

IDCF 黑客马拉松示例项目 TailwindTraders Web 前端项目部署文档

本文档提供部署说明，请严格按照说明进行部署。

部署前提条件

请确保你已经获取了以下信息和环境，以便完成此项目的部署

1. 微软账号：微软账号为 @outlook.com @hotmail.com @live.cn 结尾的邮件地址
2. 黑客松组委会已经将你的微软账号加入对应的 Microsoft Teams, Azure DevOps 以及 Azure 管理控制台，并给你赋予了足够的权限
3. 如果需要进行本地开发调试，请确保提前安装以下软件，您可以在 Windows/Mac/Linux 环境中安装以下软件，所有软件均提供跨平台版本，请确保通过官方渠道安装以下软件

- [Azure CLI](#)
- [PowerShell](#)
- [Helm](#)
- [Visual Studio Code](#)
- [Git](#)
- [.Net Core 2.2](#)

部署步骤

****注意：以下操作需要一个人独立完成，不可多人同时操作，否则会造成冲突****

Step 1.1 - Fork 本代码库

请在自己团队中协调，使用一名组员自己的 GitHub 账号作为小组共有环境，并将此 repo fork 到这个账号中。

Step 1.2 - 获取流水线配置文件

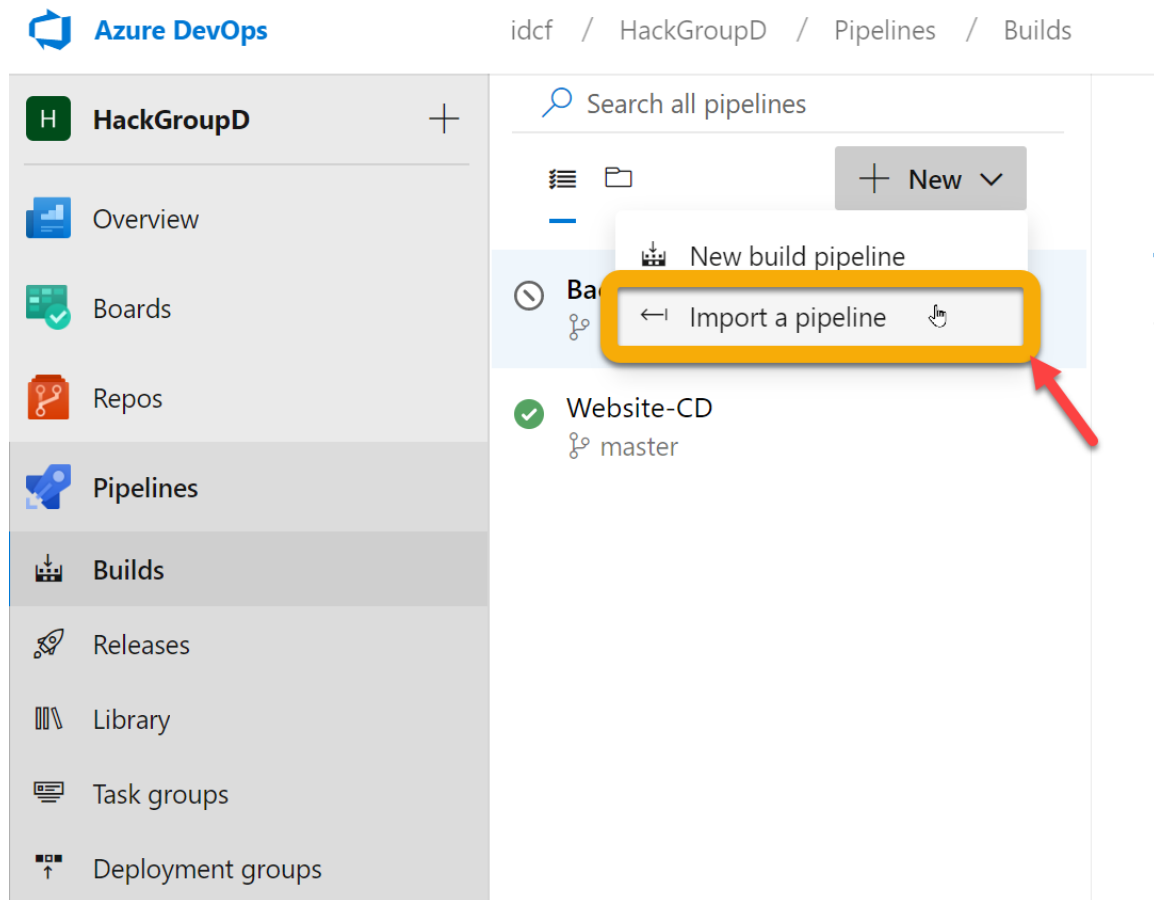
从本 repo 的 /Deploy/pipeline 目录下载以下文件

- Website-Cl.json
- Website-CD.json

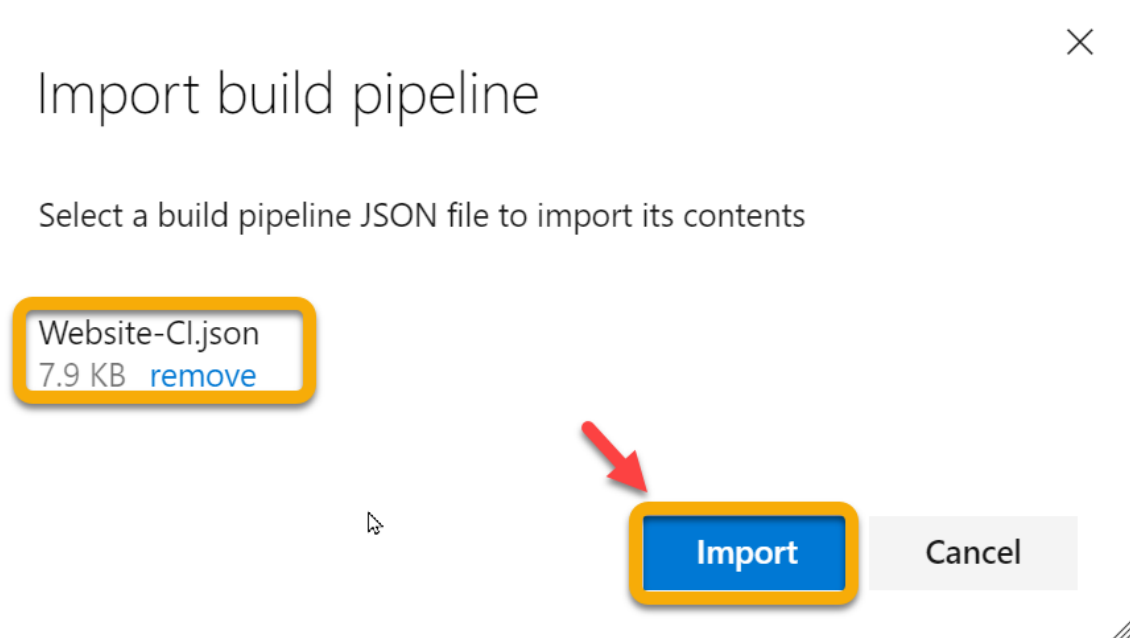
这两个文件包括全部流水线的主要配置，在后续步骤中将会用到。

Step 2.1 - 导入 Build 配置文件

使用自己的微软账号登录自己团队的 Azure DevOps 项目，进入 Pipelines | Builds 并点击 New | Import a Pipeline

