Azure_App_Services

INDEX

https://learn.microsoft. comf en-usf azuref container-apps/compare-options

* Azure Functions https://learn.microsoft.com/en-us/azurefazure-functions/
* Azure Logic apps :; https://learn.microsoft.com/en-us/azure/logic-apps/
x Azure Spring Apps https://learn.microsoft.com/en-us/azure/spring-apps/
x Azure App Service (web hosting) https://learn.microsoft.com/en-us/azure/app-service/
Azure App Service WebJobs

* Azure Container Instances https://learn.microsoft.com/en-us/azure/container-instances/
xAzureContainerApps https://learn.microsoft.com/en-us/azure/container-apps/
* Azure Kubernetes Service https://learn.microsoft.com/en-us/azure/aks/
x Azure Red Hat Openshift https://learn.microsoft.com/en-us/azure/openshift/
* Azure container Registry https://learn.microsoft.com/en-us/azure/container-registry/

Azure Functions

- Azure Functions is a serverless Functions-as-a-Service (FaaS) solution.
- It's optimized for running event-driven applications using the functions programming model.
- The Azure Functions programming model provides productivity benefits for teams looking to trigger the execution of your functions on events and bind to other data sources.
- The Azure Functions programming model is available as a base container image, making it portable to other container based compute platforms allowing teams to reuse code as environment requirements change
- It shares many characteristics with Azure Container Apps around scale and integration with events, but optimized for ephemeral functions deployed as either code or containers.

Azure Spring Apps

- Azure Spring Apps is a fully managed service for Spring developers.
- If you want to run Spring Boot, Spring Cloud or any other Spring applications on Azure, Azure Spring Apps is an ideal option.
- The service manages the infrastructure of Spring applications so developers can focus on their code.
- Azure Spring Apps provides lifecycle management using comprehensive monitoring and diagnostics, configuration management, service discovery, CI/CD integration, blue-green deployments, and more.

Azure App Service

• Azure App Service provides fully managed hosting for web applications including

1. websites and
2. web APIs

• Azure App Service is optimized for web applications
• When building web apps, Azure App Service is an ideal option.
• These web applications may be deployed using code or containers.
• Azure App Service is integrated with other Azure services including Azure Functions or Azure Container Apps.

<u>Azure Container Instances</u>

• Azure Container Instances (ACI) provides a single pod of Hyper-V isolated containers on demand
• It can be thought of as a lower-level "building block" option compared to Container Apps.
• If you need a less "opinionated" building block that doesn't align with the scenarios Azure
• Container Apps is optimizing for; Azure Container Instances is an ideal option.
• Concepts like scale, load balancing, and certificates are not provided with ACI containers
• Users often interact with Azure Container Instances through other services

<u>Azure Container Apps</u>
• Azure Container Apps is a fully managed environment that enables you to run microservices and containerized applications on a serverless platform.
• Azure Container Apps enables you to build serverless microservices based on containers.
• Azure Container Apps provide many application-specific concepts on top of containers, including certificates, revisions, scale, and environments.
• Azure Container Apps doesn't provide direct access to the underlying Kubernetes APIs.
• If you require access to the Kubernetes APIs and control plane, you should use Azure Kubernetes Service
• Howeve4 if you would like to build Kubernetes-style applications and don't require direct access to all the native Kubernetes APIs and cluster management, Container Apps provides a fully managed experience.
• For these reasons, many teams may prefer to start building container microservices with Azure Container Apps.

<u>Azure Kubernetes Service</u>

• Azure Kubernetes Service (AKS) provides a fully managed Kubernetes option in Azure.
• It supports direct access to the Kubernetes API and runs any Kubernetes workload.
• The full cluster resides in your subscription, with the cluster configurations and operations within your control and responsibility.
• * Teams looking for a fully managed version of Kubernetes in Azure, Azure Kubernetes Service is an ideal option.

Azure Red Hat OpenShift extends Kubernetes.
* Running containers in production with Kubernetes requires additional tools and resources.
x This often includes needing to juggle

* image registries,
* storage management,
* networking solutions, and
* logging and monitoring tools - all of which must be versioned and tested together.
* Building container-based applications requires even more integration work with middleware, frameworks, databases, and CI/CD tools.
* Azure Red Hat OpenShift combines all this into a single platform, bringing ease of operations to IT teams while giving application teams what they need to execute.

Azure Red Hat Openshift Service
* Azure Red Hat OpenShift is jointly engineered, operated, and supported by Red Hat and Microsoft to provide an integrated product and support experience for running Kubernetes-powered OpenShift.
x lf your team or organization is using OpenShift, Azure Red Hat OpenShift is an ideal option.
* With Azure Red Hat OpenShift, teams can choose their own
* registry,
* networking,
* storage, and
* CI/CD solutions, or
*oR

x use the built-in solutions for automated source code management, container and application builds, deployments, scaling, health management, and more from OpenShift.

