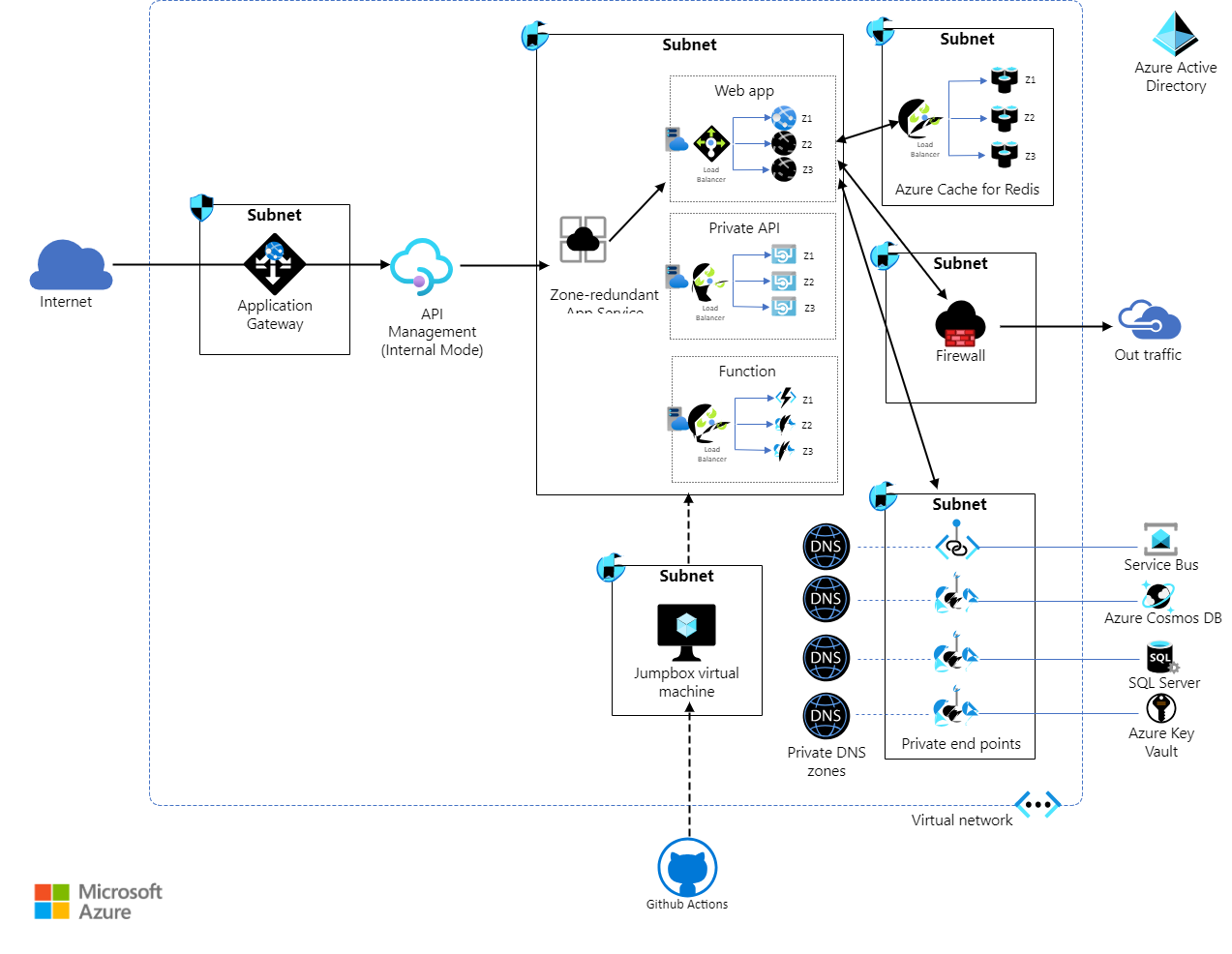
# Cloud Apps & Integration-Technical Assessment

## High-Level Design

App Service Environment v3 (ASE v3) supports High Availability (HA) and Disaster Recovery (DR) through various features and configurations:



## High Availability (HA)

### Multi-Region Deployment:

Deploy your application across multiple regions to ensure high availability. This setup allows traffic to be routed to a different region if one region becomes unavailable.

Use Azure Traffic Manager or Azure Front Door to manage traffic and provide automatic failover between regions.

### Instance Scaling:

Configure your ASE v3 to have multiple instances within a region. Azure Load Balancer distributes traffic across these instances to ensure that the failure of a single instance does not affect the application's availability.

Use autoscaling rules to automatically increase or decrease the number of instances based on demand.

### Availability Zones:

Deploy your ASE v3 in regions that support Availability Zones. This ensures that your applications are spread across physically separate datacenters within the same region, providing redundancy against datacenter-level failures.

## Disaster Recovery (DR)

### Backup and Restore:

Regularly back up your application and its data. Azure App Service provides automated backups that you can configure for your applications. These backups can be stored in different regions to ensure they are available even if the primary region is down.

Use Azure SQL Database with geo-replication or other Azure data services that support geo-redundancy to ensure that your data is replicated to a different region.

### Geo-Redundancy:

Utilize geo-redundant storage (GRS) for your application’s data storage. This ensures that your data is automatically replicated to a secondary region.

Configure your services such as Azure SQL Database, Azure Cosmos DB, and others to use geo-replication.

### Disaster Recovery Plan:

Develop and maintain a disaster recovery plan that includes procedures for failover and failback. Regularly test this plan to ensure that it works as expected and that your team is familiar with the process.

Use Azure Site Recovery to orchestrate replication, failover, and recovery of applications and data to ensure business continuity.

• What are your recommendations for their high-level requirements?

1. Resiliency, availability, recovery
2. Data protection, threat detection, and mitigation
3. Cost modeling, budgets, reduce waste
4. Holistic observability, DevOps practices
5. Scalability, load testing

• Are there other considerations you think are important for what this customer wants to achieve?

1. The Payment Card Industry Data Security Standard (PCI DSS)
2. WAF polices
3. Cost optimization vs security requirements
4. Cost optimization VS Storage requirements (Specially around Log Analytics)
5. Monitoring, Workbook and Alert requirement

• How long do you think it would take to implement your recommendations?

1. DevOps Setup & Initial Bicep module/script development – 2 days
2. Landing Zone & Networking – 1 week
3. Core service deployment bicep – 2 days
4. Application infrastructure & Configurations bicep -3 days
5. Application/Api/Workflow development – 2 weeks

• What skill sets would be required to implement your recommendations?

1. DevOps Skills – bicep/Terraform
2. Azure Infrastructure Skills
3. Azure Networking Skills
4. Azure iPaaS skills

• An example implementation of a Get Products API endpoint submitted via a GitHub repository