Write-Ups

1 - Case Study details

Deployment of Multicloud App

To deploy a multicloud app using two load balancer setups on
AWS and Azure platforms

Background of the problem statement:

A B2C company serving the global audience wants to ensure high availability for its end users using their web app which can be accessed from anywhere in the world. Since the web app gets multimillion traffic on a daily basis, the company cannot afford to have downtime on their app. So, they have decided to deploy the web app on two cloud platforms, on AWS and Azure to ensure that the resiliency is maintained.
Create an architecture and the step-by-step guide to provide a solution for the above problem statement.

Motivation

Real-World Scenario:

A logistics company wants to ensure high availability for its end users using their web application which can be accessed from anywhere in the world. Since the web application gets huge traffic everyday, the company cannot afford to have any downtime. So, they have decided to deploy the web application on two cloud platforms to maintain Resiliency.

Create an architecture and the step-by-step guide to provide a solution for this scenario.

2 - Problem understanding / Requirement Gathering details:

1. Current Scenario -
   - Logistics Company
   - Static Web App
   - Huge traffic
   - Single cloud or on-prem
   - No fail-over
   - Basic DNS hosting

2. Proposed Requirements
   a. Global Accessibility
   b. High Availability
   c. No Down time
   d. Multi-region Static web hosting
   e. Network isolation
   f. Routing setup
   g. Traffic distribution
   h. Load balancing
   i. Health monitoring
   j. DNS failover

3. Has the company provisioned a budget for deploying workloads in multiple regions within both AWS and Azure?

4. Should the deployment follow an active-active or active-passive architecture?

5. Does the company prefer DNS-based failover (e.g., Traffic Manager) or application-level routing?

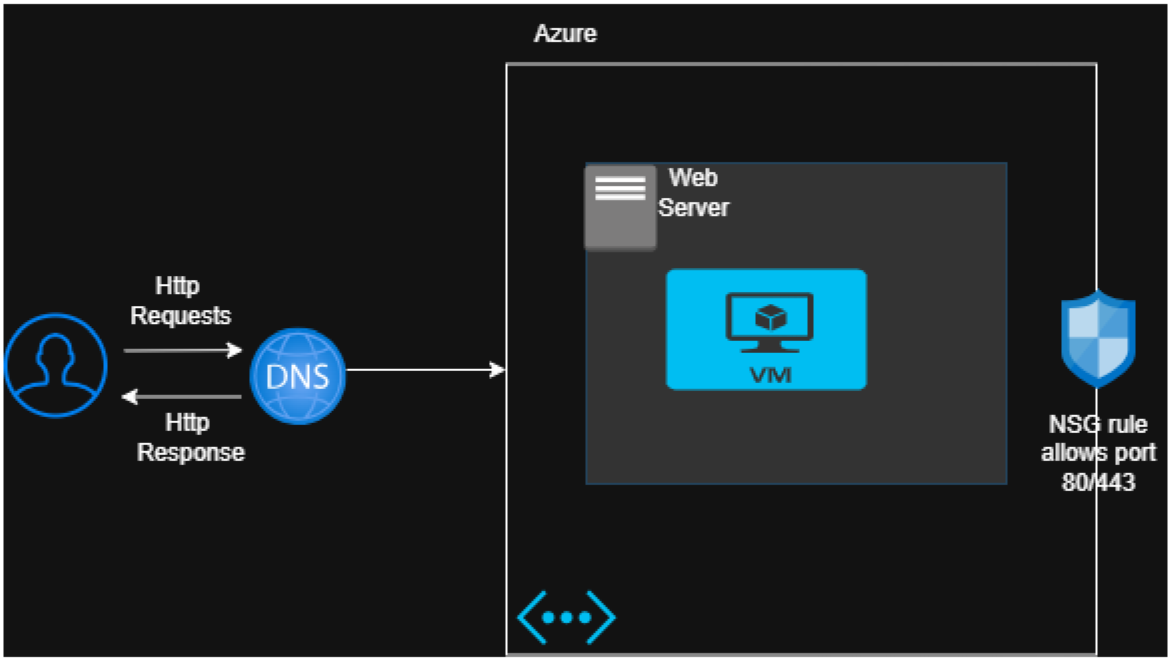
6. Are there any specific compliance requirements the company must meet?

