

# Activity 7 - Ring-based deployment

## Steps

Let's look at how a release pipeline can stage features using ring-based deployments.

When I have a new feature, I might want to release it to a few users first, just in case something goes wrong.

I could do it in authenticated systems by having those users as members of a security group and letting members of that group use the new features.

However, on a public website, I might not have logged-in users. Instead, I might want to direct a small percentage of the traffic to use the new features.

Let's see how that's configured.

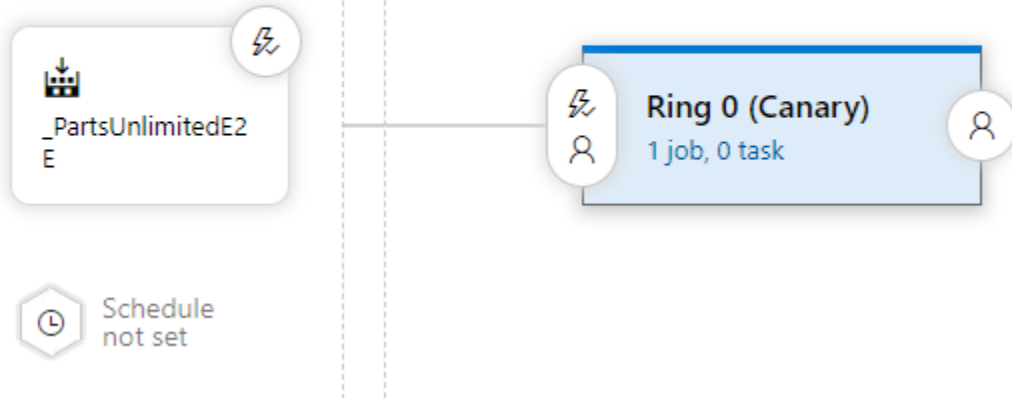
We'll create a new release pipeline that isn't triggered by code changes but manually when we slowly release a new feature.

We start by assuming that a new feature has already been deployed to the Green site (the staging slot).

1. In the main menu for the **Parts Unlimited** project, click **Pipelines**, then click **Release**, click **+New**, then click **New release pipeline**.
2. When prompted to select a template, click **Empty job** from the top of the pane.
3. Click on the **Stage 1** stage and rename it to **Ring 0 (Canary)**.

Artifacts | + Add

Stages | + Add ▾




4. Hover over the **New release pipeline** name at the top of the page, and when a pencil appears, click it, and change the pipeline name to **Ring-based Deployment**.

All pipelines > ⚙ Ring-based Deployment

Pipeline Tasks ▾ Variables Retention Options History

5. Select the **Ring 0 (Canary)** stage from the Tasks drop-down list. Click the + to add a new task, and from the list of tasks, hover over **Azure CLI** when the **Add** button appears, click it, then click to select the **Azure CLI** task in the task list for the stage.

Azure CLI ⓘ

 View YAML  Remove

Task version 1.\* ▼

Display name \*

Azure CLI

Azure subscription \* ⓘ | [Manage](#)



ⓘ This setting is required.

Script Location \* ⓘ

Script path



Script Path \* ⓘ




This setting is required.

6. In the **Azure CLI** settings pane, select your **Azure subscription**, set **Script Location** to **Inline script**, set the **Inline Script** to the following, then click **Save** and **OK**.

```
az webapp traffic-routing set --resource-group $(ResourceGroupName) --name $(WebsiteName) --distribution staging=10
```

This distribution will cause 10% of the web traffic to be sent to the new feature Site (currently the staging slot).

7. From the menu above the task list, click **Variables**. Create two new variables as shown. (Make sure to use your correct website name).

Name	Value		Scope
ResourceGroupName	ASPDOTNET		Release
WebsiteName	pule2eb3100890		Release

8. From the menu above the variables, click **Pipeline** to return to editing the pipeline. Hover over the **Ring 0 (Canary)** stage and click the **Clone** icon when it appears. Select the new stage and rename it to **Ring 1 (Early Adopters)**.

