**Prerequisites**

* Create an Azure DevOps organization. Make sure you are an administrator of the Azure DevOps project.
  + For this demo, the following project exists within the demo environment.
    - [**https://dev.azure.com/poc2pattern/POCtoPattern**](https://dev.azure.com/poc2pattern/POCtoPattern)
* Create a shared image gallery

## Preview: Register the feature

Shared Image Galleries is in preview, but you need to register the feature before you can use it. To register the Shared Image Galleries feature:

Azure CLI Copy Try It

az feature register --namespace Microsoft.Compute --name GalleryPreview

az provider register -n Microsoft.Compute

It might take a few minutes to register the feature. You can check the progress using:

Azure CLI Copy Try It

az provider show -n Microsoft.Compute

## Create an image gallery

An image gallery is the primary resource used for enabling image sharing. Gallery names must be unique within your subscription. Create an image gallery using [az sig create](https://docs.microsoft.com/en-us/cli/azure/sig#az-sig-create). The following example creates a gallery named myGallery in myGalleryRG.

Azure CLI Copy Try It

az group create --name myGalleryRG --location WestCentralUS

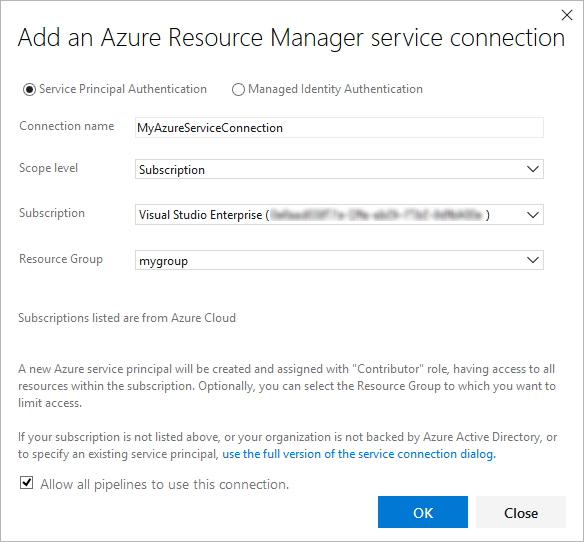
az sig create -g myGalleryRG --gallery-name myGallery

**Initialize the repository**

* Go to **Azure Repos**. (The **Code** hub in the previous navigation)
* For this demo, the follow Repo exists within the demo environment.
  + [**https://dev.azure.com/poc2pattern/POCtoPattern/\_git/POCtoPattern?path=%2F&version=GBmaster**](https://dev.azure.com/poc2pattern/POCtoPattern/_git/POCtoPattern?path=%2F&version=GBmaster)

**Create a Service Connection**

1. In Azure DevOps, open the **Service connections** page from the [project settings page](https://docs.microsoft.com/en-us/azure/devops/project/navigation/go-to-service-page?view=azure-devops#open-project-settings). In TFS, open the **Services** page from the "settings" icon in the top menu bar.
2. Choose **+ New service connection** and select the type of service connection you need.
3. Fill in the parameters for the service connection. The list of parameters differs for each type of service connection - see the [following list](https://docs.microsoft.com/en-us/azure/devops/pipelines/library/service-endpoints?view=azure-devops#ep-types). For example, this is the default **Azure Resource Manager** connection dialog. Choose Service Principal Authentication.



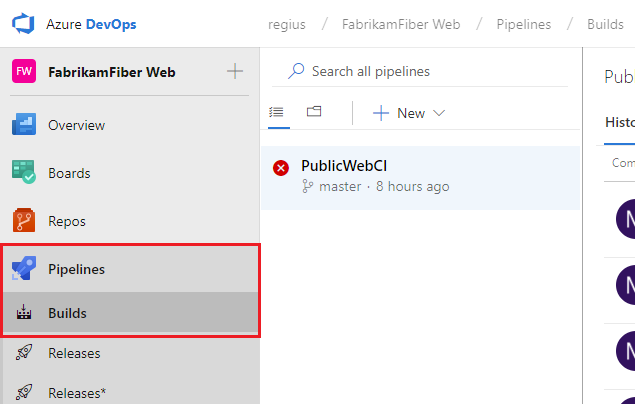
1. Decide if you want the service connection to be accessible for any pipeline by setting the **Allow all pipelines to use this connection** option. This option allows pipelines defined in YAML, which are not automatically authorized for service connections, to use this service connection. See [Use a service connection](https://docs.microsoft.com/en-us/azure/devops/pipelines/library/service-endpoints?view=azure-devops#use-connection).
2. Choose **OK** to create the connection.

