

GRIP Case Assignment

Infrastructure Design & Setup

Migrating a monolithic application into cloud-native application

Virtual machines

Azure virtual machines are an on-demand, scalable computing resource that Azure offers. When you choose a VM, you usually need more control over the computing environment than the other choices offer. However, it gives you the flexibility of virtualization without having to buy and maintain the physical hardware that runs it.

Use cases:

- Development and test - Offer a quick and easy way to create a computer with specific configurations required to code and test an application.
- Apps in the cloud – Because demand for your application can fluctuate, running it on a virtual machine in Azure might make economic sense.
- Extended data center – Connecting to your organization's network is easy.

Key Features:

Availability

We can archive 99.99% of availability when you have two or more instances deployed across two or more Availability zones in the same Azure region.

We can create and manage a group of load-balance virtual machines. The number of VM instances can automatically increase or decrease in response to demand or a defined schedule.

Scale sets provide high availability to your apps

Sizes and pricing

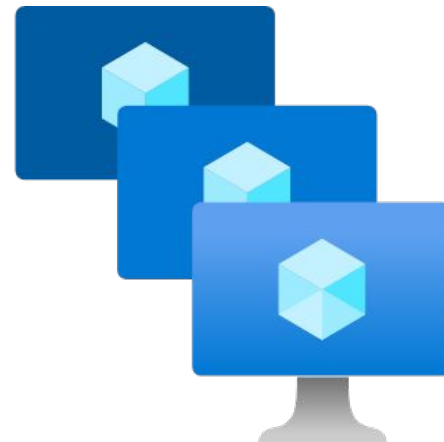
The size of the virtual machine you use is determined by the workload you want to run.

Managed Disks

Managed Disks handles Azure Storage account creation and management in the background for you and ensures that you do not have to worry about the scalability limits of the storage account.

Distributions

Microsoft Azure supports a variety of Linux and Windows distributions. In addition, you can find available distributions in the marketplace.



Azure Migrate

This tool provides a simplified migration, modernization, and optimization service for Azure. All pre-migration steps, such as discovery, assessments, and right-sizing on-premises resources, are included for infrastructure, data, and applications.

Use cases:

Unified migration platform: A single portal to start, run, and track your migration to Azure.

Tools like:

Azure Migrate tools include Azure Migrate: Discovery and assessment and Migration and modernization. Azure Migrate also integrates with other Azure services and tools and with independent software vendor (ISV) offerings.

Assessment, migration, and modernization: In the Azure Migrate hub, you can assess, migrate, and modernize:

- Databases - Assess on-premises SQL Server instances and databases to migrate them to an SQL Server on an Azure VM or an Azure SQL Managed Instance or to an Azure SQL Database.
- Web applications - Assess on-premises web applications and migrate them to Azure App Service and Azure Kubernetes Service.
- Virtual desktops - Assess your on-premises virtual desktop infrastructure (VDI) and migrate it to Azure Virtual Desktop.
- Data: Migrate large amounts of data to Azure quickly and cost-effectively using Azure Data Box products.



Azure Kubernetes Service (AKS)

This service simplifies deploying a managed Kubernetes cluster in Azure by offloading the operational overhead to Azure. A control plane is automatically created and configured when you create an AKS cluster. As a result, you only pay for and manage the nodes attached to the AKS cluster.

Key Features:

Access, security, and monitoring

For improved security and management, you can integrate with Azure AD to:

- Use Kubernetes role-based access control (Kubernetes RBAC).
- Monitor the health of your cluster and resources.

Integrated logging and monitoring

Azure Monitor for Container Health collects memory and processor performance metrics from containers, nodes, and controllers within your AKS clusters and deployed applications. You can review both container logs and the Kubernetes logs (Stored in Azure Log Analytics)

