

Demonstration Setting Up Service Hooks to Monitor the Pipeline

In this demonstration, you will investigate Service Hooks.

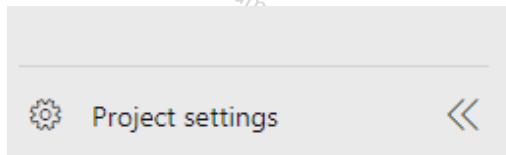
Note: Before starting this walkthrough, make sure you have performed the steps in the prerequisites section and the previous walkthroughs.

Steps

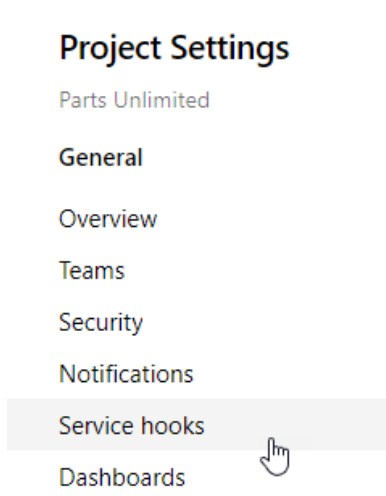
Let's now take a look at how a release pipeline can communicate with other services by using service hooks.

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**.

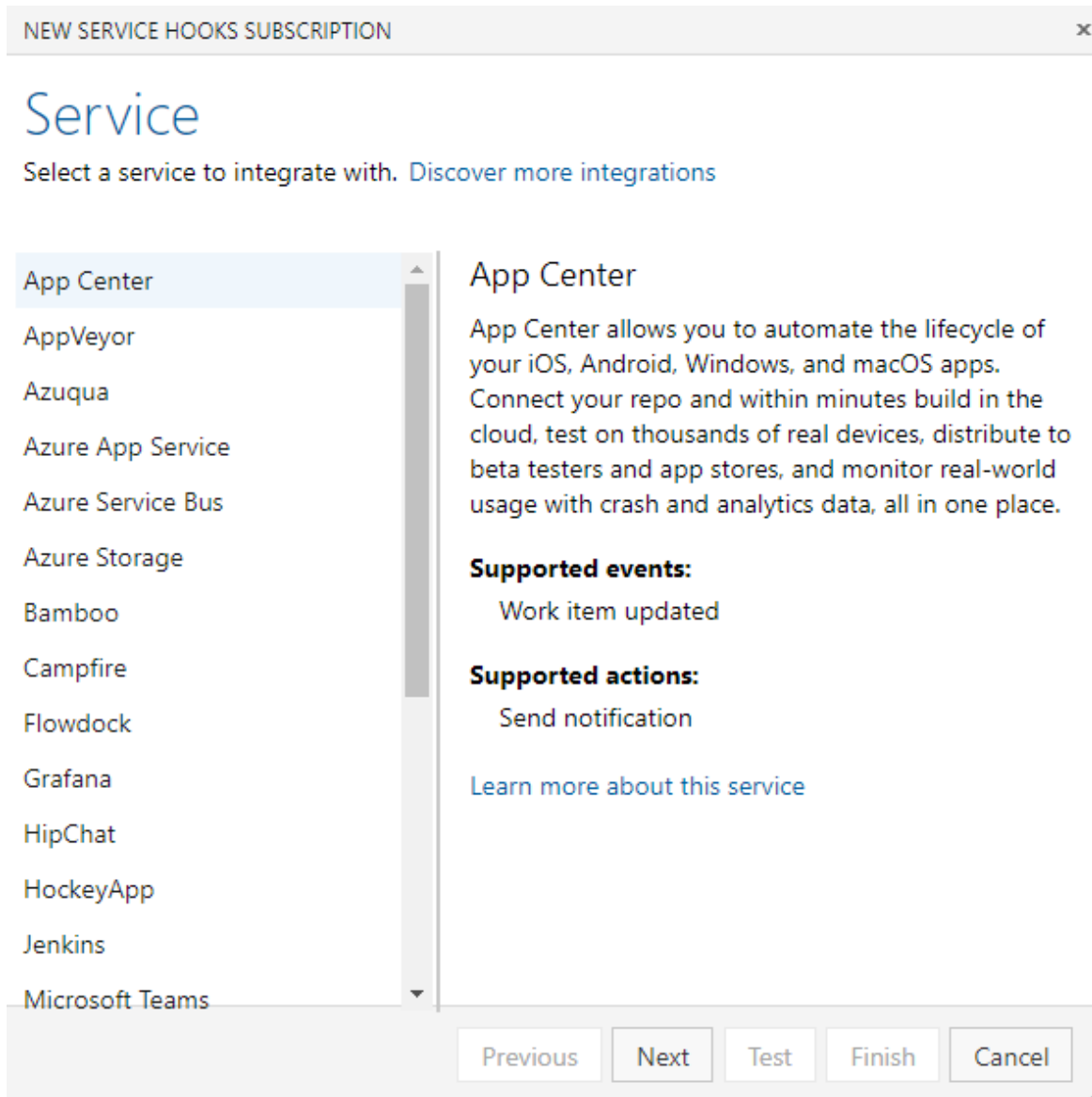


Service Hooks

Integrate with your favorite services by notifying them when events happen in your project.

