Demonstration Setting Up Service Hooks to Monitor the Pipeline

In this demonstration, you will investigate Service Hooks.

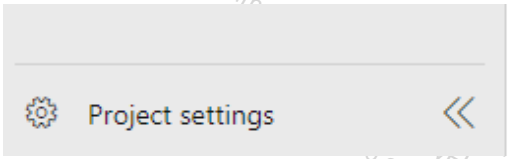
Note: Before starting this walkthrough, make sure you have performed the steps in the prerequisites section and the previous walkthroughs.

Steps

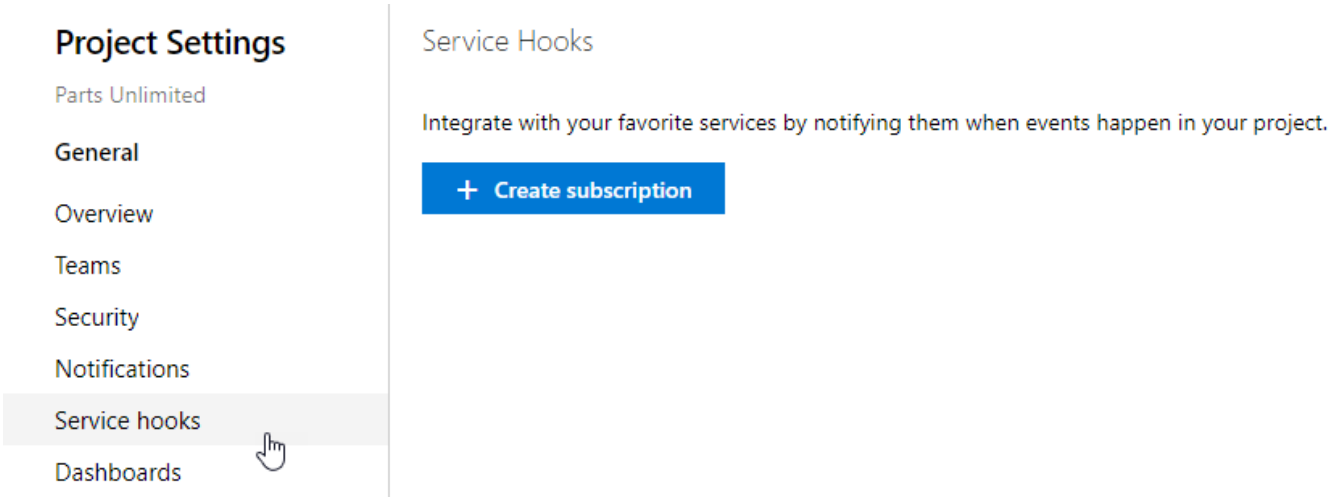
Let's now take a look at how a release pipeline can communicate with other services by using service hooks.

Azure DevOps can be integrated with a wide variety of other applications. It has built in support for many applications, and generic hooks for working with other applications. Let's take a look.

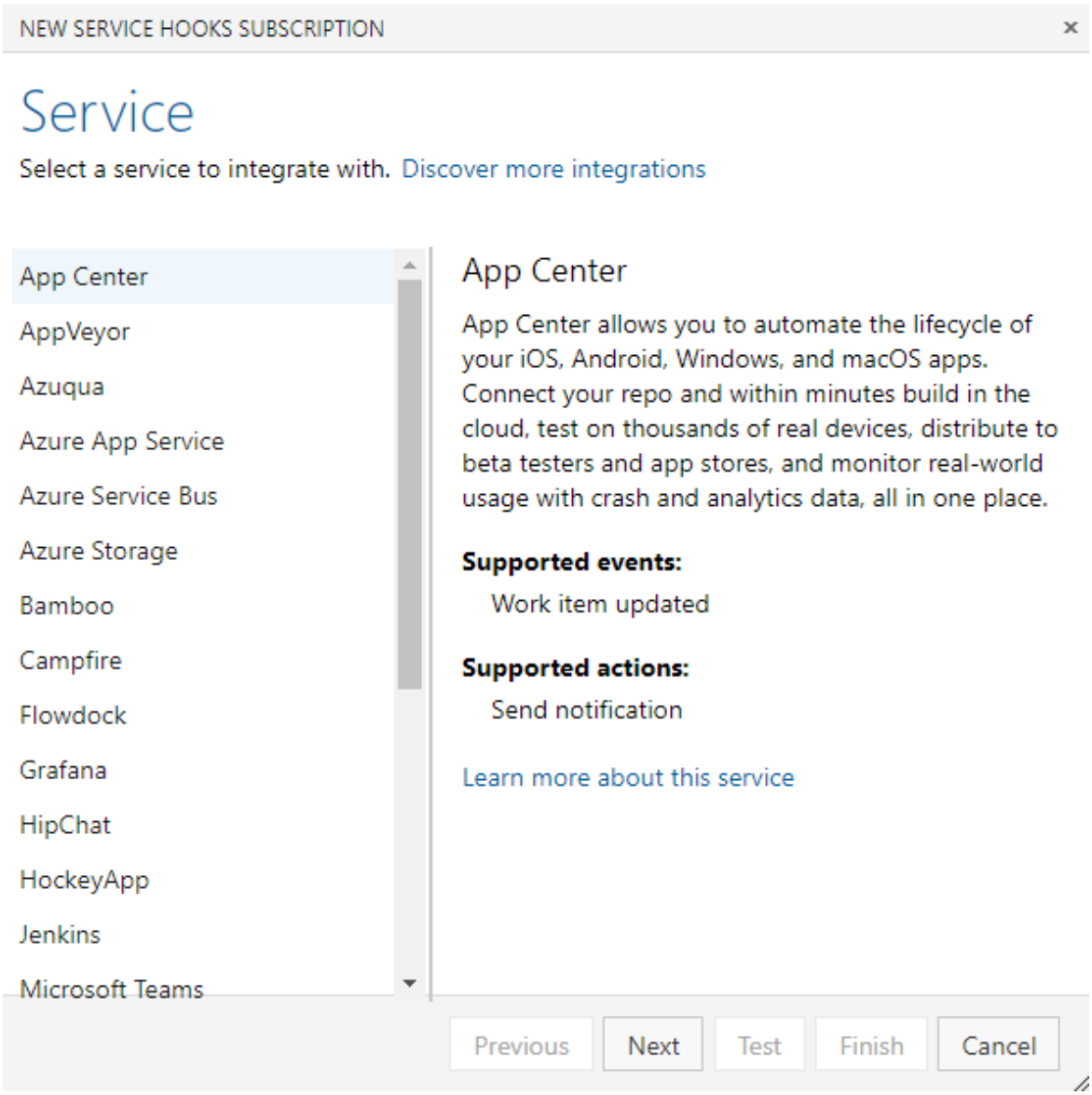
1. Below the main menu for the **Parts Unlimited** project, click **Project settings**.



