

Introduction to private Docker container registries in Azure

• 3 minutes to read •  +17

Azure Container Registry is a managed, private Docker registry service based on the open-source Docker Registry 2.0. Create and maintain Azure container registries to store and manage your private Docker container images and related artifacts.

Use Azure container registries with your existing container development and deployment pipelines, or use Azure Container Registry Tasks to build container images in Azure. Build on demand, or fully automate builds with triggers such as source code commits and base image updates.

For more about Docker and registry concepts, see the [Docker overview](#) and [About registries, repositories, and images](#).

Use cases

Pull images from an Azure container registry to various deployment targets:

- **Scalable orchestration systems** that manage containerized applications across clusters of hosts, including [Kubernetes](#) , [DC/OS](#) , and [Docker Swarm](#) .
- **Azure services** that support building and running applications at scale, including [Azure Kubernetes Service \(AKS\)](#), [App Service](#), [Batch](#), [Service Fabric](#), and others.

Developers can also push to a container registry as part of a container development workflow. For example, target a container registry from a continuous integration and delivery tool such as [Azure Pipelines](#) or [Jenkins](#) .

Configure ACR Tasks to automatically rebuild application images when their base images are updated, or automate image builds when your team commits code to a Git repository. Create multi-step tasks to automate building, testing, and patching multiple container images in parallel in the cloud.

Azure provides tooling including Azure Command-Line Interface, Azure portal, and API support to manage your Azure container registries. Optionally install the [Docker](#)

[Extension for Visual Studio Code](#) and the [Azure Account](#) extension to work with your Azure container registries. Pull and push images to an Azure container registry, or run ACR Tasks, all within Visual Studio Code.

Key features

- **Registry service tiers** - Create one or more container registries in your Azure subscription. Registries are available in three tiers: [Basic](#), [Standard](#), and [Premium](#), each of which supports webhook integration, registry authentication with Azure Active Directory, and delete functionality. Take advantage of local, network-close storage of your container images by creating a registry in the same Azure location as your deployments. 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. Azure Container Registry transfers container images over HTTPS, and supports TLS to secure client connections.

Important

Starting January 13, 2020, Azure Container Registry will require all secure connections from servers and applications to use TLS 1.2. Enable TLS 1.2 by using any recent docker client (version 18.03.0 or later). Support for TLS 1.0 and 1.1 will be retired.

You [control access](#) to a container registry using an Azure identity, an Azure Active Directory-backed [service principal](#), or a provided admin account. Use Azure role-based access control (Azure RBAC) to assign users or systems fine-grained permissions to a registry.

Security features of the Premium service tier include [content trust](#) for image tag signing, and [firewalls and virtual networks \(preview\)](#) to restrict access to the registry. Azure Security Center optionally integrates with Azure Container Registry to [scan images](#) whenever an image is pushed to a registry.

- **Supported images and artifacts** - Grouped in a repository, each image is a read-only snapshot of a Docker-compatible container. Azure container registries can include both Windows and Linux images. You control image names for all your container deployments. Use standard [Docker commands](#) to push images into a repository, or pull an image from a repository. In addition to Docker container images, Azure Container Registry stores [related content](#)

formats such as [Helm charts](#) and images built to the [Open Container Initiative \(OCI\) Image Format Specification](#) .

- **Automated image builds** - Use [Azure Container Registry Tasks](#) (ACR Tasks) to streamline building, testing, pushing, and deploying images in Azure. For example, use ACR Tasks to extend your development inner-loop to the cloud by offloading `docker build` operations to Azure. Configure build tasks to automate your container OS and framework patching pipeline, and build images automatically when your team commits code to source control.

[Multi-step tasks](#) provide step-based task definition and execution for building, testing, and patching container images in the cloud. Task steps define individual container image build and push operations. They can also define the execution of one or more containers, with each step using the container as its execution environment.

Next steps

- [Create a container registry using the Azure portal](#)
- [Create a container registry using the Azure CLI](#)
- [Automate container builds and maintenance with ACR Tasks](#)

Is this page helpful?

☐ Yes ☐ No