Azure Container Registry
* Azure Container Registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments.
* Use Azure container registries with your existing container development and deployment pipelines.
* Use Azure Container Registry Tasks to build container images in Azure on-demand, or automate builds triggered by source code updates, updates to a container's base image, or timers

Azure_Functions

Azure Functions

• Azure Functions is a serverless Functions-as-a-service (Faas) solution.
• It's optimized for running event-driven applications using the functions programming model.
• The Azure Functions programming model provides producfivity benefits for teams looking to triggerthe execution ofyourfunctions on events and bind to other data sources.
• The Azure Functions programming model is available as a base container image, making it portable to other container based compute platforms allowing teams to reuse code as environment requirements change

- It shares many characteristics with Azure Container Apps around scale and integration with events, but optimized for ephemeral functions deployed as either code or containers.
- * Functions provides serverless compute for Azure.

* You can use Functions to

1. build web APIs,
2. respond to database changes,
3. process IoT streams,
4. manage message queues, and more

* Azure Functions is a cloud service available on-demand that provides allthe continually updated infrastructure and resources needed to run your applications.
* You focus on the code that matters most to you, in the most productive language for you, and Functions handles the rest.

Azure_Logic_Apps

Azure Logic apps :; https ://lea rn. m icrosoft.co m/e n-us/azu re/logic-a pps/

Both Azure Functions and Azure Logic Apps enable serverless workloads.
Both can create complex orchestrations

1. Azure Functions is a serverless compute service, whereas
2. Azure Logic Apps is a serverless workflow integration platform

Both can create complex orchestrations
An orchestratjon is a collection of functions orsteps, that are executed to accomplish a complex task.

x (called actions in Logic Apps),
For Azure Functions, we develop orchestrations by writing code and using the Durable Functions extension.

For Logic Apps, we create orchestrations by using a GUI or edifing configuration files.

Azure_Spring_Apps

Azure Spring Apps

- Azure Spring Apps is a fully managed service for Spring developers.
- If you want to run Spring Boot, Spring Cloud or any other Spring applications on Azure, Azure Spring

Apps is an ideal option.

- The service manages the infrastructure of Spring applications so developers can focus on their code.
-
- Azure Spring Apps provides lifecycle management using comprehensive monitoring and diagnostics, configuration management, service discovery, CI/CD integration, blue-green deployments, and more.

https://learn.microsoft.com/en-us/azure/spring-apps/overview

- Azure Spring Apps makes it easy to deploy Spring Boot applications to Azure without any code changes.
- Azure Spring Apps supports both Java Spring Boot and ASP.NET Core Steeltoe apps.
- Steeltoe support is currently offered as a public preview. With public preview offerings, you can experiment with new features prior to their official release.

Why use Azure Spring Apps?

You get the following benefits when you deploy applications to Azure Spring Apps:

- Efficiently migrate existing Spring apps and manage cloud scaling and costs.
- Modernize apps with Spring Cloud patterns to improve agility and speed of delivery.
- Run Java at cloud scale and drive higher usage without complicated infrastructure.
- Develop and deploy rapidly without containerization dependencies.
- Monitor production workloads efficiently and effortlessly.

Azure_App_Service

Azure App Service

- Azure App Service provides fully managed hosting for web applications including

1. websites and
2. web APTs

- Azure App Service is optimized for web applications
- When building web apps, Azure App Service is an ideal option.
- These web applications may be deployed using code or containers.
- Azure App Service is integrated with other Azure services including Azure Functions or Azure Container Apps.

Azure_Container_Instances

Azure Container Instances

• Azure Container Instances (ACI) provides a single pod of Hyper-V isolated containers on demand
• It can be thought of as a lower-level "building block" option compared to Container Apps.
• If you need a less "opinionated" building block that doesn't align with the scenarios Azure Container Apps is optimizing for; Azure Container Instances is an ideal option.
• Concepts like scale, load balancing, and certificates are not provided with ACIcontainers
• Users often interact with Azure Container Instances through other services

Azure_Container_Apps

https://learn.microsoft.com/en-us/azure/container-apps/overview

Azure Container Apps
• Azure Container Apps is a fully managed environment that enables you to run microservices and containerized applications on a serverless platform.
• Azure Container Apps enables you to build serverless microservices based on containers.
• Azure Container Apps provide many application-specific concepts on top of containers, including certificates, revisions, scale, and environments.
• Azure Container Apps doesn't provide direct access to the underlying Kubernetes APIs.
• If you require access to the Kubernetes APIs and control plane, you should use Azure Kubernetes Service

• However, if you would like to build Kubernetes-style applications and don't require direct access to all the native Kubernetes APIs and cluster management, Container Apps provides a fully managed experience.
• For these reasons, many teams may prefer to start building container microservices with Azure Container Apps.

