Contents

[1 Azure CI/CD 2](#_Toc46260135)

[1.1 CI/CD for Azure VMs 2](#_Toc46260136)

[1.1.1 Data Flow 2](#_Toc46260137)

[1.1.2 Components 2](#_Toc46260138)

[1.2 CI/CD for Containers 3](#_Toc46260139)

[1.2.1 Data Flow 3](#_Toc46260140)

[1.2.2 Components 3](#_Toc46260141)

[1.3 Container CI/CD using Jenkins and Kubernetes on Azure Kubernetes Service (AKS) 4](#_Toc46260142)

[1.3.1 Data Flow 4](#_Toc46260143)

[1.3.2 Components 4](#_Toc46260144)

[2 WordPress and Azure DevOps 5](#_Toc46260145)

[2.1 WordPress in Azure App Service Containers using Docker 5](#_Toc46260146)

[2.1.1 Components 6](#_Toc46260147)

[2.2 WordPress in Web App Production and Staging Slots 6](#_Toc46260148)

[3 Magento and Azure DevOps 7](#_Toc46260149)

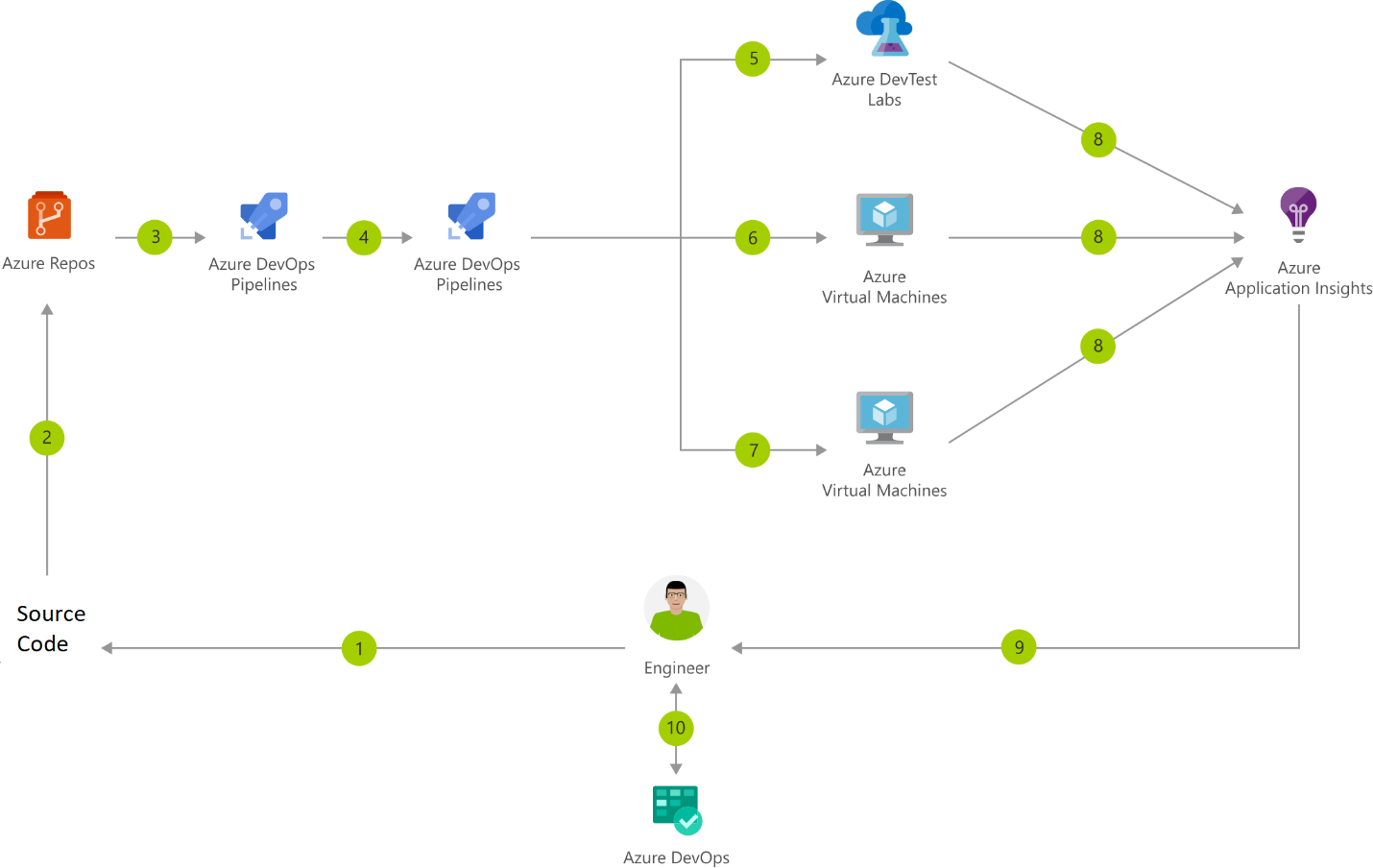
[3.1 Magento in Azure Kubernetes Service (AKS) 7](#_Toc46260150)

