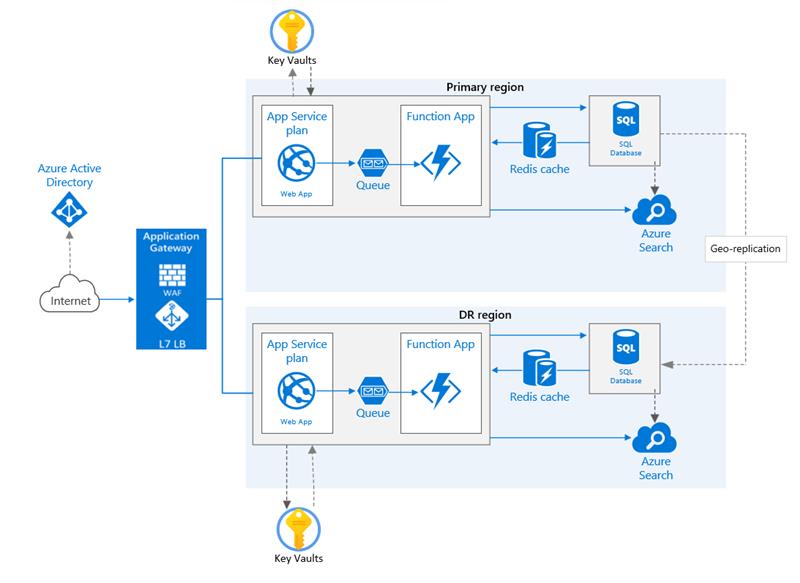
Business case:

Customer has a website running in an outdated platform hosted in datacenter. To modernize it, they plan to launch a new website Ghost Blog platform in Cloud. During the launch, the users/traffic flow will be 4 times than earlier. The customer is also interested in disaster recovery capabilities in case of a region failure. The teams want to be able to release new versions of the application multiple times per day, without requiring any downtime. The website will be exposed to the internet; thus, the security team also needs to have visibility into the platform and its operations.

Solutions Overview:

Solution 1:



Azure Services to be used for this scenario:

1. Azure AD Directory
2. Application Gateway
3. App registration
4. VNET/Subnet
5. App Services
6. SQL Database
7. Redis cache
8. Azure Keyvault
9. Function App
10. Application Insights

Scope of Work:

The above requirement can be achieved with the help of Blue green deployment in Azure. The website will be hosted in Azure within the resource group with Premium App Service Plan in two different regions. Azure AD will authenticate the users who were already registered with the Ghost Blog platform.

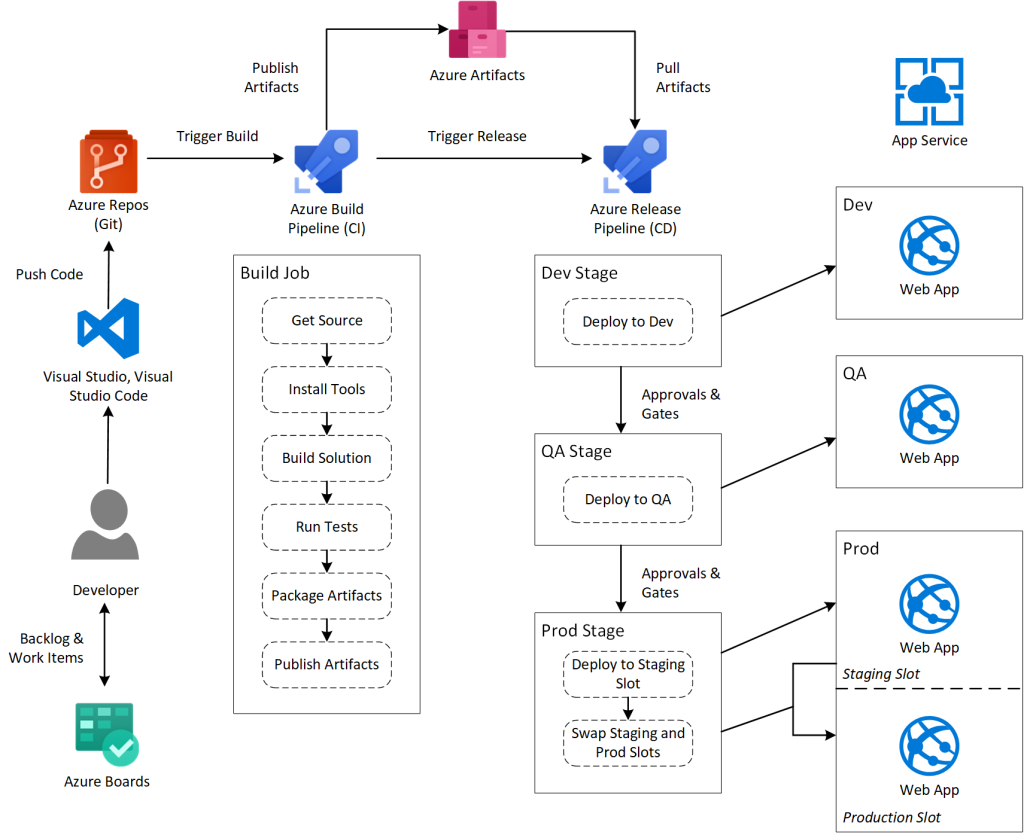
Application Gateway plays a key role to allow and distribute the traffic to the backend app services and it acts as a Web Application Firewall which protects the web app by filtering, monitoring and blocking any malicious HTTP/s traffic.

Database to be configured as Geo-redundant in Azure.

A new service principal name is created and configured in the Azure Key vault to authenticate the secret keys to host/deploy the app services in Azure and Azure Key vault will be used to store the keys and certificates.

When there are increases in the load the during new launch or peak time, the app service will be scaled out to meet the loads by increasing the number of instances and can be scaled in automatically once the load goes down below the threshold.

Geo-replication is enabled to the DR site which will be in different region and will be available active even if there is any geographical failure.



Azure Boards provides backlogs and work item tracking to help development teams collaborate and coordinate their work.

Azure Repos fires a trigger to launch a Build Pipeline. The Build Pipeline includes jobs and tasks that clone the repo, install tools, build the solution, and then package and publish artifacts to Azure Artifacts.

Release Pipeline is responsible for deploying the application artifacts to development, QA, and production environments. The Release Pipeline is organized into stages which, although executed sequentially, act independently of each other. In this scenario, the Dev stage deploys the application to a Dev environment. This environment is typically hosted in a non-production Subscription and may share an App Service Plan with other non-production environments such as QA.

Between stages, you use approvals and gates to control when the next stage is executed. This allows your team to perform testing and validation in each stage before moving onto the next.

Blue-Green Deployment, the staging slot represents your “green” deployment. The production slot represents your “blue” deployment. Once you validate that everything has been successfully deployed to the staging slot (i.e., green), the Prod stage performs a swap of green and blue. This makes the green deployment live for end-users and moves the blue deployment to your staging slot where it remains until you remove it. If problems arise with the new green deployment, then you can swap again to move blue back to production.

**Solution 2:**

Using Terraform, we can deploy a Kubernetes Cluster (Cluster auto scaler, Ingress, Argo-CD, Key clock for Authentication) and create the applications in the docker container and need to do the manifest and send to the Argo-CD in Kubernetes cluster, so that the auto deployment for the pods will be deployed in multi-region. The Argo-CD and the GitHub should be tightly integrated so that the changes done in the GitHub will automatically triggered.