Stages | + Add ▾



9. Select the **Ring 1 (Early Adopters)** stage from the Tasks drop-down list and select the **Azure CLI** task. Modify the script by changing the value from **10** to **30** to cause 30% of the traffic to go to the new feature site.

Inline Script \* ⓘ

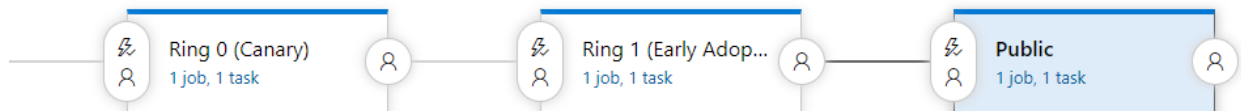
```
az webapp traffic-routing set --resource-group $(ResourceGroupName) --name $(WebsiteName) --distribution staging=30
```

```
az webapp traffic-routing set --resource-group $(ResourceGroupName) --name $(WebsiteName) --distribution staging=30
```

It allows us to move the new feature into broader distribution if it works ok in smaller users.

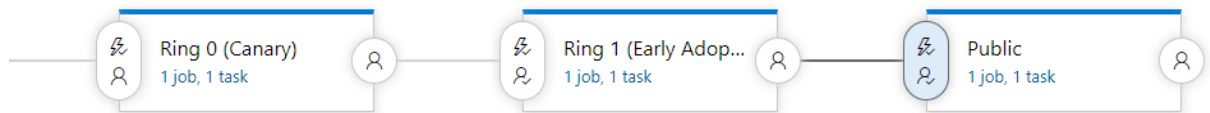
10. From the menu above the tasks, click **Pipeline** to return to editing the release pipeline. Hover over the **Ring 1 (Early Adopters)** stage and when the **Clone** icon appears, click it. Click to select the new stage and rename it to **Public**. Click **Save** and **OK**.

Stages | + Add ▾



11. Click the **Pre-deployment conditions** icon for the **Ring 1 (Early Adopters)** stage and add yourself as a pre-deployment approver. Do the same for the **Public** stage—Click **Save** and **OK**.

Stages | + Add ▾



The first step in releasing the new code to the public is to swap the new feature site (that is, the staging site) with the production so that production is now running the new code.

12. From the **Tasks** drop-down list, select the **Public** stage. Select the **Azure CLI** task, change the **Display name** to **Swap sites** and change the **Inline Script** to the following command:

```
az webapp deployment slot swap -g $(ResourceGroupName) -n $(WebsiteName) --slot staging --target-slot production
```

Inline Script \* ⓘ

```
az webapp deployment slot swap -g $(ResourceGroupName) -n $(WebsiteName) --slot staging --target-slot production
```

```
az webapp deployment slot swap -g $(ResourceGroupName) -n $(WebsiteName) --slot staging --target-slot production
```

Next, we need to remove any traffic from the staging site.

13. Right-click the **Swap sites** task and click **Clone task(s)**. Select the **Swap sites copy** task, change its **Display name** to **Stop staging traffic**, and set the **Inline Script** to the following:

```
az webapp traffic-routing set --resource-group $(ResourceGroupName) --name $(WebsiteName) --distribution staging=0
```

