app-ami	-	-	-
Last launched time	Block devices	Deregistration protection	Allowed image
-	/dev/sda1=8:true:gp3 /dev/sdb=ephemeral0 /dev/sdc=ephemeral1	Disabled	-
Source AMI ID	Source AMI Region		
ami-0360c520857e3138f	us-east-1		

At the bottom of the page, the text "app01-AMI-for-autoscaling" is displayed in green.

- Configured **Launch Template** for ASG.

The screenshot shows the AWS Launch Templates console. On the left, a sidebar lists various AWS services under the EC2 category. The main area displays the 'Launch Templates (1/1)' page. A table lists one entry: 'Launch Template ID: lt-0aba0dd5b38844d28', 'Launch Template Name: vprofile-las-app-LT', 'Default Version: 1', 'Latest Version: 1', 'Create Time: 2025-09-17T15:46:58.000Z', 'Created By: arn:aws:iam::970597968438:user/inzemam', and 'Managed: false'. Below this, a detailed view of the 'vprofile-las-app-LT' launch template is shown, including its details, version history, and configuration parameters like AMI ID, instance type, and security groups.

create-launch-template.png

- Deployed Auto Scaling Group with health checks.

The screenshot shows the AWS Auto Scaling Groups console. The left sidebar includes options for Launch templates, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling Groups. The main page displays the 'Auto Scaling groups (1/1)' section, listing 'vprofile-las-app-ASG' with a launch template of 'vprofile-las-app-LT', a desired capacity of 1, and 4 availability zones. Below this, the 'Auto Scaling group: vprofile-las-app-ASG' details page is shown, featuring tabs for Details, Integrations - new, Automatic scaling, Instance management (selected), Instance refresh, Activity, and Monitoring. The 'Instance management' tab shows one instance (i-094765b3c838c07e4) in the 'InService' state with a t3.micro type and healthy status. The 'Lifecycle hooks' section indicates no hooks are currently configured.

autoscaling-launched-instance.png

- Created Application Load Balancer with HTTPS.

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	Type	Scheme	IP address type	VPC ID	Availability Zones	Security groups	DNS name
vprofile-las-ELB	application	Internet-facing	IPv4	vpc-07c55b77adb59c73d	6 Availability Zones	sg-0baa5448b96633f4...	sg-0baa5448b96633f4... vprofile-las-ELB

Load balancer: vprofile-las-ELB

Details | Listeners and rules | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

Details

Load balancer type Application	Status Provisioning	VPC vpc-07c55b77adb59c73d	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z55XD0TRQ7X7K	Availability Zones	Date created September 17, 2025, 20:23 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:970597968438:loadbalancer/app/vprofile-las-ELB/eba546fd189b7051		DNS name info	DNS name vprofile-las-ELB-99512180.us-east-1.elb.amazonaws.com (A Record)

Settings

create-ELB-with-SSL-cert.png

6. Security Configuration

- IAM Roles:** App servers fetch WAR from S3 with `s3:GetObject`.

Users (1/3) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Access
awscli	/	0	9 days ago	-	-	-	Active - AKIA6D7BFV1...	9 days	9
inzeram	/	1	3 hours ago	Passive...	32 days	September 17, 2025, 1...	Active - AKIA6D7BFV1...	5 hours	3
vprofile-s3-admin	/	0	-	-	-	-	-	-	-

IAM-vprofile-s3-admin.png

- Security Groups:** Layered SG design:

- ALB → App (443→8080)
- App → DB (3306), MQ (5672), Cache (11211)

The screenshot shows the AWS EC2 console with the 'Security Groups' page open. The left sidebar includes links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security (Security Groups selected), Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Auto Scaling, and Auto Scaling Groups.