[3.2 Components 7](#_Toc46260151)

# Azure CI/CD

Azure DevOps provides the CI/CD pipeline, starting with a Git repository for managing your application source code and infrastructure code (ARM templates), a Build system for producing packages and other build artifacts, and a Release Management system for setting up a pipeline to deploy your changes through dev, test, and production environments. The pipeline uses ARM templates to provision or update your infrastructure as necessary in each environment, and then deploys the updated build. You can also use Azure DevTest Labs to automatically tear down test resources that are not in use.

## CI/CD for Azure VMs



### Data Flow

1. Change application source code

2. Commit Application Code and Azure Resource Manager (ARM) Template

3. Continuous integration triggers application build and unit tests

4. Continuous deployment trigger orchestrates deployment of application artifacts with environment-specific parameters

5. Deployment to QA environment

6. Deployment to staging environment

7. Deployment to production environment

8. Application Insights collects and analyses health, performance, and usage data

9. Review health, performance and usage information

10. Update backlog item

### Components

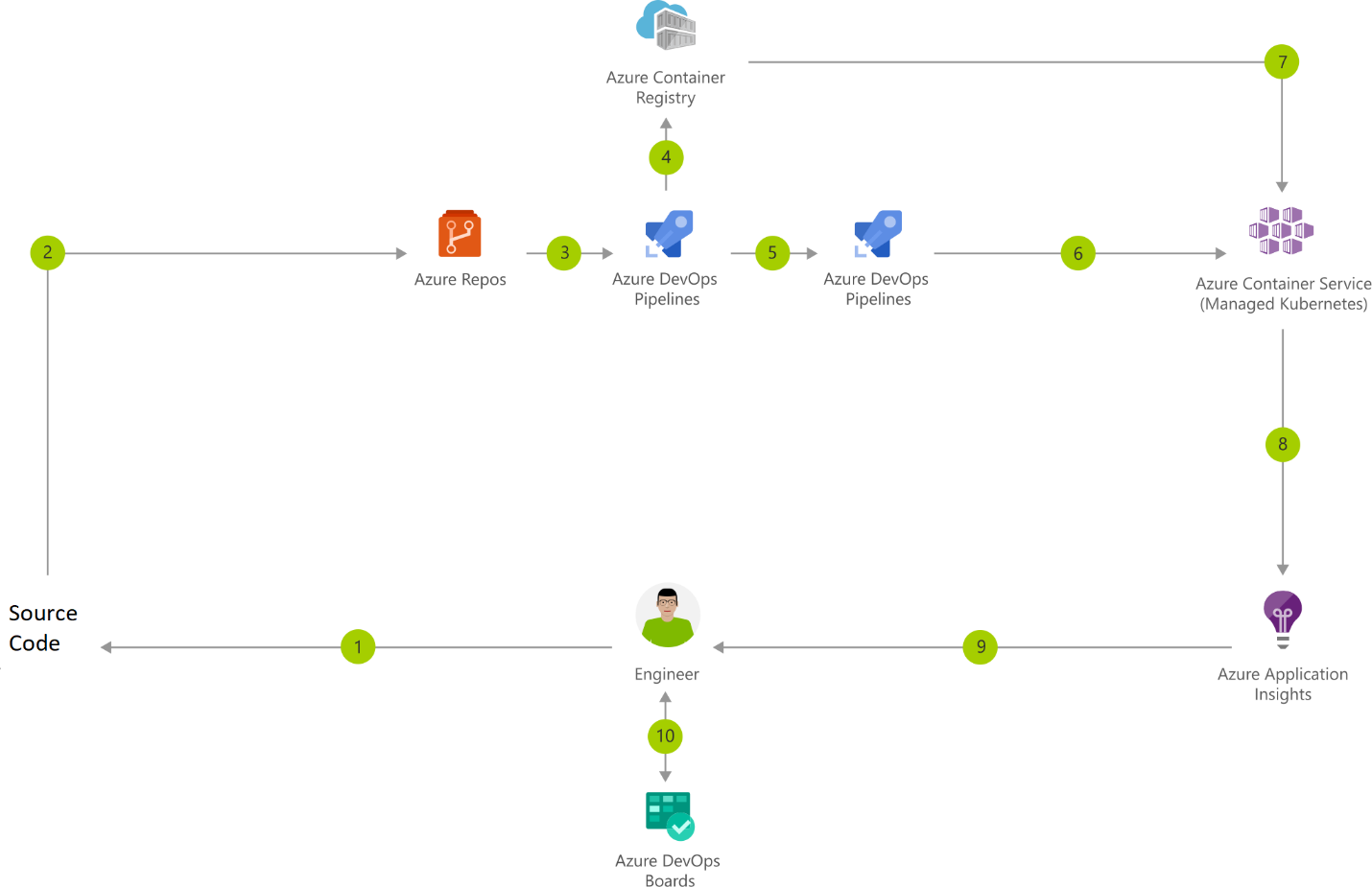
• Virtual Machines: Provision Windows and Linux virtual machines in seconds

• Azure DevTest Labs: Quickly create environments using reusable templates and artifacts

• Application Insights: Detect, triage, and diagnose issues in your web apps and services.

• Azure DevOps: Build and deploy multi-platform apps to get the most from Azure services