For more information about Azure Resource Manager service connections, see [Connect to Microsoft Azure](https://docs.microsoft.com/en-us/azure/devops/pipelines/library/connect-to-azure?view=azure-devops). You can also create your own [custom service connections](https://docs.microsoft.com/en-us/azure/devops/extend/develop/service-endpoints?view=azure-devops).

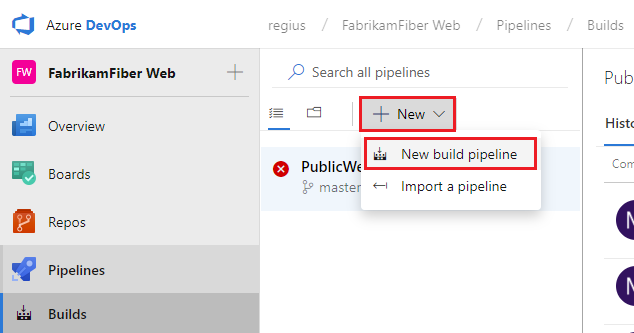
**Create a Build Pipeline**

Create a build pipeline that downloads an Azure Martketplace image, patches it, creates a managed image using packer and stores it in an existing shared image gallery.

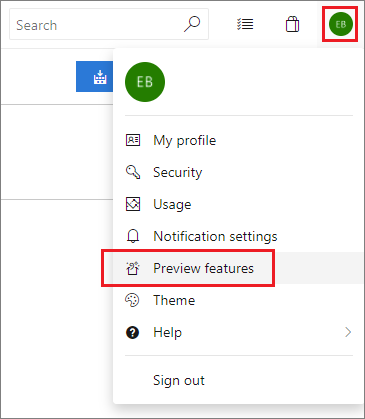
1. Select **Azure Pipelines**, it should automatically take you to the **Builds** page.

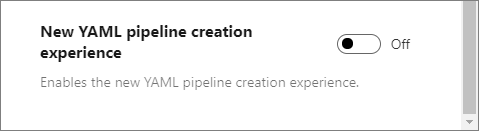


1. Create a new pipeline.

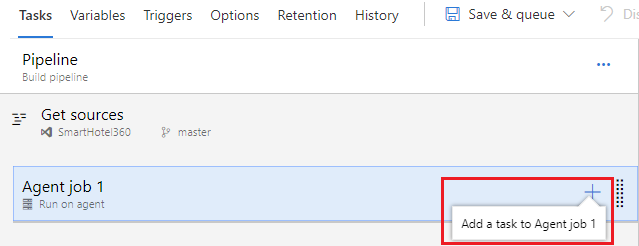


For new Azure DevOps accounts, this will automatically take you to the YAML pipeline creation experience. To get to the classic editor and complete this guide, you must turn off the **preview feature** for the New YAML pipeline creation experience:

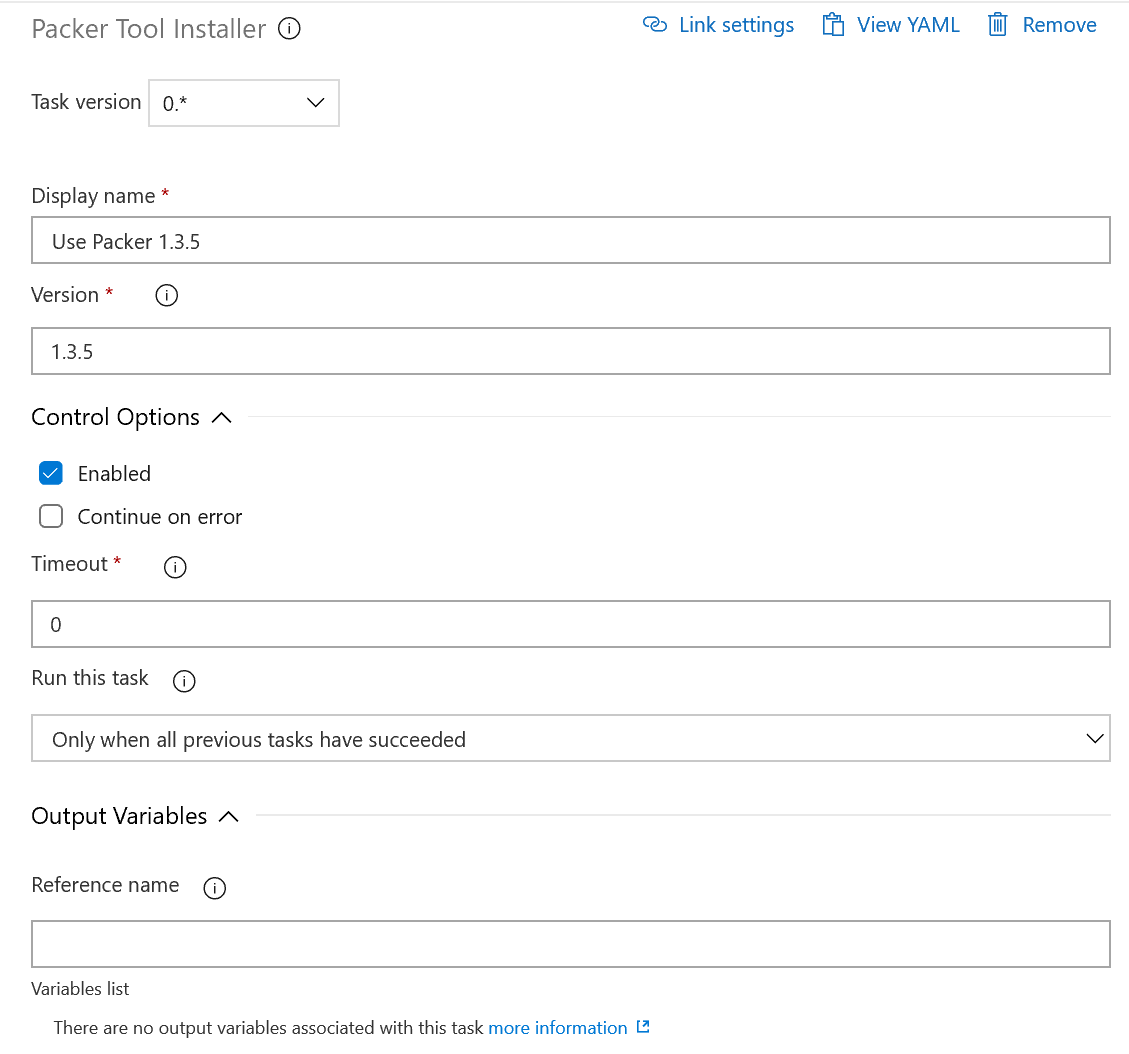




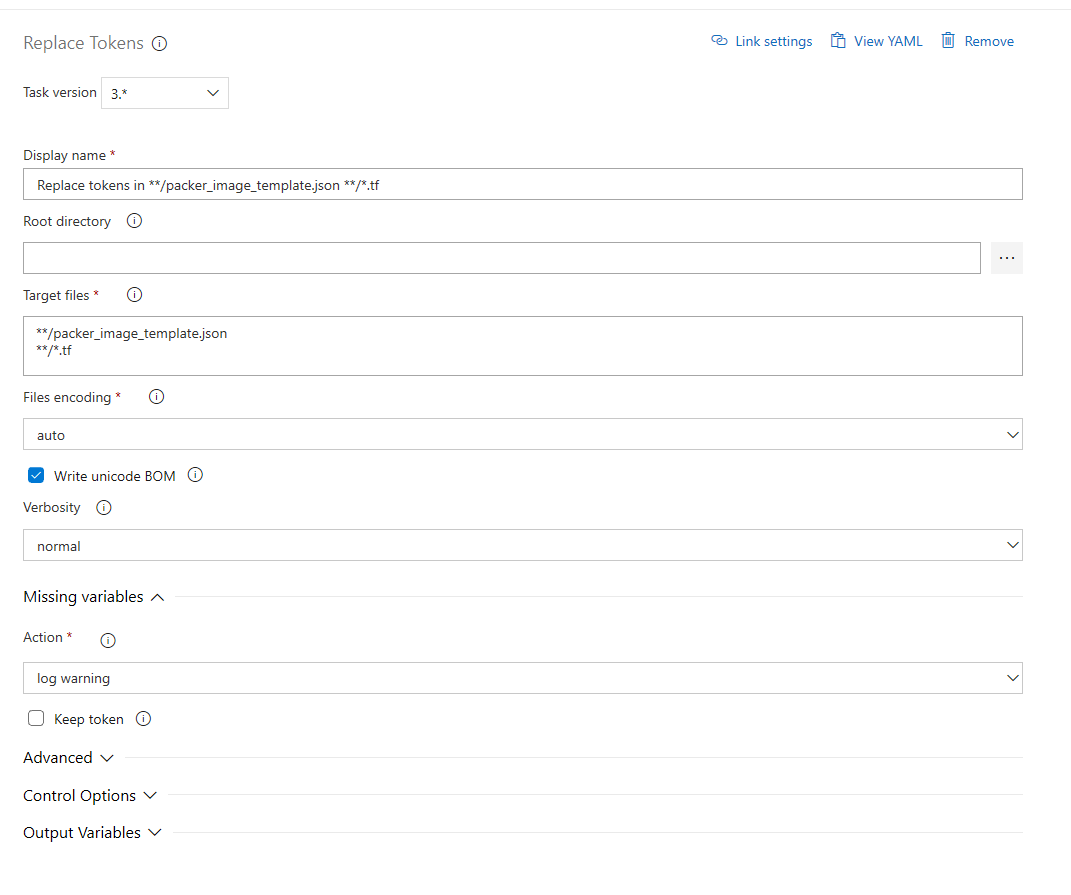
1. Make sure that the **source**, **project**, **repository**, and default **branch** match the location where the scripts are located.
2. Start with an **Empty job**.
3. On the left side, select **Pipeline** and specify whatever **Name** you want to use. For the **Agent pool**, select **Hosted VS2017**.
4. On the left side, select the plus sign **( + )** to add a task to **Job 1**. On the right side, select the **Utility** category, select the **Packer Tool Installer** task from the list, and then choose **Add**.



1. On the left side, select your new **Packer Tool Installer** task.
2. Fill out the configuration information below to utilize the latest version Packer.