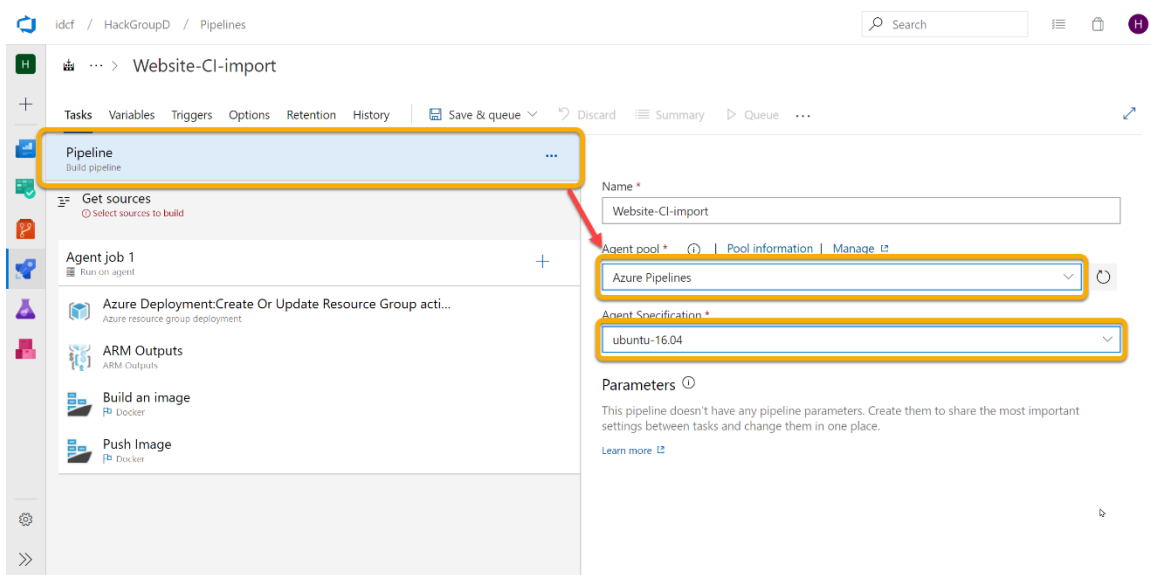


在弹出的对话框中选择 Website-Cl.json 文件，并导入

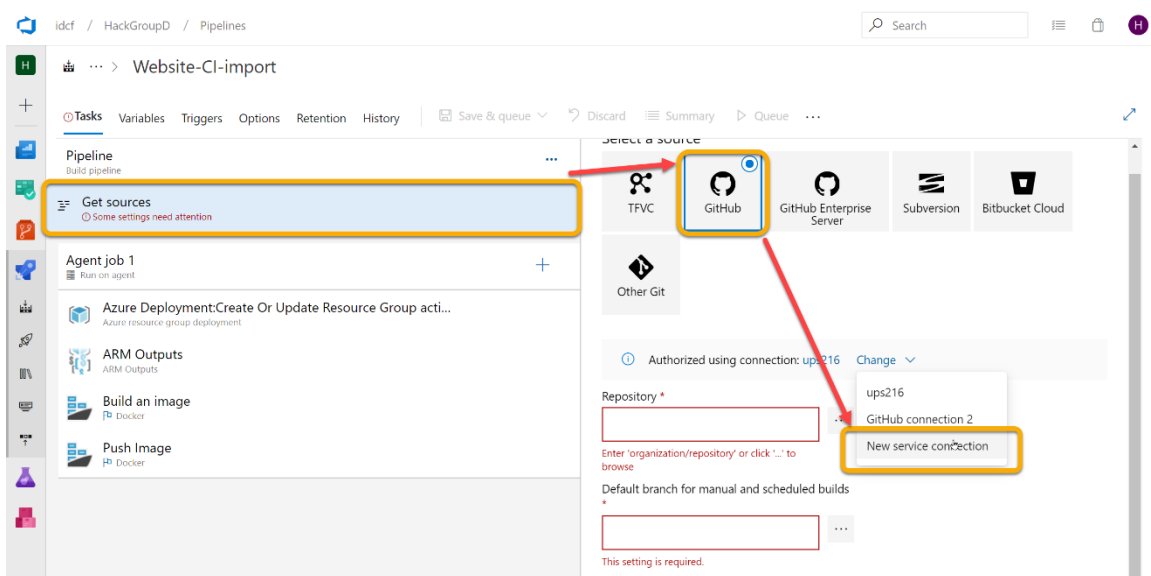


Step 2.2 - 构建代理服务和构建代码源配置

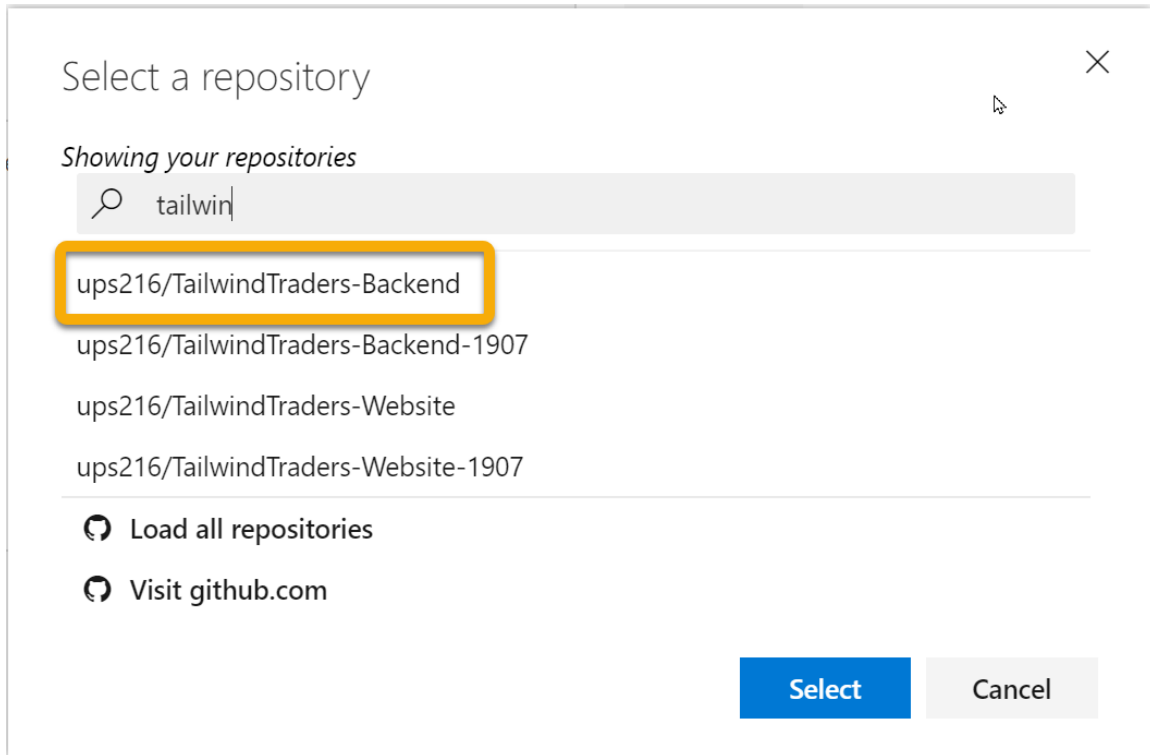
在 Pipeline 设置中选择 Azure Pipeline 作为 Agent Pool, Ubuntu 16.04 作为 Agent Specification



在 Get Sources 设置中选择 Github 作为代码源, 并点击 New Service Connection 创建一个新的指向到本组的 Github 账号的 service connection.

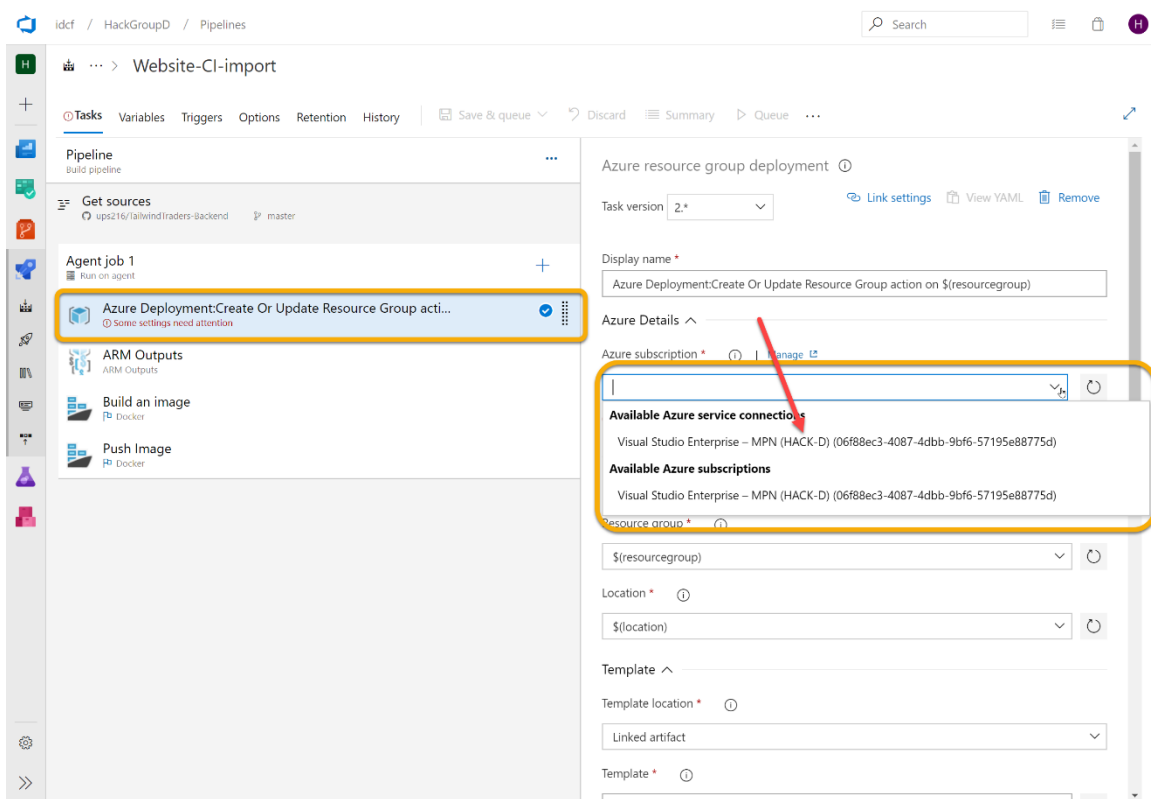


使用新创建的 service connection 获取 Github 中的 repo 列表, 并选择 TailwindTraders-Backend 作为本构建的代码源, 并选择 master 分支。

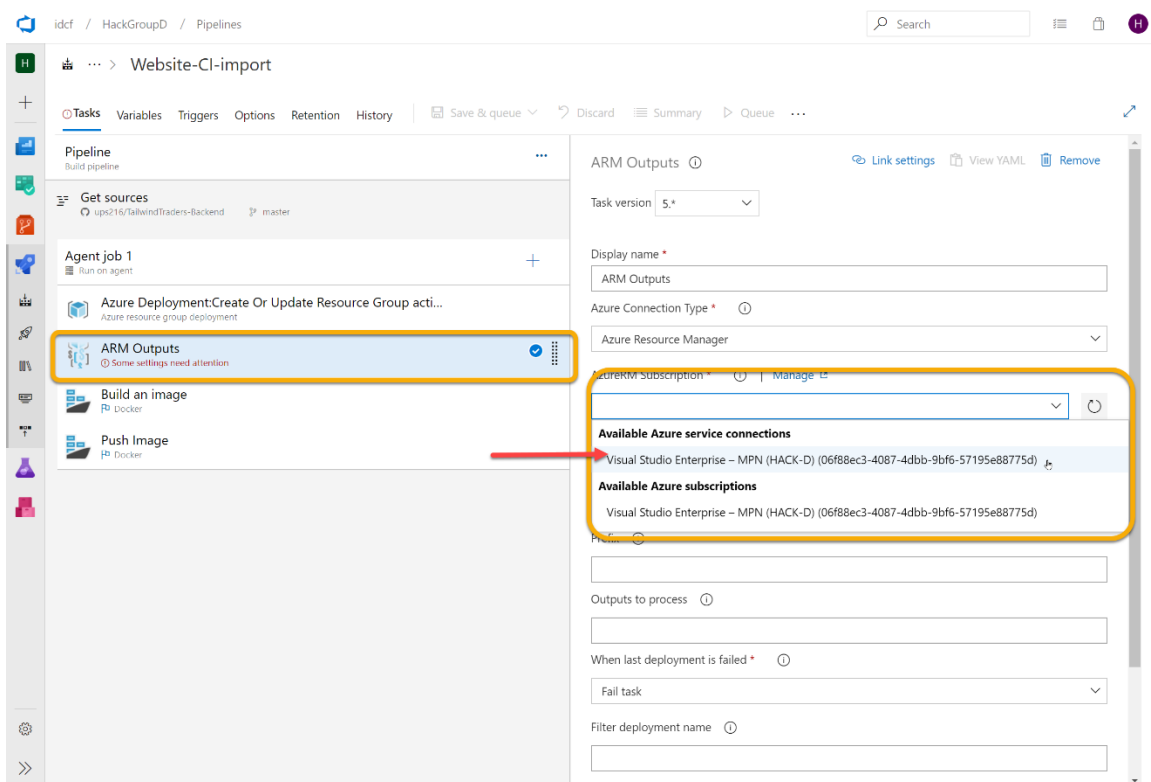


Step 2.3 - 构建步骤配置

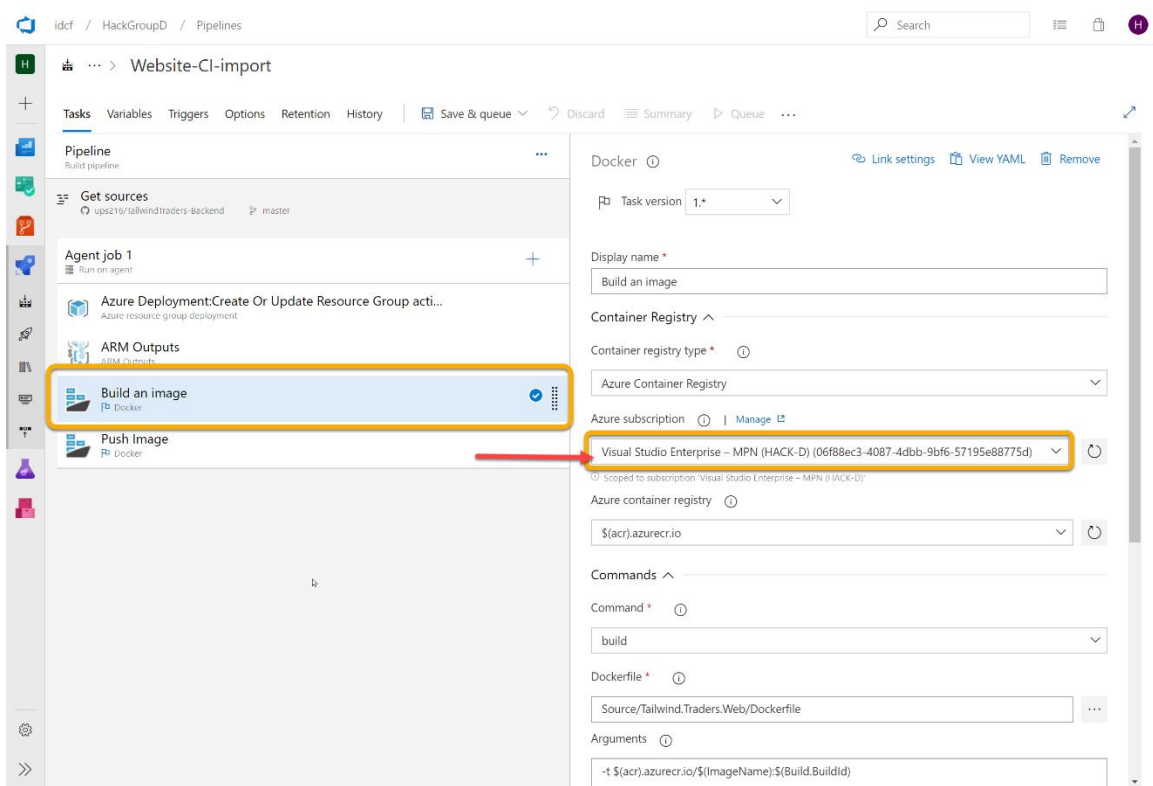
选择 Azure Deployment... 步骤，并设置 Azure Subscription 为分配给本组的订阅（注意订阅名称中包含的组别信息）



