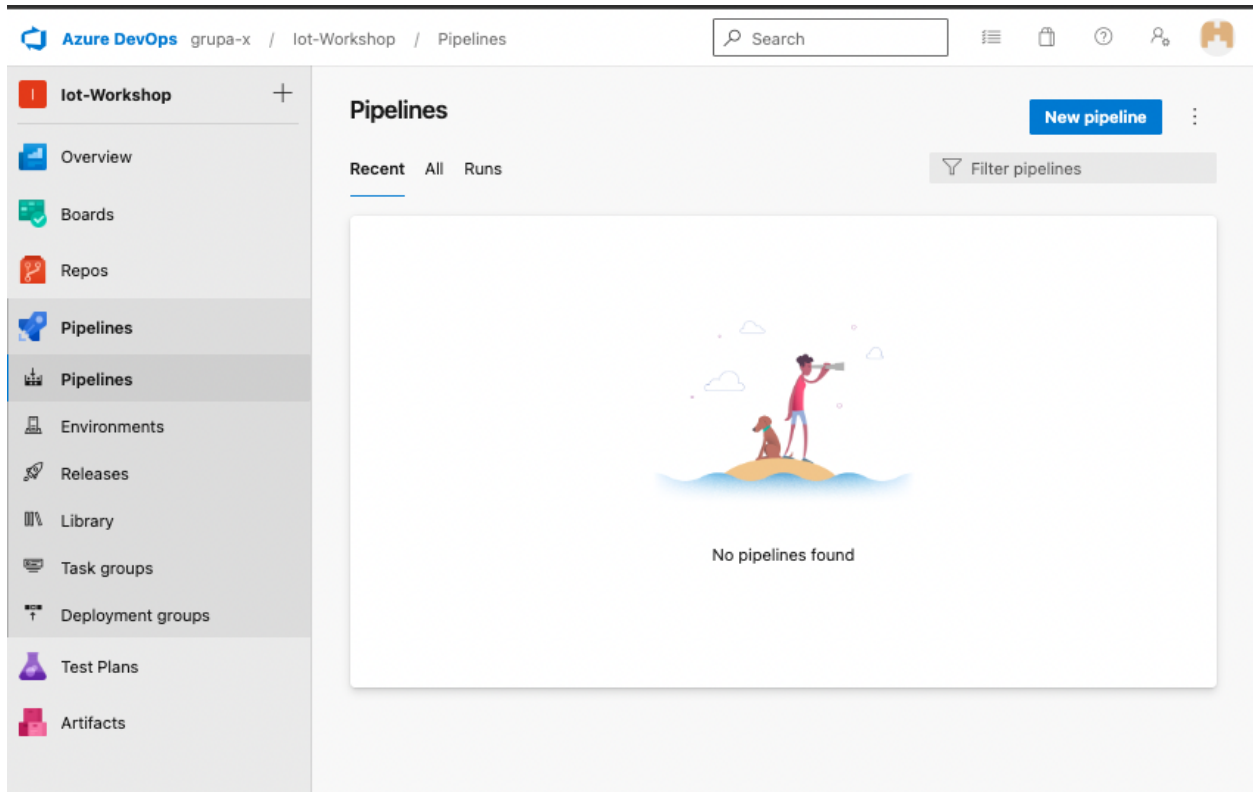


In this document we will create a basic CI/CD pipeline in Azure DevOps that will be triggered at each commit to the repo on GitHub - preferably the forked repo provided by Jacek should be used.

Prerequisite: Initial setup with an added agent needs to be completed to proceed with those instructions.