## CI/CD for Containers



### Data Flow

1. Change application source code

2. Commit Application Code

3. Continuous integration triggers application build, container image build and unit tests

4. Container image pushed to Azure Container Registry

5. Continuous deployment trigger orchestrates deployment of application artifacts with environment-specific parameters

6. Deployment to Azure Kubernetes Service (AKS)

7. Container is launched using Container Image from Azure Container Registry

8. Application Insights collects and analyses health, performance, and usage data

9. Review health, performance and usage information

10. Update backlog item

### Components

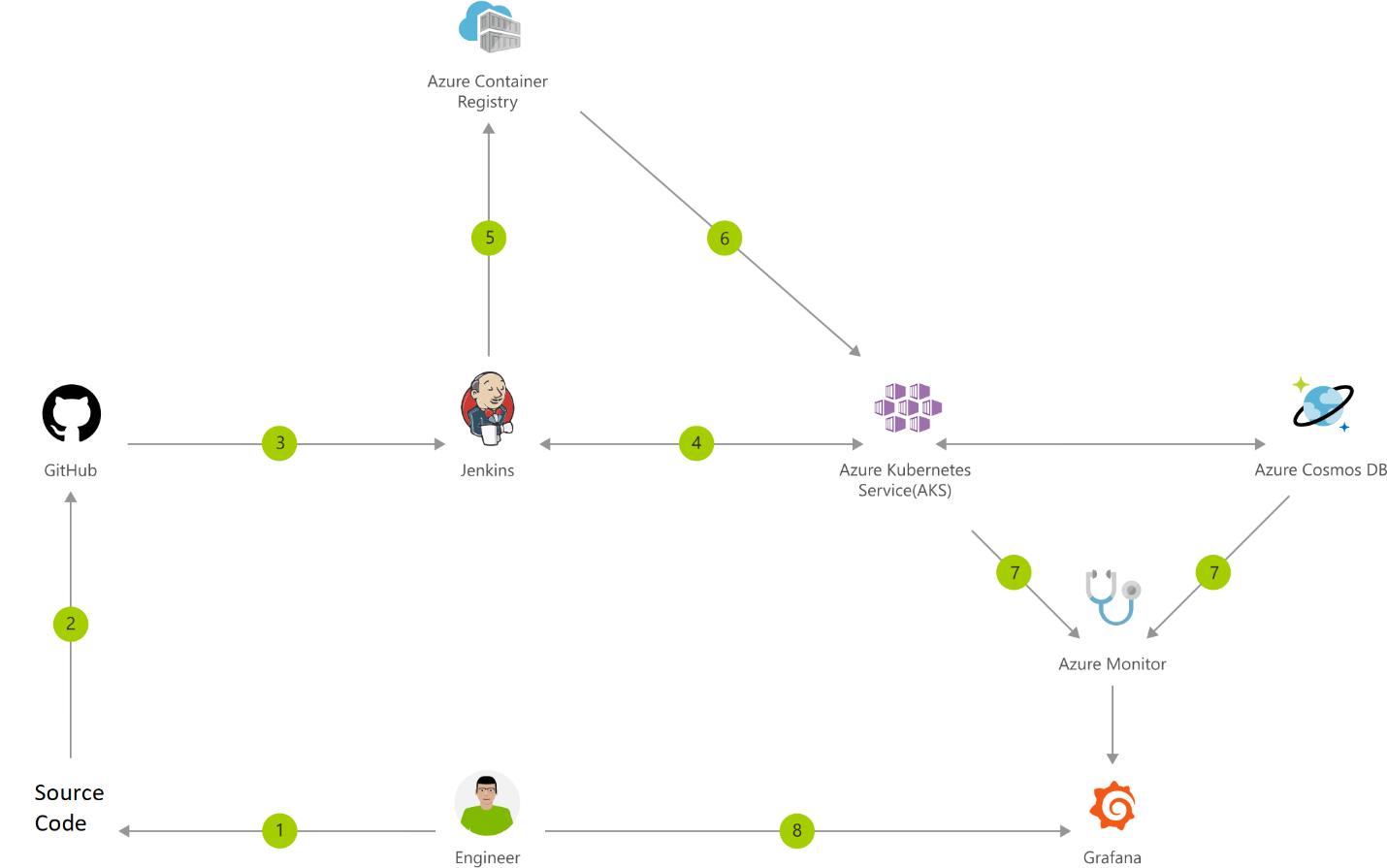
• Container Registry: Store and manage container images across all types of Azure deployments

• Azure Kubernetes Service (AKS): Simplify the deployment, management, and operations of Kubernetes

• Application Insights: Detect, triage, and diagnose issues in your web apps and services

• Azure DevOps: Build and deploy multi-platform apps to get the most from Azure services

## Container CI/CD using Jenkins and Kubernetes on Azure Kubernetes Service (AKS)



### Data Flow

1. Change application source code.

2. Commit code to GitHub.

3. Continuous Integration Trigger to Jenkins.

4. Jenkins triggers a build job using Azure Kubernetes Service (AKS) for a dynamic build agent.

5. Jenkins builds and pushes Docker container to Azure Container Registry.

6. Jenkins deploys your new containerized app to Kubernetes on Azure.

7. Container Service (AKS), backed by Azure Cosmos DB.

8. Grafana displays visualization of infrastructure and application metrics via Azure Monitor.

9. Monitor application and make improvements.

### Components

• Azure Kubernetes Service (AKS): Simplify the deployment, management, and operations of Kubernetes.