2. In the **Project settings** menu, click **Service hooks**.

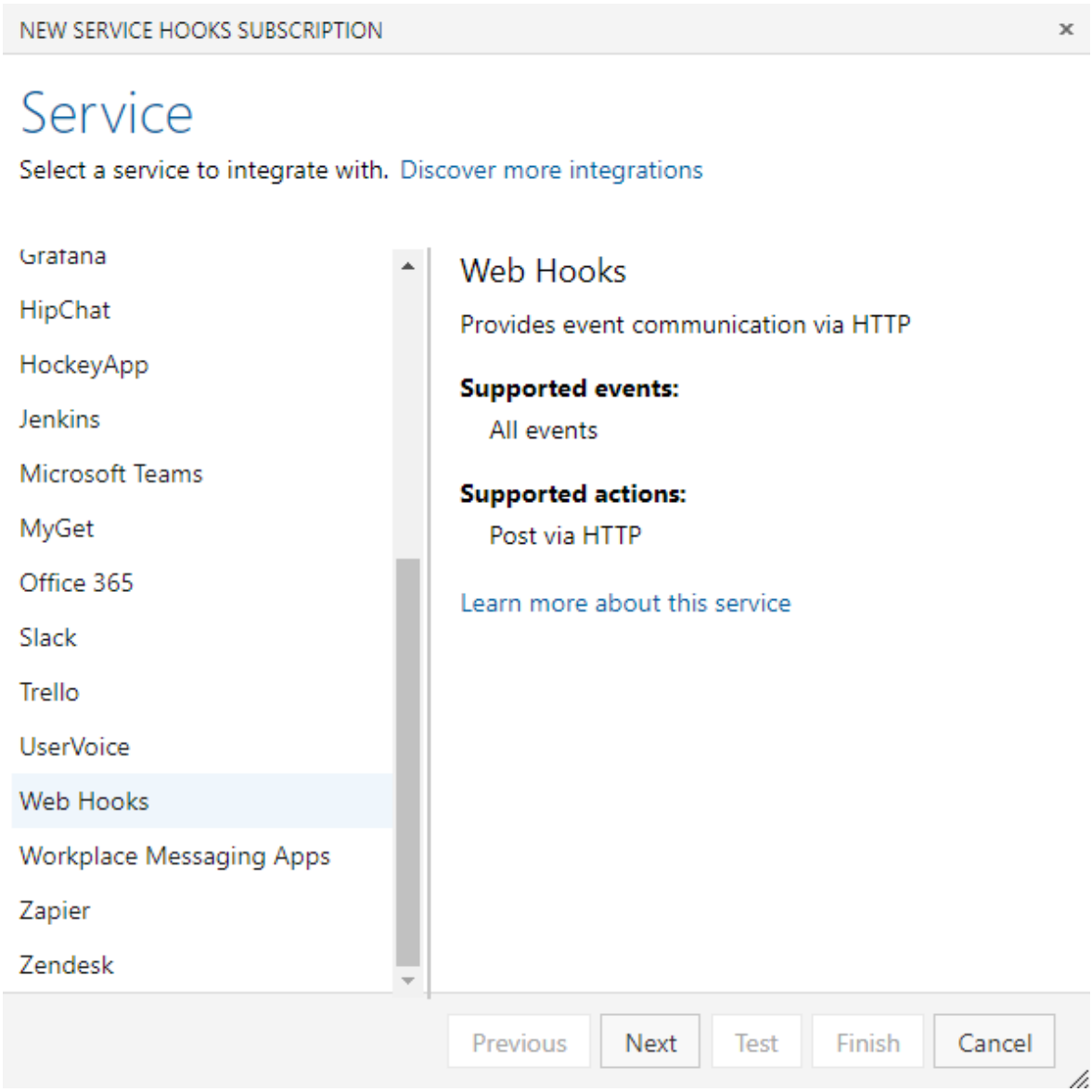


3. Click **+Create subscription**.



By using service hooks, we can notify other applications that an event has occurred within Azure DevOps. We could also send a message to a team in **Microsoft Teams** or **Slack**. We could also trigger an action in **Bamboo** or **Jenkins**.