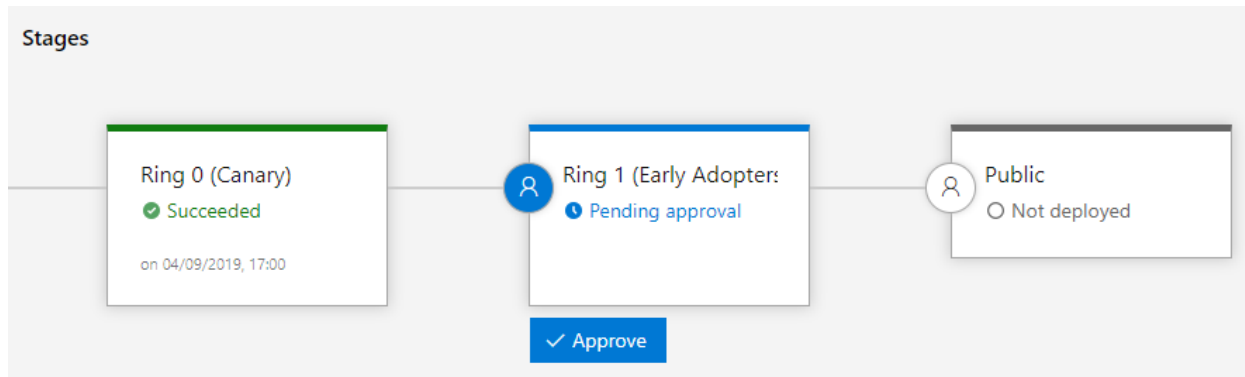
1. Click **Save**, then **OK** to save your work.
2. Click **Create release** and **Create** to start a release. Click the release link to see the progress.

## All pipelines > Ring-based Deployment

Release [Release-1](#) has been created

Pipeline Tasks Variables Retention Options History

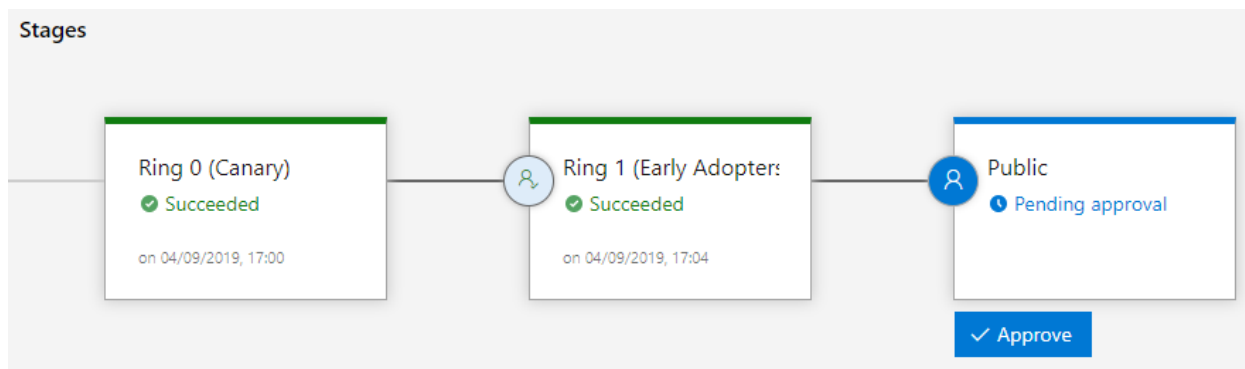
Wait until Ring 0 (Canary) has succeeded.



Currently, 10% of the traffic will go to the new feature site.

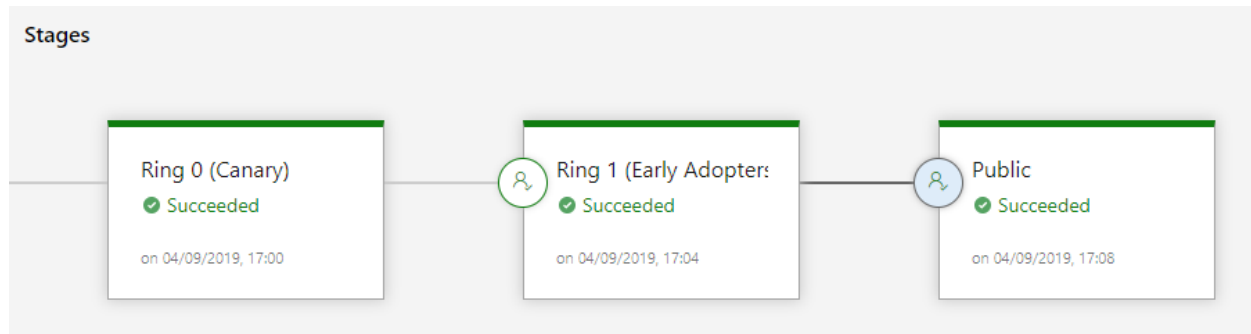
- Click **Approve** on the **Ring 1 (Early Adopters)** stage, and then **Approve**.