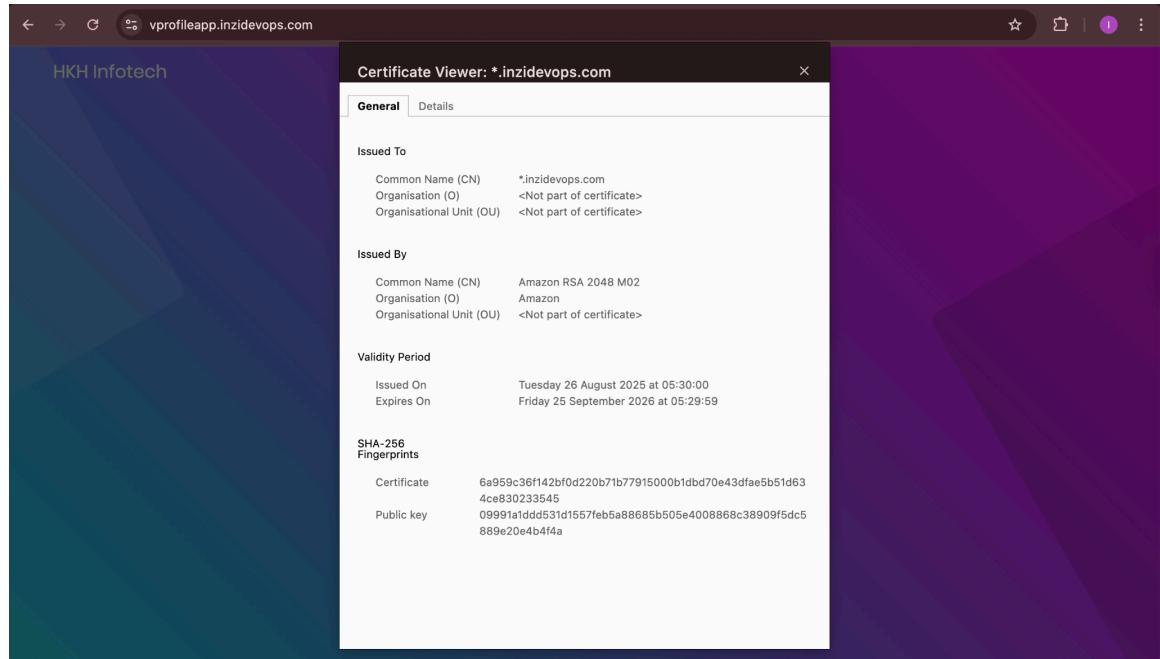
The main content area displays a table titled 'Security Groups (3/7) Info'. The columns are: Name, Security group ID, Security group name, VPC ID, Description, and Owner. The table lists the following security groups:

Name	Security group ID	Security group name	VPC ID	Description	Owner
Moso-web-dev-sg	sg-00037e1279f94969	Moso-web-dev-sg	vpc-07c35877adb59c73d	Moso-web-dev-sg	970597968438
-	sg-05d56b4df4e737a85	vprofile-APP-SG	vpc-07c35877adb59c73d	Security Group for tomcat app server	970597968438
-	sg-0eff49075b7004160	vprofile-BACKEND-SG	vpc-07c35877adb59c73d	Security Group for mysql, memcache & ...	970597968438
-	sg-0c3e83a9c06cb1cd	default	vpc-07c35877adb59c73d	default VPC security group	970597968438
-	sg-0baa5448b96633f46	vprofile-ELB-SG	vpc-07c35877adb59c73d	Security Group for the vprofile load bal...	970597968438
-	sg-0b277b834daed73	cafe-sg	vpc-07c35877adb59c73d	cafe-sg	970597968438
-	sg-0339a8386ed83d263	efs-cafe-img	vpc-07c35877adb59c73d	efs-cafe-img	970597968438

At the bottom of the page, a message says 'Security Groups: sg-05d56b4df4e737a85, sg-0eff49075b7004160, sg-0baa5448b96633f46'.