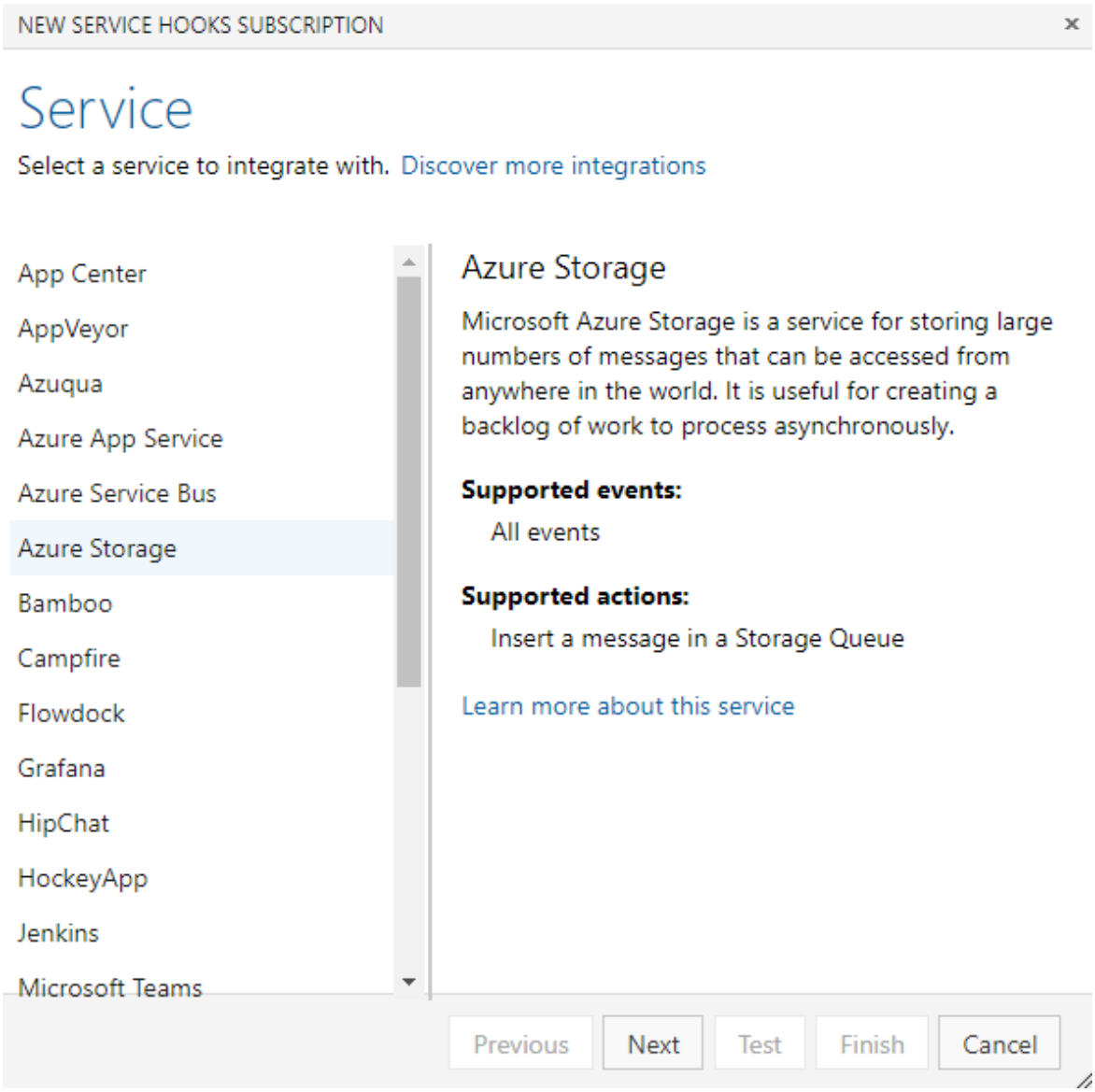
4. Scroll to the bottom of the list of applications and click on **Web Hooks**.