7. Is the company subject to any particular regulatory or compliance obligations?

8. For static web application, why does the company want to use VMs instead of AWS Amplify Hosting or Azure Static Web App? Paas solutions are scalable, low-maintenance, operational simplicity and cost-effective.
   Static web apps hosted in VM involves more complexity and maintenance, high budget, etc… Is the company tied to legacy infrastructure? Or do they need full VM control for the web app? Then using virtual machines could be a good fit for this use-case.

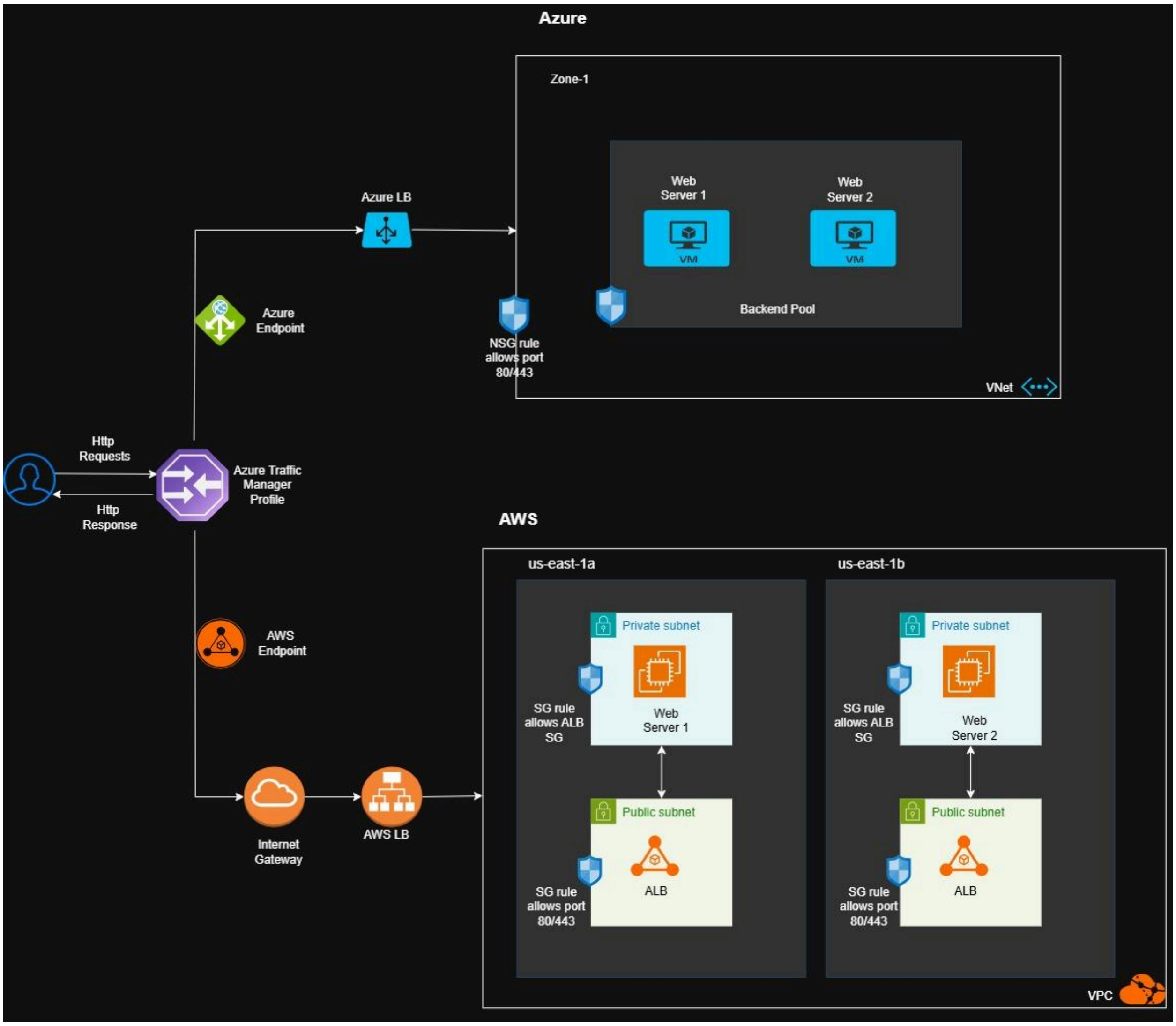
Assumption: The company opted for VMs over PaaS due to legacy dependencies and the need for full OS-level control for compliance testing.