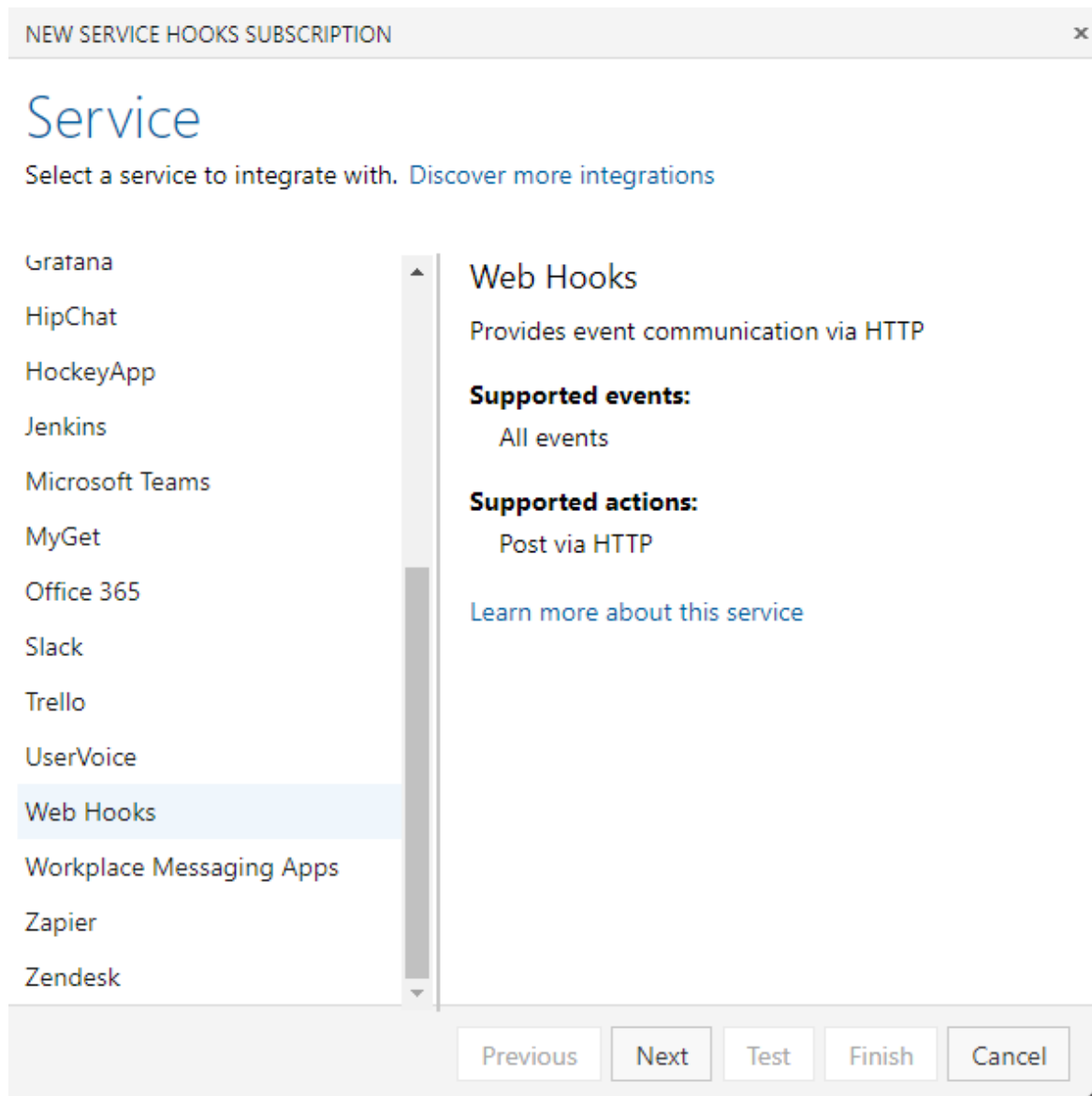
[+ Create subscription](#)

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**.

NEW SERVICE HOOKS SUBSCRIPTION ✕

Service

Select a service to integrate with. [Discover more integrations](#)

App Center

AppVeyor

Azuqua

Azure App Service

Azure Service Bus

Azure Storage

Bamboo

Campfire

Flowdock

Grafana

HipChat

HockeyApp

Jenkins

Microsoft Teams

Azure Storage

Microsoft Azure Storage is a service for storing large numbers of messages that can be accessed from anywhere in the world. It is useful for creating a backlog of work to process asynchronously.