Common uses of Azure Container Apps include:
• Azure Container Apps enables execufing application code packaged in any container and is unopinionated about runtime or programming model.
• With Container Apps, you enjoy the benefits of running containers while leaving behind the concerns of managing cloud infrastructure and complex container orchestrators

1. Deploying API endpoints
2. Hosting background processing applications
3. Handling event-driven processing
4. Running microservices

Applications built on Azure Container Apps can dynamically scale based on the following characteristics:

1. HTTP traffic
2. Event-driven processing
3. CPU or memory load
4. Any KEDA-supported scaler

https://learn.microsoft.com/en-us/azure/container-apps/media/overview/azure-container-apps-example-scenarios.png

https://learn.microsoft.com/en-us/azure/container-apps/media/overview/azure-container-apps-example-scenarios.png

Azure_Kubernetes_Service

08 Azure Kubernetes Service

https://learn. m icrosoft. com I en-us/ azure /a ks/i ntro-kubernetes

Azure Kubernetes Service

• Azure Kubernetes Service (AKS) provides a fully managed Kubernetes option in Azure.
• It supports direct access to the Kubernetes API and runs any Kubernetes workload.
• The full cluster resides in your subscription, with the cluster configurations and operations within your control and responsibility.
• Teams looking for a fully managed version of Kubernetes in Azure, Azure Kubernetes Service is an ideal option.

• Azure Kubernetes Service (AKS) simplifies deploying a managed Kubernetes cluster in Azure by offloading the operational overhead to Azure.

• As a hosted Kubernetes service, Azure handles critical tasks, like health monitoring and maintenance.

• When you create an AKS cluster, a control plane is automatically created and configured.
• This control plane is provided at no cost as a managed Azure resource abstracted from the user.
• You only pay for and manage the nodes attached to the AKS cluster.

• AKS also supports Windows Server containers.

* This service supports Azure Lighthouse, which lets service providers sign in to their own tenant to manage subscriptions and resource groups that customers have delegated.
Azure Lighthouse:: https://learn.microsoft.com/en-us/azure/lighthouse/

AKS with Azure AD

- * You can configure an AKS cluster to integrate with Azure AD.
- * With Azure AD Integration, you can set up Kubernetes access based on existing identity and group membership.
- * Your existing Azure AD users and groups can be provided with an integrated sign-on experience and access to AKS resources.

Access, security, and monitoring

* For improved security and management, you can integrate with Azure AD to:
1. Use Kubernetes role-based access control (Kubernetes RBAC).
2. Monitor the health of your cluster and resources.

Kubernetes RBAC

1. To limit access to cluster resources, AKS supports Kubernetes RBAC.
2. Kubernetes RBAC controls access and permissions to Kubernetes resources and namespaces.

Terminologies

**AKS Terminologies **

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.
* AKS supports Kubernetes clusters that run multiple node pools to support mixed operating systems and Windows
Server containers.

Cluster node and pod scaling

* As demand for resources change, the number of cluster nodes or pods that run your services automatically scales
up or down.

* You can adjust both the horizontal pod autoscaler or the cluster autoscaler to adjust to demands and only run
necessary resources.

Storage volume support

* To support application workloads, you can mount static or dynamic storage volumes for persistent data.

* Depending on the number ofconnected pods expected to share the storage volumes, you can use storage backed
by:

* Azure Disks for single pod access
* Azure Files for multiple, concurrent pod access

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 and can directly
communicate with other pods in the cluster and other nodes in the virtual network.

* Pods can also connect to other services in a peered virtual network and on-premises networks over ExpressRoute
or site-to-site (S2S) VPN connecfions.

Ingress with HTTP application routing
* The HTTP application routing add-on helps you easily access applications deployed to your AKS cluster.
* When enabled, the HTTP application routing solution configures an ingress controller in your AKS cluster,

* As applications are deployed, publicly accessible DNS names are auto-configured.
* The HTTP application routing sets up a DNS zone and integrates it with the AKS cluster.
* You can then deploy Kubernetes ingress resources as normal.

Development tooling i ntegration
* Azure provides several tools that help streamline Kubernetes.
* Kubernetes has a rich ecosystem of development and management tools that work seamlessly with AKS.
* These tools include Helm and the Kubernetes extension for Visual Studio Code.

Helm

Docker image support and private container registry
* AKS supports the Docker image format.
* For private storage of your Docker images, you can integrate AKS with Azure Container Registry (ACR).

**AKS Terminologies

Loggin Monitoring

02 Loggin Monitoring
Created Wednesday 21, June 2023
Integrated logging and monitoring
* Container Insights is a feature in Azure Monitor that monitors the health and performance of managed
Kubernetes clusters hosted on AKS

* Container Insights has native integration with AKS, like collecting critical metrics and logs, alerting on identified issues, and providing visualization with workbooks or integration with Grafana

* Container Insights captures platform metrics and resource logs from containers, nodes, and controllers
within your AKS clusters and deployed applications that are available in Kubernetes through the Metrics API

* Contalner Insights provides interactive views and workbooks that analyze collected data for a variety of
monitoring scenarios.