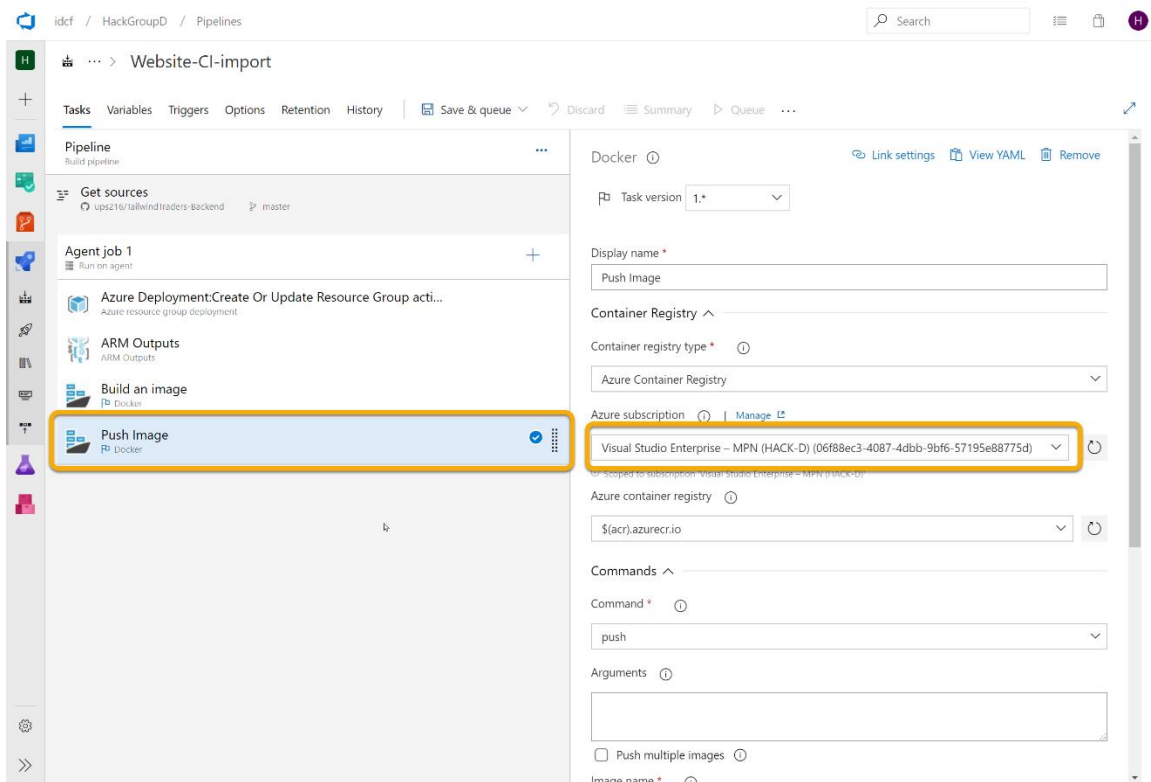
选择 ARM Outputs...步骤，并设置 Azure Subscription 为分配给本组的订阅（注意订阅名称中包含的组别信息）



选择 Build an image...步骤，并设置 Azure Subscription 为分配给本组的订阅（注意订阅名称中包含的组别信息）

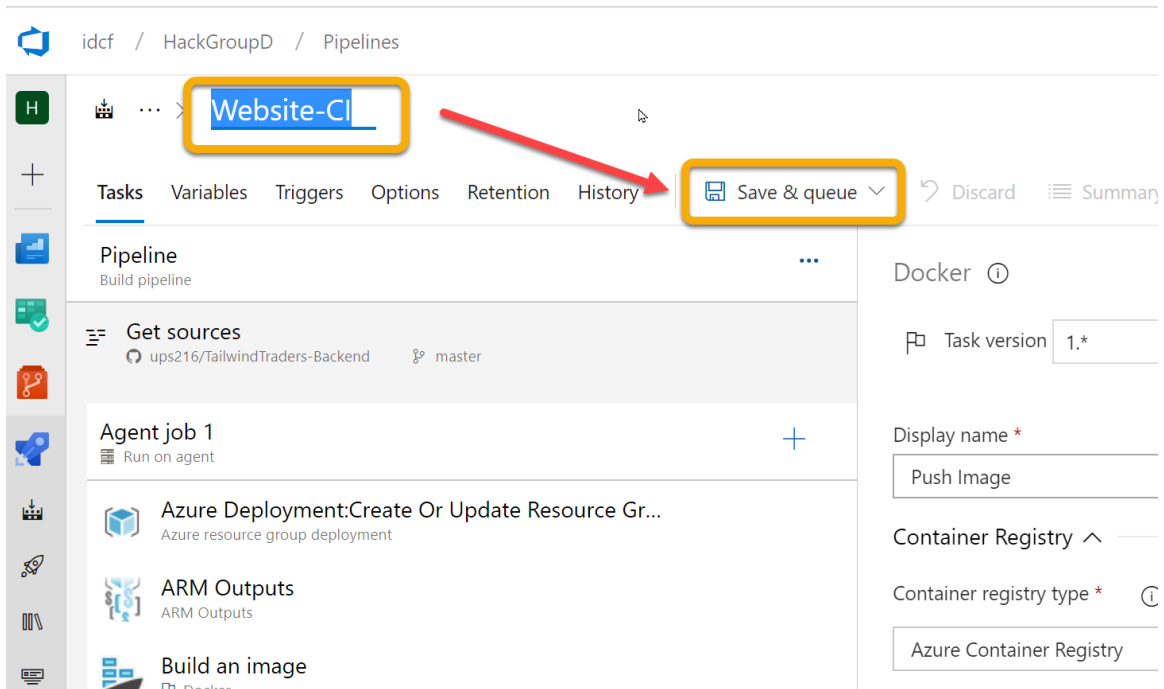


选择 Push an image...步骤，并设置 Azure Subscription 为分配给本组的订阅（注意订阅名称中包含的组别信息）

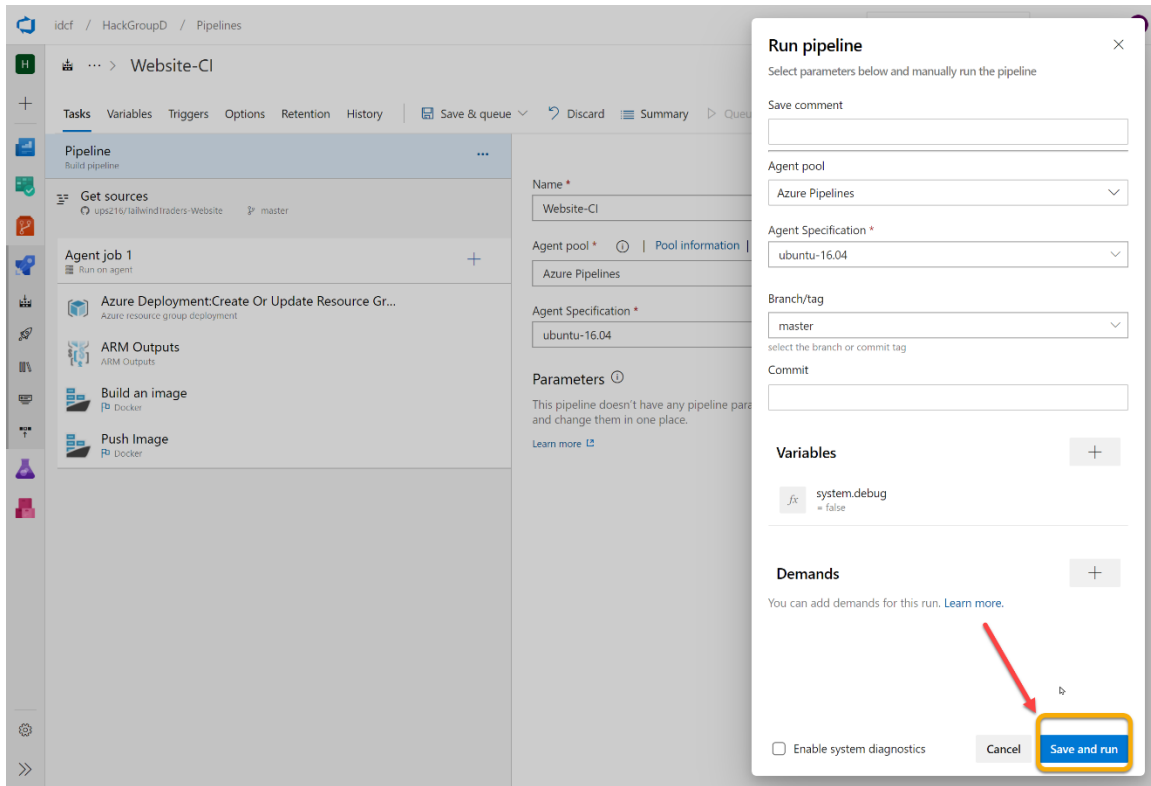


Step 2.4 - 保存构建并触发 CI

点击顶部的构建名称并修改成 Website-Cl，点击 Save & queue 按钮保存并触发构建



在弹出的对话框右下角点击 Save and run



Step 2.5 - 监控构建进度

在构建运行界面查看构建进展

idcf / HackGroupD / Pipelines / Builds / Website-CI / #8

Search

#8: revert configuration

Manually run just now by Hack Group D @ ups216/TailwindTraders-Website @ master @ 73af2fc

Cancel build

Logs Summary Tests

Agent job 1

Pool: Azure Pipelines · Agent: Azure Pipelines 2

Started: 9/5/2019, 10:58:29 PM

13s

- Initialize job · succeeded 1s
- Checkout · succeeded 6s
- Azure Deployment:Create Or Update Resource Group action on TailwindTraderWeb 4s

```
=====
Starting: Azure Deployment:Create Or Update Resource Group action on TailwindTraderWeb
=====
Task       : Azure resource group deployment
Description: Deploy an Azure Resource Manager (ARM) template to a resource group and manage virtual machines
Version    : 2.156.0
Author     : Microsoft Corporation
Help       : https://docs.microsoft.com/azure/devops/pipelines/tasks/deploy/azure-resource-group-deployment
=====
Checking if the following resource group exists: TailwindTraderWeb.
Resource group exists: true.
Creating deployment parameters.
The detected encoding for file '/home/vsts/work/1/s/Deploy/deployment.json' is 'utf-8'
Starting Deployment.
Deployment name is deployment-20190905-145841-288f
=====
```

