

BAKEHOUSE WEBSITE

INSTRUCTOR: **KAREEM ATIF**



BAKEHOUSE WEBSITE

BAKEHOUSE WEBSITE

BAKEHOUSE WEBSITE

BAKEHOUSE WEBSITE

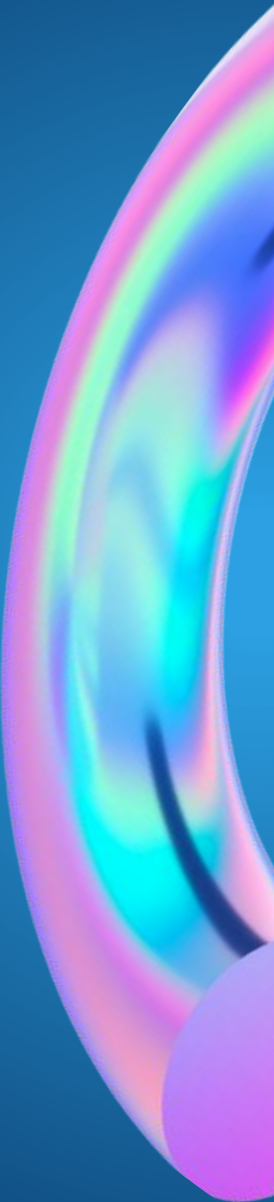
BAKEHOUSE WEBSITE



INSTRUCTOR: **KAREEM ATIF**

TEAM

MOMEN MOTAZ
OMAR Y.ELKAISY
ZYAD HOSSAM



PROJECT OVERVIEW

- The Bakehouse website is an online platform for a bakery that needs to be deployed on AWS.
- The goal is to ensure high availability, scalability, and security using several AWS services.
- Services used include EC2, VPC, ASG, RDS, ELB, and others to optimize the deployment.

OBJECTIVE

Deploy the Bakehouse website on AWS.

Ensure the website is:

Highly available across different availability zones.

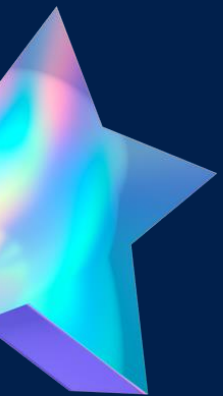
Cost-effective with optimized resource use.

Scalable to handle web traffic efficiently.