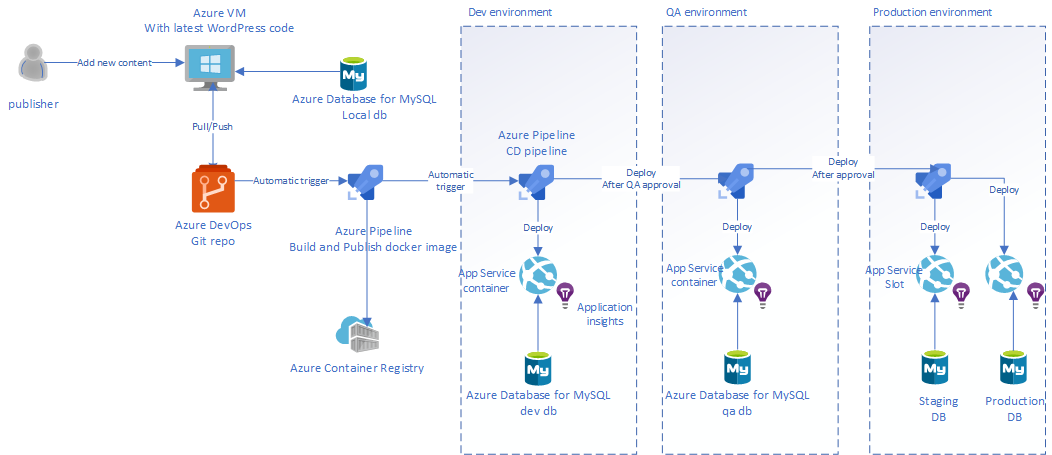
• Container Registry: Store and manage container images across all types of Azure deployments.

• Azure Cosmos DB: Globally distributed, multi-model database for any scale.

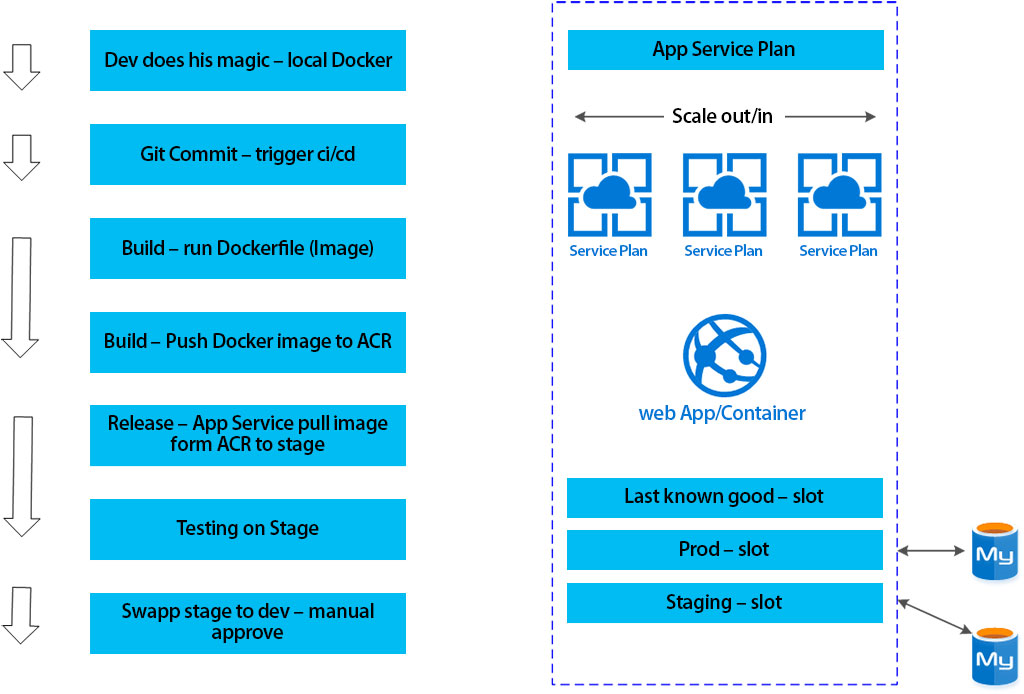
• Azure Monitor: Highly granular and real-time monitoring data for any Azure resource.