When this stage completes, 30% of the traffic will go to the early adopters in ring 1.



- Click **Approve** on the **Public** stage, and then **Approve**.

When this stage completes, all the traffic will go to the swapped production site, running the new code.



The new feature has been fully deployed.

## Activity 8 - Integration between Azure DevOps and Teams

### Lab requirements

- Microsoft Teams. This will be installed as part of prerequisites for this lab.
- Set up a Microsoft 365 subscription.

### Objectives

After you complete this lab, you will be able to:

- Integrate Microsoft Teams with Azure DevOps.
- Integrate Azure DevOps Kanban boards and Dashboards in Teams.
- Integrate Azure Pipelines with Microsoft Teams.
- Install the Azure Pipelines app in Microsoft Teams.
- Subscribe for Azure Pipelines notifications.

Estimated timing: 60 minutes

### Instructions

#### Exercise 0: Configure the lab prerequisites

In this exercise, you will set up the prerequisites for the lab, which consist of the preconfigured **Tailwind Traders** team project based on an Azure DevOps Demo Generator template and a team created in Microsoft Teams.

### Task 1: Configure the team project

In this task, you will use Azure DevOps Demo Generator to generate a new project based on the **Tailwind Traders** template.

1. On your lab computer, start a web browser and navigate to [Azure DevOps Demo Generator](#). This utility site will automate the process of creating a new Azure DevOps project within your account that is prepopulated with content (work items, repos, etc.) required for the lab.
2. Click **Sign in** and sign in using the Microsoft account associated with your Azure DevOps subscription.
3. If required, on the **Azure DevOps Demo Generator** page, click **Accept** to accept the permission requests for accessing your Azure DevOps subscription.
4. On the **Create New Project** page, click **Choose template**.
5. In the list of templates, in the toolbar, click **General**, select **Tailwind Traders** and click **Select Template**.
6. Back on the **Create New Project** page, in the **New Project Name** textbox, type **Tailwind Traders** and, in the **Select organization** dropdown list, select your Azure DevOps organization.
7. On the **Create New Project** page, if prompted to install a missing extension, select the checkbox below the **ARM Outputs** and click **Create Project** (Ignore GitHub forking).

**Note:** Wait for the process to complete. This should take about 2 minutes. In case the process fails, navigate to your DevOps organization, delete the project, and try again.

8. On the **Create New Project** page, click **Navigate to project**.

### Task 2: Create a team in Microsoft Teams

In this task, you will create a team in Microsoft Teams.

1. On the lab computer, start a web browser, navigate to the [Microsoft Teams download page](#), and, from there, download and install Microsoft Teams with the default settings.
2. On the lab computer, launch **Microsoft Teams** by using the desktop app.



**Note:** Alternatively, you can use a web browser and navigate to the [Microsoft Teams launch page](#)

3. When prompted to sign in, sign in with a user account that is part of the Microsoft 365 subscription and has access to your Azure DevOps organization.
4. In Microsoft Teams, in the toolbar on the left side of the page, click **Teams** and then, at the bottom of the teams list, click **Join or create a team**.
5. On the **Join or create team** pane, click **Create team**.
6. On the **Create a team** panel, click **\*From scratch\*** and, on the **What kind of team will this be?** panel, click **Private**
7. On the **Some quick details about your private team** panel, replace **Give your team a name** with **Tailwind Traders** and click on **Create**.
8. On the **Add members to Tailwind Traders** panel, click **Skip**.

## Exercise 1: Integrate Azure Boards with Microsoft Teams

In this exercise, you will implement integration between Azure Boards and Microsoft Teams.

### Task 1: Install and configure Azure Boards app in Microsoft Teams

In this task, you will install and configure Azure Boards app in the newly created team in Microsoft Teams.