* It can also collect Prometheus metrics and send them to Azure Monitor managed service for Prometheus, and all together deliver end-to-end observability

Azure Lighthouse : : https://learn.microsoft .com/en-us/azure/lighthouse/

Azure Container Insights : https://learn.microsoft.com len-us/azure/azure-monitor/containers/ container-insights-
overview

Prometheus :: https://learn.microsoft.com/en-us/azure/azure-monitor/containers/container-insights-prometheus?ta bs=cl uster-wide

Grafana ::https://learn.microsoft.com/en-us/azure/azure-monitor/visualize/qrafana-plugin

Azure_OpenShift

09 Azure OpenShift

https://learn. microsoft.com/en-us/azure/openshift/

Azure Red Hat OpenShift extends Kubernetes.

* Running containers in production with Kubernetes requires additional tools and resources.
* This often includes needing to juggle
* image registries,
* storage management,
* networking solutions, and
* logging and monitoring tools - all of which must be versioned and tested together.

- Building container-based applications requires even more integration work with middleware, frameworks, databases, and CI/CD tools.
- Azure Red Hat OpenShift combines all this into a single platform, bringing ease of operations to ITteams while giving application teams what they need to execute.

Azure Red Hat Openshift Service

- Azure Red Hat OpenShift is jointly engineered, operated, and supported by Red Hat and Microsoft to provide an integrated product and support experience for running Kubernetes-powered OpenShift.
-
- If your team or organization is using OpenShift, Azure Red Hat OpenShift is an ideal option.
- With Azure Red Hat OpenShift, teams can choose thelr own
  1. registry,
  2. networking,
  3. storage, and
  4. CI/CD solutions, or

  *oR

  * use the built-in solutions for automated source code management, container and application builds, deployments, scaling, health management, and more from OpenShift.

* Master, infrastructure, and application nodes are patched, updated, and monitored on your behalf by Red Hat and Microsoft

* Azure Red Hat OpenShift provides an integrated sign-on experience through Azure Active Directory.
* You can choose your own registry, networking, storage, and CI/CD solutions, or use the built-in solutions for automated source code management, container and application builds, deployments, scaling, health management, and more.

Access, security, and monitoring

* For improved security and management, Azure Red Hat OpenShift lets you integrate with Azure Active Directory (Azure AD) and use Kubernetes role-based access control (Kubernetes RBAC).
* You can also monitor the health of your cluster and resources.

Cluster and node

* Azure Red Hat OpenShift nodes run on Azure virtual machines.
* You can connect storage to nodes and pods and upgrade cluster components.

Service Level Agreement

Azure_Container_Registry

20 Azure Container Registry
https ://learn. m icrosoft . com / en-us / azur e /conta i ner-registry/

Azure Container Registry
• Azure Container Registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments.
• Use Azure container registries with your existing container development and deployment pipelines.
• Use Azure Container RegistryTasks to build container images in Azure on-demand, or automate builds triggered by source code updates, updates to a container's base image, or timers

----

TEMP

12_Deploy and manage Azure compute resources

01_02_03_Configure VMs for high availability and scalability

Azure Virtual Machines

There are 3major delivery models when it comes to Cloud services. They are:
1. **SAAS – Software as a Service**

2. **PAAS – Platform as a Service**

3. **IAAS - Infrastructure as a Service**

• Azure Virtual Machines are part of the **IAAS** offering from Azure.

• As customers, we are responsible for managing the virtual machine, and just the hardware will be provided to us by the cloud provider. We can *start, stop, and delete* the virtual machine.

• If we find that the capacity is insufficient or too high, we can change to a different machine type. We can install any software we like.

• Also, please note that this is the most expensive e of the three offerings.

• We can create **Windows or Linux VMs**, and there are multiple locations throughout the world where resources can run from.

• When we create a VM, we need to attach a virtual hard disk, and the location that we specify is where the hard disks are stored.

This is	**A**	**Table**
a	a	a
a	a	a
asdfasfd	asdf	asdf
asdf	asdf	asdf

Here is the SLA table:

Sl No.	**VM**	**Disk**	**SL**
1	2or mor e VMs across 2or mor e AZs		9.9
2	2or mor e VMs in a same Availability set		9.9
3	Single VM	Premium or Ultra disk for all disks	9.9
4	Single VM	Standard SSD	9.5
5	Single VM	Standard HDD	95

Please see below details for VM types:

Sl No	Type	Sizes	Short Description	Best for
1	GP (General Purpose)	B, Dsv, Dasv, Dav, Av2, DC , Dsv	Balanced CPU to memory	Testing/ Dev, small DB, low traffic servers
2	Compute Optimized	F, Fs, Fsv2	High CPU to memory	Medium traffic servers, batch processes, ap servers
3	Memory-Optimized	Esv, Ev, Eav, Mv2, M, DSv2, Dv2	High memory to CPU ratio	RDBMS servers
4	Storage Optimized	Lsv2	High disk throughputand IO	Big data/ DB warehousing/ Large DB
5	GPU	NC, NCv2, ND , NV	Specialized VMs for heavy graphics	Model training with deep learning
6	HPC(High-perf ormance Compute)	HB, HBv2, HC , H	Fastest and most op werful CPU	Real-time processing

