

ature is in a pre-release state and might change or have limited support. For more information, see the [product launch](#) ([/products/#product-launch-stages](#)).

This document shows you how to set up Visual Studio's Publishing Wizard to simplify deployment of .NET apps to Google Cloud.

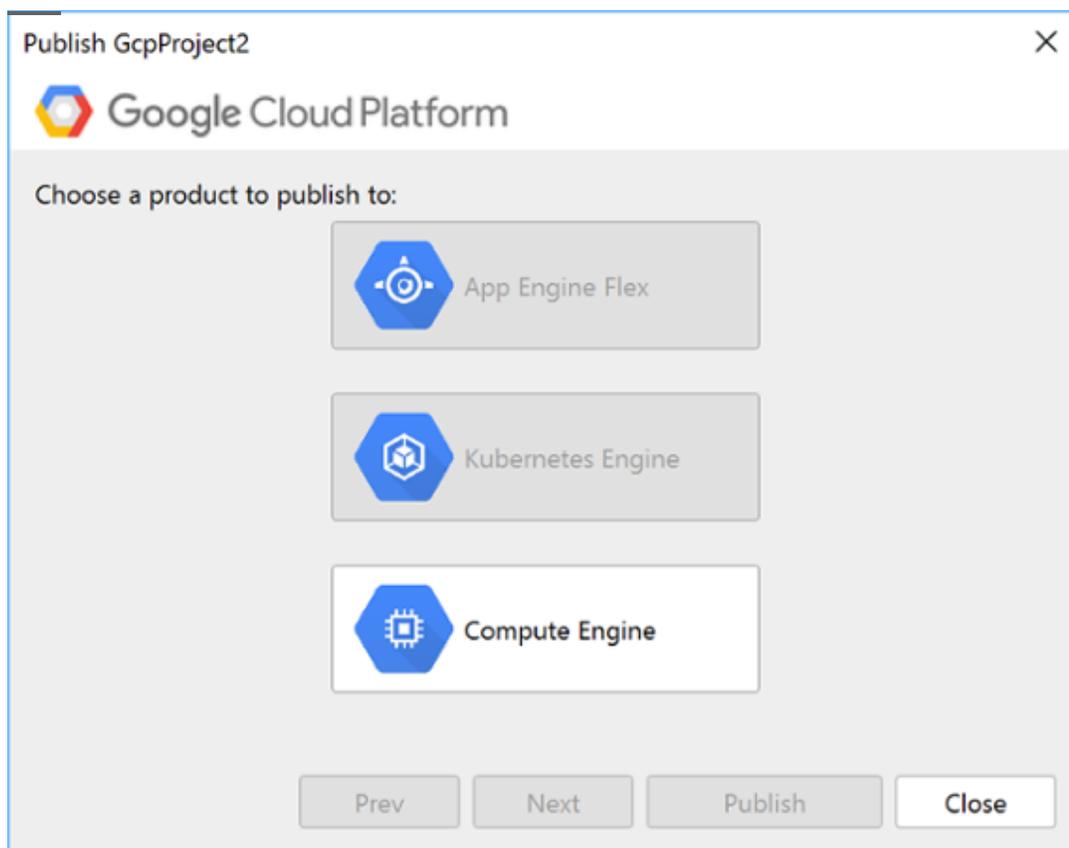
Deploying .NET apps to Google Cloud is easier than ever with the latest release of Cloud Tools for Visual Studio. We now have a deployment wizard built into the extension that will deploy your app to the following supported targets:

- Compute Engine for ASP.NET 4.x apps
- App Engine flexible environment and Google Kubernetes Engine for ASP.NET Core apps

The deployment wizard will automatically detect what targets are supported for your project and guide you through the deployment process.

The deployment wizard can be invoked by either going to **Tools > Google Cloud Tools > Publish [PROJECT-NAME] to Google Cloud** or by right-clicking on the project node in the Solution Explorer and clicking on **Publish [PROJECT-NAME] to Google Cloud**.

The **Publish [PROJECT-NAME] to Google Cloud** menu item will only be enabled if the startup project for the solution is compatible with Google Cloud.



The deployment wizard will display Google Cloud deployments targets that are compatible with the selected project.

If you want to change the project, open the Cloud Explorer by clicking on **Tools > Google Cloud Tools > Open Cloud Explorer** and select the project you want to deploy.

ASP.NET 4.x apps only run on Windows VMs on Compute Engine.