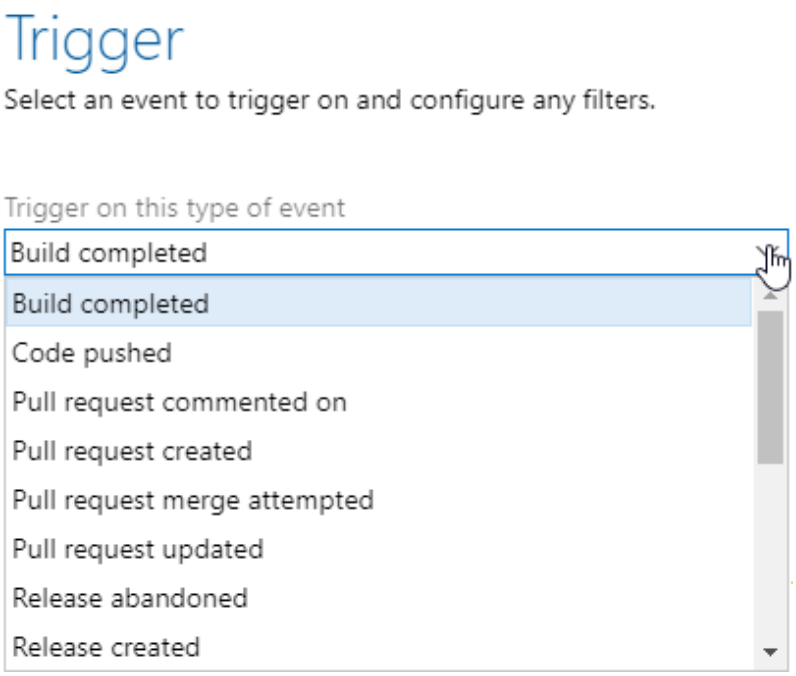
If the application that you want to communicate with isn't in the list of available application hooks, you can almost always use the **Web Hooks** option as a generic way to communicate. It allows you to make an HTTP POST when an event occurs. So, if for example, you wanted to call an Azure Function or an Azure Logic App, you could use this option.

To demonstrate the basic process for calling web hooks, we'll write a message into a queue in the Azure Storage account that we have been using.

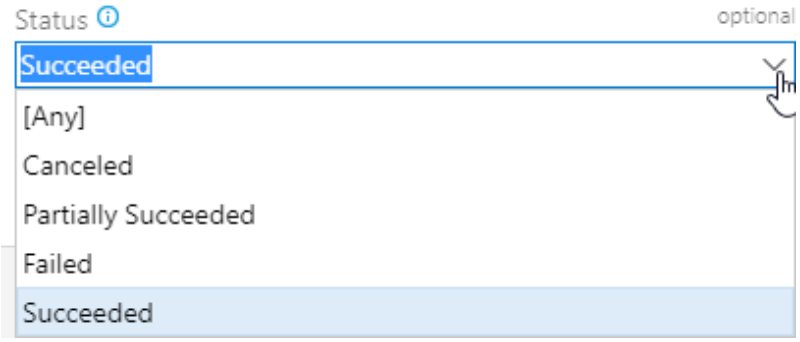
5. From the list of available applications, click **Azure Storage**.



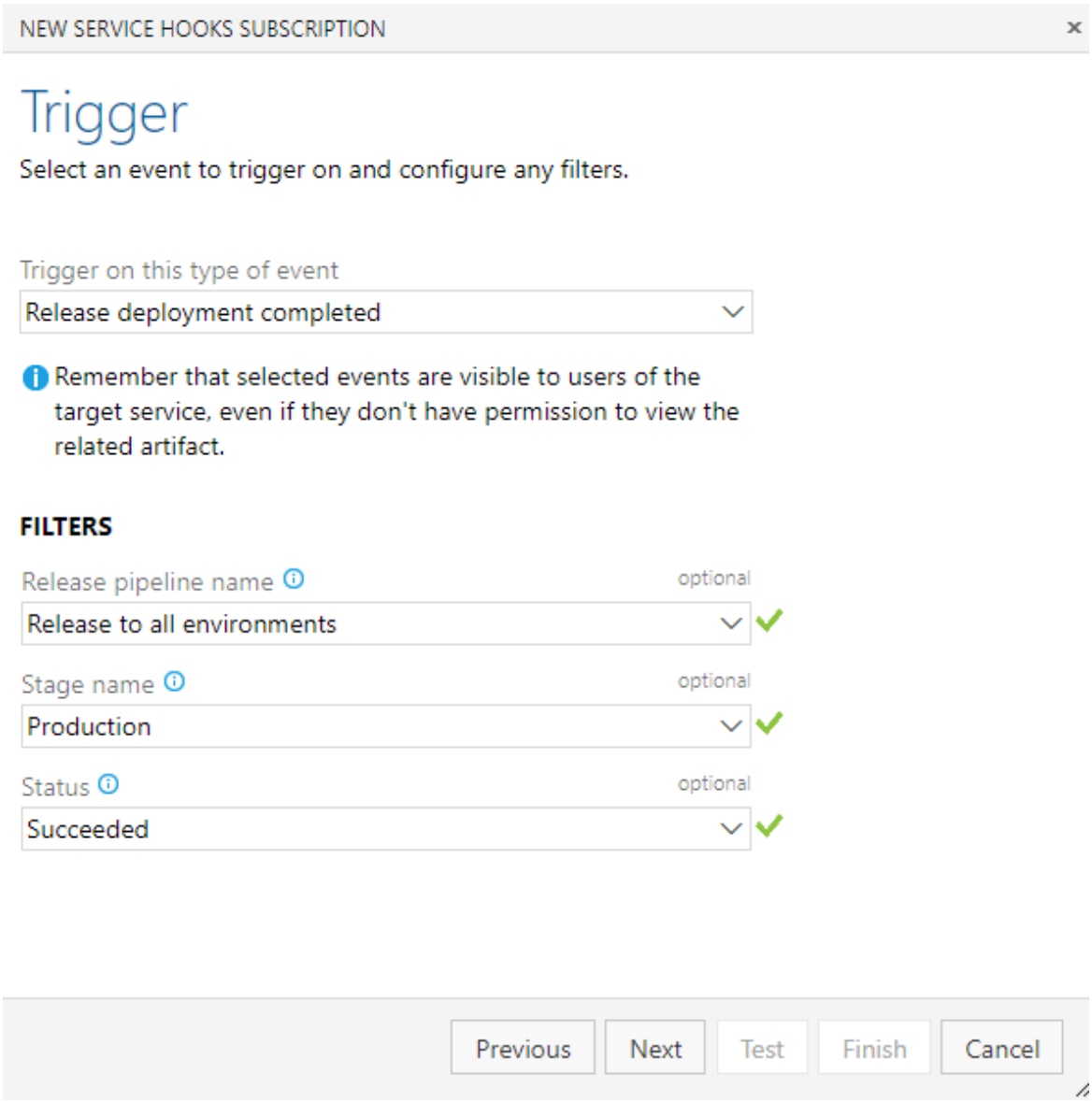
6. Click **Next**. In the **Trigger** page, we determine which event causes the service hook to be called. Click the drop down for **Trigger on this type of event** to see the available event types.