Clusters and nodes

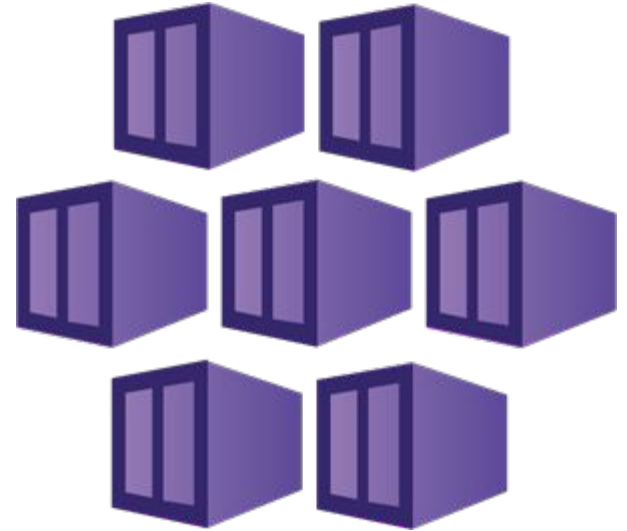
AKS nodes run on Azure virtual machines (VMs). With AKS nodes, you can connect storage to nodes and pods, upgrade cluster components, and use GPUs.

Virtual networks and ingress

An AKS cluster can be deployed into an existing virtual network. In this configuration, every pod in the cluster is assigned an IP address in the virtual network. It can directly communicate with other pods in the cluster and other nodes in the virtual network.

Ingress with HTTP application routing

The HTTP application routing add-on helps you quickly access applications deployed to your AKS cluster. The HTTP application routing solution configures an ingress controller in your AKS cluster when enabled.



Azure App Service

This Service is an HTTP-based service for hosting web applications, REST APIs, and mobile backends. You can develop in your favorite language. Applications run and scale with ease on both Windows and Linux-based environments.

App Service not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. In addition, you can also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and TLS/SSL certificates.

With App Service, you pay for the Azure compute resources you use. The compute resources you use are determined by the App Service plan on which you run your apps. Key features



Azure Container Registry (ACR)

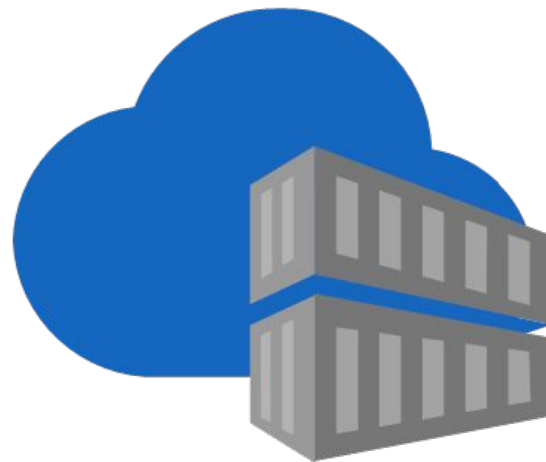
ACR is a managed registry service based on Docker Registry, used to store and manage your container images and related artifacts.

Use cases

- Scalable orchestration systems that manage containerized applications across clusters as Kubernetes.
- Azure services that support building and running applications at scale, including Azure Kubernetes Service (AKS), App Service, and Service Fabric.
- Developers can also push to a container registry as part of a container development workflow. For example, target a container registry from continuous integration and delivery tools such as Azure Pipelines or Jenkins.

Key features

- **Registry service tiers** - Create one or more container registries in your Azure subscription. Use the geo-replication feature of Premium registries for advanced replication and container image distribution scenarios.
- **Security and access** - You log in to a registry using the Azure CLI or the standard docker login command. ACR transfers container images over HTTPS, and supports TLS to secure client connections.
- **Supported images and artifacts** - Grouped in a repository, each image is a read-only snapshot of a Docker-compatible container.
- **Automated image builds** - Use ACR Tasks to streamline building, testing, to push, and deploying images in Azure.



Migration (Option 1)

1. Create a new project on Azure Migrate
2. Click in Discover (Assessment tools)
3. We are using the RedHat Linux distribution we have two types of options
 - a. With VMware vSphere Hypervisor
 - b. With Hyper-V

To discover Azure provides the Azure Migrate appliance to grant access to read details, about server size, as well as running processes.

Under the covers, the appliance measures utilization over time for VM sizing recommendations.

4. After Discover, we have to create an assessment, that comprises a few things.
 - a. Azure Readiness, which measures which servers are suitable for migration to Azure
 - b. Size recommendations, that avoid paying more than we need
 - c. Monthly cost post-migration
5. Target the machine found to migrate into Azure (Manage > Replicated machines)
Once the initial replication for a VM is complete, the service will continue to replicate those changes, this progress is block-level and near continuous.
To prevent any data loss, the final process will shut down your VM on-premises and do one more replication sync into Azure
6. Once complete you have to point your network traffic to do it.