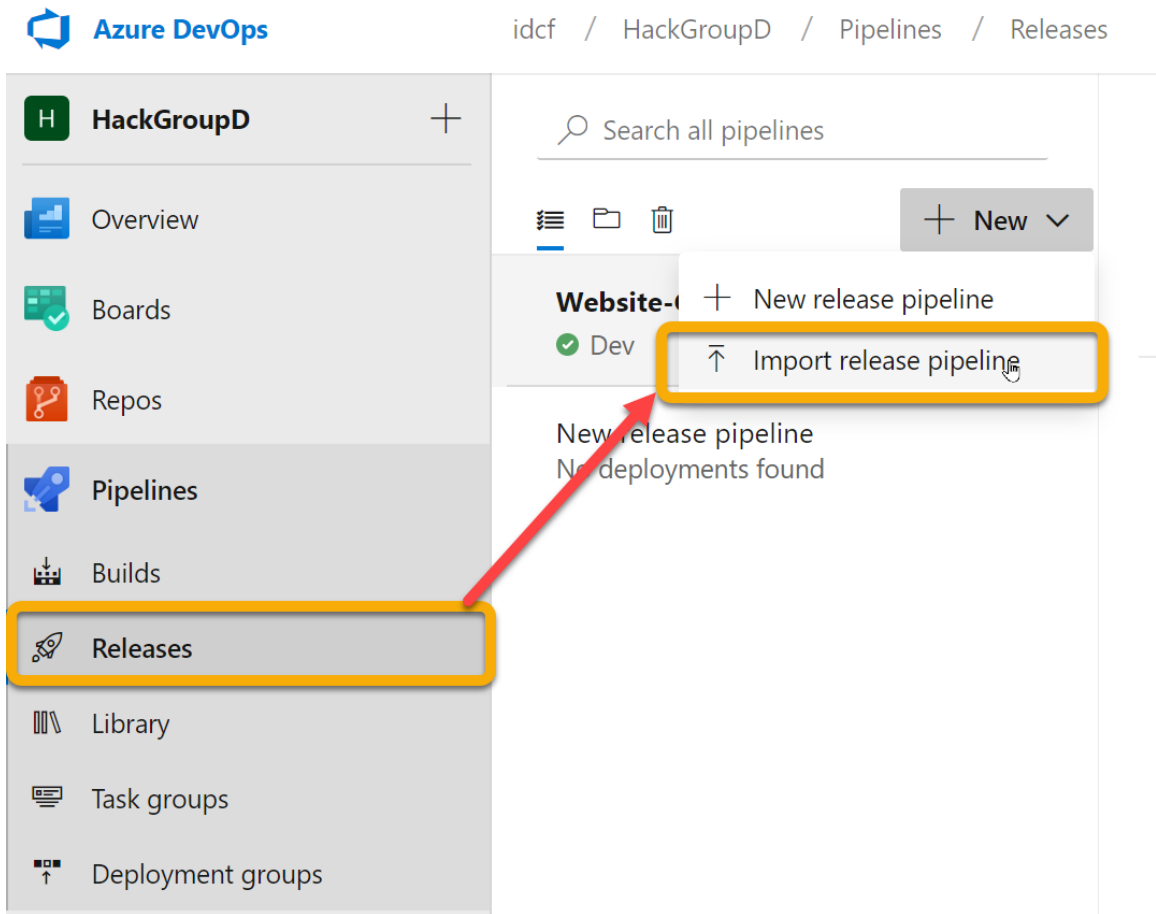
ARM Outputs · pending

Build an image · pending

以下步骤需要等待构建完成方可继续。

Step 3.1 - 导入 Release 配置

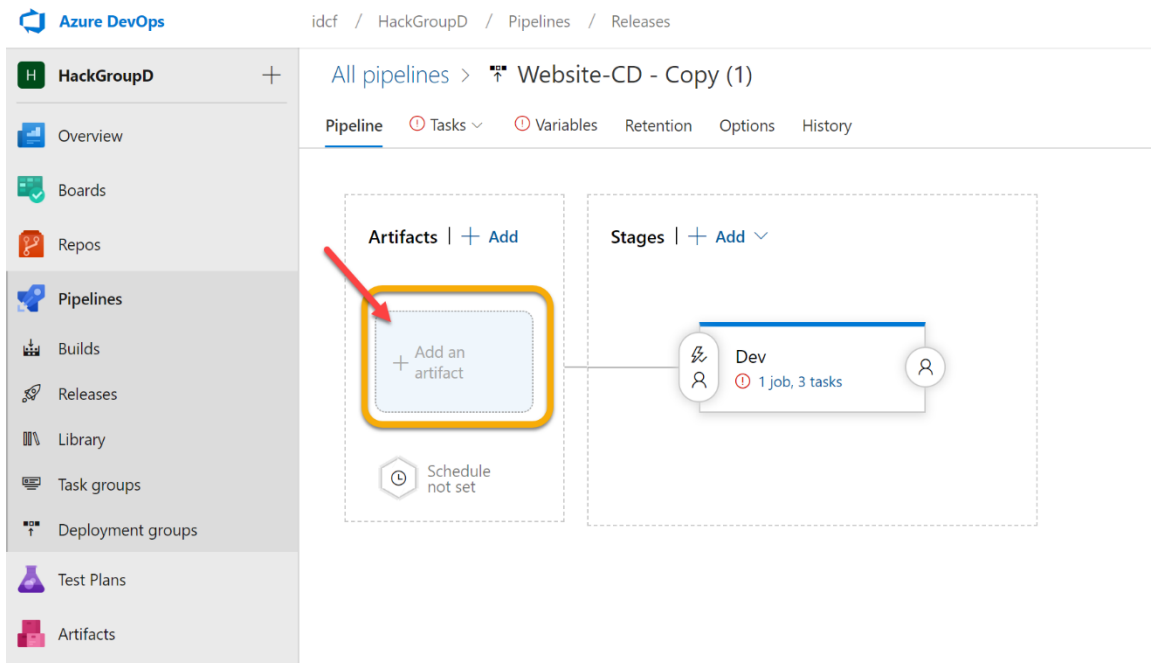
切换到 Pipeline | Release，并点击 New | Import release pipeline