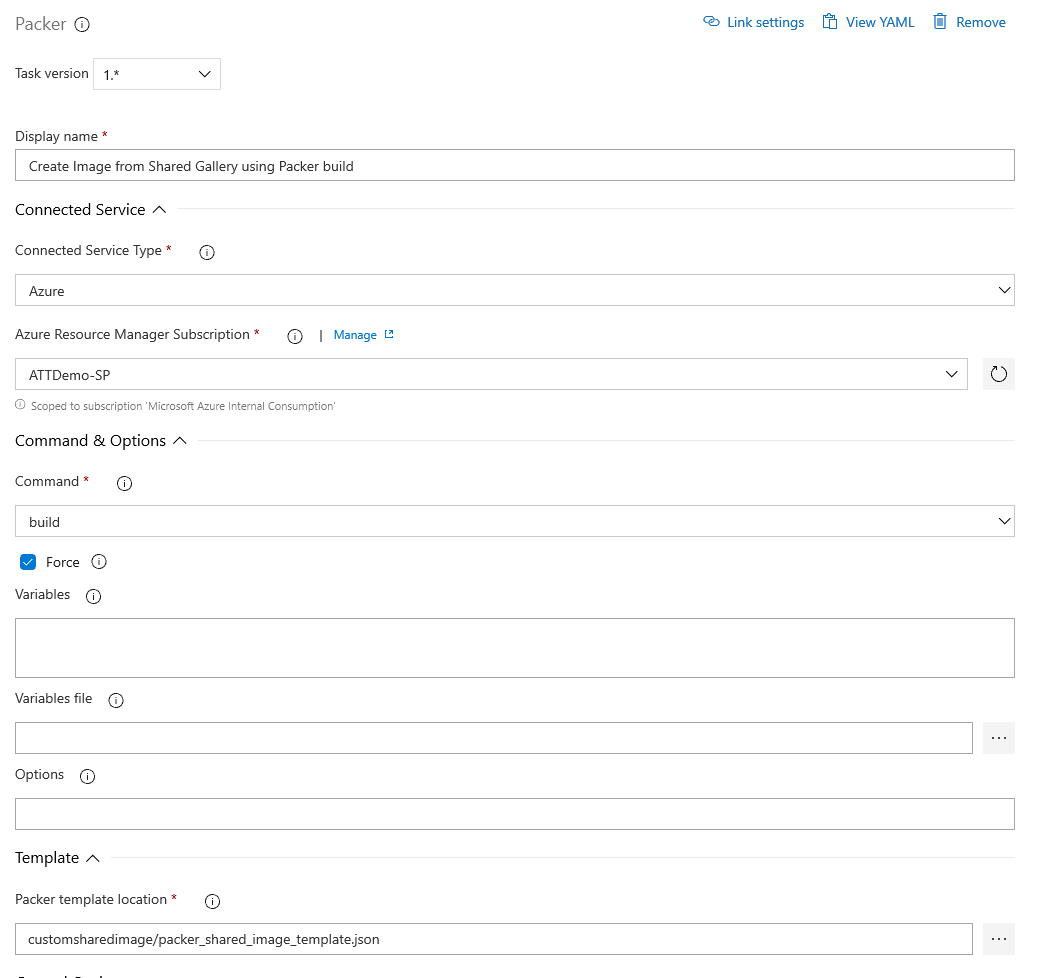


1. Select **Save**, and then select **Save**.
2. On the left side, select the plus sign **( + )** to add a task to **Job 1**. On the right side, select the **Utility** category, select the **Replace tokens** task from the list, and then choose **Add**.
3. On the left side, select your new **Replace tokens** task.
4. Fill out the configuration information as shown below.