## 1. Go to Pipelines and click “New pipeline”



## 2. Choose GitHub

Connect


Select

Configure

Review


New pipeline

**Where is your code?**

 Azure Repos Git 


YAML

Free private Git repositories, pull requests, and code search

 Bitbucket Cloud 


YAML

Hosted by Atlassian

 GitHub 


YAML


Home to the world's largest community of developers

 GitHub Enterprise Server 

YAML

The self-hosted version of GitHub Enterprise

 Other Git  
Any generic Git repository


 Subversion  
Centralized version control by Apache



3. It will auto-populate with all the repos added to the GitHub account that was used to create this Azure DevOps Project. Choose the forked repo provided by Jacek


✓ Connect **Select** Configure Review


New pipeline


**Select a repository**


 Filter by keywords

My repositories  

 PiotrPietruszk/fiot\_workshop\_test fork private  
43m ago

 PiotrPietruszk/DevOps-test  
Yesterday

 Simlandir/fiot\_workshop private  
26 Nov

 PiotrPietruszk/Snipe-it-final private  
28 Jul

11 Dec 2022 at 18:16 CET

① Showing the most recently used repositories where you are a collaborator.  
If you can't find a repository, make sure you [provide access](#).  
You may also select a specific [connection](#).

*(at this point you might be taken to GitHub and prompted to log in again and install a webservice in your Github account to handle Azure Pipelines - do it)*

#### 4. Choose “Starter pipeline”|

✓ Connect


✓ Select


**Configure**


Review


New pipeline


## Configure your pipeline


**Node.js**  
Build a general Node.js project with npm.


**Node.js Express Web App to Linux on Azure**  
Build a Node.js Express app and deploy it to Azure as a Linux web app.


**Node.js with Vue**  
Build a Node.js project that uses Vue.


**Node.js with webpack**  
Build a Node.js project using the webpack CLI.

**Node.js with React**  
Build a Node.js project that uses React.

**Node.js React Web App to Linux on Azure**  
Build a Node.js React app and deploy it to Azure as a Linux web app.

**Node.js with Angular**  
Build a Node.js project that uses Angular.

**Starter pipeline**  
Start with a minimal pipeline that you can customize to build and deploy your code.

**Existing Azure Pipelines YAML file**  
Select an Azure Pipelines YAML file in any branch of the repository.

Show more

5. You will be taken to a final Pipeline creation window (pipelines in Azure DevOps are kept a YAML file in the repo)

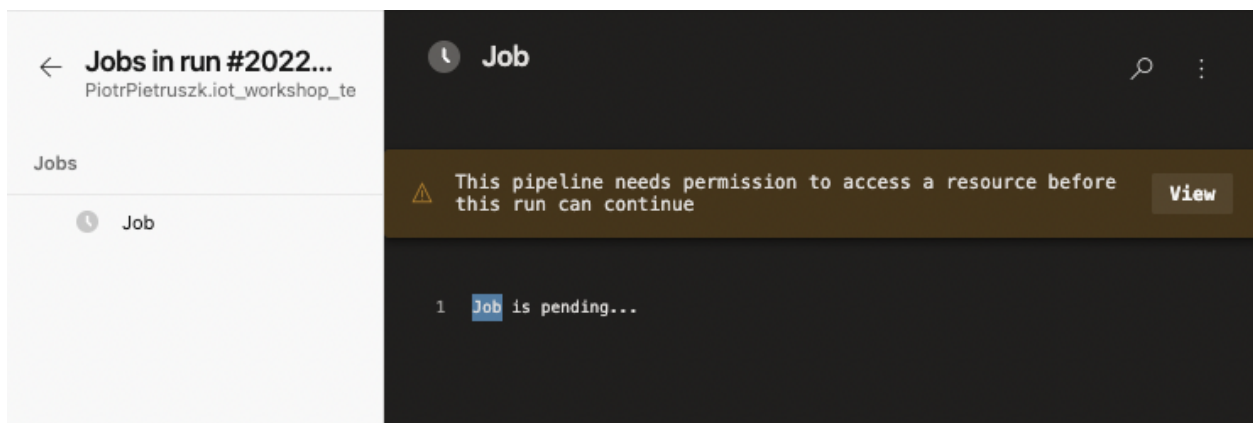
```
PiotrPietruszk/iot_workshop_test / azure-pipelines.yml *  
  
1  # Starter pipeline  
2  # Start with a minimal pipeline that you can customize to build and deploy your code.  
3  # Add steps that build, run tests, deploy, and more:  
4  # https://aka.ms/yaml  
5  
6  trigger:  
7  - master  
8  
9  pool: default  
10  
11 steps:  
12 - script: echo Hello, world!  
13   displayName: 'Run a one-line script'  
14  
15 - script: |  
16     echo Add other tasks to build, test, and deploy your project.  
17     echo See https://aka.ms/yaml  
18   displayName: 'Run a multi-line script'  
19
```