Supported events:

All events

Supported actions:

Insert a message in a Storage Queue

[Learn more about this service](#)

Previous

Next

Test

Finish

Cancel

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.

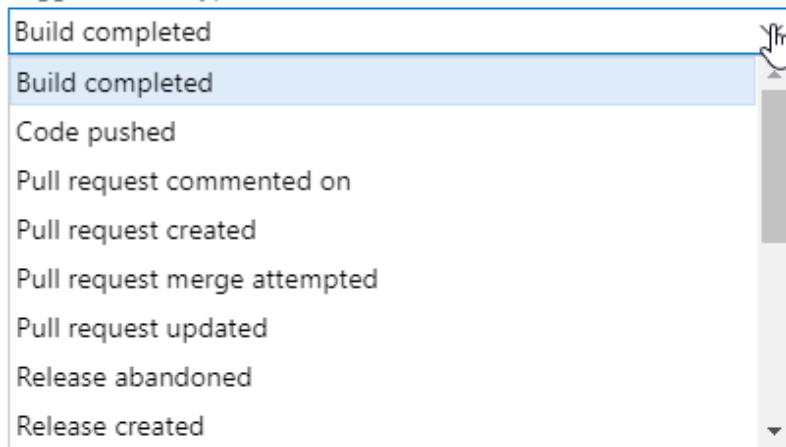
This document belongs to CHRISTOPHER M PEREZ.
cperez@academyFL.com
No unauthorized copies allowed!

This document belongs to
cpere-
No u

Trigger

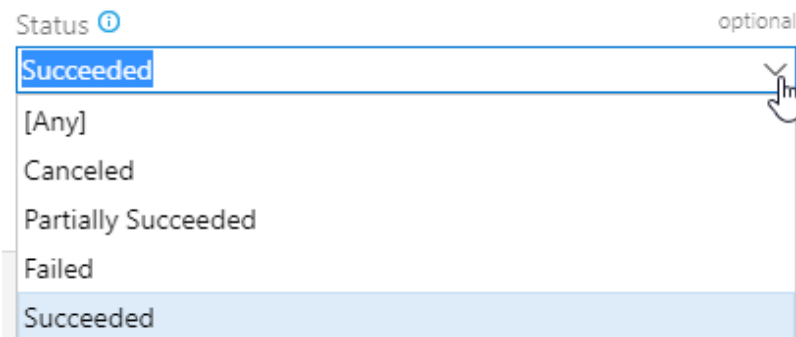
Select an event to trigger on and configure any filters.

Trigger on this type of event



- Build completed
- Build completed
- Code pushed
- Pull request commented on
- Pull request created
- Pull request merge attempted
- Pull request updated
- Release abandoned
- Release created

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.



Status ⓘ optional

- Succeeded
- [Any]
- Canceled
- Partially Succeeded
- Failed
- Succeeded

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

NEW SERVICE HOOKS SUBSCRIPTION

Trigger

Select an event to trigger on and configure any filters.

Trigger on this type of event

Release deployment completed

i Remember that selected events are visible to users of the target service, even if they don't have permission to view the related artifact.

FILTERS

Release pipeline name **i** optional

Release to all environments

Stage name **i** optional

Production

Status **i** optional

Succeeded

Previous
Next
Test
Finish
Cancel

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**.

Home > SQLDEMO > devopsoutput - Access keys

devopsoutput - Access keys
Storage account

Search (Ctrl+/)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Data transfer
Events
Storage Explorer (preview)
Settings
Access keys

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. [Learn more](#)

Storage account name
devopsoutput

key1
Key
ApXXJ9Dvs2dguErzh/VY...

Connection string
DefaultEndpointsProtocol=https;AccountName=devopsoutput;AccountKey=ApXXJ9Dvs2dguErzh/VY...

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

SETTINGS

Storage account name ? required
devopsoutput ✓

Storage account key ? required
ApxxJ9Dvs2dguErzh/vYgsbkz ✓

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

The screenshot shows the 'NEW SERVICE HOOKS SUBSCRIPTION' dialog in Azure DevOps. The 'Action' section is selected, showing 'Insert a message in a Storage Queue'. The 'SETTINGS' section is visible in the background, showing the configuration for the storage account and queue. A 'TEST NOTIFICATION' window is overlaid on the dialog, showing the results of the test.

TEST NOTIFICATION

Azure Storage (Insert a message in a Storage Queue)

Summary Request Response Event

✓ **Succeeded**

Sent at: Wednesday, 4 September 2019 12:21:17 AM

Message




Deployment on stage **Dev** Succeeded.

Previous Next Test Finish Cancel

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

Service Hooks

Integrate with your favorite services by notifying them when events happen in your project.