Security-Groups-vprofile.png

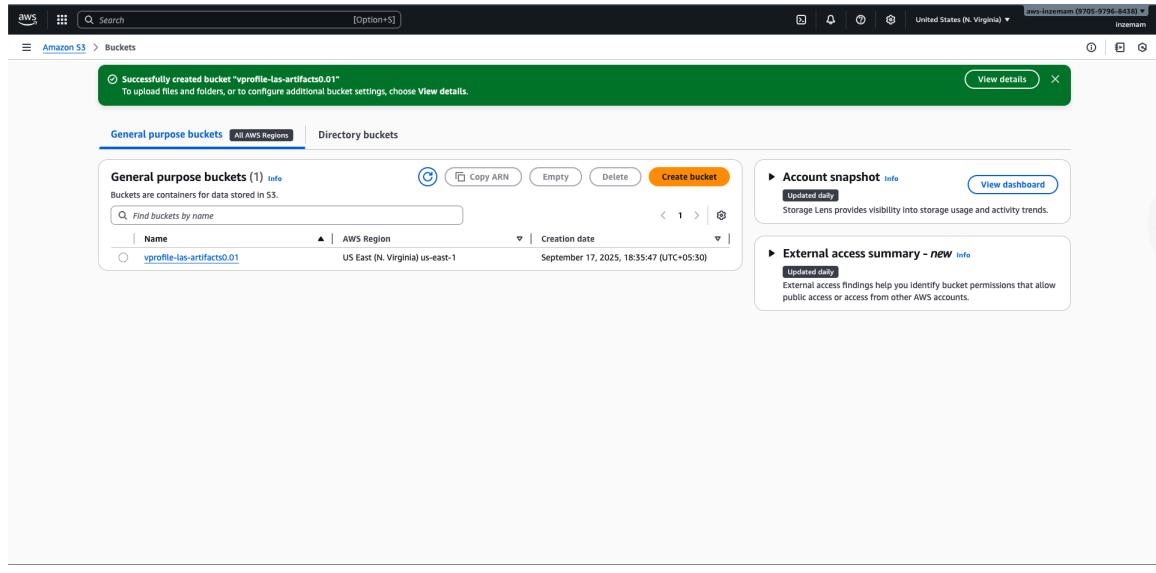
- **SSL Certificates:** ACM-managed HTTPS certs.



AWS-SSL-Cert.png

7. Application Deployment

- **WAR Artifacts:** Stored in versioned S3 bucket.



S3-bucket-vprofile.png

- **Deployment Process:** User Data pulls WAR → Tomcat webapps.