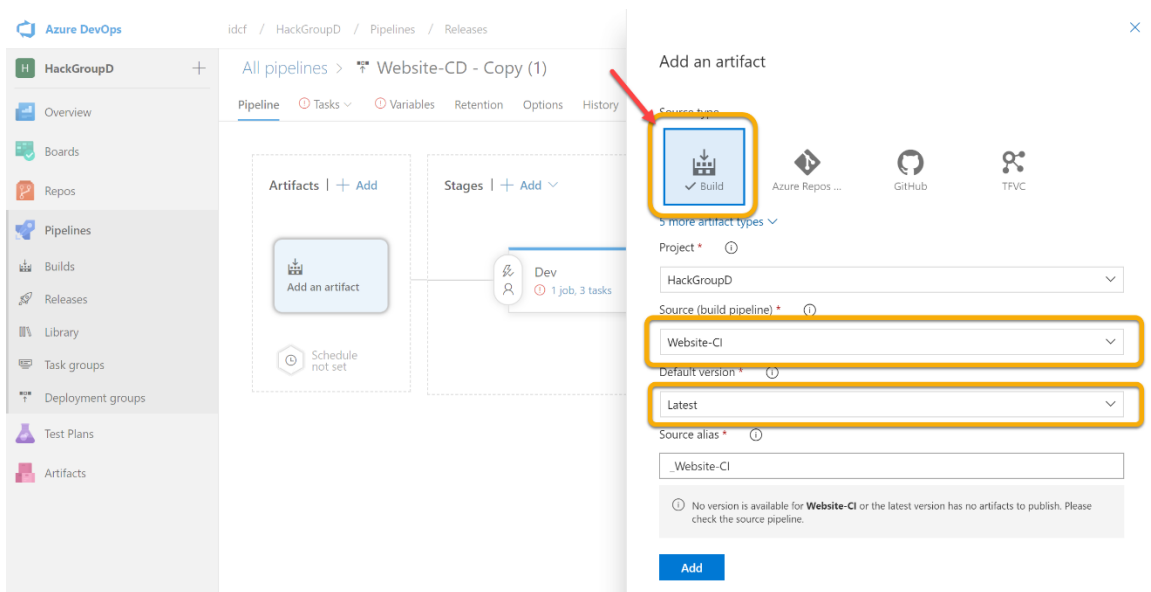
在弹出的对话框选择 Website-CD.json 并导入

Step 3.2 - 配置需要进行部署的制品(artifact)版本

点击 Add an artifact 按钮

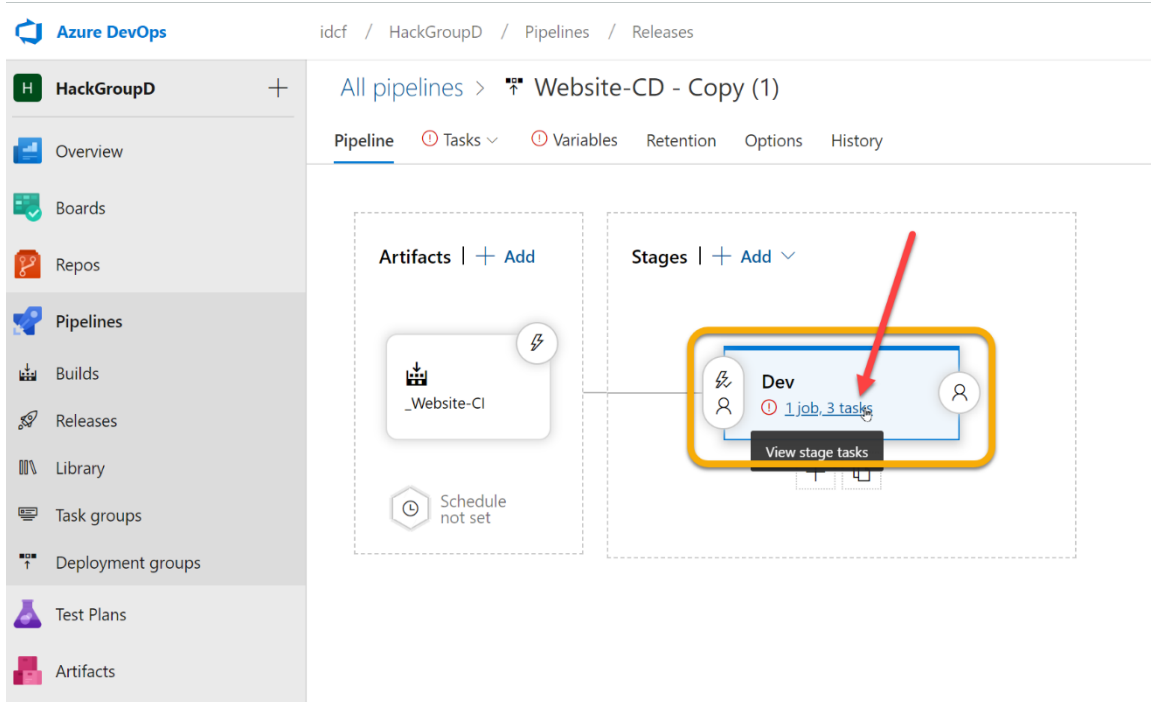


在弹出的对话框中选择以上步骤中创建的 Website-CI 构建配置作为制品来源，同时选择 latest 作为制品版本，点击 Add

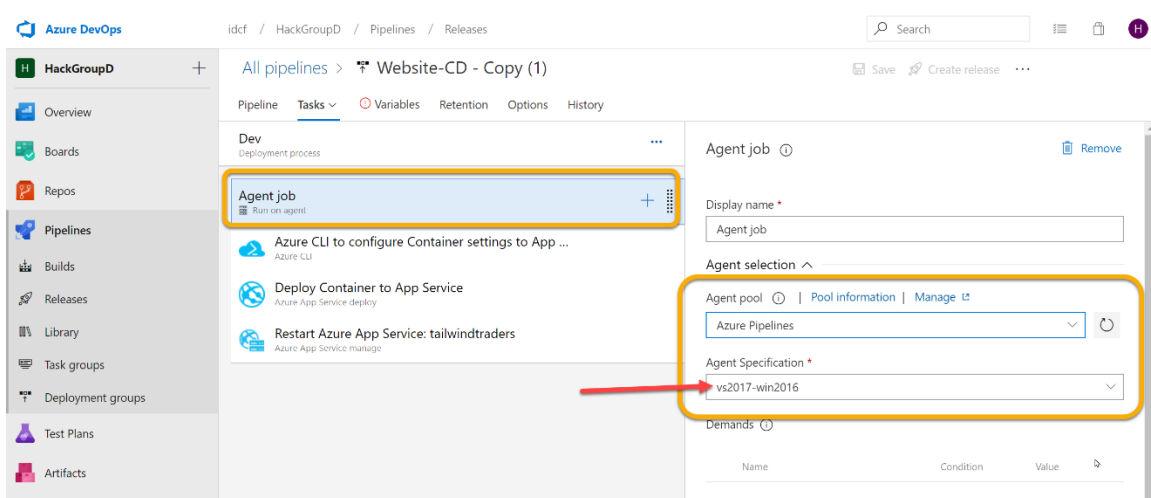


Step 3.3 - 配置部署步骤

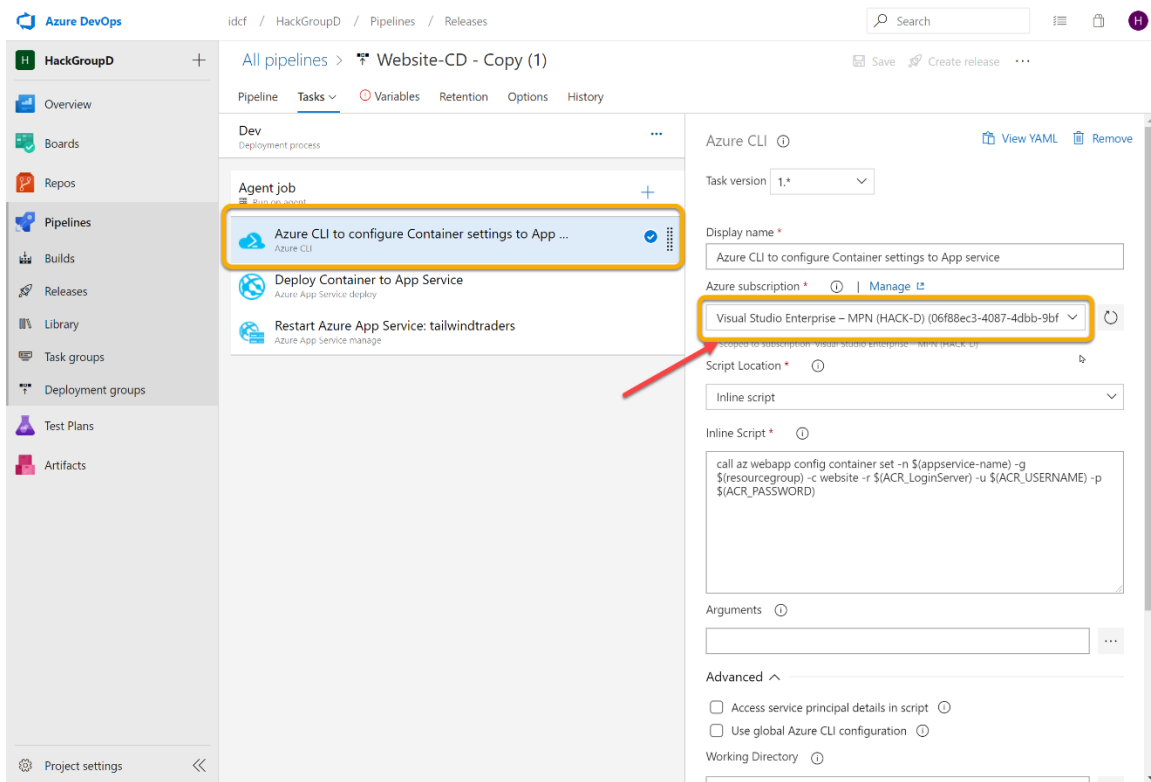
点击 Dev 阶段中的告警信息进入此阶段的部署步骤配置



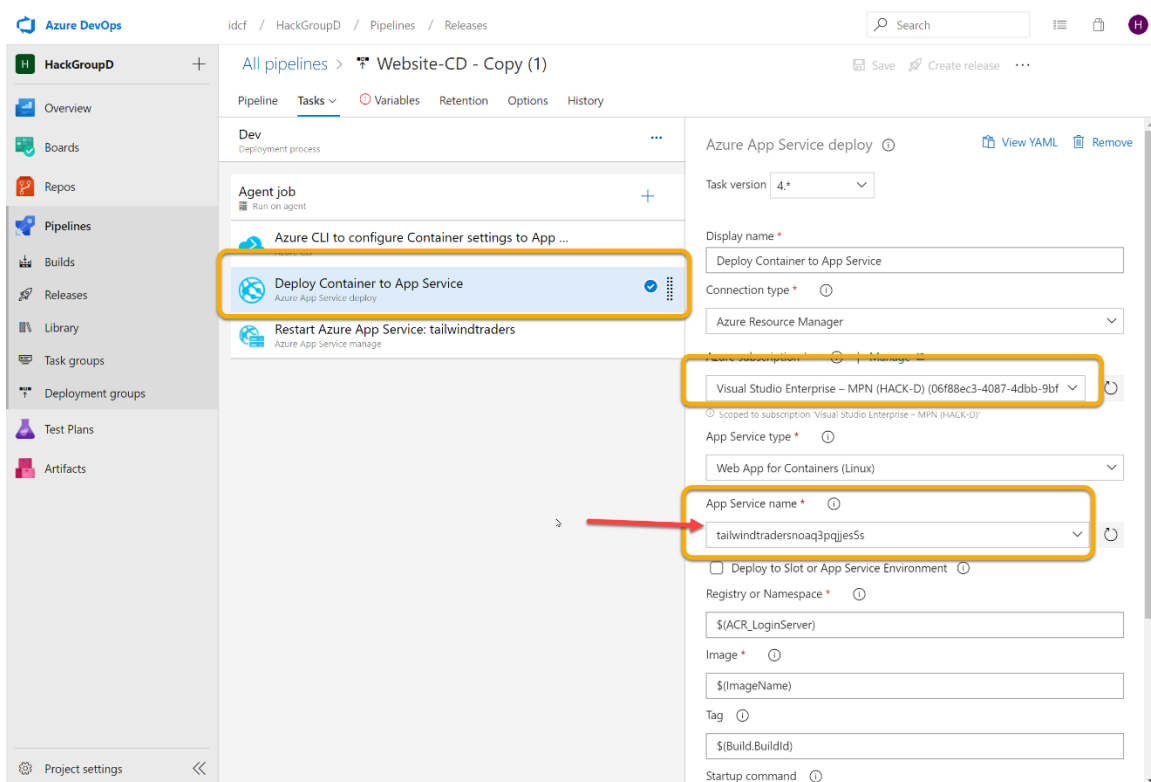
选择 Azure Pipeline | vs2017-win2016 作为 Agent Job 配置



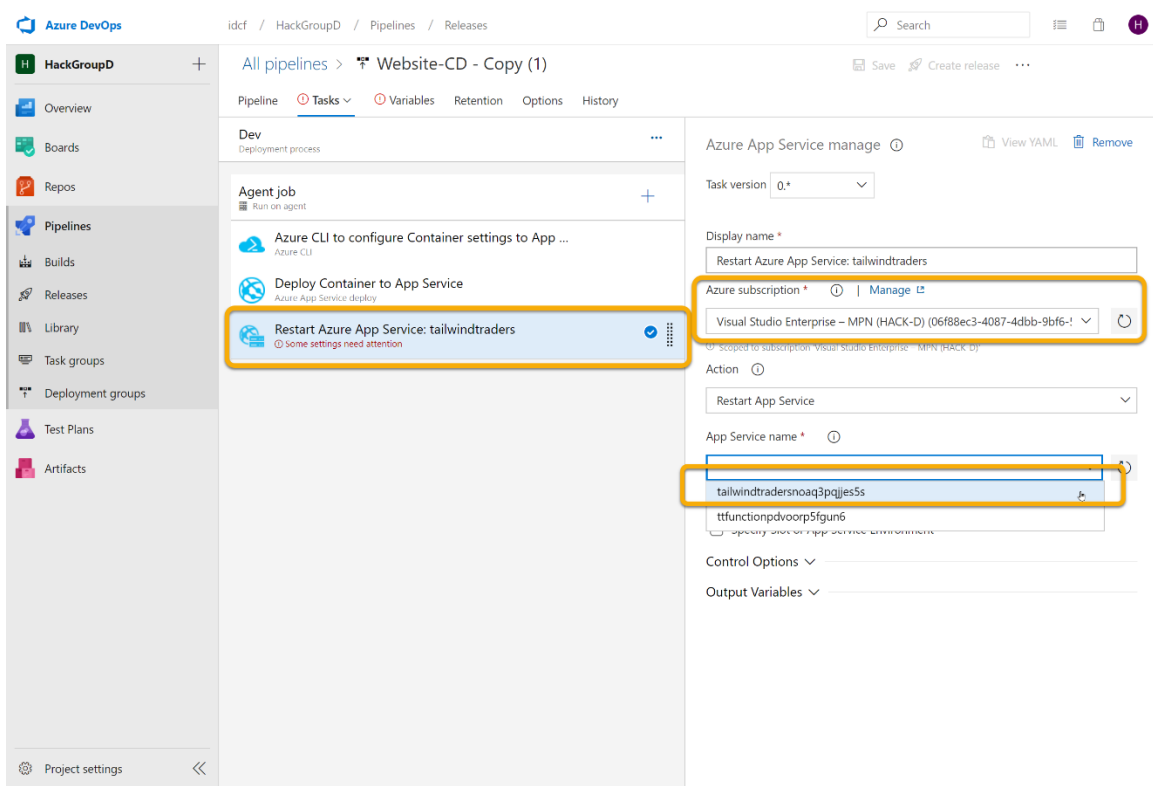
选择 Azure CLI...步骤，并设置 Azure Subscription 为分配给本组的订阅（注意订阅名称中包含的组别信息）