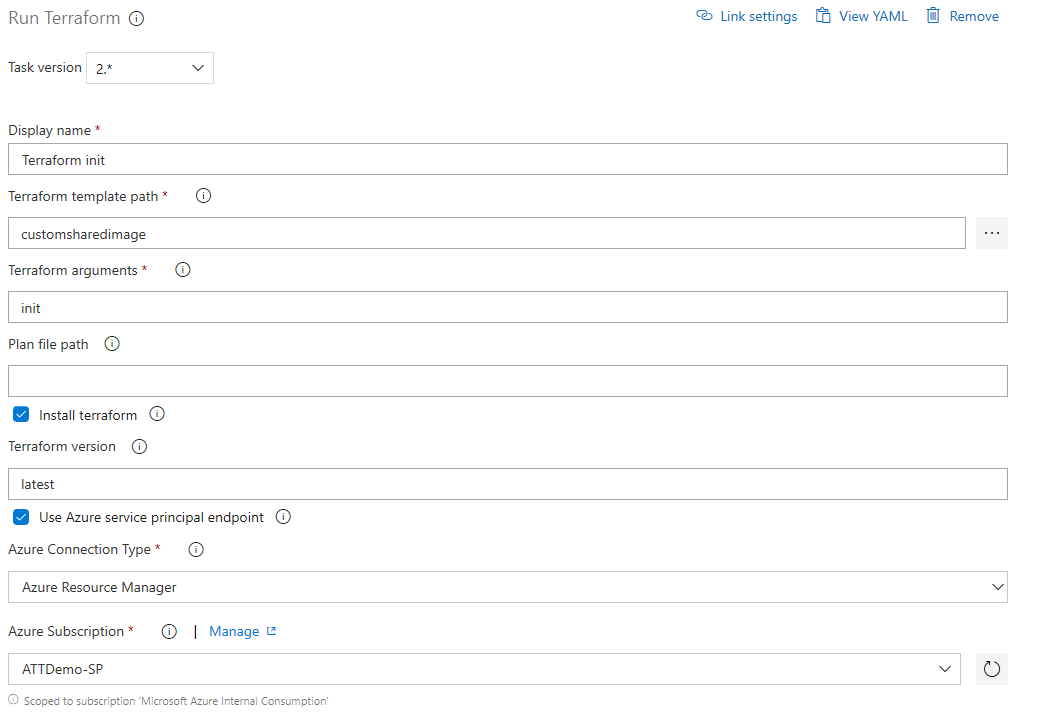




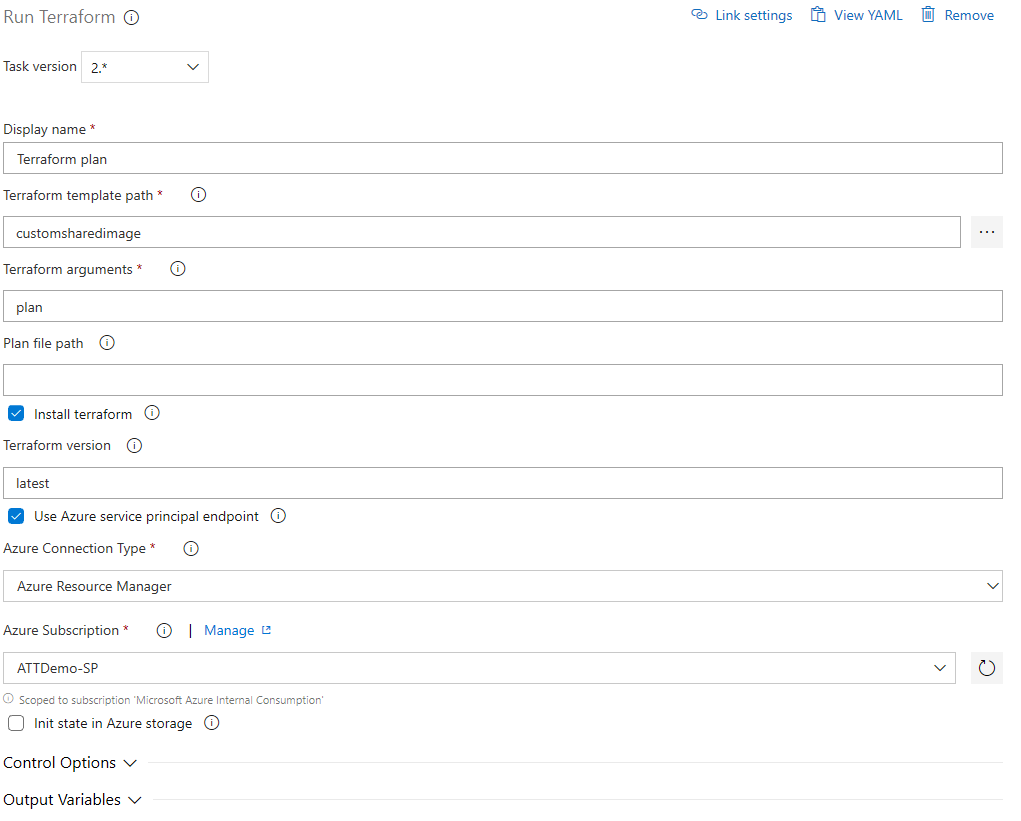
1. Select **Save**, and then select **Save**.
2. On the left side, select the plus sign **( + )** to add a task to **Job 1**. On the right side, select the **Utility** category, select the **Packer** task from the list, and then choose **Add**.
3. On the left side, select your new **Packer** task.
4. Fill out the configuration information as shown below. Source repo folder and Service Connection.