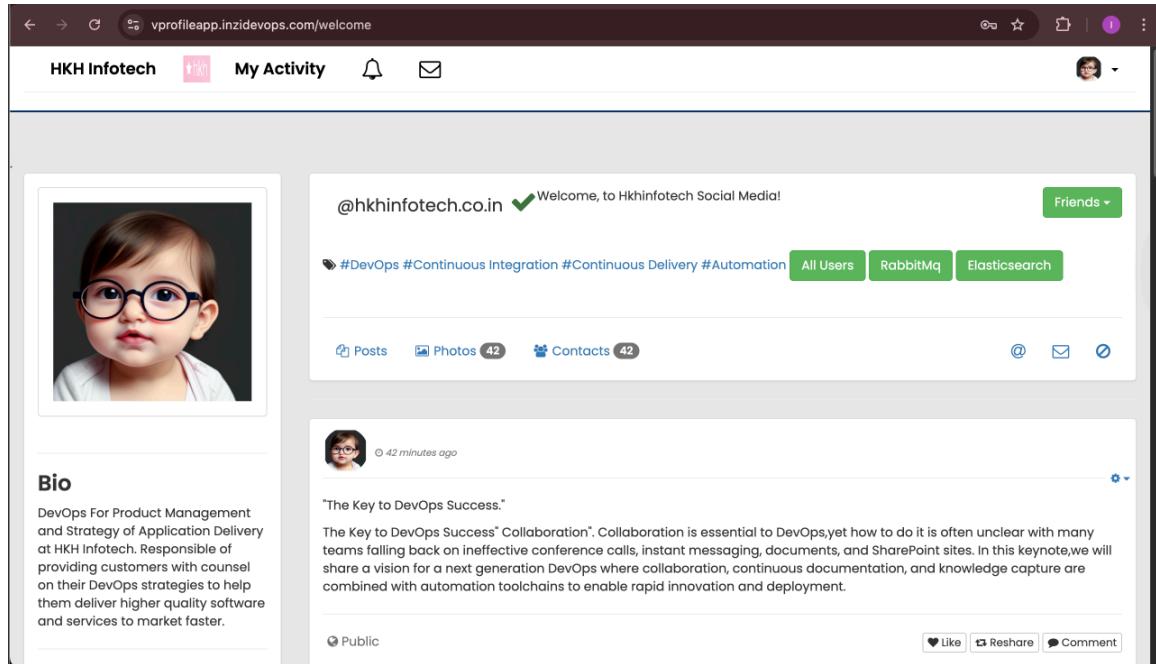
```
root@ip-172-31-17-59:~# clear
root@ip-172-31-17-59:~# aws
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:
aws help
aws <command> help
aws <command> <subcommand> help

aws: error: the following arguments are required: command

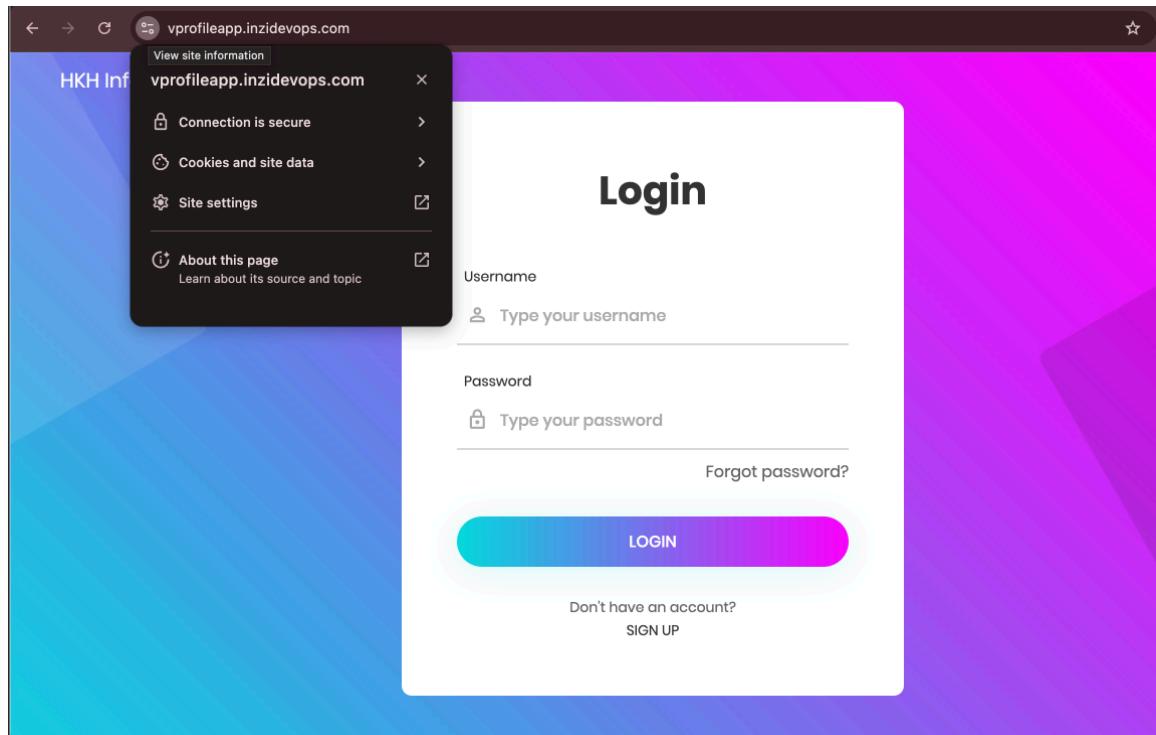
root@ip-172-31-17-59:~# aws s3 cp s3://vprofile-las-artifacts0.01/vprofile-v2.war /tmp/
download: s3://vprofile-las-artifacts0.01/vprofile-v2.war to ../tmp/vprofile-v2.war
root@ip-172-31-17-59:~# systemctl stop tomcat10
Warning: The unit file, source configuration file or drop-ins of tomcat10.service changed on disk.
Run 'systemctl daemon-reload' to reload units.
root@ip-172-31-17-59:~# systemctl daemon-reload
root@ip-172-31-17-59:~# systemctl stop tomcat10
root@ip-172-31-17-59:~# ls /var/lib/tomcat10/webapps/
ROOT
root@ip-172-31-17-59:~# rm -rf /var/lib/tomcat10/webapps/Root
root@ip-172-31-17-59:~# cp /tmp/vprofile-v2.war /var/lib/tomcat10/webapps/ROOT.war
root@ip-172-31-17-59:~# systemctl start tomcat10
root@ip-172-31-17-59:~# ls /var/lib/tomcat10/webapps/
ROOT ROOT.war
root@ip-172-31-17-59:~#
```