3 - Existing Architecture:



4 - Solution Architecture:

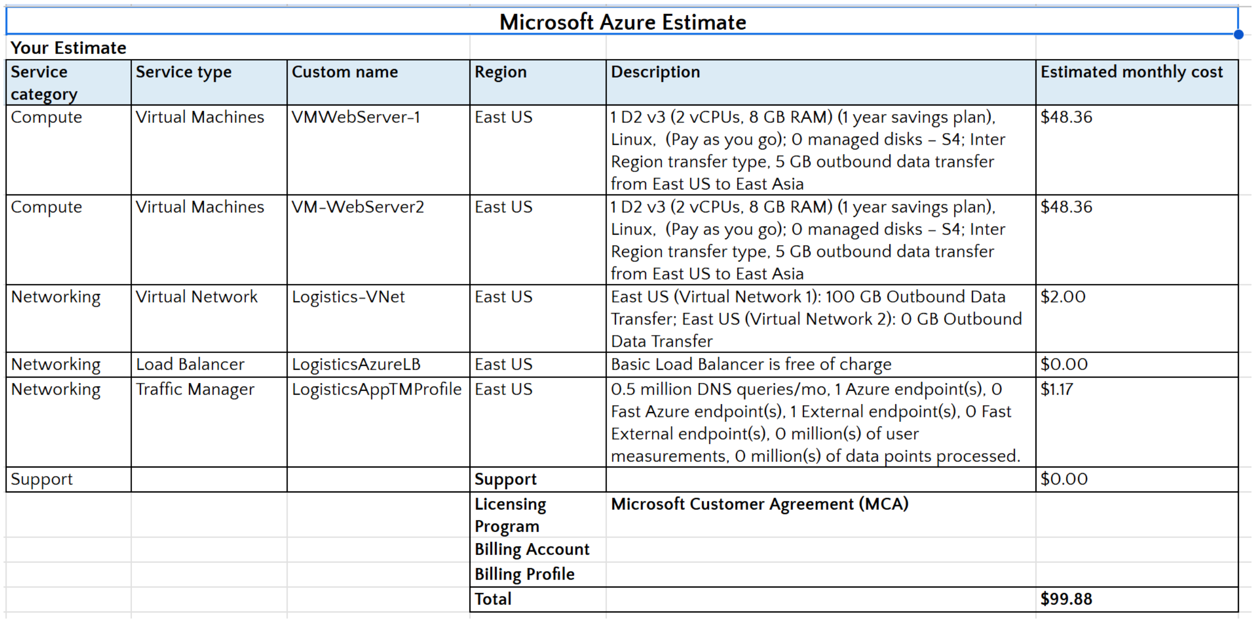
Active-Active with Weighted Failover



5 - Component Mapping:-

<table>
<tr><th colspan="2">Required AWS Components</th></tr>
<tr><th>Component</th><th>Purpose</th></tr>
<tr><td><b>Amazon EC2</b></td><td>Hosts the static web app</td></tr>
<tr><td><b>Vitual Private Cloud</b></td><td>Provides isolated network environment for EC2 and other resources</td></tr>
<tr><td><b>Internet Gateway</b></td><td>Enables internet access for resources in public subnets</td></tr>
<tr><td><b>Route tables and Subnets</b></td><td>Define traffic flow and segmentation within the VPC</td></tr>
<tr><td><b>Elastic Load Balancer (ELB)</b></td><td>Distributes incoming traffic across EC2 instances</td></tr>
<tr><td><b>Target Group</b></td><td>Registers EC2 instances for load balancing and health checks</td></tr>
<tr><th colspan="2">Required Azure Components</th></tr>
<tr><th>Component</th><th>Purpose</th></tr>
<tr><td><b>Resource Group</b></td><td>Logical container for managing and grouping Azure resources</td></tr>
<tr><td><b>Virtual Network + Subnet</b></td><td>Provides isolated network environment and segmentation for Azure resources</td></tr>
<tr><td><b>Azure Virtual Machine (VM)</b></td><td>Hosts the static web app</td></tr>
<tr><td><b>Azure Frontend IP Configuration</b></td><td>Public IP used by the Load Balancer to receive traffic</td></tr>
<tr><td><b>Backend Pool</b></td><td>Contains Azure VMs targeted by the Load Balancer</td></tr>
<tr><td><b>Health probe</b></td><td>Monitors VM health for load balancing decisions</td></tr>
<tr><td><b>Azure Load Balancer (Standard)</b></td><td>Distributes incoming traffic across Azure VMs</td></tr>
<tr><td><b>Traffic Manager Profile</b></td><td>Provides geo-redundant DNS-based routing and failover across regions</td></tr>
</table>