1. Select **Save**, and then select **Save**.
2. On the left side, select the plus sign **( + )** to add a task to **Job 1**. On the right side, select the **Utility** category, select the **Run Terraform** task from the list, and then choose **Add**.
3. On the left side, select your new **Run Terraform** task. Fill out the following information for Terraform init. Source repo folder and Service Connection.



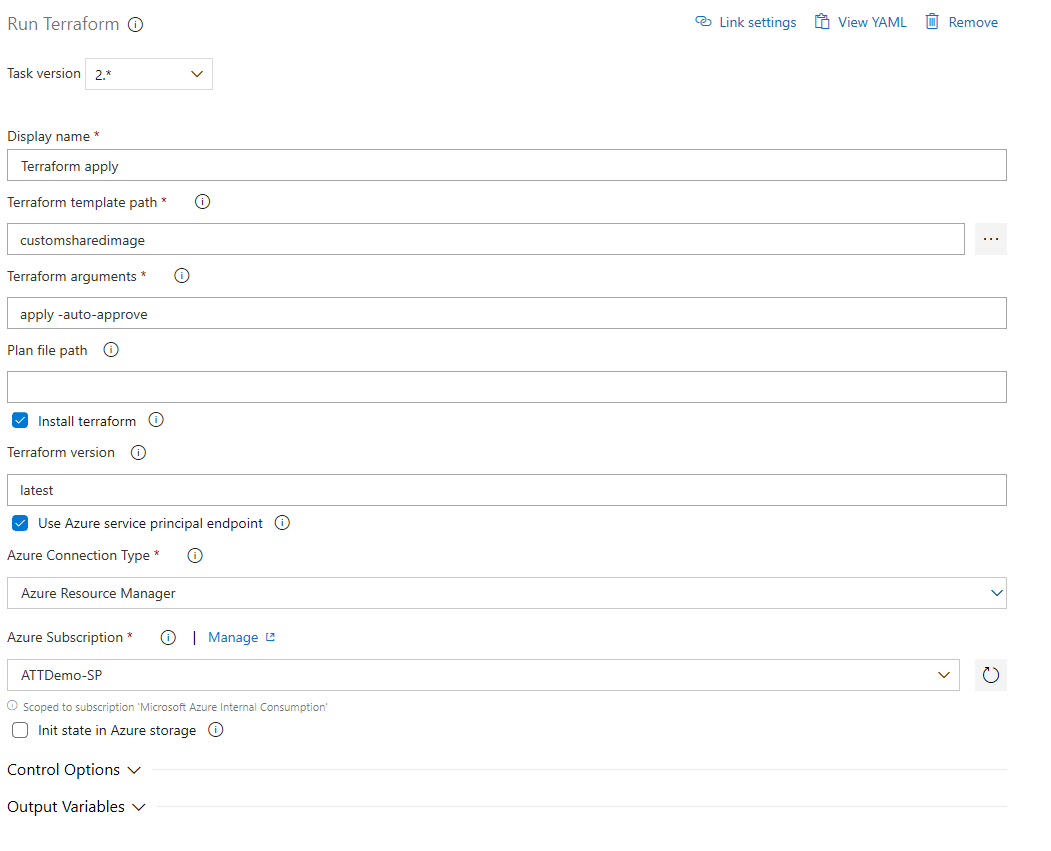
1. Select **Save**, and then select **Save**.
2. On the left side, select the plus sign **( + )** to add a task to **Job 1**. On the right side, select the **Utility** category, select the **Run Terraform** task from the list, and then choose **Add**.
3. On the left side, select your new **Run Terraform** task. Fill out the following information for Terraform plan.



