



Container Groups

examlabpractice.com



<https://t.me/learningnets>



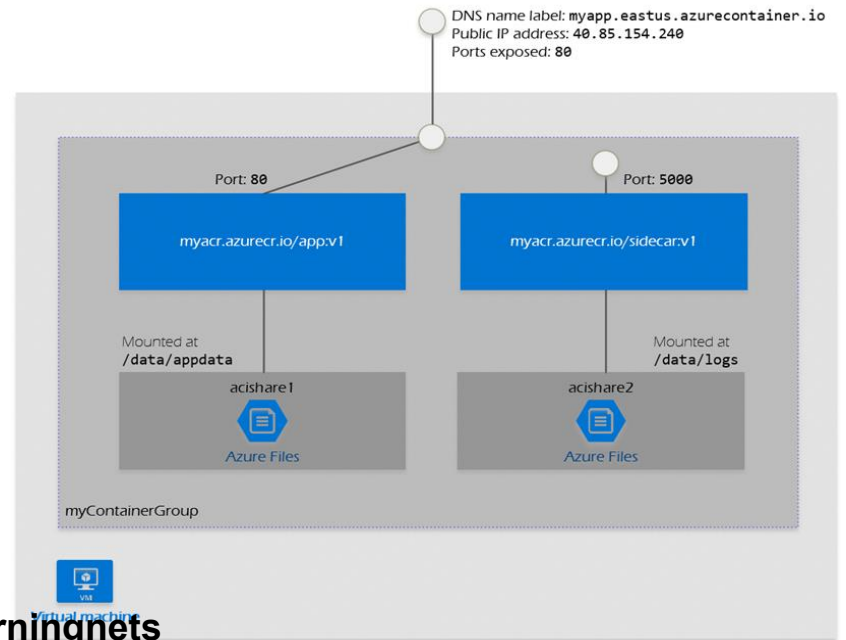
What is a container group?

A container group is a collection of containers that get scheduled on the same host machine. The containers in a container group share a lifecycle, resources, local network, and storage volumes.



This example container group:

- Is scheduled on a single host machine.
- Is assigned a DNS name label.
- Exposes a single public IP address, with one exposed port.
- Consists of two containers. One container listens on port 80, while the other listens on port 5000.
- Includes two Azure file shares as volume mounts, and each container mounts one of the shares locally.



<https://t.me/learningnets>

Deployment

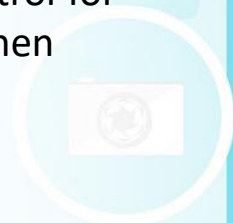
There are two common ways to deploy a multi-container group:

- Use a Resource Manager template
- YAML file.

A Resource Manager template is recommended when you need to deploy additional Azure service resources (for example, an Azure Files share) when you deploy the container instances. Due to the YAML format's more concise nature, a YAML file is recommended when your deployment includes only container instances.

To preserve a container group's configuration, you can export the configuration to a YAML file by using the Azure CLI command **az container export**. Export allows you to store your container group configurations in version control for "configuration as code." Or, use the exported file as a starting point when developing a new configuration in YAML.

<https://t.me/learningnets>



Resource Usage by Container Instances

Each container instance in a group is allocated the resources specified in its resource request. However, the maximum resources used by a container instance in a group could be different if you configure its optional resource limit property. The resource limit of a container instance must be greater than or equal to the mandatory resource request property.

- If you don't specify a resource limit, the container instance's maximum resource usage is the same as its resource request.
- If you specify a limit for a container instance, the instance's maximum usage could be greater than the request, up to the limit you set. Correspondingly, resource usage by other container instances in the group could decrease. The maximum resource limit you can set for a container instance is the total resources allocated to the group.



Networking

Container groups can share an external-facing IP address, one or more ports on that IP address, and a DNS label with a fully qualified domain name (FQDN). To enable external clients to reach a container within the group, you must expose the port on the IP address and from the container. A container group's IP address and FQDN are released when the container group is deleted.

Within a container group, container instances can reach each other via localhost on any port, even if those ports aren't exposed externally on the group's IP address or from the container.

Optionally deploy container groups into an Azure virtual network to allow containers to communicate securely with other resources in the virtual network.

<https://t.me/learningnets>



Common Usage Scenarios

Multi-container groups are useful in cases where you want to divide a single functional task into a small number of container images. These images can then be delivered by different teams and have separate resource requirements.

Example usage could include:

- A container serving a web application and a container pulling the latest content from source control.
- An application container and a logging container. The logging container collects the logs and metrics output by the main application and writes them to long-term storage.
- An application container and a monitoring container. The monitoring container periodically makes a request to the application to ensure that it's running and responding correctly, and raises an alert if it's not.
- A front-end container and a back-end container. The front end might serve a web application, with the back end running a service to retrieve data.