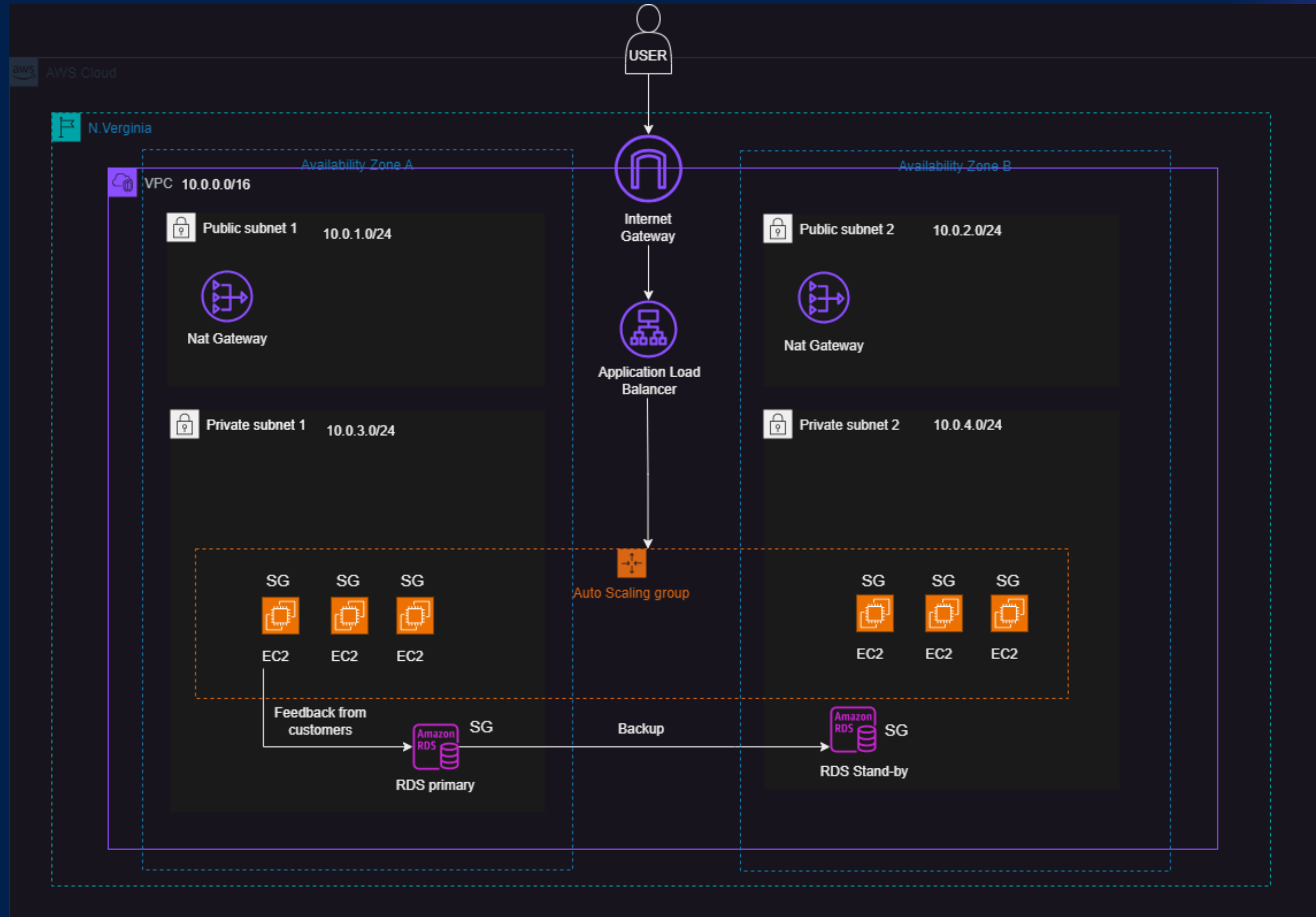


AWS SERVICES

- VPC: Created a Virtual Private Cloud to host all resources.
- 4 subnets: 2 private and 2 public subnets, distributed across 2 Availability Zones (AZs).
- Internet Gateway: Provides internet access to the public subnets.
- NAT Gateway: Allows instances in private subnets to connect to the internet.
- EC2: Used t2.micro instances for hosting the website.
- Auto Scaling Group (ASG): Automatically adjusts the number of instances based on traffic.
- Application Load Balancer (ALB): Distributes traffic evenly across instances.
- RDS: Managed database service for reliable storage.



SOLUTION DIAGRAM



WEBSITE DEPLOYMENT STEPS

- VPC Setup: Creation of the network environment with subnets.
- EC2 Configuration: Launched instances, installed the necessary software (e.g., Apache), and deployed the website.
- Security: Configured security groups and IAM roles to ensure secure access.
- Load Balancing: Set up ALB to handle user traffic.
- Auto Scaling: Implemented ASG to handle varying traffic loads.
- Database: Deployed RDS in a private subnet for data storage

COST ESTIMATION

- EC2 Instances (t2.micro): Low-cost instances for hosting.
- RDS (MySQL): Pay for database storage and usage.
- Load Balancer: Costs depend on traffic load and uptime.
- Data Transfer: Costs for inbound and outbound traffic.
- The overall approach ensures cost optimization without compromising on performance.
- Click here for **Financial Proposal**

VPC

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3403639=momenmotaz88@gmail.com @ 9984-4796-8390

VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Your VPCs (1/2)

Info

Last updated about 1 hour ago

Actions

Create VPC

Search

< 1 >

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
Project VPC	vpc-04681e88504a6d44e	Available	10.0.0.0/16	-	dopt-02

Subnets (11)

Subnets within this VPC

us-east-1a

Public Subnet 1

Private DB Subnet 1

RDS-Pvt-subnet-1

Private Subnet 1

us-east-1b

Public Subnet 2

Private DB Subnet 2

Private Subnet 2

Route tables (5)