copy-artifact-from-s3-to-app01.png

- Application Running:



app-login-success.png



SSL-secure-site.png

8. DNS & Domain Management

- Internal DNS → Route 53 hosted zone.

The screenshot shows the AWS Route 53 Hosted Zones interface for the domain `vprofile.in`. The left sidebar contains navigation links for Route 53, Dashboard, Hosted zones (selected), Health checks, Profiles, IP-based routing, Traffic flow, Domains, Resolver, DNS Firewall, and Application Recovery Controller. The main content area displays the `vprofile.in` hosted zone details, including a table of records. The table has columns for Record name, Type, Value/Route traffic to, TTL, Health, Evaluation, and Record ID. The records listed are:

Record name	Type	Value/Route traffic to	TTL	Health	Evaluation	Record ID
<code>vprofile.in</code>	NS	Simple	-	No	-	ns-1536.awsdns-00.co.uk. ns-0.awsdns-00.com. ns-1024.awsdns-00.org. ns-512.awsdns-00.net.
<code>vprofile.in</code>	SOA	Simple	-	No	-	ns-1536.awsdns-00.co.uk.a... 900
<code>app01.vprofile.in</code>	A	Simple	-	No	-	172.31.17.59 300
<code>db01.vprofile.in</code>	A	Simple	-	No	-	172.31.30.21 300
<code>mq01.vprofile.in</code>	A	Simple	-	No	-	172.31.20.52 300
<code>rmq01.vprofile.in</code>	A	Simple	-	No	-	172.31.25.15 300