# WordPress and Azure DevOps

## WordPress in Azure App Service Containers using Docker



Environments: Development, QA, and Production. The Development environment is for the developer to run the Docker images locally, commit the required changes, and push the code to the Azure DevOps Git repo. The push action will initiate a CI process, which will build and push a new Docker image to the Azure Container Registry. Then WordPress Docker image is built on the basis of a Dockerfile. Azure Database for MySQL is the WordPress database and each environment will have a separate database that will be synced with one another by performing database export from one environment to another. Azure Application Insights is used for logging and monitoring purposes.



### Components

• WordPress Docker image built on the basis of a Dockerfile

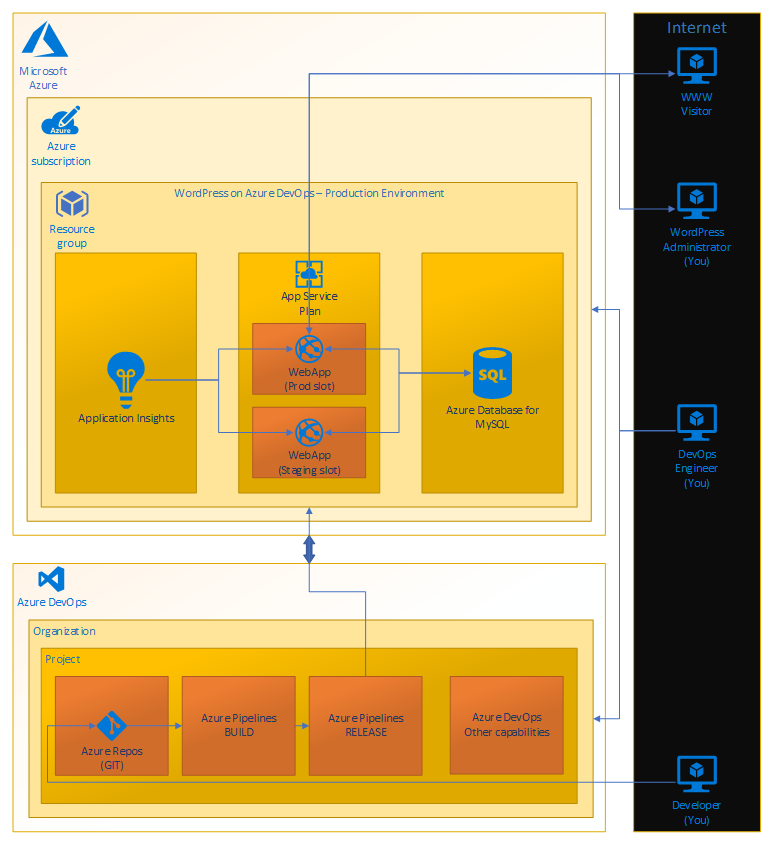
• Azure Repos - Docker WordPress project and WordPress development project

• Developer station with Docker Containers and Git installed on it - Git clone from Azure DevOps Git repo for local development (or work directly with Azure App Service)

• Azure Container Registry (ACR) for Private Registry - Link Azure Container Registry (ACR) with Azure App Service which hosts the site

