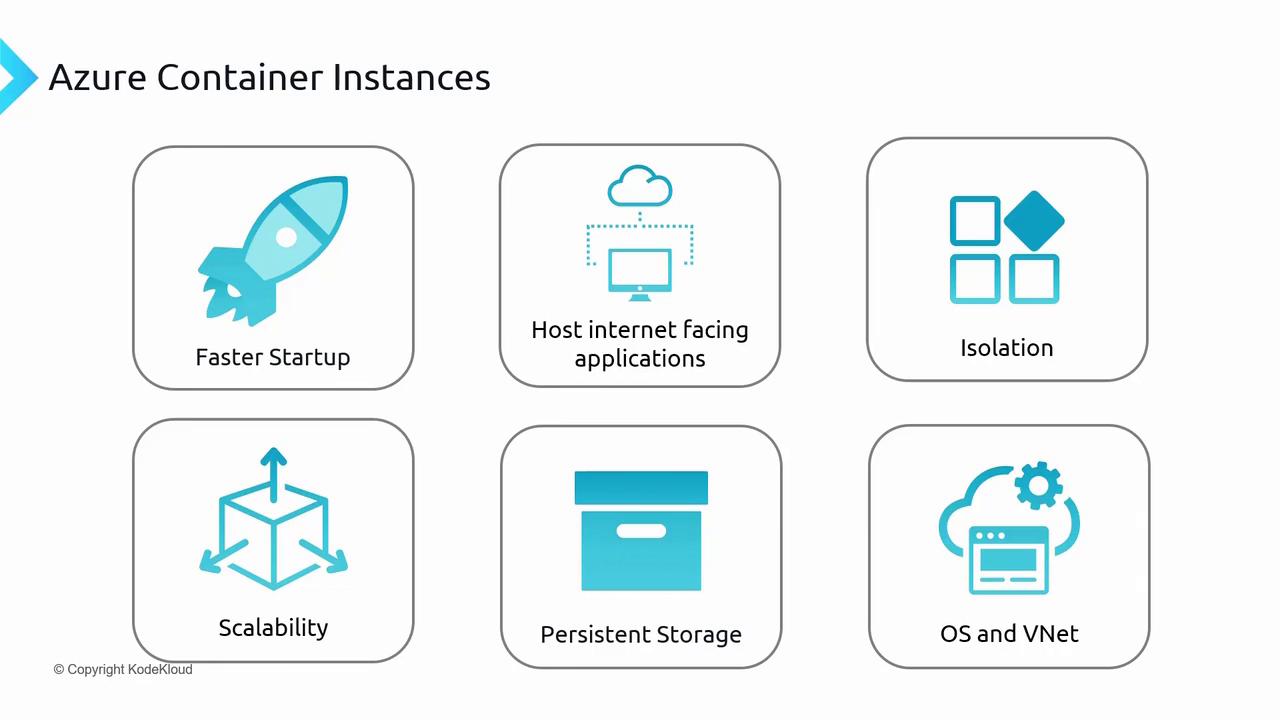
**Azure Container Instances**

Azure Container Instances (ACI) offer a simplified and efficient way to deploy containerized applications directly on Azure. This guide covers the ACI architecture, key benefits, security best practices, and step-by-step instructions to deploy container instances via the Azure Portal.



**Key features of Azure Container Instances**

Some of the main features of the Azure service are the following:

* **Public IP connectivity.** A developer can expose containers to the internet with a [fully qualified domain name](https://www.techtarget.com/whatis/definition/fully-qualified-domain-name-FQDN) and an [IP address](https://www.techtarget.com/whatis/definition/IP-address).
* **Customization.**A developer can specify the number of [central processing unit (CPU) cores](https://www.techtarget.com/searchitoperations/feature/How-to-choose-the-best-CPU-for-virtualization) and memory required for a container instance.
* **Persistent storage.** Container instances are stateless by default, but an organization can choose to mount an Azure file share to a container to enable [persistent storage](https://www.techtarget.com/searchstorage/definition/Persistent-storage).
* **Container groups.** A developer can schedule multiple containers to deploy as a group that shares the same host machine, storage, network and other resources. This feature is beneficial when a developer wants to split one functional task among several container images. For example, a container group could include one application container and then one logging or monitoring container.
* **Infrastructure as a service (IaaS).** Given the structure of ACI and the fact that it can contain all resources needed for one or more applications and operating environments, it can be considered an [IaaS](https://www.techtarget.com/searchcloudcomputing/definition/Infrastructure-as-a-Service-IaaS).

Azure Container Instances (ACI) is a fully managed service for deploying and running containerized applications in Azure. It allows you to specify the exact number of CPU cores and amount of memory that your container needs, and it automatically allocates the resources to your container when it is deployed. ACI is designed to be easy to use and requires no upfront investment in infrastructure. It is well-suited for development and testing scenarios, as well as for deploying small, bursty workloads that start and stop quickly. ACI integrates with other Azure services such as Azure Functions and Azure Monitor, and it supports a wide range of container orchestration platforms including Docker, Kubernetes, and Azure Batch.

**Key Features of Azure Container Services**

ACI is different from other Azure Container Services in a few key ways:

* **Scale and Resource Allocation:**ACI allows you to specify the exact number of CPU cores and amount of memory that your container needs, and it automatically allocates the resources to your container when it is deployed. This makes it easy to scale up or down based on the needs of your workload. Other Azure container services, such as AKS and ACR, require you to specify the number of nodes (virtual machines) in your cluster and allocate resources to containers based on the capacity of those nodes.
* **Deployment Model:** ACI is designed to be used for short-lived, ephemeral workloads that start and stop quickly. It is not intended for long-running, always-on workloads. In contrast, AKS and ACR are more geared towards deploying and managing long-running containerized applications in production.
* **Cost Model:**ACI is generally more expensive on a per-hour basis than AKS or ACR, but it requires no upfront infrastructure investment and is easier to use. This makes it well-suited for development and testing scenarios, as well as for deploying small, bursty workloads.

**Ways to Create and Manage ACI Containers**

To Create and Manage Azure Container Instances (ACI) Containers, we can use these ways

1. **Azure Portal:** You can use the Azure portal to create and manage ACI containers through a graphical user interface. To create a new ACI container, select the "Create a resource" button in the portal, and then search for "Container Instance". Follow the prompts to specify the details of your container, such as the image to use, the number of CPU cores and amount of memory, and any environment variables or secrets that your container requires.
2. **Azure CLI:** You can use the Azure command-line interface (CLI) to create and manage ACI containers from the command line. The Azure CLI is available on Windows, Linux, and macOS, and it can be used to perform a wide range of Azure tasks.
3. **Azure Resource Manager Templates:**You can use Azure Resource Manager templates to create and manage ACI containers as part of a larger Azure deployment. Resource Manager templates are JSON files that define the resources and settings for your Azure environment.

Azure Container Instances (ACI) is a great way to run container workloads and positions itself between Azure Functions (FaaS) & Azure Kubernetes Service (Cluster PaaS). However, it does not provide any autoscaling out-of-the-box which can be a show-stopper.

In this blog post, I’ll walk you through the process of building our autoscaler for ACI by using Azure Monitor & Azure Serverless that allows you to scale in/out and publish events to Azure Event Grid for every scaling action to create awareness.