

Background jobs in cloud-native applications: step-by-step plan

1. Make sure that you've installed Visual Studio 2022 and Docker Desktop.
2. In Visual Studio 2022, create a .NET 6 C# **console application**.
3. Right-click on the project and select *Add > Docker Support* to generate a **Docker file**.
If you add additional Nuget packages or project dependencies later, you also need to add them to your Docker file. You can regenerate the Docker file by adding Docker support again or you can update the Docker file manually.
4. In the Azure Portal, create a **Container Registry**.
5. In Visual Studio 2022, **publish** your console application to the container registry.
Visual Studio will use the Docker file to build and push the image to the container registry.
6. In the Azure Portal, create a **Container Instance**.
 - Select the container registry as the image source.
 - Select the image from the container registry as the image and image tag.
 - Set the VCPU and RAM size depending on your needs.
 - Set the networking type to *None*, since the image won't need a public or private IP address.
 - In this guide, we will trigger the Azure Container Instance using a service principal. You could also try to use a managed identity. However, this didn't work at the time of writing – probably because managed identities for Azure Container Instances are still in preview.
7. Manually **trigger** the container instance and check the **logs** to ensure the image is working.
8. To **schedule** the container instance with a logic app, we will need a service principal or managed identity to connect to the container instance from the logic app.
 - If you want to use a **service principal** to connect to the container instance, create an app registration in Azure Active Directory and add a client secret to use in the logic app.
 - If you would like to use a **managed identity**, enable the system-assigned managed identity in the Azure Container Instance.
9. Add the service principal or managed identity in the Access Control (**IAM**) of the container instance and assign it the **Contributor** role.
10. Create a **logic app**.
 - Use the logic app designer to add a recurrence trigger and a *Start containers* in a *container group* action.
 - Use the service principal or managed identity to connect to the container instance.
 - Save the logic app.
11. Congratulations! You've just created a timer triggered background job running in a container.