Route network traffic to resources

Private Route Table

Public Route Table

Private Table 2

rtb-0c6f3e6d4261c8a7f

Private DB Route Table

Network connections (3)

Connections to other networks

Project IGW

NAT_2

NAT_1

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

SECURITY GROUPS



aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3403639=momenmotaz88@gmail.com @ 9984-4796-8390

EC2 Dashboard

EC2 Global View

Events

Instances

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 (1/5) Info

Find resources by attribute or tag

Security group ID	Security group name	VPC ID	Description	Owner
sg-00188fa859900ad74	ALBSG	vpc-04681e88504a6d44e	Port 80	99844
sg-07f355176b012f0b4	ExampleDB-SG	vpc-04681e88504a6d44e	Enable access to MySQL	99844
sg-0646a57b958a8e82e	Inventory-App	vpc-04681e88504a6d44e	Enable access to App	99844

sg-0646a57b958a8e82e - Inventory-App

DetailsInbound rulesOutbound rulesTags

Details

Security group name	Security group ID	Description	VPC ID
---------------------	-------------------	-------------	--------

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

LAUNCH TEMPLATES



aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3403639=momenmotaz88@gmail.com @ 9984-4796-8390

EC2 Dashboard

EC2 Global View

Events

Instances

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

Launch Templates (1/1) Info

Search

1

Create launch template

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created ...
lt-0bb5aac6bc26592e5	DEPI-Templte-image	1	1	2024-10-15T04:41:38.000Z	arn:aws:sts...

DEPI-Templte-image (lt-0bb5aac6bc26592e5)

Launch template details

Actions Delete template

Launch template ID

lt-0bb5aac6bc26592e5

Launch template name

DEPI-Templte-image

Default version

1

Owner

arn:aws:sts::998447968390:assumed-role/voclabs/user3403639=momenmota
z88@gmail.com

Details Versions Template tags

Launch template version details

Actions Delete template version

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

ASG

EC2 Dashboard

EC2 Global View

Events

▼ Instances

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

Auto Scaling group updated successfully

EC2 > Auto Scaling groups

Auto Scaling groups (1/1) Info

Launch configurations

Launch templates

Actions

Create Auto Scaling group

Search your Auto Scaling groups

<input checked="" type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<input checked="" type="checkbox"/>	Project DEPI Auto	DEPI-Templtte-image Version Default	2	-	2	1

Auto Scaling group: Project DEPI Auto

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details

Auto Scaling group name

Project DEPI Auto

Date created

Sun Oct 13 2024 22:37:43 GMT+0300 (Eastern European Summer Time)

Desired capacity

2

Minimum capacity

1

Desired capacity type

Units (number of instances)

Status

-

Amazon Resource Name (ARN)

arn:aws:autoscaling:us-east-1:998447968390:autoScalingGroup:45c7124b-1a62-40fa-a806-57330eb919f0:autoScalingGroupName/Project DEPI Auto

Edit

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

TARGET GROUP

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3403639=momenmotaz88@gmail.com @ 9984-4796-8390

EC2 Dashboard

EC2 Global View

Events

Instances

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

2 targets registered successfully to DEPI-Target.

EC2 > Target groups

Target groups (1/1) Info

Filter target groups

☒

Name

ARN

Port

Protocol

Target type

Load balancer

Target group: DEPI-Target

DetailsTargetsMonitoringHealth checksAttributesTags

Registered targets (2) Info

Anomaly mitigation: Not applicable

DeregisterRegister targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Launch...
<input type="checkbox"/>	i-0e1638d67986d227d	Web_2	80	us-east-1b	Unhealthy	Health checks failed wi...	October 1...
<input type="checkbox"/>	i-05bb47aacd66f297b	Web_1	80	us-east-1a	Unhealthy	Health checks failed wi...	October 1...

