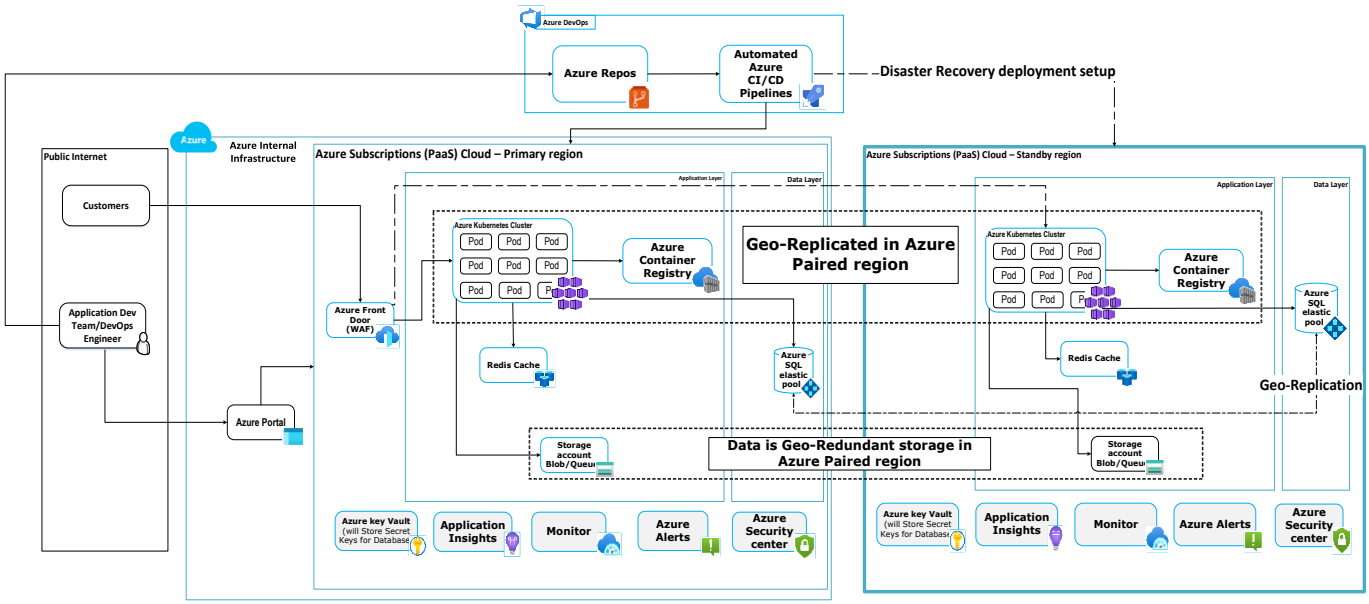
Solution Architecture Design – Assignment



Solution Overview:

1. The monolithic application will be broken down into microservice based on the business functionality and it will follow a microservice architecture.
2. The application code will be refactored using modern technology stack and containerized.
3. All container images will be pushed to Azure Container Registry through Azure CI/CD pipelines.
4. All images are hosted in Azure Kubernetes a container orchestration platform.
5. Binary objects are stored in Azure Blob storage.

Technology stack:

- Angular/.Net C#/Node JS/Python.
- Azure SQL elastic pool.

Requirement consideration:

1. ***Scaling to meet the demand:***
    a. Kubernetes supports both manual and auto scale for Pods & nodes.
    b. 'Horizontal pod autoscaler' (HPA) feature can used for to scale the Pods. HPA checks the Metric API for every 30 seconds and based on CPU usage it can increase or decrease accordingly.
    c. 'Cluster autoscaler' feature can be used to scale the nodes.

2. ***CI/CD setup:***
    a. Azure DevOps service can be leveraged to share code, track work and ship software.
    b. Developers will maintain code in Azure Repos.
    c. DevOps engineer will create service connections with target azure platform and created automated Build/Release Pipelines. Azure Artifacts can be fed into the pipelines.
    d. Azure Test plans can be used to do manual test, UAT or created automated test workflow.
    e. Team members will use Azure boards to track user stories/work Items.
3. ***Disaster recovery setup:***
    a. Azure SQL Elastic pool has built-in availability SLA of 99.99%.
    b. Cross region read replicas can be leveraged to enhance business continuity and disaster recovery planning.
    c. Backups can be used to restore server to a point-in-time.
4. ***Distribute load:***
    a. Azure Front door has in-built capability to load balance between the backends.
    b. Load-balancing will be done based on the load balancing rules configured in front door.
5. ***Reducing latency:***
    a. Azure Front door will send the user requests to the "closest" set of backends in respect to network latency.
    b. Azure CND has several POP locations to cache the static content near to the user.
    c. Azure Redis to cache the page output for better performance.
6. ***Encryption in transit and rest:***
    a. Azure SQL elastic pool encrypts data in motion with Transport Layer security (TLS) and data in rest with Transparent Data Encryption (TDE).
    b. The service uses the AES 256-bit cipher included in Azure storage encryption, and the keys are system managed. Option for Customer managed keys is also available.
    c. Storage encryption is always on and can't be disabled.
7. ***Secure accessing:***
    a. All web-traffic over HTTPS (TLS1.2).
    b. Developer, administrators, DevOps engineer are authenticated to the platform using Azure Active directory.
    c. Developer, administrators, DevOps engineer are authorized to resources using azure role-based access (RBAC) controls.
    d. Azure services use ‘Managed identity’ authentication to communicate to each other.
    e. Customers accessing website using Azure AD B2C login.
8. ***Monitoring:***
    a. Azure monitor maximizes the availability and performance of the application and services by delivering a comprehensive solution for collecting, analyzing, and acting on telemetry from cloud environments.
    b. Azure monitor can be used to monitor Kubernetes nodes/Pods, setting alerts and sending notifications.
    c. Application insights monitors the availability, performance and usage of the microservices.
    d. Azure SQL Elastic pool will leverage the following monitoring features.
        - Query store

- Metrics
- Server logs
- Query performance insights
- Performance recommendation
- Planned maintenance notification.

9. Application Lifecycle:

a. Azure Resource manager (ARM) template/Terraform scripts can be leveraged to manage Infrastructure as a code
b. Use Azure DevOps repos for source control.
c. Automate build and deployments using Azure CI/CD pipelines to Azure.

10. Deployment:

a. Deployments will target 2 subscriptions.
    - Dev, QA.
    - Production.
b. Pre-Prod subscription will house Dev, Test.
c. Production azure subscription will house production environment only.

11. Replicate on-demand environment based on the Azure blueprint architecture:

a. Azure blueprints package the different infrastructure components into a blueprint. Once the blueprint is created it is easy to assign the blueprint to stand up a new environment.
b. Azure blueprints can reside at the management group level to be assigned for multiple subscriptions under this group or at the Subscription level to be assigned for multiple resource groups.
c. Azure blueprint takes care of packaging Azure policy definitions, Role based access controls, ARM templates.