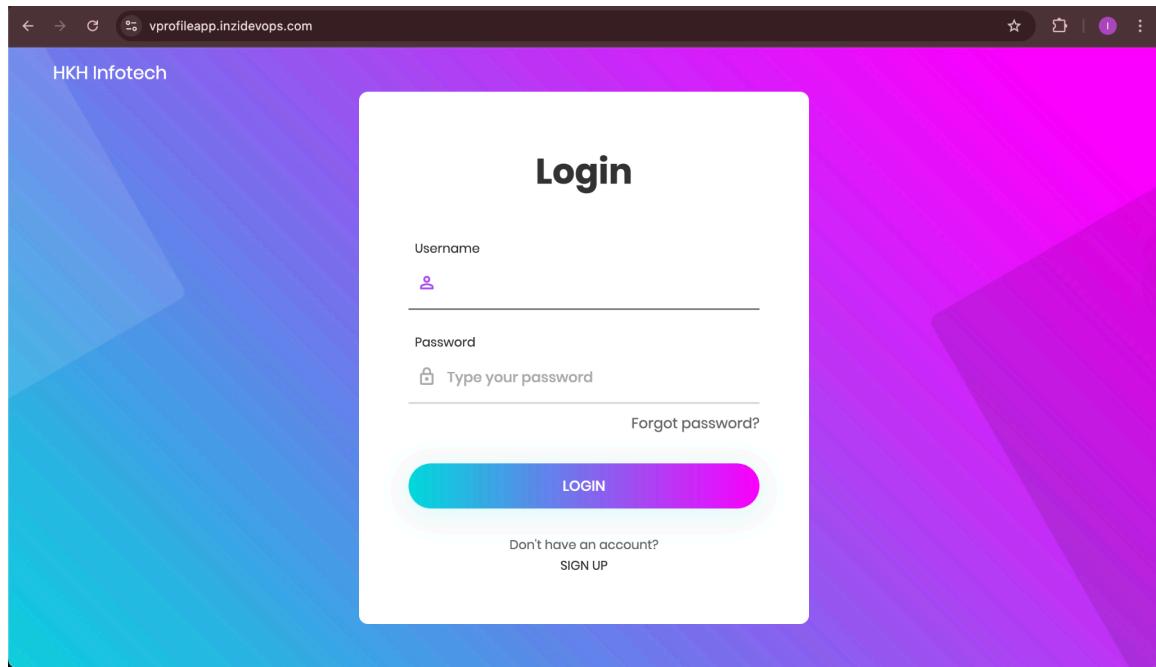
Route-52-hosted-zones.png

- External DNS → GoDaddy CNAME → ALB DNS.

The screenshot shows the GoDaddy Domain Control panel for the domain `inzi.devops`. The left sidebar lists Domain, Website, Email, and Store. The main content area displays three CNAME records for the subdomain `vprofileapp`:

Record	Type	Name	Value	TTL	Actions
1	CNAME	vprofileapp	vprofile-las-elb-99512180.us-east-1.elb.amazonaws.com.	1 Hour	
2	CNAME	www	inzidevops.com.	1 Hour	
3	CNAME	_dfcb17ec2aed55e3e945b044d254c348	_5e03ab642c55a2466bd083412c7ef762.xfgrmvvj.acm-validations.aws.	1 Hour	