选择 Deploy Container ...步骤，并设置 Azure Subscription 为分配给本组的订阅（注意订阅名称中包含的组别信息），同时选择 App Service name 下拉菜单中唯一的选项

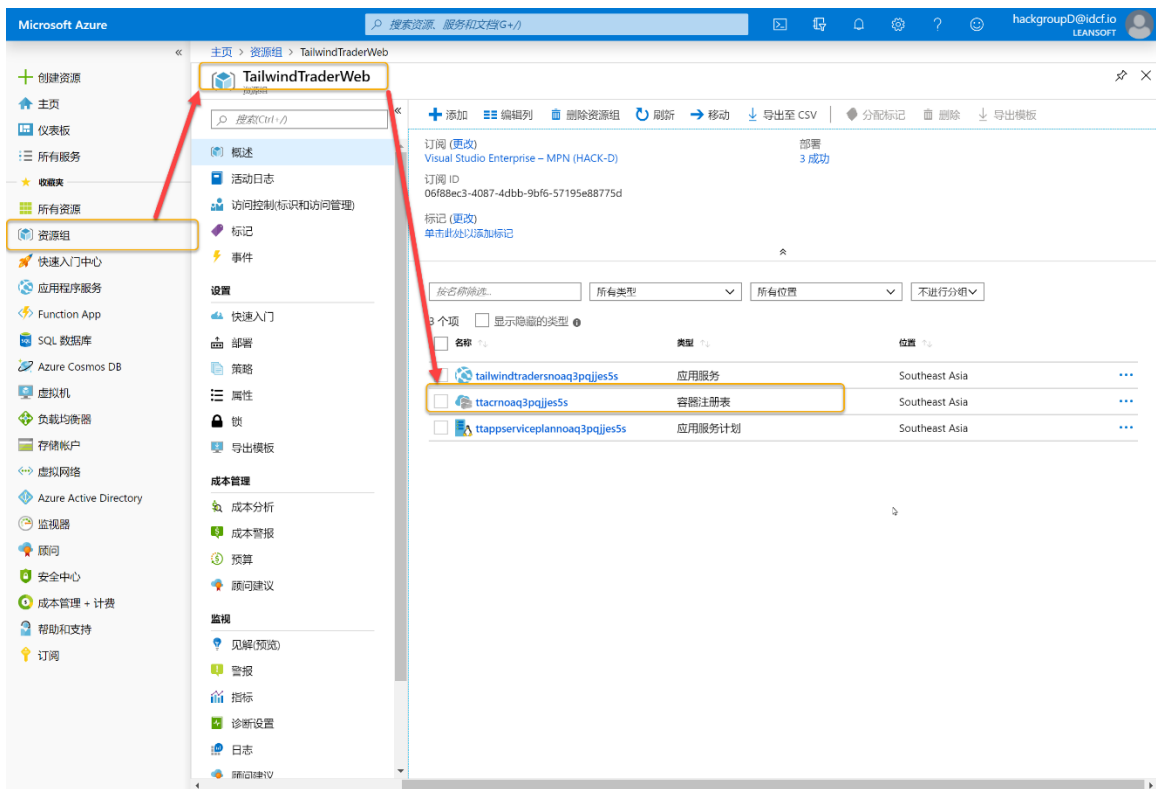


选择 Restart Azure App Service ...步骤，并设置 Azure Subscription 为分配给本组的订阅（注意订阅名称中包含的组别信息），同时选择 App Service name 下拉菜单中不含 func 关键字的选项

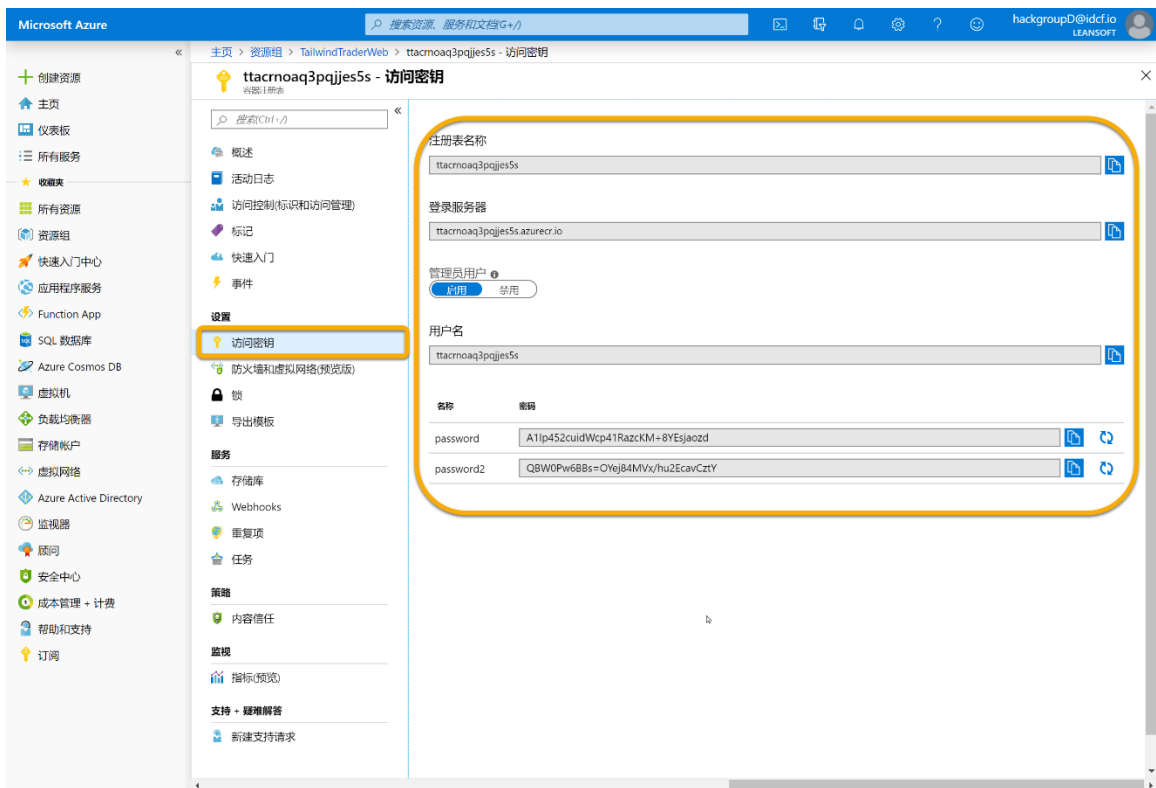


Step 3.4 - 配置部署变量

进入 Azure Portal (<https://portal.azure.com>) 获取容器注册表相关信息，在 Azure Portal 中选择 **资源组** | **TailwindTraderWeb** | **容器注册表**

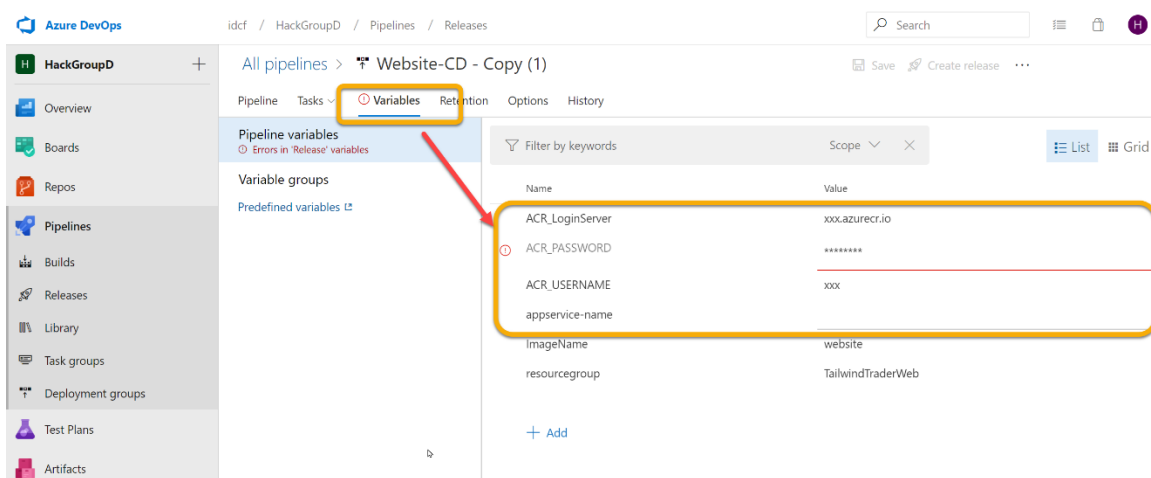


进入 访问密钥 页面获取右侧所有信息，备用。



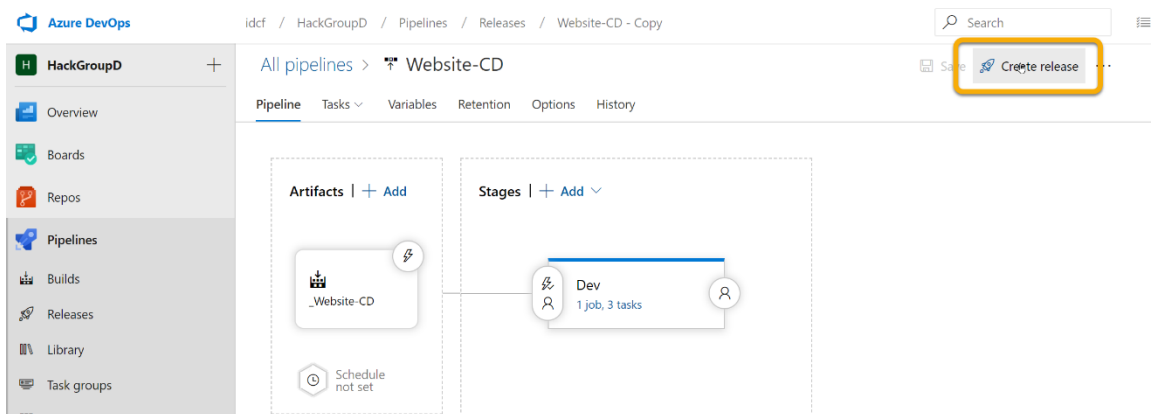
回到 Azure DevOps 的部署流水线配置中的 Variable 页面，按照从 Azure Portal 中获取的信息填写以下变量

- ACR_LoginServer: 注册表名称
- ACR_PASSWORD: 注册表 password
- ACR_USERNAME: 用户名
- appservice-name: 从步骤 3.3 中复制 App Service name