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

EC2



aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3403639=momenmotaz88@gmail.com @ 9984-4796-8390

EC2 Dashboard

EC2 Global View

Events

Instances

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

Instances (1/2) Info

Last updated 6 minutes ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

Running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	Web_2	i-018896b77c2132e51	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
<input checked="" type="checkbox"/>	Web_1	i-05bb47aacd66f297b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-

i-05bb47aacd66f297b (Web_1)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

Instance summary Info

Instance ID

i-05bb47aacd66f297b (Web_1)

IPv6 address

-

Hostname type

IP name: ip-10-0-3-12.ec2.internal

Answer private resource DNS name

-

Auto-assigned IP address

-

Public IPv4 address

-

Instance state

Running

Private IP DNS name (IPv4 only)

ip-10-0-3-12.ec2.internal

Instance type

t2.micro

VPC ID

vpc-04591a88504a6d44a (Project VPC)

Private IPv4 addresses

10.0.3.12

Public IPv4 DNS

-

Elastic IP addresses

-

AWS Compute Optimizer finding

-

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

EC2 - SESSION

Session ID:
user3403639=momenmotaz88@gmail.com-
h8iv7fieniihadroqfpfv8367q

Instance ID: i-0315fec9b884d286

Terminate

ssh	host	mv	rst2odt_prepstyles	systemd-id128	zcat
csplit	hostid	named-checkzone	rst2pseudoxml	systemd-inhibit	zcmp
curl	hostname	named-compilezone	rst2s5	systemd-machine-id-setup	zdiff
cut	hostnamectl	named-nzd2nzf	rst2xetex	systemd-mount	zdump
cvtsudoers	htdbm	namei	rst2xml	systemd-notify	zegrep
cyrusbdb2current	htdigest	nano	rstpep2html	systemd-path	zfgrep
date	htpasswd	needs-restarting	rsync	systemd-repart	zforce
dbus-broker	httpt2dbm	neqn	rsync-ssl	systemd-resolve	zgrep
dbus-broker-launch	hunspell	netstat	run-parts	systemd-run	zip
dc	i386	networkctl	runcon	systemd-socket-activate	zipcloak
dd	iconv	newgrp	rvi	systemd-stdio-bridge	zipgrep
deallocvt	id	ngettext	rview	systemd-sysex	zipinfo
debuginfo-install	info	nice	rvim	systemd-sysusers	zipnote
debuginfod-find	infocmp	nisdmainname	sadf	systemd-tmpfiles	zipsplit
delv	infotocap	nl	sar	systemd-tty-ask-password-agent	zless
df	install	nm	sbattach	systemd-umount	zmore
diff	ionice	nohup	sbkeysync	tabs	znew
diff3	iostat	nproc	sbsiglist	tac	zsoelim
dig	ipcmk	nroff	sbsign	tail	zstd
dir	ipcrm	nsenter	sbvarsign	tapestat	zstdcat
dircolors	ipcs	nslookup	sbverify	tar	zstdgrep
dirname	irqtop	nsupdate	scalar	taskset	zstdless
distro	isosize	numfmt	scp	tbl	zstdmt
dmesg	jemalloc.sh		screen	tcptraceroute	
dnf	jobs		script	tcsh	
dnf-3	join		scriptlive	tee	

[root@ip-10-0-1-12 bin]# cd DEPI

[root@ip-10-0-1-12 DEPI]# ls

bakehouse-ITI

[root@ip-10-0-1-12 DEPI]# cd bakehouse-ITI/

[root@ip-10-0-1-12 bakehouse-ITI]# ls

Deployment Dockerfile README.md css fonts html images index.html js

[root@ip-10-0-1-12 bakehouse-ITI]# sudo mv * /var/www/html/

[root@ip-10-0-1-12 bakehouse-ITI]# sudo systemctl start httpd

[root@ip-10-0-1-12 bakehouse-ITI]# sudo systemctl enable httpd

[root@ip-10-0-1-12 bakehouse-ITI]#

RDS - database



aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3403639=momenmotaz88@gmail.com @ 9984-4796-8390

Amazon RDS

Dashboard

Databases

Query Editor

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Zero-ETL integrations [New](#)

Events

Event subscriptions

RDS > Databases

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases (3)