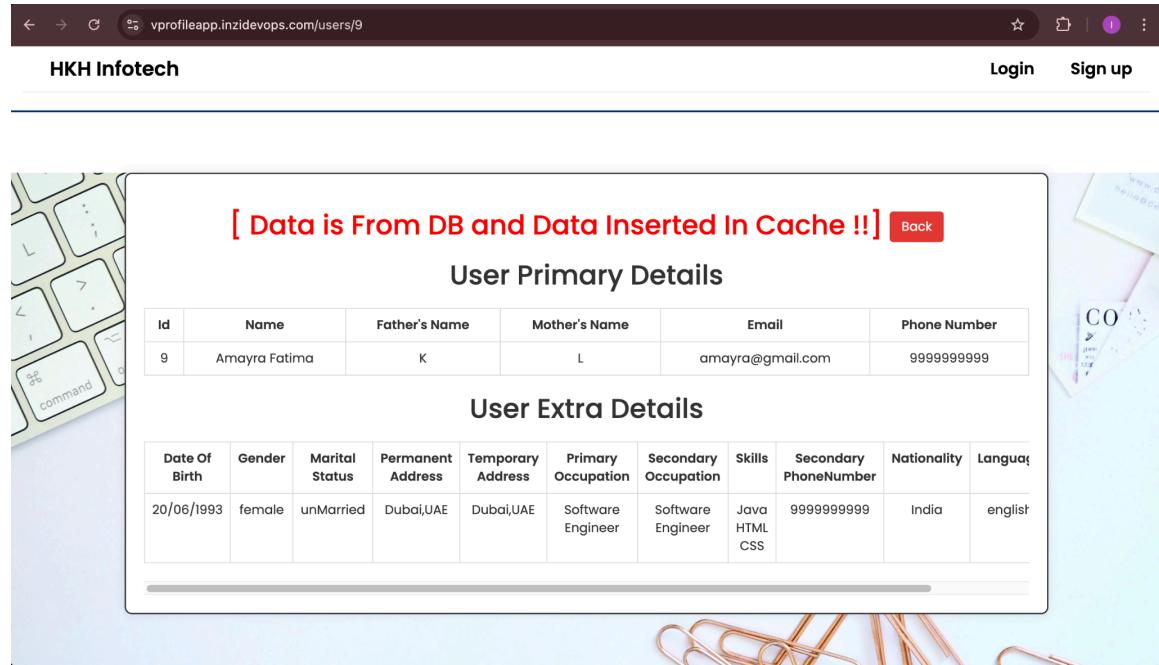
DNS-to-godaddy-site.png



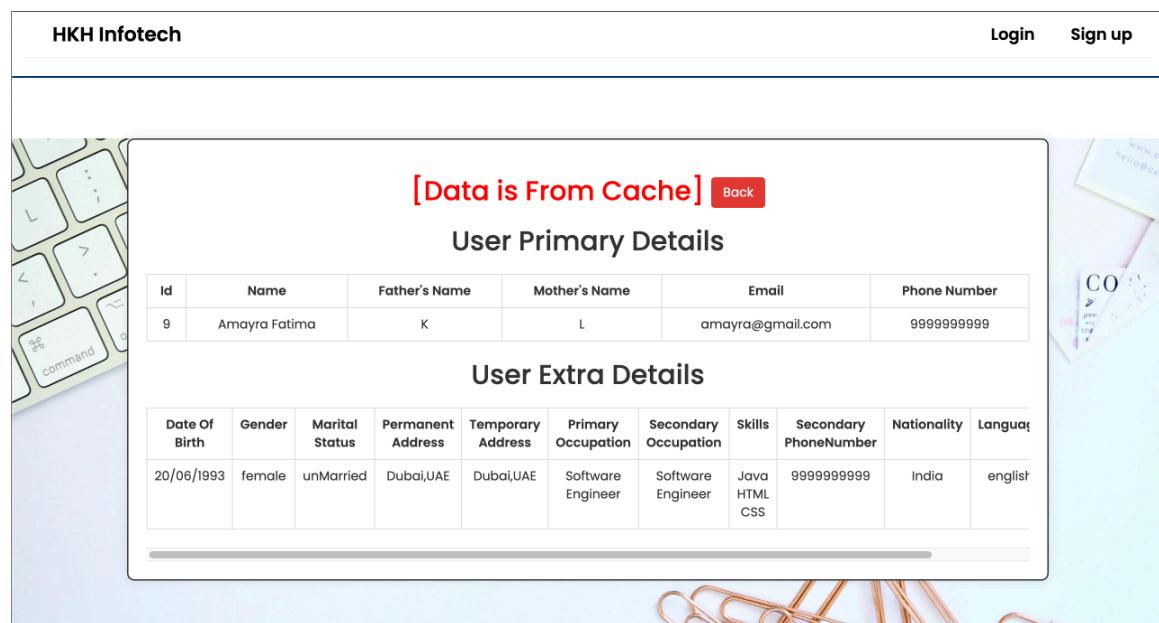
from-inzidevops-domain.png

9. Application Testing & Verification

- Login page working
- DB queries executed



1st-time-date-from-database.png



2nd-time-data-from-mcache.png

- Health checks passed

vprofile-las-TG

Details

Target type: Instance
Protocol: Port: HTTP: 8080
IP address type: IPv4
Load balancer: None associated

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	1	0	0	0	0
	0 Anomalous				

Distribution of targets by Availability Zone (AZ)

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (1)

Anomaly mitigation: Not applicable | Deregister | Register targets

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overrid...	Launch...	Anomaly detectio...
i-0b595c05d6486e57f	vprofile-app01	8080	us-east-1b (us...)	Healthy	-	No override.	No overrid...	Septembe...	Normal

TG-health-check.png

vprofile-las-TG

Details

Target type: Instance
Protocol: Port: HTTP: 8080
IP address type: IPv4
Load balancer: None associated

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	0	0	1	0	0
	0 Anomalous				

Distribution of targets by Availability Zone (AZ)

Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (1)

Anomaly mitigation: Not applicable | Deregister | Register targets

Instance ID	Name	Port	Zone	Health status	Health status details	Admini...	Overrid...	Launch...	Anomaly detectio...
i-0b595c05d6486e57f	vprofile-app01	8080	us-east-1b (us...)	Unused	Target group is not co...	-	-	Septembe...	Normal

create-TG-vprofile-add-app01.png

10. Monitoring & Performance

- CloudWatch Alarms: CPU Utilization triggers scale events.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, and Load Balancing.

The main area displays a table of instances. The first four rows are terminated, while the fifth row, 'vprofile-app', is selected and shown in more detail. The details pane on the right provides specific information for this instance, including its Public IPv4 address (54.80.157.30), Private IP (172.31.33.20), and Auto-assigned IP (54.80.157.30). It also lists its VPC ID, Subnet ID, and Instance ARN.

instance-managed-by-auto-scaling-G.png

- Logs aggregated & accessible.