1. In the Microsoft Teams window, in the lower left corner, click the **Apps** icon. This will open the **Apps** pane.
2. On the **Apps** pane, in the **Search all apps** textbox, type **Azure Boards** and, in the list of apps, click **Azure Boards**. Click the down facing arrowhead directly to the right of the **Open** button and select the **Add to a team** entry in the dropdown.
3. On the **Set up Azure boards for a team** panel, in the **Search** text box, type **Tailwind Traders**. Select the **Tailwind Traders > General** entry from the search result, and click **Set up a bot**.
4. From the list of posts in the **General** channel of the **Tailwind Traders** team, select the post titled **Azure Boards** and press the **Enter** key to review additional messages posted by the bot, as below:

Here are some of the things you can do:  
[link \[project url\]](#) - [Link to a project to create work items](#) and receive notifications

subscriptions - Add or remove subscriptions for this channel  
addAreapath [area path] - Add an area path from your project to this channel  
signin - Sign in to your Azure Boards account  
signout - Sign out from your Azure Boards account  
unlink - Unlink a project from this channel  
feedback - Report a problem or suggest a feature  
To know more see documentation.

5. Open a **New conversation** and type @ and as you type **Azure**, you will be prompted with results, select **Azure Boards** followed by **signin** in the command panel. Follow the steps to make sure you have access to the Azure DevOps organizations.
6. Copy the url of your Azure DevOps **Tailwind Traders** project.  
Example: <https://dev.azure.com/myorg/myproject> . Open **New conversation** in the Teams channel and type @ and as you type **Azure**, you will be prompted with results, select **Azure Boards** followed by **link** in the command panel. At this point, paste your project url that you had copied. Your post should look like, '**Azure Boards** link <https://dev.azure.com/myorg/myproject> '. Press **Enter**.

Review the message received:

NAME has linked this channel to project Tailwind Traders. Create work items using @Azure Boards create.  
To monitor work items, please add subscription  
Add subscription

7. Click on **Add subscription**. From the dropdown, select **works item created** from the list of events and click **Next**. Leave defaults and click **Submit**. click **OK** and close. A message will be posted in the channel with details about the newly added subscription.
8. Switch to the web browser displaying the **Tailwind Traders** project in the Azure DevOps portal, click on **Boards > Work items**. Click on **New work item** and choose **User Story** on the dropdown. Give a title to the user story, something like **Test Azure Boards integration with Teams** and **Save**. The Teams channel we recently set up will post a notification/card with information about the created user story.

## Task 2: Add Azure Boards Kanban boards to Microsoft Teams

In this task, you will add Azure Boards Kanban boards to tabs in Microsoft Teams.

**Note:** Kanban board turns your backlog into an interactive signboard, providing a visual flow of work. As work progresses from idea to completion, you update the items on the board. Each column represents a

work stage, and each card represents a user story (blue cards) or a bug (red cards) at that stage of work. You can bring in your teams kanban board or favorite dashboard directly into Microsoft Teams by using Tabs. **Tabs** allow team members to access your service on a dedicated canvas, within a channel or in user's personal app space. You can leverage your existing web app to create a great tab experience within Teams.

1. On your lab computer, switch to the web browser displaying the **Tailwind Traders** project in the Azure DevOps portal, in the vertical menu bar at the far left of the Azure DevOps portal, click **Boards**, and, in the **Boards** section, click **Boards**.
2. On the **Boards** pane, in the **My team boards (1)** section, click **Tailwind Traders Team boards** entry.
3. While on the **Tailwind Traders Team** pane, in the web browser window, copy its URL into Clipboard.
4. Switch to the Microsoft Teams window, ensure that the **General** channel of the newly created team **Tailwind Traders** is selected, and, in the upper section of the **General** pane, click the plus sign, next to **Posts, Files and Wiki tabs**. This will display the **Add a tab** panel.
5. On the **Add a tab** panel, click **Website**, on the **Website** panel, set **Tab name** to **Tailwind Traders Team boards**, set the **URL** to the URL you just copied into Clipboards, and then click **Save**.
6. In the Microsoft Teams window, with the **General** channel of the **Tailwind Traders** team selected, in the list of tabs in the top menu, click the newly added **Tailwind Traders Team boards** tab and ensure that it contains the content matching the **Tailwind Traders Team** board available in the Azure DevOps portal (you may need to Sign in).