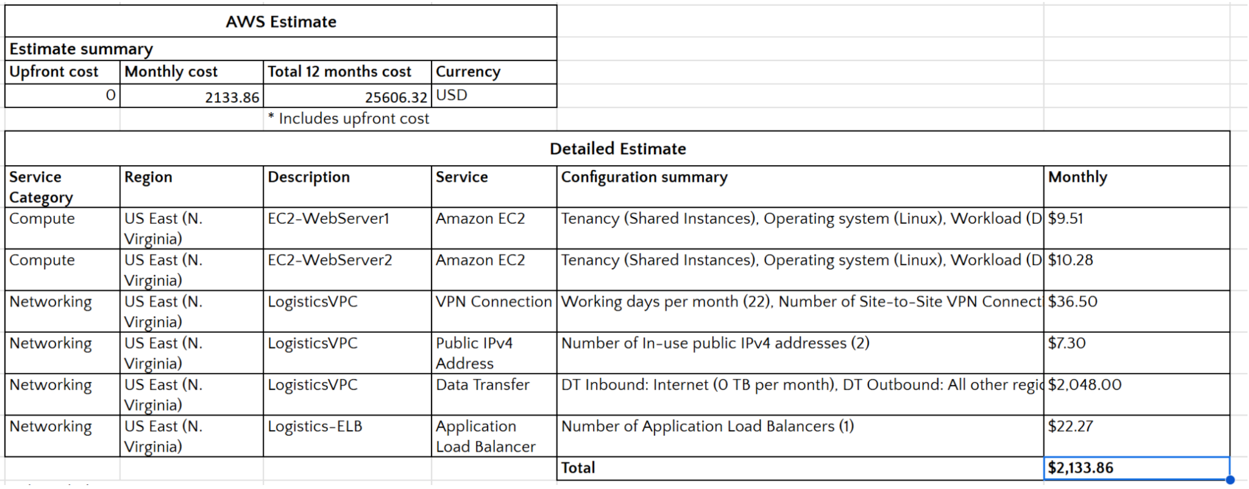
6 - Bill of Materials:-



Microsoft Azure Estimate

Your Estimate

Service category	Service type	Custom name	Region	Description	Estimated monthly cost
Compute	Virtual Machines	VMWebServer-1	East US	1 D2 v3 (2 vCPUs, 8 GB RAM) (1 year savings plan), Linux, (Pay as you go); 0 managed disks – S4; Inter Region transfer type, 5 GB outbound data transfer from East US to East Asia	$48.36
Compute	Virtual Machines	VM-WebServer2	East US	1 D2 v3 (2 vCPUs, 8 GB RAM) (1 year savings plan), Linux, (Pay as you go); 0 managed disks – S4; Inter Region transfer type, 5 GB outbound data transfer from East US to East Asia	$48.36
Networking	Virtual Network	Logistics-VNet	East US	East US (Virtual Network 1): 100 GB Outbound Data Transfer; East US (Virtual Network 2): 0 GB Outbound Data Transfer	$2.00
Networking	Load Balancer	LogisticsAzureLB	East US	Basic Load Balancer is free of charge	$0.00
Networking	Traffic Manager	LogisticsAppTMProfile	East US	0.5 million DNS queries/mo, 1 Azure endpoint(s), 0 Fast Azure endpoint(s), 1 External endpoint(s), 0 Fast External endpoint(s), 0 million(s) of user measurements, 0 million(s) of data points processed.	$1.17
Support			**Support**		$0.00
			Licensing Program	**Microsoft Customer Agreement (MCA)**	
			Billing Account		
			Billing Profile		
			Total		**$99.88**