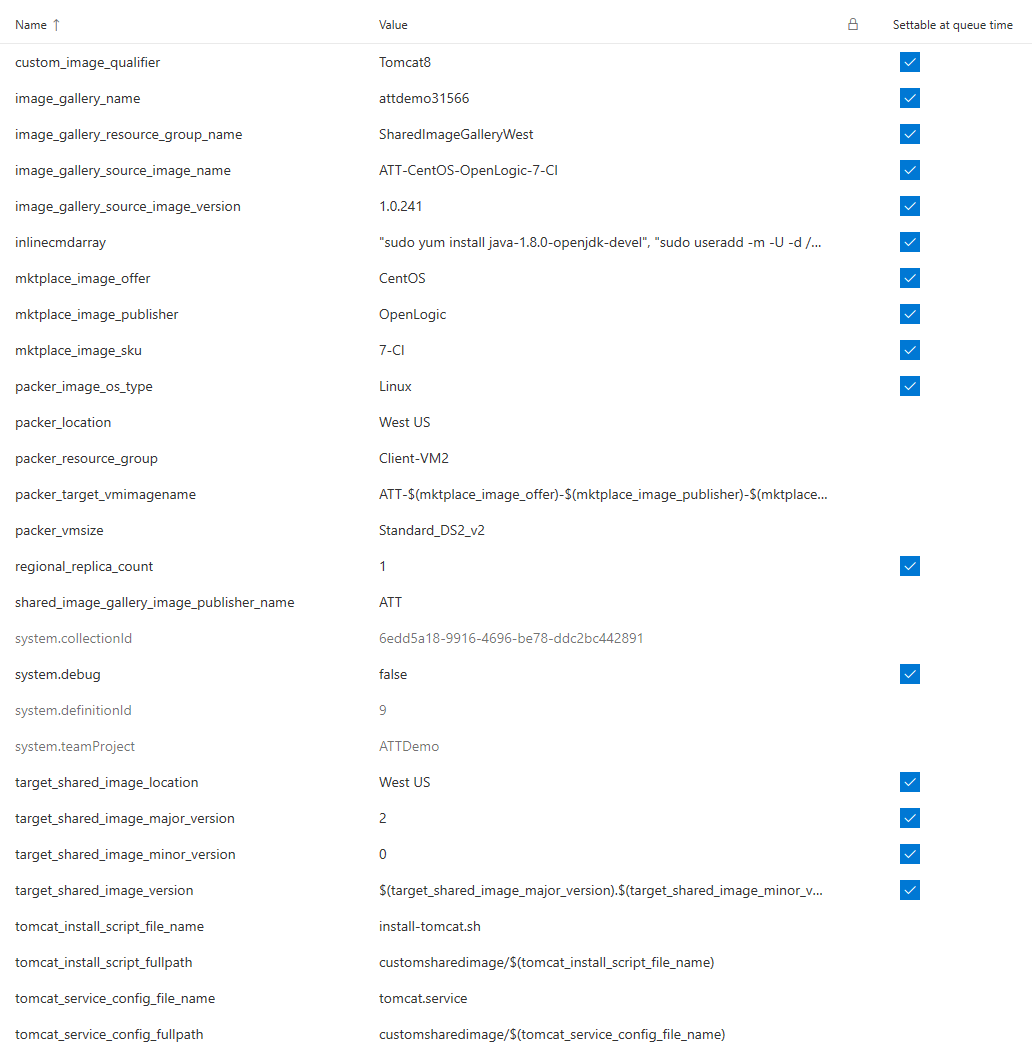
23. Select **Save**, and then select **Save**.

24. On the left side, select the plus sign **( + )** to add a task to **Job 1**. On the right side, select the **Utility** category, select the **Run Terraform** task from the list, and then choose **Add**.

25. On the left side, select your new **Run Terraform** task. Fill out the following information for Terraform apply.



1. Select **Save**, and then select **Save**.
2. Fill out Pipeline Variables



1. Queue the build pipeline by clicking the Queue button
2. The output is a managed image in a resource group and a shared image gallery version in the location specified in the environment variables.

A build pipeline is the entity through which you define your automated build pipeline. In the build pipeline, you compose a set of tasks, each of which perform a step in your build. The task catalog provides a rich set of tasks for you to get started. You can also add PowerShell or shell scripts to your build pipeline.