FAQs

1. **How do I resize a VM?**

◇ You can first run the list-vm-resize-options and see available sizes. If you find the size, you can run the resize command

◇ az vm resize –resource-group WLRG –name WLVM1 –size Standard_DS3_2

◇ Else you need to deallocate the VM, which will allow you to use any size. You need to deallocate, resize and start a VM.

```
az vm deallocate –resource-group WLRG –name WLVM1
az vm resize –resource-group WLRG –name WLVM1 –size Standard_DS3_2
az vm start –resource-group WLRG –name WLVM1
```

1. **What are Azure Dedicated hosts?**

◇ We usually shared the physical hardware with other tenants. If we want exclusively to use the physical server, then we can choose dedicated hosts.

1. **What are Azure Spot instances?**

◇ This feature allows us to take advantage of the unused CPU at a significantly lower cost at almost 0%9 savings.

◇ If there are workloads that can tolerate disruption and can be restarted, then we can choose this option.

◇ If there is another bidder who bids more than our price, we will be vacated on 30 seconds' notice. So we need to be prepared with proper scripts to save the data or any other process from exiting gracefully.

4 **How can we save costs on VMs other than Spotinstances?**
◇ There are two other ways we can save on costs.–

■ **Reserved Instances** – We can commit to -1 year or 3- year and choose to pay upfront or monthly to buy RIs. We have the flexibility to change size if needed.

■ **Azure Hybrid Benefit** – If you have a license already, you can use the license on Azure and get this benefit.

1. **What are Azure Images?**

◇ If there is a custom image that we want every VM to have when created, we can choose to create a standard VM and syspr ep and then create an image. We can then use this image to create VMs.

1. **How can we make VMs highly available?**

◇ We had discussed in the excel above with SLAs. We can use multiplemachines either in availability or in more than 1 availability zone.

◇ In addition to this, we can use Azure VMSS (Virtual machine scale sets).

◇ VMSS is automatically created from a central configuration using a standard template.

◇ More VMs will be added during peak and will be brought down when the demand goes down based on our auto-scaling options.

1. **How can we back up VMs?**

We have 3 options:
1. **Azure Backup** – We can create recovery vaults and configure Azure Backupt o backup ourVMs
2. **ASR (Azure Site Recovery)** – Here, our VMs are replicated to another region, and our entire production region fails; we can failover to the backupareas with the click of a button
3. **Managed Snapshots** – If we have managed disks, we can take a snapshot of our disks, a read-only copy. We leveraged this feature for quickback ups in dev and test environments.

0.1 **How can we monitor VMs?**
◇ Under Monitoring tabs, we have metrics to see various parameters. We can also set alerts. We can also Log analytics by enabling the Logs option in Monitoring. We need to create a log analytics workspace.

04_Create and configure Web Apps

01_Azure App Service

Azure AppService

• Azure AppService allows us to run applications on the cloud. Here are some features:

◇ HTTP based Service for hosting web applications, REST APIs, and mobile backends.

◇ Supports NET, .NET Core, Java, Ruby, Node.js, PHP, Python

◇ Run and Scale on Windows/Linux

◇ *App Services run under a pa service plan.*

◇ *App service plan is the logical abstraction that represents one or more VMs that runs the pa service. It consists of compute resources like CPU, memory and disk space.*

◇ *We pay for pa service plans and not the pa service.*

◇ *Also, we can have more than one pa service running inside an pa service plan. The number of pa services that can run inside an pa service plan depends on the pa service plan. Also, the amount of resources like CPU, RAM and disk space depends on the paservice plan.*

Plan	**Comput e typeDomain**	**Custom**	**Scaling**	**Workload**		**SpaceBackup / Restore**	**No of Ap s (max)**
Free	Shared	No	No		Nil	No	010
Shared	Shared	Yes	Yes	Dev		1GBNo	01
Basic	Dedicated	Yes	Yes	Dev/Test	10GBNo		Unlimited
Premium	Dedicated	Yes	Yes	Prod	0G25B	Yes	Unlimited
Isolated	Isolated	Yes	Yes	Prod		1TBYes	Unlimited

Let's look at some features of Apservices:

Deployment Slots	**This concept is used for zero downtime deployments. There will be a production slot and a Staging slot. New version of the Production deployment will be done in the Staging slot. Either all at once deployment or in stages(canary) will be done.**
Deployment Center	This allows for Continuous integration / Continuous deployment (CI / CD)
Custom Domains	By default, the website will be xxxx.azurewebsites.net. We can buy a domain in your company name and use that name.