AWS Estimate

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost	Currency
0	2133.86	25606.32	USD
		* Includes upfront cost	

Detailed Estimate

Service Category	Region	Description	Service	Configuration summary	Monthly
Compute	US East (N. Virginia)	EC2-WebServer1	Amazon EC2	Tenancy (Shared Instances), Operating system (Linux), Workload (D	$9.51
Compute	US East (N. Virginia)	EC2-WebServer2	Amazon EC2	Tenancy (Shared Instances), Operating system (Linux), Workload (D	$10.28
Networking	US East (N. Virginia)	LogisticsVPC	VPN Connection	Working days per month (22), Number of Site-to-Site VPN Connect	$36.50
Networking	US East (N. Virginia)	LogisticsVPC	Public IPv4 Address	Number of In-use public IPv4 addresses (2)	$7.30
Networking	US East (N. Virginia)	LogisticsVPC	Data Transfer	DT Inbound: Internet (0 TB per month), DT Outbound: All other regi	$2,048.00
Networking	US East (N. Virginia)	Logistics-ELB	Application Load Balancer	Number of Application Load Balancers (1)	$22.27
				Total	$2,133.86

7. Steps to Achieve Multi-Cloud Deployment for Static Web App in Virtual Machines

Phase 1: AWS Setup

1. Create VPC
2. Create Internet Gateway and attach to VPC
3. Create 2 Route tables
    - Public route table with route to internet gateway
    - Private route table with default local routes
4. Create 4 subnets
    - 1 public and 1 private subnet in us-east-1a
    - 1 public and 1 private subnet in us-east-1b
5. Create 2 SecurityGroups
    - Security Group for ApplicationLoadBalancer :-
        i. - Allow Inbound Rule - Http, port 80 from anywhere
    - Security Group for for the EC2 instances:-
        i. Allow Inbound rule for SSH from My Ip. (to install web server)
        ii. Allow Inbound Rule from ALB Sec Group id.
6. Launch 2 EC2 instances (WebApp1 & WebApp2) and place in private subnets in each zone
    - Amazon Linux 2 OS
    - Install Apache web server
    - Serve a static page (e.g., “Hello from AWS WebApp1”)
7. Create TargetGroup and register the 2 EC2 instances (Target type - Instance) with Health check on /index.html
8. Create ApplicationLoadBalancer and associate the TargetGroup and add listeners to port 80. Load balancer must be deployed in at least 2 AZs.
9. Test using ELB DNS name

Phase 2: Azure Setup

1. Create Resource Group
2. Create Virtual Network with default subnet.
   - Allow Inbound rules for Http, SSH (to connect and install web server)
3. Create 2 Azure VMs (WebApp1 & WebApp2)
   - Ubuntu Machine
   - Install web server
   - Serve static page (e.g., “Hello from Azure WebApp1”)
4. Create Azure Load Balancer
   - Front end IP config
   - Backend pool with the VMs
   - Health probe - Http, port 80, path /index.html
5. Create a Traffic Manager Profile (Weight-Based Routing)
   - Add AWS Load balancer DNS full name (External endpoint)
   - Add Azure Load balancer public ip address
   - Choose numeric value for weight (50-50)

Weighted routing will control traffic distribution (e.g., 70% AWS, 30% Azure)

Test the Setup

- Access the Traffic Manager DNS name (e.g., globalwebapptm.trafficmanager.net)
- Validate that traffic is distributed according to weights
- Simulate failover in either AWS or Azure endpoint to confirm automatic rerouting

Conclusion

This capstone project delivers a multi-cloud architecture for a logistics company’s static web application, deployed across AWS and Azure using virtual machines, load balancers, and Azure Traffic Manager with weight-based DNS routing. The solution includes secure network isolation, health probes, and traffic distribution logic to ensure high availability and global accessibility.

By leveraging weighted routing, the architecture enables flexible traffic control across cloud platforms while maintaining resilience through health-aware endpoint monitoring. This ensures that users are served from the most responsive and available region, even during partial outages.

Through this project, I gained hands-on experience with cross-cloud networking, load balancing strategies, and DNS-based traffic distribution. I also strengthened my skills in scenario-based troubleshooting, cost modeling, and documenting resilient cloud architectures for real-world deployment.