To deploy your ASP.NET 4.x app on Compute Engine:

1. Open the deployment wizard by clicking **Tools > Google Cloud Tools > Publish [PROJECT-NAME] to Google Cloud**.
2. Select **Compute Engine**
3. Select the Windows VM instance and deployment credentials

- The Windows VM has to be running IIS and capable of running ASP.NET 4.x apps, such as an ASP.NET VM created from the [Deployment Manager](https://console.cloud.google.com/marketplace/details/click-to-deploy-images/aspnet?q=asp.net) (<https://console.cloud.google.com/marketplace/details/click-to-deploy-images/aspnet?q=asp.net>).
- Select credentials for the deployment. To create Windows credentials click on **Manage Credentials**.

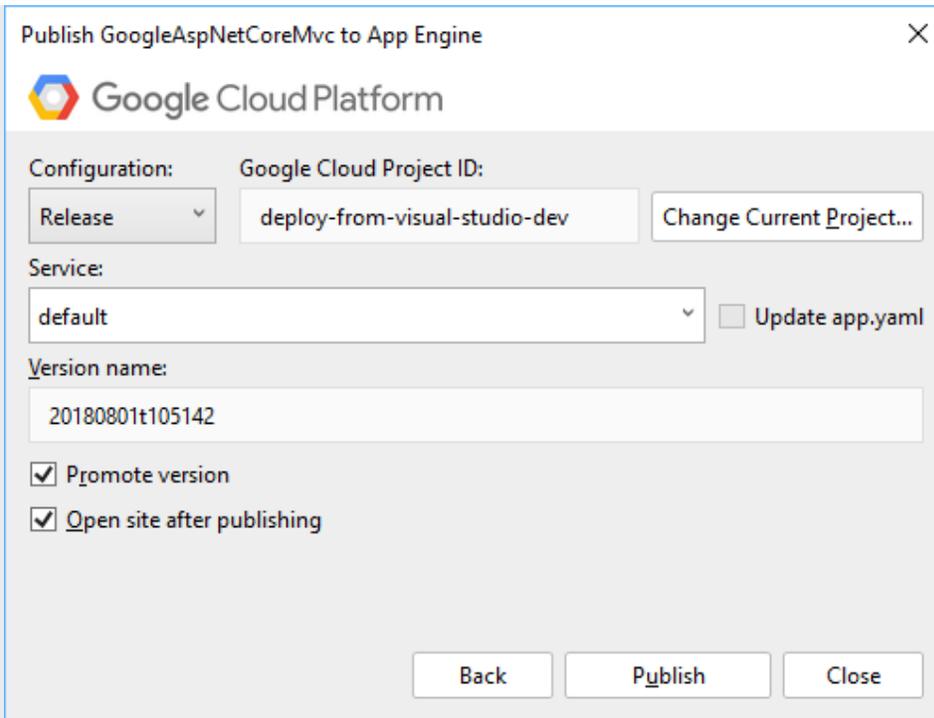
4. Click **Publish** to build and deploy your app to the selected VM.

Deployment progress will be displayed in the Visual Studio output window and a progress indicator will be displayed in the Visual Studio shell status bar.

ASP.NET Core apps can run inside of a Docker container so your app can be deployed on the App Engine flexible environment and GKE.

To deploy on flexible environment:

1. Open the deployment wizard by clicking **Tools > Google Cloud Tools > Publish [PROJECT-NAME] to Google Cloud**.
2. Select **App Engine Flex**



Publish GoogleAspNetCoreMvc to App Engine

Google Cloud Platform

Configuration: Google Cloud Project ID:

Release Change Current Project...

Service:

default Update app.yaml

Version name:

Promote version

Open site after publishing

Back Publish Close

3. Enter your app's version name and traffic management choice.

- The default version name is based on the current system time. You can specify a different name.

★ **Note:** Specifying an existing version name will cause the previous version to be overwritten.

- **Promoting version** lets you choose if this app version should receive 100% of the traffic. Checking this box will result in the new app receiving all of the traffic immediately after deployment.