7. Ensure that **Release deployment completed** is selected, then in the **Release pipeline name** select **Release to all environments**. For **Stage**, select **Production**. Drop down the list for **Status** and note the available options.

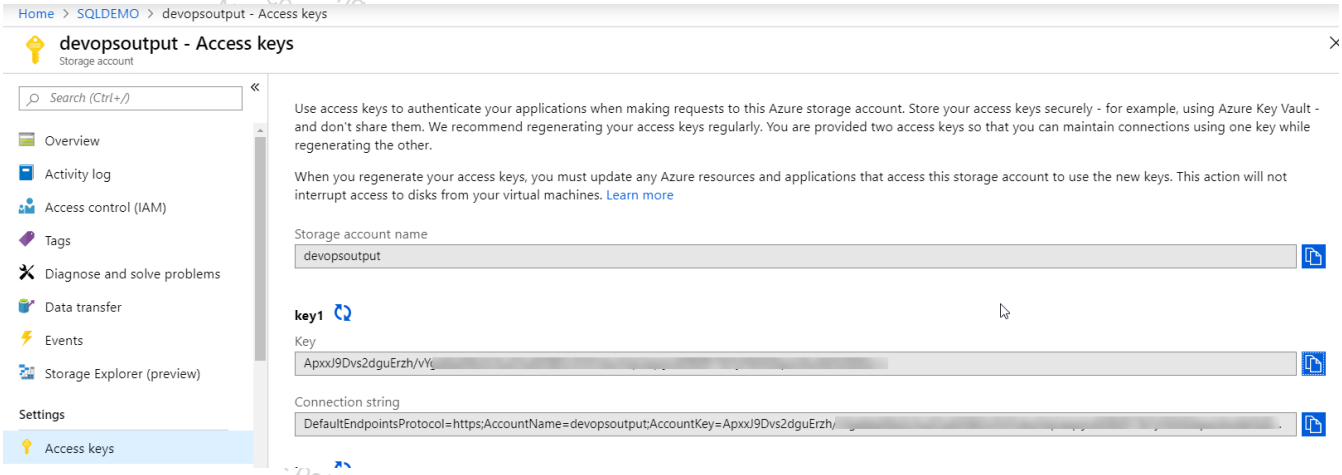


8. Ensure that **Succeeded** is selected, then click **Next**.

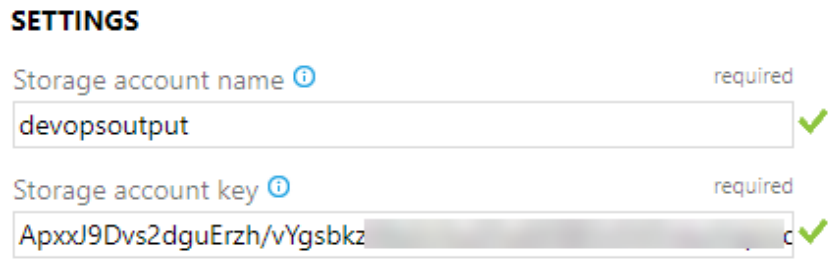