Deployment Slots	This concept is used for zero downtime deployments. There will be a production slot and a Staging slot. New version of the Production deployment will be done in the Staging slot. Either all at once deployment or in stages(canary) will be done.
SSL Settings	You can certificates and ensure encrypted data transmission between client and Server
Scale up (ApService Plan)	You can increase the size of your VM if you need more resources

Deployment Slots	**This concept is used for zero downtime deployments. There will be a production slot and a Staging slot. New version of the Production deployment will be done in the Staging slot. Either all at once deployment or in stages(canary) will be done.**
Scale out (AppService Plan)	You can also increase the number of instances. You can either do this manually with a slider or set uprules/ schedule to scale automatically on schedule or CPU usage (like >70%)

FAQs

1 **How does app service plan work?**

◇ App service plan is supported by Service Fabric.

◇ Service fabric replacesinstances if an existing one fails. Also, it adds instances if there is a requirement.

2. What are the typesof Ap Services?

There are 4typesof services as follows:

Sl no	Type	Purpose
1	WebAp (previously Azure Websites)	Hosting websites and web aplications
2	API Ap	Used for hosting the RESTful APIs
3	Logic Ap	Used for business process automation, system integration and sharing data across clouds
4	Mobile Ap (previously delivered b y Azure Mobile services)	Used for hosting mobile app back ends

App Service

◇ HTTP based Service for hosting web aplications, REST APIs, and mobile backends.

◇ Suports NET , .NET Core, Java, Ruby, Node.js, PHP, Python

◇ Run and Scale on Windows/Linux

Features

◇ **PAAS** – Patches/OS Maintenance done by Azure ●

◇ Suport for Containerization and Docker

◇ Serverless

◇ **Deployments Slots** – Swapaplication content in Prod and avoid downtimes

◇ Grouped under ApService plans with following tiers

●

Plan	Compute type	Custom Domain	Scaling	Workload	Space	Backup/ Restore	Others
Free	Shared	No	N o		N il	N o	
Shared	Shared	Yes	Yes	Dev	G1 B	N o	
Basic	Dedicated	Yes	Yes	Dev/Test	0G1 B	N o	
Premium	Dedicated	Yes	Yes	Prod	0G25 B	Yes	
Isolated	Isolated	Yes	Yes	Prod	1TBYes		Private Endpoints

App Service types

◇ **Webapps** – Websites/Online Apps

◇ **Webapps for Containers** – Containerization

◇ **API apps** – backend data

Can add – V net Integration /Hybrid Connections/Security , but these are not asked in the exams.

Tips

◇ When you move an Apservice from one RG to another, the App Service plan doesn't change.

◇ Destination RG cannot contain ApSer vice resources like Web apor ApService plan.

◇ **.Net** Core aplication can be deployed on Windows or Linux OS

◇ **ASP .Net** app CANNOT be deployed on Linux OS. Only Windows OS

◇ Multiple Web Apps can be hosted on a single ApService plan.

◇ Web Ap and Ap Service plansmust exist in the same region

02_Application_Service_Environment

Application Service Environments

• There are 3 components for hosting *web pa s/ Docker containers/ Mobile pa s* and functions. There are appservice plans which host the spservices.

• When we host the regular app services, the apps are directly exposed to the internet, and the resources are shared.

• Some organizations prefer to host the services in the internal network, and security features like firewalls and security groups could be aplied t o protect the apps.

• For such scenarios, there is a feature called the **Azure App Service Environment**, which provides a fully isolated and dedicated environment for securely running App Service apps at a high scale.

• **Ap Service environments (ASEs)** provide very high scaling with isolation and secure network access with high memory utilization.

• We can create multiple ASEs within a single Azure region or across multiple Azure regions, making it ideal for horizontally scaling stateless application tiers when we have high **requests per second** (**RPS**) workloads.

◇ There are three types of workloads available when choosing the workload tier. They are *Dev/test, Production, and Isolated.*

• Of these, the isolated offering provides the ASE environments which host applications within the client's VNets. As stated, we have fine-grained control over inbound and outbound application network traffic.

• While the other category of app services has a fixed suffix of azurewebsties.net, we can create our own domain name.

• Also, ASEs come with op werful computers, which is twice as op werful as the regular app service plans. They also come with **1TB** Storage as compared to **50GB** of space for the regular ones.

• We can host up to 01 instances which are sufficient to host a miniature web service hub. We can expect the service to cost us about **250-30$** per month, which is very cheapf or the services being provided.

Steps to creating App Service Environment

◇ In the first screen, we select if the service is public-facing or internal

◇ Then we select whether we are hosting Windows-based or Linux-based OS.

◇ On the second screen, we select the Vnet where we want to host the service. *(Since*

■ *services are being created in our private infrastructure, it takes much longer time to*

■ *create)*

◇ ‚óŹ Then we can DNS resolution. We can create our own private zone and use that name. This is not possible when choosing the other ap service plans

Steps to creating Web Apps under ASE

◇ Please note that the process is similar except that we dropdown the region and select the ASE which we just created.