**Note:** All the work can be monitored during the daily standup's and the updates are reflected in real-time, whenever the corresponding work items states change. You also have the option to modify the Kanban board from Microsoft Teams.

## Exercise 2: Integrate Azure Pipelines with Microsoft Teams

In this exercise, you will implement integration between Azure Pipelines and Microsoft Teams.

### Task 1: Install and configure Azure Pipelines app in Microsoft Teams

In this task, you will install and configure Azure Pipelines app in the designated team in Microsoft Teams.

**Note:** Azure Pipelines app on Microsoft Teams enables you to monitor the events for your pipelines. You can set up and manage subscriptions for such events as releases, pending approvals, and completed builds, with notifications being posted directly into your Teams channel. You can also approve releases from within the Teams channel.

1. Go to Microsoft Teams window, in the lower left corner, click the **Apps** icon. This will open the **Apps** pane.
2. On the **Apps** pane, in the **Search all apps** textbox, type **Azure Pipelines** and, in the list of apps, click **Azure Pipelines**.
3. On the **Azure Pipelines** panel, click the down facing arrowhead directly to the right of the **Open** button and, in the dropdown list, click the **Add to a team** entry.
4. On the **Set up Azure Pipelines for a team** panel, in the **Search** text box, type **Tailwind Traders**, in the list of results, select the **Tailwind Traders > General** entry, and click **Set up a bot**. You will be redirected automatically to the post view in the **General** channel of the **Tailwind Traders** team.
5. In the list of posts in the **General** channel of the **Tailwind Traders** team, open **New conversation** and type **@** and as you type **Azure**, you will be prompted with results, select **Azure Pipelines** and press **Enter** key to review additional messages posted by the bot, such as below:

Here are some of the things you can do:  
subscribe [pipeline url/ project url] - Subscribe to a pipeline or all pipelines in a project to receive notifications  
subscriptions - Add or remove subscriptions for this channel  
feedback - Report a problem or suggest a feature  
signin - Sign in to your Azure Pipelines account  
signout - Sign out from your Azure Pipelines account  
To know more see documentation.

## Task 2: Subscribe to the Azure Pipeline notifications in Microsoft Teams

In this task, you will subscribe to the Azure Pipeline notifications in Microsoft Teams

**Note:** You can use the **@Azure Pipelines** handle to start interacting with the app.

1. With the **Posts** tab selected, in the **General** channel of the **Tailwind Traders** team, type **@** and as you type **Azure**, you will be prompted with results, select **Azure Pipelines** followed by **signin** in the command panel and press **Enter**.
2. In the **Azure Pipelines Sign in** pane, click **Sign in**.
3. If prompted to grant **Service hooks (read and write)**, **Build (build an execute)**, **Release (read, write, execute, and manage)**, **Project and team**

(read), Identity picker (read), and Teams Integration permissions, click **Accept** and then click **Close**.

**Note:** Now you can run the `@azure pipelines subscribe [pipeline url]` command to subscribe to an Azure Pipeline.

- Switch to the web browser displaying the **Tailwind Traders** project in the Azure DevOps portal, in the vertical menu bar at the far left of the Azure DevOps portal, click **Pipelines**, on the **Pipelines** pane, click the **Website-CI** entry, and, while on the **Website-CI** pane, in the web browser window, copy its URL into Clipboard.

**Note:** The URL will be in the format `https://dev.azure.com/<organization_name>/Tailwind%20Traders/_build?definitionId=<number>`, where the `<organization_name>` is the placeholder representing the name of your DevOps organization.

**Note:** The pipeline URL can be to any page within your pipeline that has a `definitionId` or `buildId`/`releaseId` present in the URL.