9. In the **Action** page, enter the name of your Azure storage account.

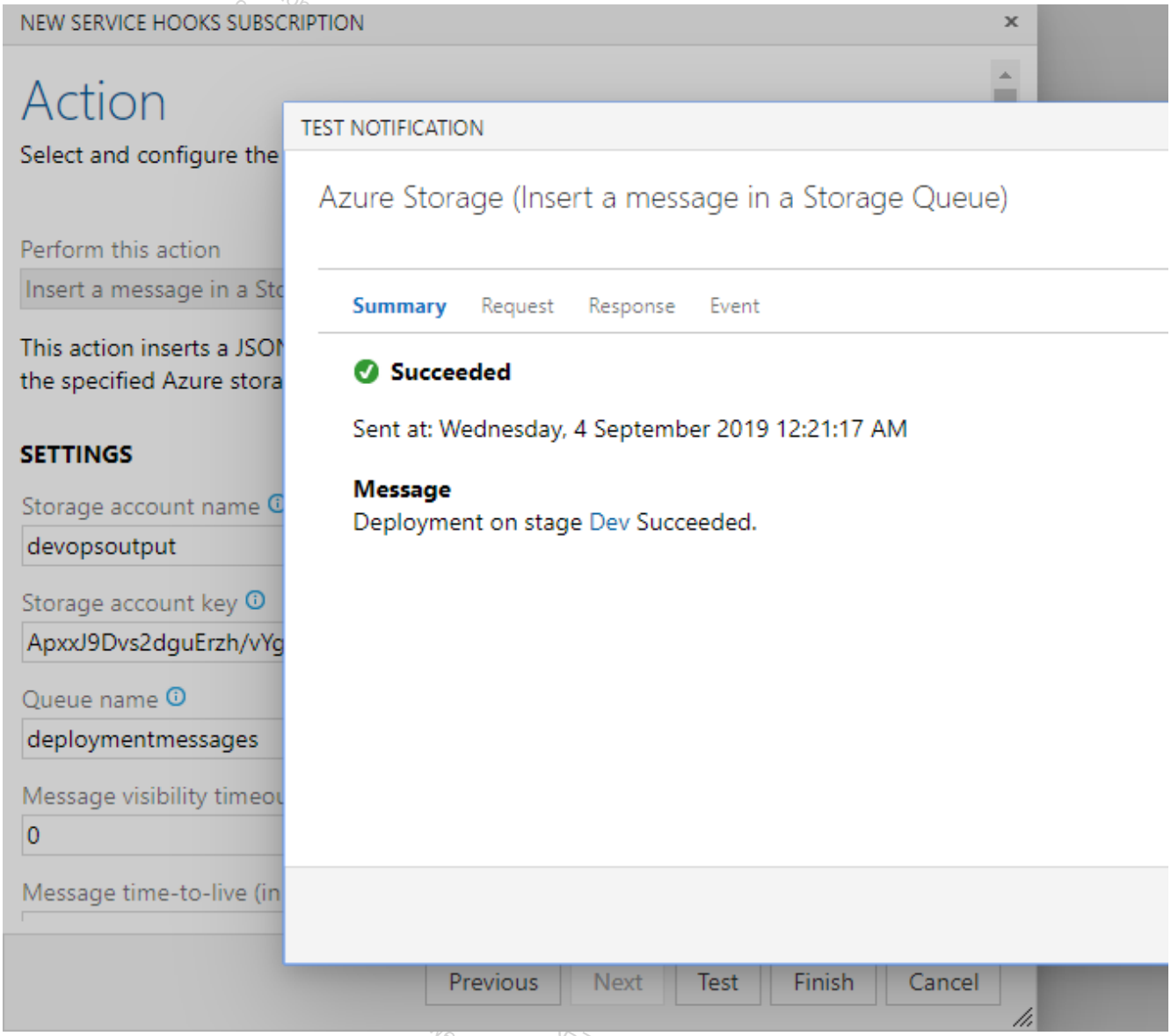
10. Open the Azure Portal, and from the settings for the storage account, in the **Access keys** section, copy the value for **Key**.



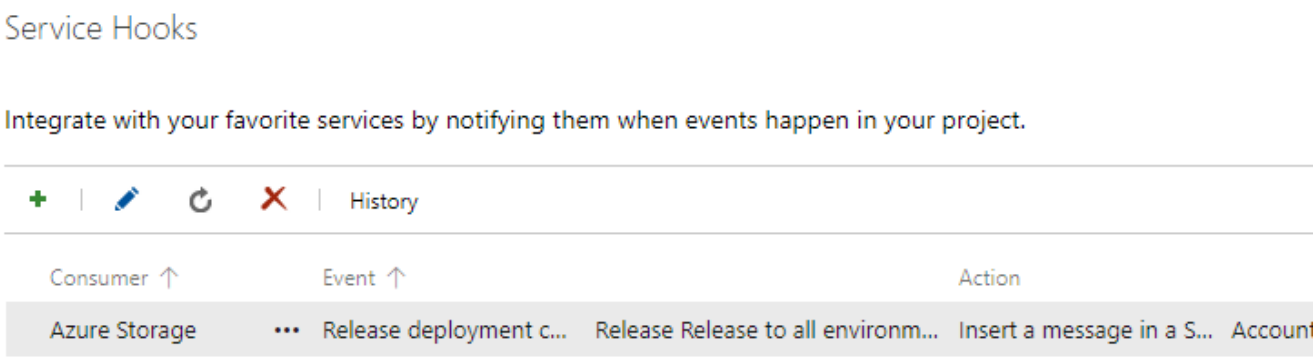
11. Back in the **Action** page in Azure DevOps, paste in the key.