• Azure DevOps - Execute the pipeline

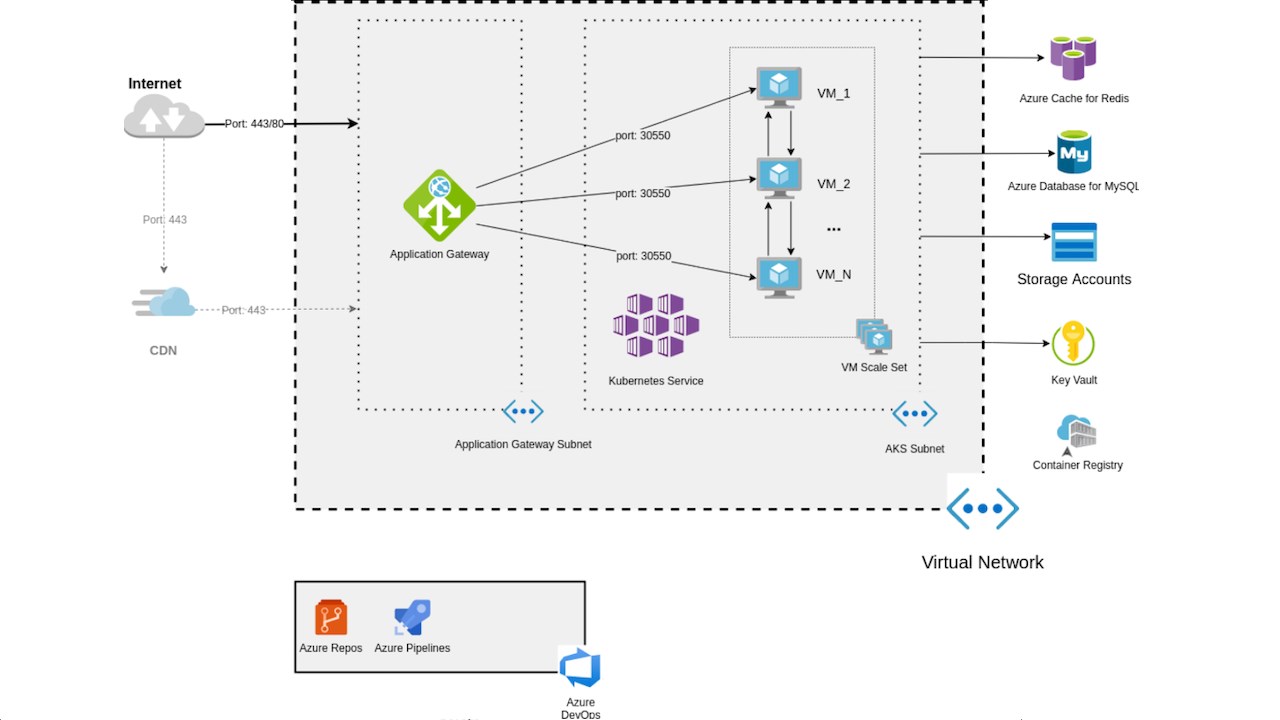
## WordPress in Web App Production and Staging Slots



The main part of the WordPress application is in the Web App Production slot. This is basically both front-end and back-end in one. MySQL database is running on extra server, where the WordPress articles are stored. On the left side there is Application Insights which monitors the Web App performance. Besides the Web App Production slot, there is a Web App Staging slot. It is utilized during application deployment process and ensures there is no Production outage. It provides an easy roll-back option as well. These infrastructure components lie within Microsoft Azure. Below the Microsoft Azure there is Azure DevOps. Azure Repos holds the WordPress source code. This is input for the build pipeline which composes the application into an installation package. The last step is release pipeline which takes the package and deploys it into our Production environment.

# Magento and Azure DevOps

## Magento in Azure Kubernetes Service (AKS)



## Components

• Magento application components deployed into Azure Kubernetes Service (AKS) managed Kubernetes cluster for horizontal workload scalability and load balancing in order to respond to outages, peak or incidental traffic

• Azure Kubernetes orchestration using Azure Kubernetes Service (AKS)

• Azure CI/CD pipelines for Development, QA, and Production environments in Kubernetes cluster