Migration (Option 2)

1. Create a new project on Azure Migrate
2. Click in Discover (Assessment tools)
3. We are using the RedHat Linux distribution we have two types of options
 - a. With VMware vSphere Hypervisor
 - b. With Hyper-V

To discover Azure provides the Azure Migrate appliance to grant access to read details, about server size, as well as running processes.

Under the covers, the appliance measures utilization over time for VM sizing recommendations.

4. After Discover, we have to create an assessment, that comprises a few things.
 - a. Azure Readiness, which measures which servers are suitable for migration to Azure
 - b. Size recommendations, that avoid paying more than we need
 - c. Monthly cost post-migration
5. Using the Azure Migration tools menu click on Replicate
6. Select our assessment on the Virtual machines tab and specify the machine that you want to migrate
7. In the Target settings tab, we have the possibility to include Availability options and Disk encryption types., very important options to increase security and redundancy.
8. Select the VM size and click “Replicate”
9. Backing to Azure Migrate, click on Replicating servers, select your server and click “Migrate”

Containerize the app and migrate to AKS

1. Download the Azure Migrate: App Containerization installer and execute with Powershell
2. At the sign-in screen, use the local administrator account on the machine to sign in.
3. Select .NET web apps as the type of application you want to containerize.
4. To specify the target Azure service, select Containers on AKS.
5. Complete tool pre-requisites
 - a. Connectivity: The tool checks that the machine has internet access
 - b. Install updates: The tool will automatically check for the latest updates and install them
 - c. Enable PowerShell remoting: The tool will inform you to ensure that PowerShell remoting is working on the application servers running the .NET apps to become a container.
6. Sign in to Azure (Specify the subscription and tenant that you want to use)
7. Discover .NET applications
 - a. Specify the IP address/FQDN and the credentials of the server running the .NET app used to connect to the server for application discovery remotely.
 - b. Click Validate to verify that the app server is reachable from the machine running the tool and that the credentials are valid. Upon successful validation, the status column will show the status as Mapped.
 - c. Click Continue to start application discovery on the selected application servers.
 - d. Upon successful completion of application discovery, you can select the list of applications to containerize.
 - e. Select on the checkbox the apps to containerize and specify the container name
8. Parameterizing the configuration makes it available as a deployment time parameter (this option is helpful for parameters like database connection strings)
 - a. Click app configurations to review detected configurations.
 - b. Select the checkbox to parameterize the detected application configurations.
 - c. Click Apply after selecting the configurations to parameterize.
9. Externalize file system dependencies, and you can add other folders that your app uses.
10. Build a container image by selecting your Azure Container Registry or create a new one
 - a. Review the Dockerfile
 - b. Trigger build process: Select the applications to build images for and click Build.
 - c. Track build status: You can also monitor the progress of the build step by clicking the Build in Progress link under the status column.
11. Deploy the containerized app on AKS
 - a. Select the AKS Cluster
 - b. Specify secret store: If you had opted to parameterize application configurations, specify the secret store used for the application.
 - c. Application deployment configuration: Once you've completed the steps above, you'll need to specify the deployment configuration for the app.
 - d. Deploy the application: Once our deployment configuration has been saved, the tool will generate the Kubernetes deployment YAML for the application.
12. Once the app is deployed, you can click the Deployment status column to track deployed resources.

Containerize the app and migrate to App Service

Follow all the last steps on the last slide “**Containerize the app and migrate to AKS**” until **step 10** and follow the instructions below.

1. Select the Azure App Service plan that the application should use or create a new one
2. If you parameterize application configurations, specify the secret store for the application. You can choose Azure Key Vault or App Service app settings to manage your secrets.
3. Specify the deployment configuration for the application, and provide this information:
 - a. Name, a unique app name for the app, is used to generate the application URL.
 - b. For any parameterized app configurations, provide the values for the current deployment.
 - c. Storage configuration. Review the information for any application folders configured for persistent storage.
4. After you save the deployment configuration for the application, the tool will generate the Kubernetes deployment YAML for the application.
 - a. Review the deployment configuration for the app.
 - b. Select the app that you want to deploy.
 - c. Select Deploy to start deployment for the selected app.
5. After the app is deployed, you can select the Deployment status column to track the deployed resources.
6. If you had any issues, you could verify the logs in “C:\ProgramData\Microsoft Azure Migrate App Containerization\Logs”.