12. For **Queue name** enter **deploymentmessages**, then click **Test**.



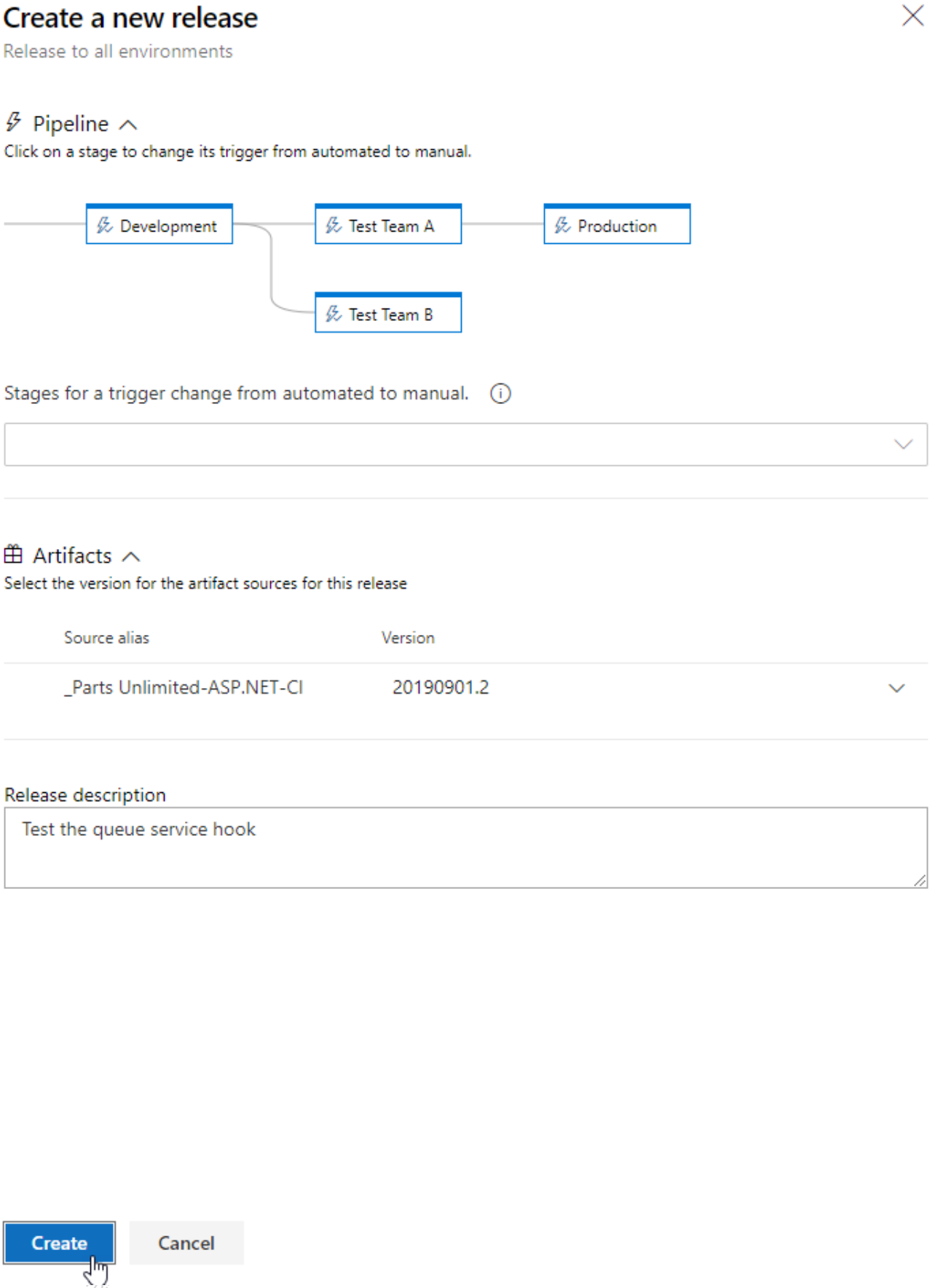
13. Make sure that the test succeeded, then click **Close**, and on the **Action** page, click **Finish**.



Create a release to test the service hook

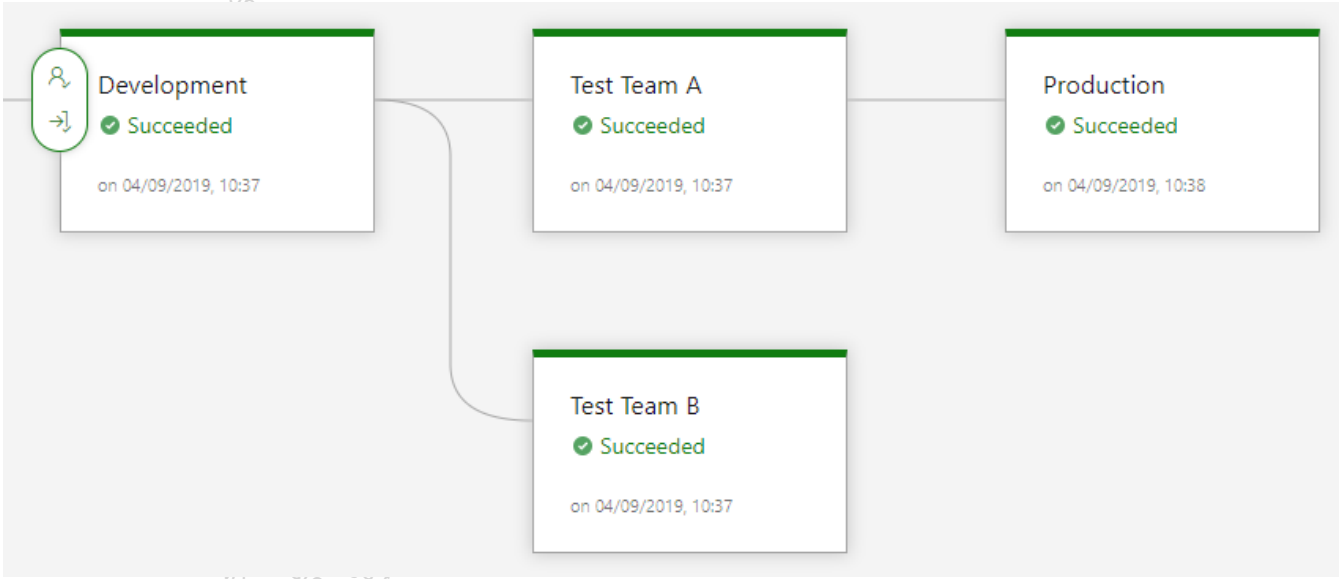
Now that you have successfully added the service hook, it's time to test it.