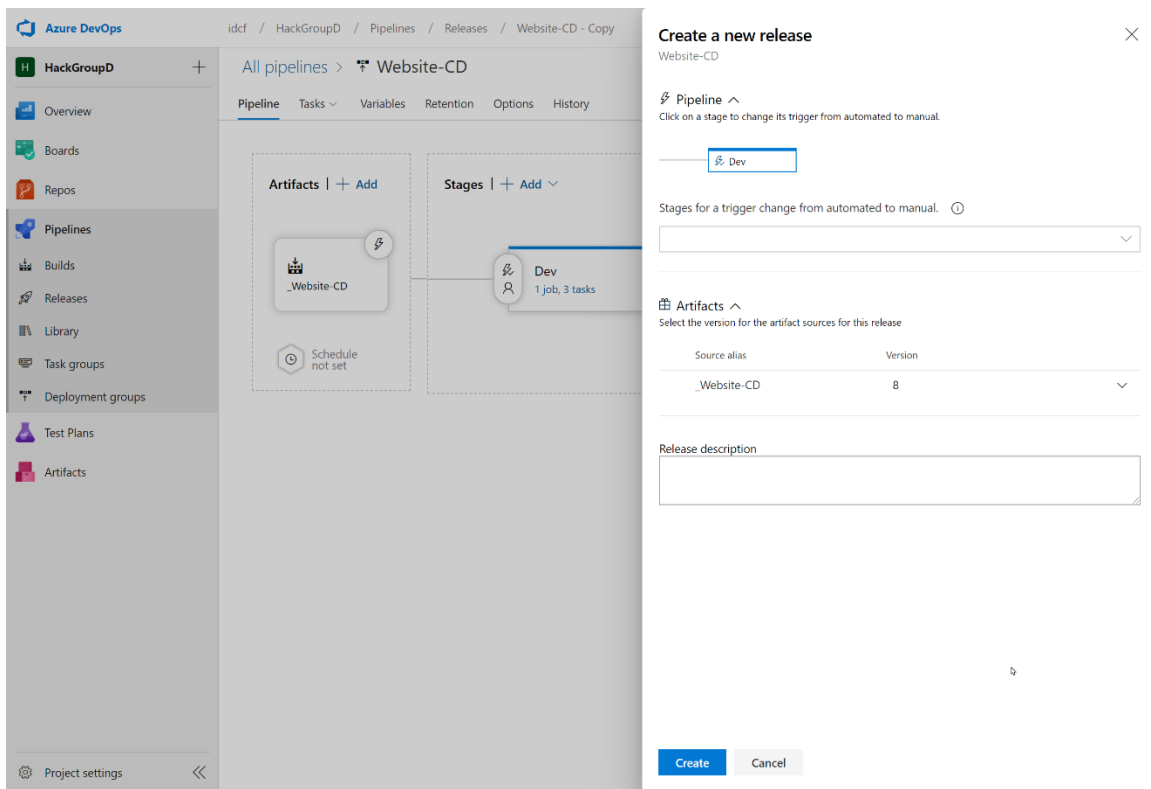


Step 3.5 - 保存触发并监控进度

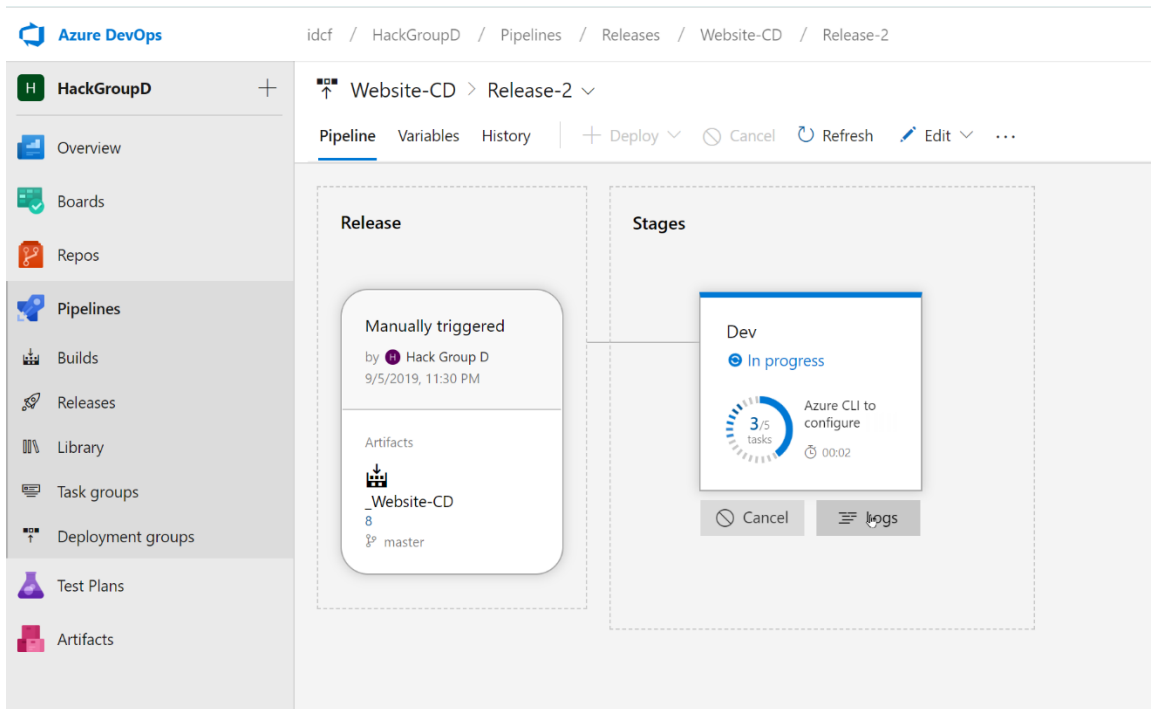
完成修改后点击 Create release 按钮触发部署



在弹出的对话框中选择 Create



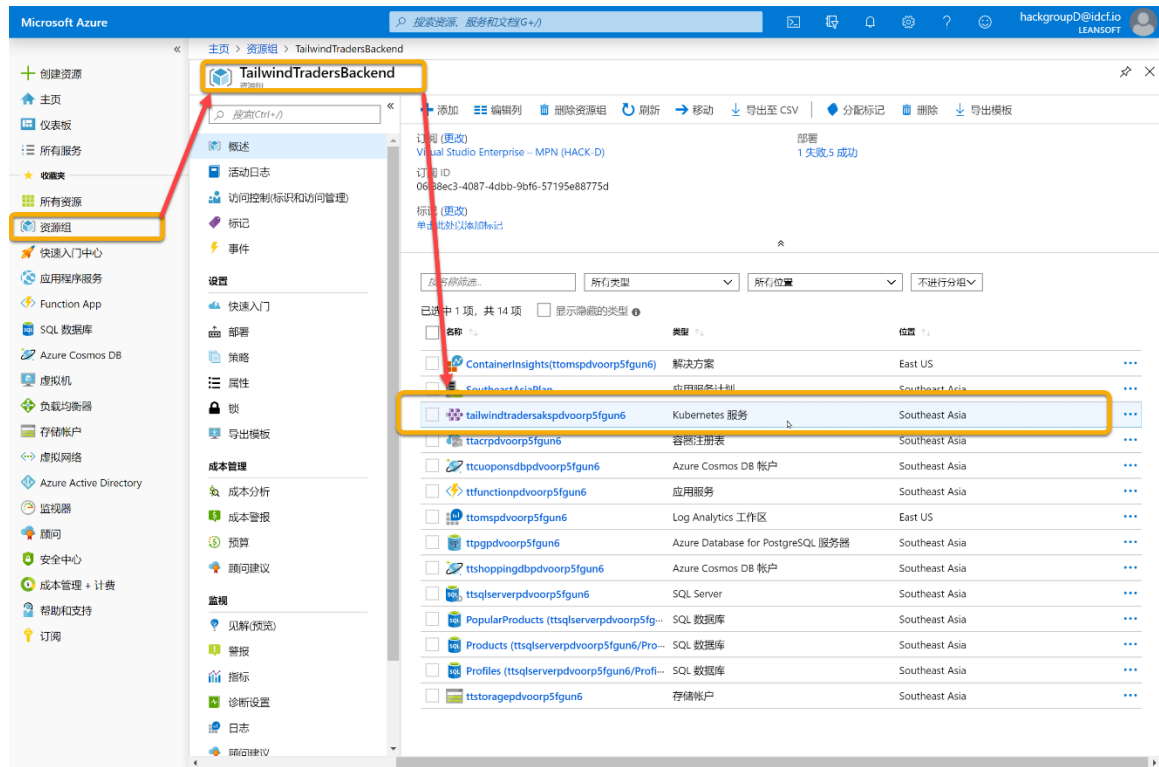
等待一段时间后部署启动，可以通过点击 logs 查看日志



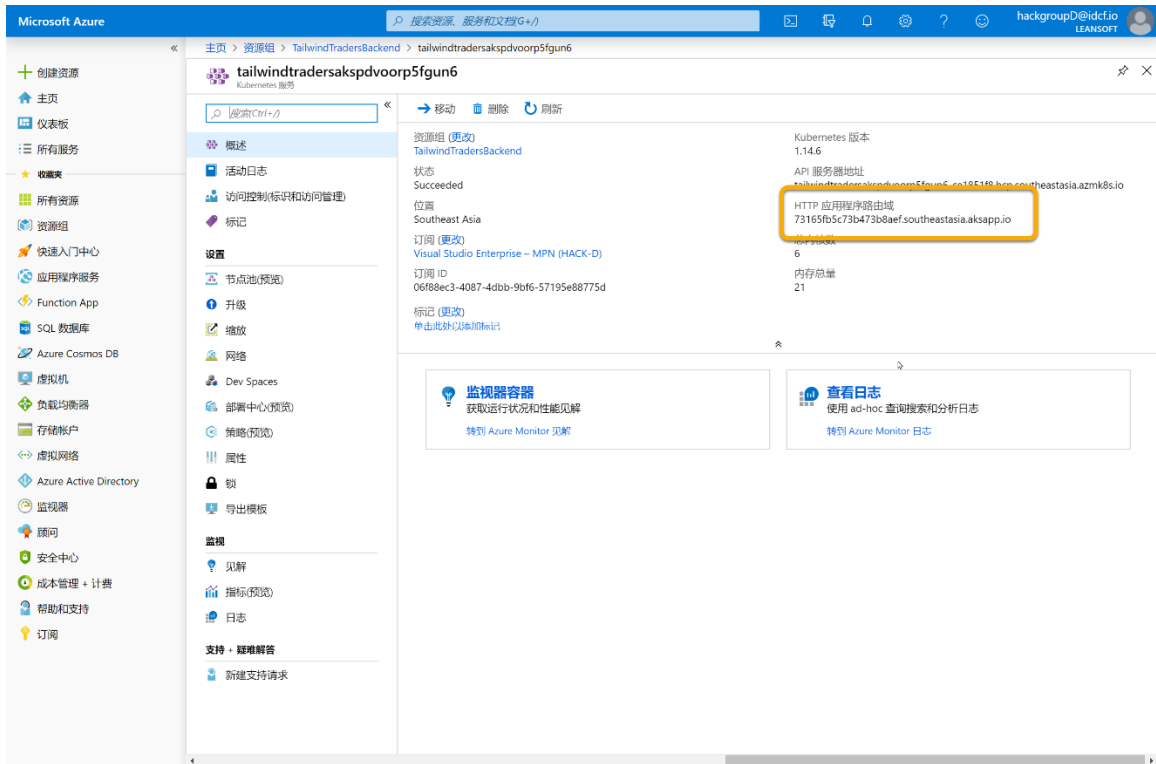
等待部署成功后进行后续步骤。

Step 4.1 - 获取后台应用 API 地址

进入 Azure Portal | 资源组 | TailwindTradersBackend 并点击 Kubernetes 服务

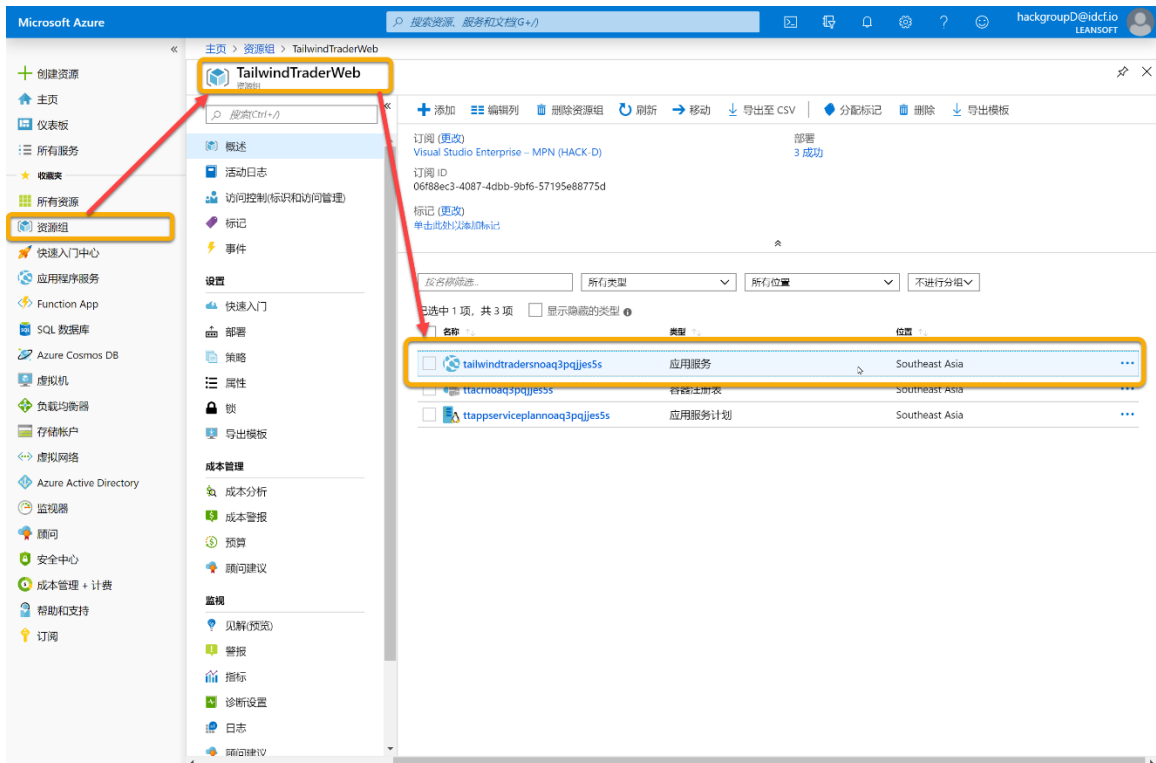


获取 **HTTP 应用程序路由域**字段的值