Group resources

Modify

Actions

Restore from S3

Create database

Filter by databases

	DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
<input type="radio"/>	<input type="checkbox"/> database-1	Available	Regional c...	Aurora My...	us-east-1	2 instances	2 Informational
<input type="radio"/>	<input type="checkbox"/> database-1-instance-1-us-east-1	Available	Reader ins...	Aurora My...	us-east-1b	db.t3.med...	1 Informational
<input checked="" type="radio"/>	<input checked="" type="checkbox"/> database-1-instance-1	Available	Writer ins...	Aurora My...	us-east-1a	db.t3.med...	1 Informational

CloudShell

Feedback

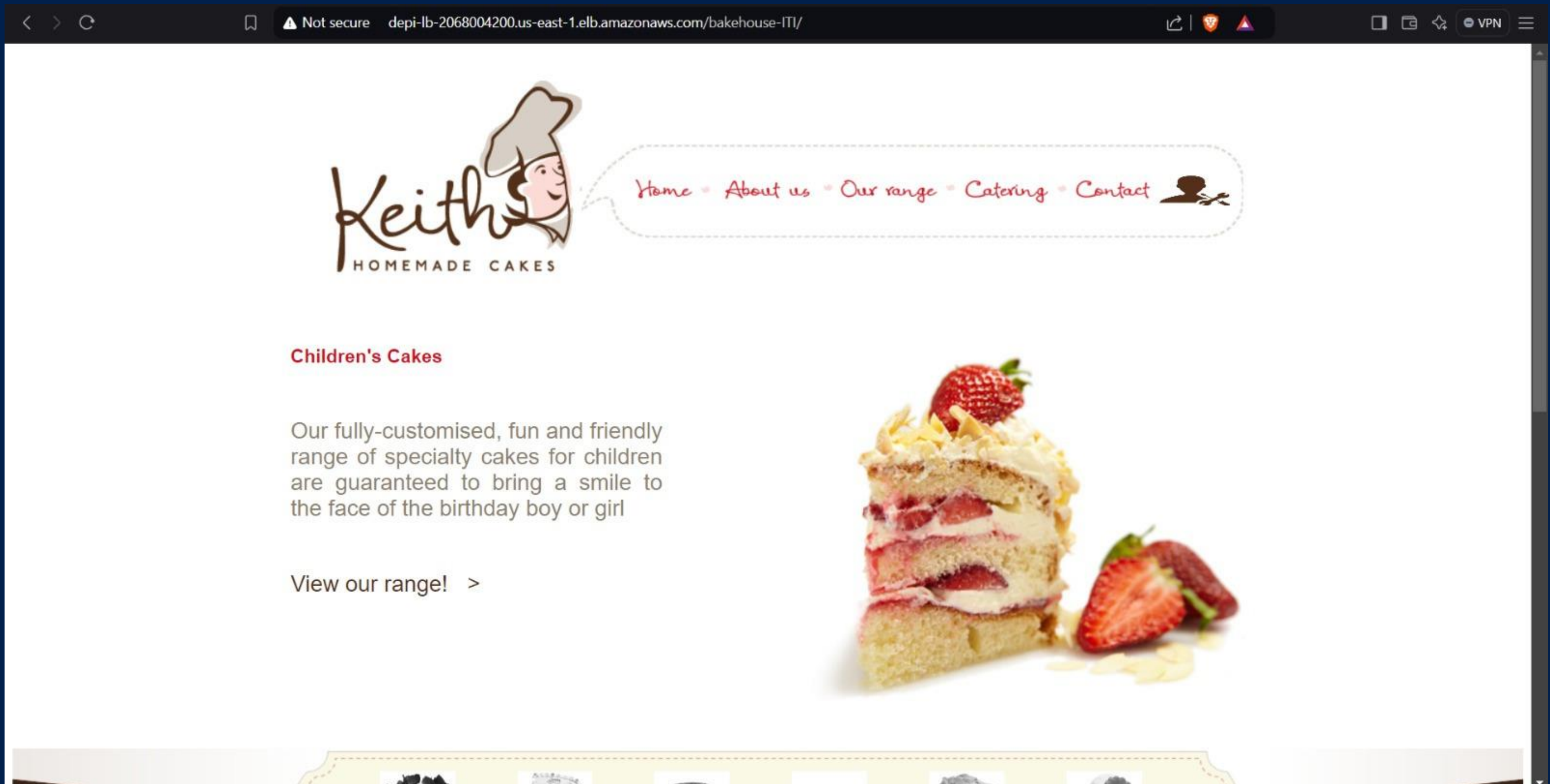
© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