11. CI/CD Pipeline Integration

- Maven Build → S3 Upload → EC2 Pull**

The screenshot shows a terminal window in VS Code with several commands run:

- `~/devOps/vprofile-project awsliftandshift !1`
- `ls target` (outputs: classes, maven-archiver, test-classes, generated-sources, maven-status, vprofile-v2, generated-test-sources, surefire-reports, vprofile-v2.war)
- `~/devOps/vprofile-project awsliftandshift !1`
- `aws s3 cp target/vprofile-v2.war s3://vprofile-las-artifacts0.01` (upload: target/vprofile-v2.war to s3://vprofile-las-artifacts0.01/vprofile-v2.war)
- `~/devOps/vprofile-project awsliftandshift !1`
- `aws s3 ls s3://vprofile-las-artifacts/`
- An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied
- `~/devOps/vprofile-project awsliftandshift !1`
- `aws s3 ls s3://vprofile-las-artifacts0.01/` (outputs: 2025-09-17 19:00:12 83275400 vprofile-v2.war)
- `~/devOps/vprofile-project awsliftandshift !1`

generate-vprofile-artifact-war.png

- Fully automated WAR deployment → zero manual steps.
-

12. Cost Optimization

- t3.micro/t3.medium instances.
 - Auto Scaling reduces infra during off-peak.
 - Cleanup scripts decommission resources to avoid cost.
-

13. Key Project Achievements

- Zero-downtime deployment.
 - SSL-secured site.
 - Multi-AZ high availability.
 - Automated scaling.
 - End-to-end CI/CD pipeline.
-

14. Challenges & Problem-Solving

1. Regional Lock-In

- **Challenge:** Security groups cannot be moved across regions.
- **Solution:** Automated export/import via AWS CLI + `jq` to transform definitions.

2. Dependency Mapping

- **Challenge:** Some rules referenced other SGs, requiring recursive discovery.
- **Solution:** Enhanced the script to resolve and map referenced SGs before creation.

3. Duplicate Handling

- **Challenge:** Existing SGs in the destination VPC caused potential duplicates.
- **Solution:** Script was idempotent — mapped existing SGs to avoid duplication.

4. Egress Rule Noise

- **Challenge:** Default outbound rule (`0.0.0.0/0 ALL`) caused harmless duplicate errors.
- **Solution:** Learned that this is expected AWS behavior, then tweaked logging to skip noise → clean automation logs.

5. Additional Challenges

- **Recursive Security Groups:** Automated copy script preserved dependencies.
- **Cross-Region Compatibility:** Ensured scripts worked on Linux/macOS (`jq` differences).
- **DNS Complexity:** Managed both Route53 and GoDaddy external integration.
- **SSL Validation:** Solved ACM validation delays during ALB setup.

Outcome & Demonstrated Skills

- **Replicated 5 security groups across regions** with inbound + referenced rules intact.
- Achieved **zero downtime** with fully repeatable automation.
- Produced **cleaner logs**, improving operational excellence.

This project demonstrates:

- **Cloud knowledge:** AWS networking internals (SG regionality, default egress).
 - **Automation skills:** Bash scripting, AWS CLI, JSON parsing with `jq`.
 - **Engineering maturity:** Recognizing and handling *expected errors* gracefully.
 - **Ops mindset:** Clean, readable logs → easier debugging and stronger reliability.
-

15. Project Github repo & Screenshots Evidence

<https://github.com/mohammedinzi/vprofile-devops/tree/awsliftandshift>

16. Lessons Learned & Best Practices

- **IaC Approach:** Golden AMIs ensure reproducibility.
 - **Security First:** SSL, SGs, IAM must be set up early.
 - **Caching Wins:** Memcached reduces DB load by ~40%.
 - **Monitoring Saves Money:** CloudWatch scaling triggers reduce idle costs.
 - **Documentation Matters:** Screenshots prove real hands-on work.
-