- Back in Teams, with the **Posts** tab selected, in the **General** channel of the **Tailwind Traders** team, type **@** and as you type **Azure**, you will be prompted with results, select **Azure Pipelines** followed by **subscribe** in the command Panel. At this point, paste the pipeline URL that you had copied. Your post should look like, `@Azure Pipelines subscribe https://dev.azure.com/<organization_name>/Tailwind%20Traders/_build?definitionId=<number>` (make sure to replace the `<organization_name>` placeholder with the name of your DevOps organization). Press **Enter**.
- Wait for the confirmation that the subscription has been successfully created.

**Note:** For Build pipelines, the channel is subscribed to the **Run stage state changed** and **Run stage waiting for approval** notifications.

- Switch to the web browser displaying the **Tailwind Traders** project in the Azure DevOps portal, in the vertical menu bar at the far left of the Azure DevOps portal, click **Pipelines**, in the **Pipelines** section, click **Releases**, in the list of releases, click the **Website-CD\* entry, and, with the \*\*Website-CD** entry selected, in the web browser window, copy its URL into Clipboard.

**Note:** The URL will be in the format `https://dev.azure.com/<organization_name>/Tailwind%20Traders/_release?_a=releases&view=mine&definitionId=2`, where the `<organization_name>` is the placeholder representing the name of your DevOps organization.

- With the **Posts** tab selected, in the **General** channel of the **Tailwind Traders** team, type **@** and as you type **Azure**, you will be prompted with results, select **Azure Pipelines** followed by **subscribe** in the command Panel. At this point, paste the release URL that you had copied. Your post should look

like, `@Azure Pipelines subscribe https://dev.azure.com/<organization_name>/Tailwind%20Traders/_release?_a=releases&view=mine&definitionId=2` to subscribe to the release pipeline (make sure to replace the `<organization_name>` placeholder with the name of your DevOps organization). Press **Enter**.

**Note:** For Release pipelines, the channel is subscribed to the **Deployment started, Deployment completed** and **Deployment approval pending** notifications.

### Task 3: Using filters to customize subscriptions to Azure Pipelines in Microsoft Teams

In this task, you will use customize subscriptions to Azure Pipelines in Microsoft Teams.

**Note:** When a user subscribes to any pipeline, a few subscriptions are created by default without any filters being applied. Often, users have the need to customize these subscriptions. For example, users may want to get notified only when builds fail or when deployments are pushed to a production environment. The Azure Pipelines app supports filters to customize what you see on your channel.

**Note:** You can list and manage subscriptions by using `@Azure Pipelines subscriptions` command. This command lists all of the current subscriptions for the channel.

1. With the **Posts** tab selected, in the **General** channel of the **Tailwind Traders** team, type `@` and as you type **Azure**, you will be prompted with results, select **Azure Pipelines** followed by **subscriptions** and press **Enter**. In the reply from the **Azure Pipelines** bot, select **Add Subscription**.
2. In the **Azure Pipelines Add subscription** panel, in the **Choose event** dropdown list, ensure that **Build completed** is selected and click **Next**.
3. In the **Azure Pipelines Add subscription** panel, in the **Choose pipeline** dropdown list, ensure that **Website-CI** is selected and click **Next**.
4. In the **Azure Pipelines Add subscription** panel, in the **Build status** dropdown list, ensure that **[Any]** is selected and click **Submit**.
5. In the **Azure Pipelines Add subscription** panel, click **OK** to acknowledge the confirmation message.
6. In the **Azure Pipelines View subscriptions** panel, review the list of subscriptions and close the panel.
7. Switch to the web browser displaying the **Tailwind Traders** project in the Azure DevOps portal, in the vertical menu bar at the far left of the Azure DevOps portal, click **Pipelines**, on the Pipelines pane, click the **Website-CI** entry, and, while on the Website-CI pane, click **Run Pipeline > Run**.
8. The Teams channel will post notifications about the failed execution of the pipeline, as an expected behavior (the pipeline has missing setup).