◇ Also, the below screen shows various features under ASE and pricing under each of the pricing tier s I1and I2 and I3. 13.

Note: *The Private link vnetLink (wlase1.ap[serviceenvironment.net/vnetLink)](serviceenvironment.net/vnetLink) is also created below. You can go to the Resource group and click on "Show hidden types" to see this resource.*
Note: *Please see the Ap Service plan as I1:1 in the screenshot below to identify the isolated service plan.*

05_Create and configure containers

01_Azure Container Instance_ACI

Azure Container Instance (ACI)

• Containerization is the buzzword today. Instead of spinning Physical servers and installing all the dependencies, and installing the application, we can create a container containing all the required dependencies.

• We then package and create an image and deploy it into a container.

• **Docker** is one of the platforms where we can run these containers in the Open-source world. Azure has two solutions. One of those is the ACI.

• ACI is a great solution in scenarios where we need to run isolated containers.

◇ Examples are simple applications, task automation, and build jobs.

• The drawback of ACI is that it cannot be used for full orchestration like multiple containers, auto-scaling, and coordinated application upgrades. Please consider AKS for such scenarios, which is the other offering from Azure.

• In simple terms, for Production, use **AKS (Azure Kubernetes Service)**, and for simple and isolated containers, use ACI.

• One of the other best use cases for ACI is where we have production issues, and we need to troubleshoot AKS, ACI comes to our rescue where we deploy the trouble-making container in ACI and try to debug.

Advantages of ACI

◇ *Fast Startup Startup times*

◇ *Container access*

◇ *Custom Sizes*

◇ *Persistent Storage* – We do this by mounting Azure file shares.

◇ *Virtual Network deployment* – When deployed in a Vnet, ACI can securely communicate with other resources in the Vnet.

What are container groups?

◇ Like AKS for orchestration, we can use container groups to combine and manage containers.

◇ They get scheduled on the same host machine.

◇ The concept is like pods in Kubernetes. The use case for this is in scenarios where we want to divide a single functional task into a smaller number of container images.

■ An example is front-end container and a back-end container.

◇ The front end might serve as a web application, with the back end running a service to retrieve data.

FAQs

1. **What are probes in ACI?**

1. **Liveness probe**

2. **Readiness probe**

o You can configure the liveness probe. We check the liveness probe to see if the container is healthy. If the container is not healthy, we need to restart. There are common scenarios when containers run for a long time.
o You can configure the readiness probe. Here we might have a scenario where the container (maybe DB for the backend) is just coming up. We run the readiness probe and send requests to the container only if the probe succeeds.
1. **How can we monitor ACI?**

◇ We use Azure Monitor. Here are the available metrics currently.

◇ CPU Usage measured in millicuries (One millicore is 1/10th of a CPU core)

◇ o Memory Usage in bytes

◇ o Network bytes received persec ond.

◇ o Network bytes transmitted persec ond

02_Azure Container Registry

Azure Container Registry

What is a Container Registry?

• A Container Registry is a central repository to store and distribute container images. A container image includes all the data needed to start a container - **for example,** the operating system, libraries, runtime environments, and the application itself.

• We first build an image, and then we push the image to the repository. When needed, we pull the image into the target environment. With versioning as a feature, we have multiple versions of the container, and different versions like the stable version would be used for Production.

• Versions being tested would be in non-production regions. In the example below, v2 is a stable version, and the developer makes changes and creates v3. Once v3 is tested, it would be then pulled into Production.

Providers

Few providers provide the container registry services, and they are:

◇ **Docker Hub**

◇ **Azure ACR (Azure Container Registry)**

◇ **AWS ECR (Elastic Container Registry)**

◇ **GitHub Container Registry**

◇ **Google Container Registry**

	Amazon ECR	**Docker Hub**	**GitHub Container Registry**	**Azure Container Registry (ACR)**
Public Repository	No	YES	YES	No
Private Repository	Yes	YES	YES	Yes

Pricing (Public Repository)		**$0**	**$0**	**$0**	
Pricing (Private Repository)		$	$	$	
Storage: $0.10 per GB, Data Transfer: $0.09 per GB	>= $7 peuser/ month	Storage: $0.25 perperGB, Outgoing Data Transfer: $0.50 perGB	Storage: $0.09 perGB		
Authentic-ation	AWS IAM	Password or Access Token	Personal Access Token (PAT)	PAT	

Pricing (Public Repository)		**$0**	**$0**	**$0**	
MFA for Image Push/Pull	Yes	NO	NO	NO	
SLA Availability	9.9%	N/A	N/A	9.9%	
General Available	YES	YES	Beta	YES	
Immutable Images	YES	NO	NO	YES	
Image Scanning	YES	YES (paid plans only)	NO	YES	
Regions	Choose between one of 52regions worldwide	Not Known	Not Known	3regions	
Rate Limits	Pull: 1,0 persecond, Push: 01 per second	Pull: /0102 (FreePlan), unlimited (Paid Plan)	n/a	Pull: 1,0per second, Push: 01 per second	

ACR Service Tiers

ACR is available in 3ser vice tiers, also called SKUs.