14. From the main menu of the **Parts Unlimited** project, click **Pipelines**, then click **Releases**, then click **Create release**, and in the **Create a new release** pane, enter **Test the queue service hook** for **Release description**, and click **Create**.



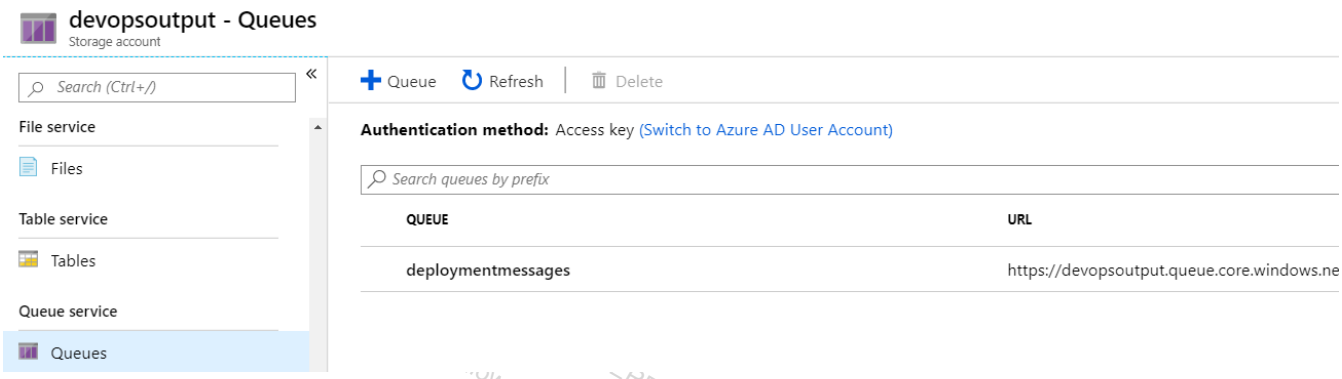
15. Click to view the release details.

16. If the release is waiting for approval, click to approve it and wait for the release to complete successfully.

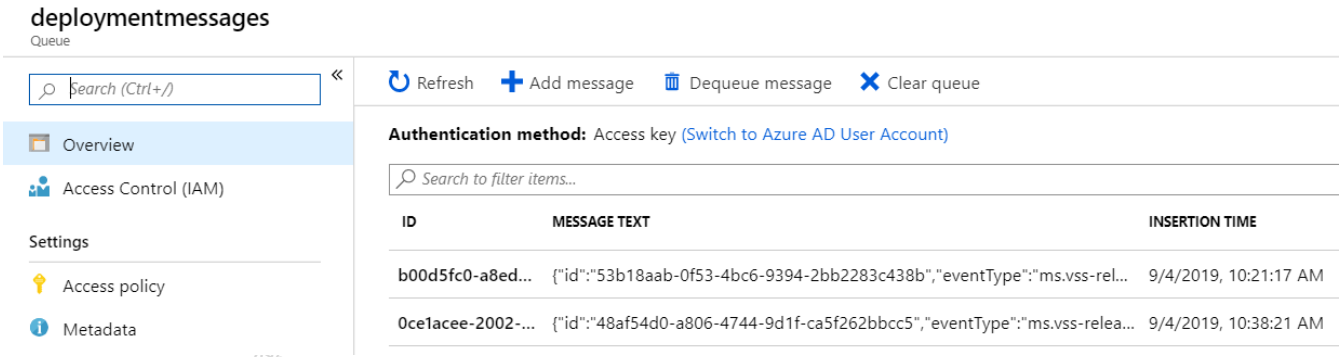


Check the queue contents

17. In the **Azure Portal**, in the blade for the storage account, click **Queues** from the **Queue service** section.



18. Click to open the **deploymentmessages** queue.



Note: if you have run multiple releases, you might have multiple messages

19. Click the latest message (usually the bottom of the list) to open it and review the message properties, then close the **Message properties** pane.

Message properties

ID

0ce1acee-2002-4950-889d-
9677518271d6

MESSAGE BODY

{"id":"48af54d0-a806-4744-9d1f-
ca5f262bbcc5","eventType":"ms.vss-
release.deployment-completed-
event","publisherId":"rm","message":
{"text":"Deployment of release
Release-4 on stage Production
succeeded.","html":"Deployment on
stage <a
href='https://greglowdevopslab.visualstudio.com/Parts%20U
_a=environment-
summary&definitionId=2&definitionEnvironmentId=7'>Proc
succeeded.","markdown":"Deployment
on stage [Production]
(https://greglowdevopslab.visualstudio.com/Parts%20Unlimi
_a=environment-
summary&definitionId=2&definitionEnvironmentId=7)
succeeded."},"detailedMessage":
{"text":"Deployment of release
Release-4 on stage Production
succeeded. Time to deploy:
00:00:26.","html":"Deployment on
stage <a

You have successfully integrated this message queue with your Azure DevOps release pipeline.