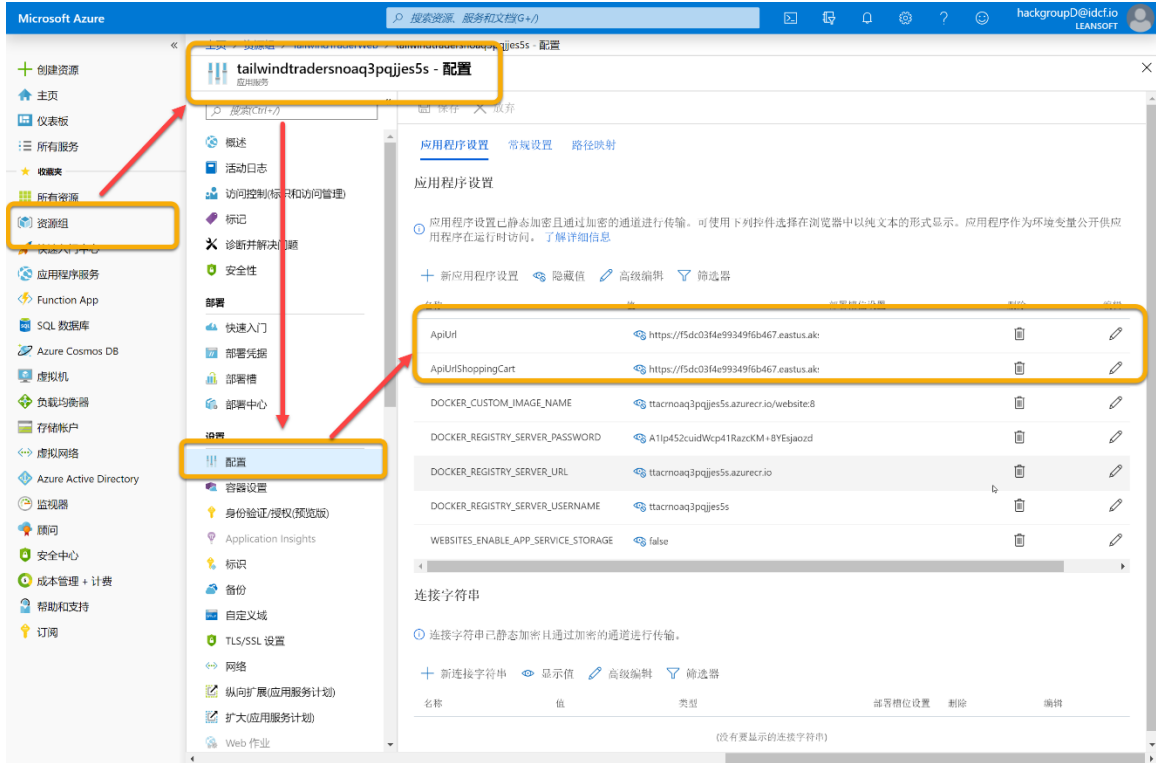
Step 4.2 - 更新 Web 应用程序配置，指向以上获取的 HTTP 应用程序路由域地址

进入 **Azure Portal | 资源组 | TailwindTradersWeb* 并点击 应用服务



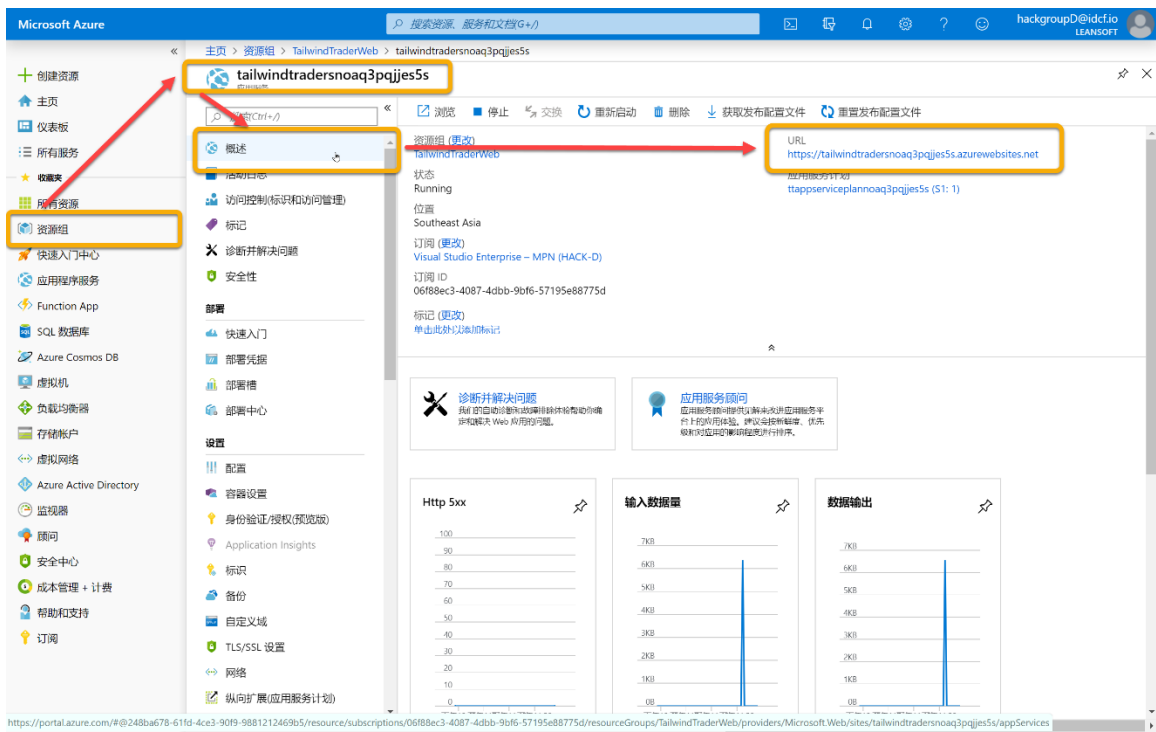
进入 **配置** 页面，点击 **显示值** 并更新以下地址

- ApiUrl = https://[HTTP 应用程序路由域]/webbff/v1
- ApiUrlShoppingCart = https://[HTTP 应用程序路由域]/cart-api

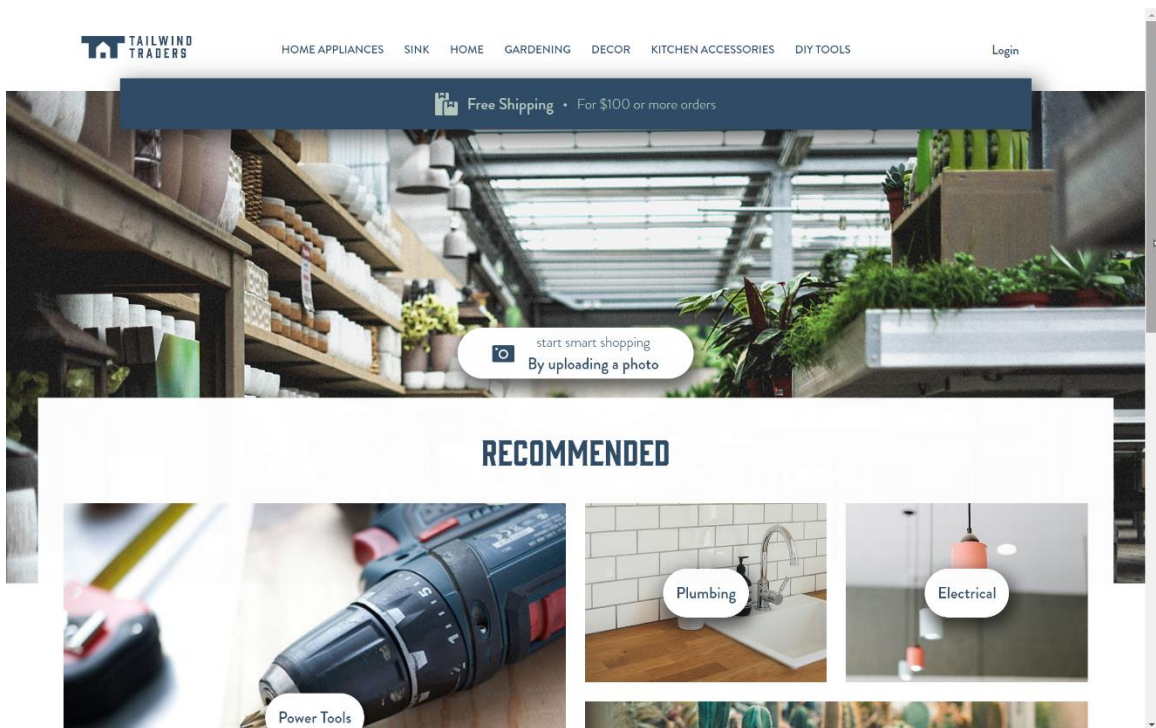


Step 5 - 测试站点工作正常

通过 **应用服务** 的页面获取网站地址



点击以上 URL 即可访问站点



恭喜，部署完成！