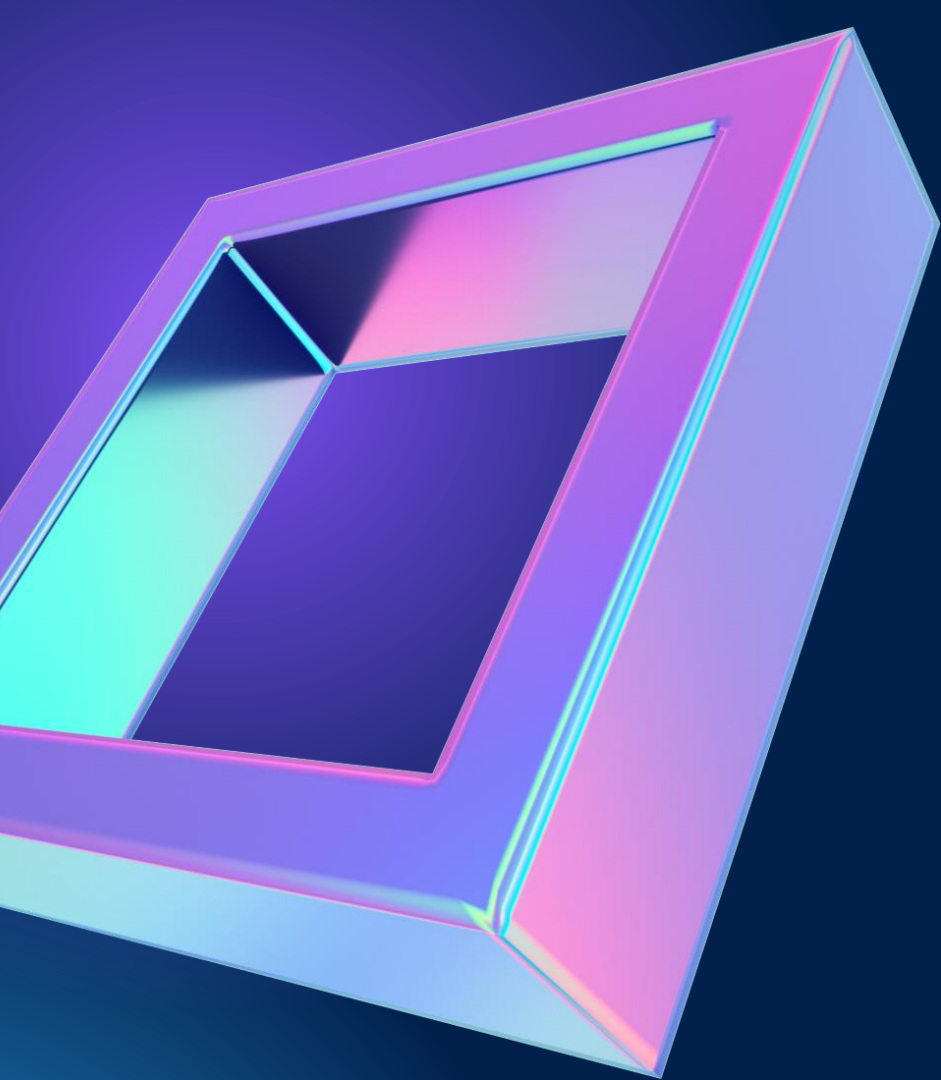
WEB RUNNING



Documentation

CONCLUSION

Successfully deployed the Bakehouse website on AWS.
Achieved high availability with multi-AZ deployment.
Scaled horizontally using ASG and ELB.
Optimized cost with efficient resource allocation.



Thank You