4. Click **Publish** to build and deploy your app to the App Engine flexible environment.

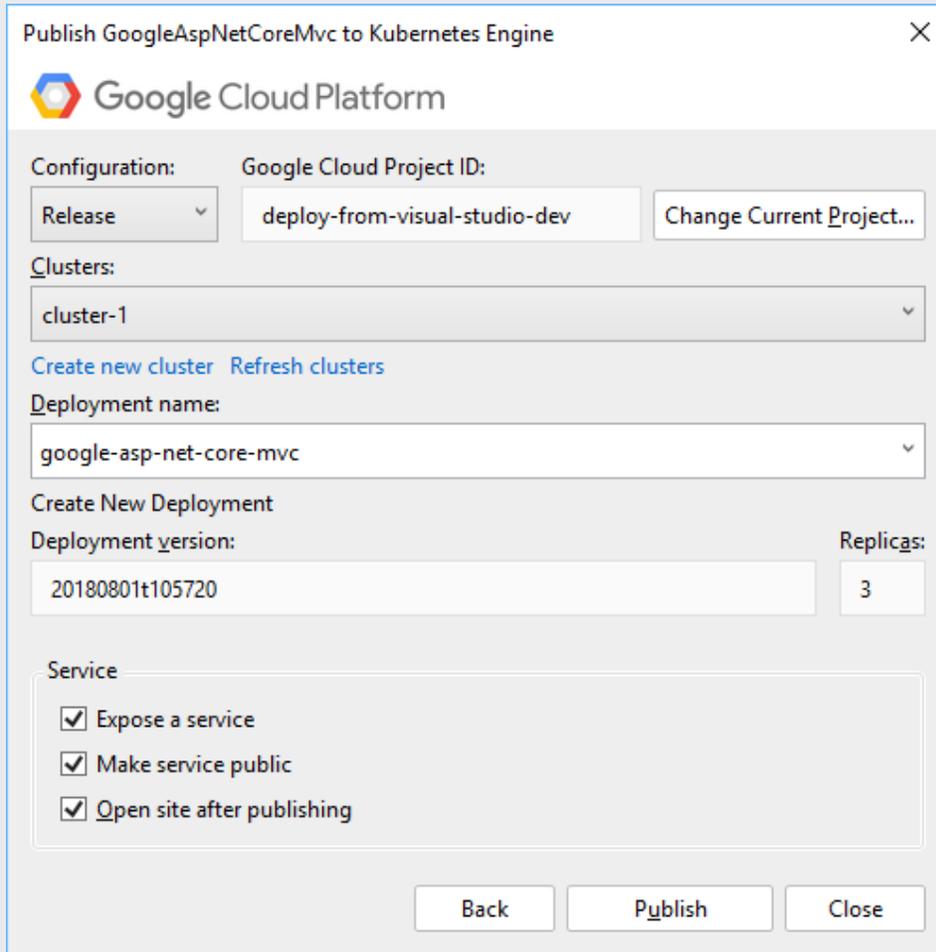
That's it! Your app will be deployed and progress will be displayed in the Visual Studio output window.

Custom App Engine settings can be done by creating an `app.yaml` file in the root folder of your project.

To generate `app.yaml`, right click on the project you want to deploy and click **Generate app.yaml and Dockerfile for [PROJECT-NAME]**. You can delete the Dockerfile if you want to run the default Docker image.

To deploy on GKE:

1. Open the deployment wizard by clicking **Tools > Google Cloud Tools > Publish [PROJECT-NAME] to Google Cloud**.
2. Select **Container Engine**.



Publish GoogleAspNetCoreMvc to Kubernetes Engine

Google Cloud Platform

Configuration: Google Cloud Project ID:
Release | deploy-from-visual-studio-dev | Change Current Project...

Clusters:
cluster-1

Create new cluster Refresh clusters

Deployment name:
google-asp-net-core-mvc

Create New Deployment

Deployment version: 20180801t105720 | Replicas: 3

Service

- Expose a service
- Make service public
- Open site after publishing

Back Publish Close

3. Select a cluster for deployment, enter your app's deployment name, version and the number of replicas.

- To create a cluster, click **Create new cluster**. This will take you to the cluster creation page on Google Cloud Console. To view the cluster in Visual Studio, click **Refresh clusters**.
- The deployment name is used when creating the Kubernetes deployment and, if optionally selected, the kubernetes service to run your app on the cluster. You can change the name to something more descriptive.

★ **Note:** If you use a name that already exists then the old deployment will be updated instead of creating a new deployment.

- The default version name is based on the current system time. You can specify a different name.

★ **Note:** Specifying an existing version name will cause the previous version to be overwritten.

- The number of replicas can be changed, the default is three.
- You can choose to expose a Kubernetes service on the Internet. By exposing a public service on the Internet you will get a public IP address that you can use to access your service from outside your cluster.

4. Click **Publish**

That's it! Your app will be packaged in a Docker image and deployed to your container. If your app is an exposed service then Visual Studio will wait until the service's IP address is available.