1. **Basic** – Cost Optimized for developers.

2. **Standard** – All features of Basic plus increased storage and image throughput. For Production

3. **Premium** – highest amount of storage and concurrent operations. It also includes geo-replication, content trust, and private link.

ACR Roles

Role/ Permission	**Create/ Delete ACR**	**Push**	**Pull**	**Signature Signing**
Owner	X	X	X	
Contributor	X	X	X	
Reader			X	

Role/ Permission	Create/ Delete ACR	Push	Pull	Signature Signing
AcrPush		X	X	
AcrPull			X	
AcrImage-Signer				X

FAQs

◇ **Can we change Service tiers –**

■ **YES**

◇ **What is geo-replication?**

■ With this feature, a replica of the ACR will be created for DR purposes and local use.

◇ **How can we secure the images in ACR?**

■ There is a concept called CONTENT TRUST. With this, images will be signed with certificates.

■ To enable this feature, enable registry content trust.

■ It is available under **Policies ->Content Trust ->Enabled and then save.**

03_Azure Kubernetes Service

Azure Kubernetes Service (AKS)

What is Containerization?

• In the traditional computing system, we had to install an Operating system and install all dependencies for an application to work. Only a single OS could be installed.

• Then came Virtualization where we could install multiple OS by introducing another layer between the hardware and the OS and this was called Virtualization. So only physical machines appeared as multiple systems.

• Then came a lightweight alternative to virtualization, which was called Containerization. This removed the drawback of having a full machine, and this had only the necessary components.

• Containers will encapsulate an application with its operating system. This would contain all the dependencies that were needed for an application to run. So we take the container and run it on any operating system, and it will run.

• Some of the containerization options are Docker, which is the most popular and

◇ sometimes equated to containers. But there are others like **LXC/LXD, ContainerD, Rocket**.

Orchestration

◇ Orchestration is the system that is used to manage the deployment of containers. We use Orchestrators as tools to achieve this. Some of the performed activities are automating the maintenance of those applications, replacng failed containers automatically, and managing the rollout of updates and reconfigurations of those containers during their lifecycle.

◇ The popular tools are

■ *Docker Swarm by Docker,*

■ *Nomad by Hashicorp,*

■ *Flocker, and*

■ *Kubernetes by Google.*

◇ Kubernetes, also stylized as K8s, is an open-source container orchestration system. It is used for automating comput er aplic ation deployment, scaling, and management.

◇ It was originally designed by Google and influenced by Google's Borg System and is now maintained by the Cloud Native Computing Foundation. It is a cluster management software for Docker containers mainly but suports others also.

AKS

◇ Kubernetes has become very popular , and many cloud service providers offer a Kubernetes-based platform or infrastructure as a PaS or IaS offering.

◇ Google has GKE (Google Kuberne tes Engine),

◇ AWS has EKS (Elastic Kuberne tes Service), and

◇ Azure has AKS (Azure Kuberne tes Service)

Components of Kubernetes

1 . The Cluster

o The Cluster contains 2c omponen ts

■ Control Plane – this consists of kube-apiser ver, etcd, kube-scheduler and kube-controller-manager

■ Nodes that run the applications

2 . Persistent Volumes

o Since the nodes are added and removed on-demand and the storage associated with it is temporary, we need to create storage outside of the cluster. Hence we create persistent volumes.

3 . Node

◇ We create Node pools in Kubernetes (as shown below). Here we choose a VM size, and that will be the unit size of the nodes within the pool.

◇ We can add node pools as needed. The first node poolcreated is the **system node** pool which hosts critical system podslk e coreDNS and tunnel front.

◇ We then add user node pools for aplic ation suportand create different pools based on theaplication requirements.

◇ Pods will be created within the nodes, and the max pod setting is configured at the node pool level.

4 . Containers

◇ We store our code that is going to be run inside containers. There are readily available pr e-built containers stored in container repositories or we can create our own containers.

◇ One or more programs can be run from the containers

5. Pods

◇ Nodes create Pods, and kubernetes use Pods to run instances.

◇ Usually, only one container is run within a pod, but multiple containers could run in a pod if there was a requirement from the application.

◇ We scale based on pods. When we can scale, we simply use pod replicas.

◇ A new pod will be spun up in another node, and we now have an additional pod.

◇ Sameway, we can remove the pods to scale down.

Deployments

◇ We don't launch podsdir ectly. Instead, we create deployments.

◇ A deployment will state how many replicas should run and the system manages that.

7 . Ingress

◇ By default, Kubernetes provides isolation between pods and the outside world. If you want to communicate with the service running in the pods, you need to open the communication. This is called Ingress.

◇ You can achieve this communication in several ways. The most common ways are Ingress controller or a load balancer. Please see the sample service.yaml file which creates an external load balancer. We get the IP of this service and connect.