6. Change your file to look like the screenshot above (change the agent pool to *pool: default*)

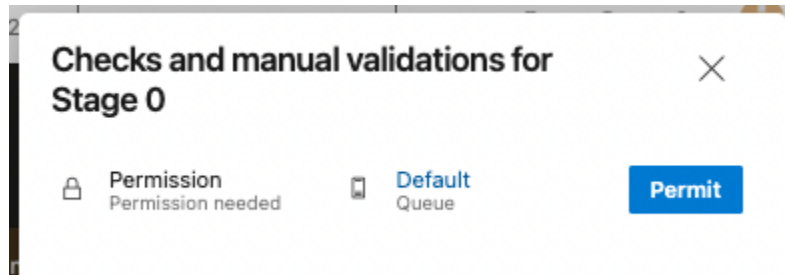
7. Click “Save and run” and “Save and run” again - this will commit the pipeline YAML file into the repo in GitHub and run it on the agent you created in the initial configuration.

(From now on you can edit your pipeline either in Azure DevOps GUI, or from your GitHub repository)

8. Go to the job that just started and click “view” on the right



9. Click “Permit” (this will allow to the pipeline to use the default agent pool)



10. The Job will run the pipeline

← **Jobs in run #20221211.1**  
PiotrPietruszk.iot\_workshop\_test

Jobs

✓ Job	6s
Initialize job	<1s
✓ Checkout PiotrPietruszk/iot_wor...	1s
✓ Run a one-line script	<1s
✓ Run a multi-line script	<1s
✓ Post-job: Checkout PiotrPietru...	<1s
Finalize Job	<1s

✓ **Run a one-line script**

```
1 Starting: Run a one-line script
2 =====
3 Task      : Command line
4 Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
5 Version   : 2.212.0
6 Author    : Microsoft Corporation
7 Help      : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
8 =====
9 Generating script.
10 Script contents:
11 echo Hello, world!
12 ===== Starting Command Output =====
13 /usr/bin/bash --noprofile --norc /home/piotr/myagent/_work/_temp/c2eb9e6a-1b28-4698-81e6-c7c0f9c3f91a.sh
14 Hello, world!
15 Finishing: Run a one-line script
```

This time the trigger was manual, but be advised that this job will run every single time you make a commit to the master branch!

## How to adjust this job to do what I want it to?

1. If you want to change the branch that this pipeline is triggered for change the {branch name} in the following line in azure-pipelines.yml:

```
[...]
trigger:
- master
[...]
```

2. There is a lot of extra functionality in Azure Pipelines, but for the most part you will stick to running bash commands using a step called - *script*:

For our purposes you can edit you azure-pipelines.yml to do everything in one step, like on the screenshot attached: (be sure to have the pipe symbols ( | ) after - *script*: it allow for multi-line scripts)



```
master  ▼  PiotrPietruszk/iot_workshop_test / azure-pipelines.yml *

1  # Starter pipeline
2  # Start with a minimal pipeline that you can customize to build and deploy your code.
3  # Add steps that build, run tests, deploy, and more:
4  # https://aka.ms/yaml
5
6  trigger:
7  - master
8
9  pool: default
10
11  steps:
12
13  - script: |
14    cd terraform
15    terraform init
16    terraform apply -auto-approve
17    make
18    sleep 5
19    echo "done"
20    displayName: 'Run a multi-line script'
21
```