+    History				
Consumer ↑	Event ↑		Action	
Azure Storage	...	Release deployment c...	Release Release to all environm...	Insert a message in a S... Account

Create a release to test the service hook

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

- 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**.

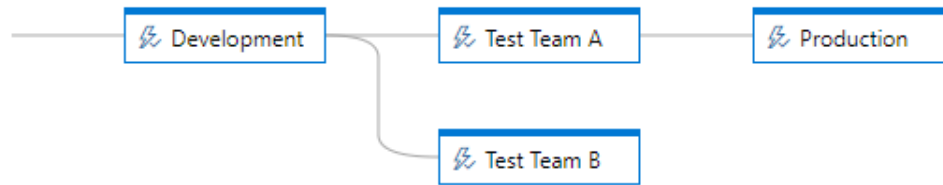
Create a new release



Release to all environments

Pipeline ^

Click on a stage to change its trigger from automated to manual.



Stages for a trigger change from automated to manual.

Artifacts ^

Select the version for the artifact sources for this release

Source alias	Version	
_Parts Unlimited-ASP.NET-CI	20190901.2	

Release description

Test the queue service hook

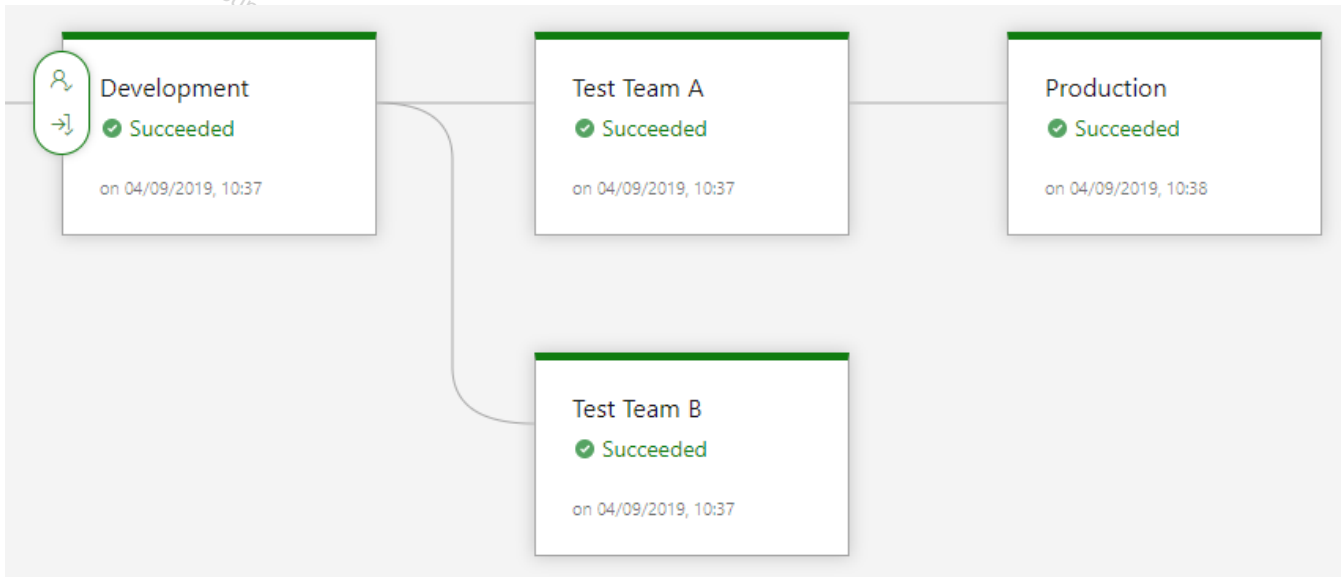
Create Cancel

15. Click to view the release details.

✓ Release [Release-4](#) has been queued

🔍 Search all pipelines

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.

QUEUE	URL
deploymentmessages	https://devopsoutput.queue.core.windows.net

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

deploymentmessages
Queue

Search (Ctrl+/)

Refresh Add message Dequeue message Clear queue

Authentication method: Access key ([Switch to Azure AD User Account](#))

Search to filter items...

ID	MESSAGE TEXT	INSERTION TIME
b00d5fc0-a8ed...	{"id":"53b18aab-0f53-4bc6-9394-2bb2283c438b","eventType":"ms.vss-rel..."}	9/4/2019, 10:21:17 AM
0ce1acee-2002-...	{"id":"48af54d0-a806-4744-9d1f-ca5f262bbcc5","eventType":"ms.vss-relea..."}	9/4/2019, 10:38:21 AM

Overview

Access Control (IAM)

Settings

Access policy

Metadata

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

- 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%20L_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 href='https://greglowdevopslab.visualstudio.com/Parts%20L_a=environment-summary&definitionId=2&definitionEnvironmentId=7'>Proc succeeded."
    }
  }
}
```

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