

This document will show how to perform initial configuration of Azure DevOps and add a self-hosted agent to this service (agent can be understood as a virtual computer that will execute our code.)

Prerequisite: git needs to be installed on the Linux VM before you start. You might have done it already before you started working on this runbook, or it might have been provided already preinstalled in your VM's image.

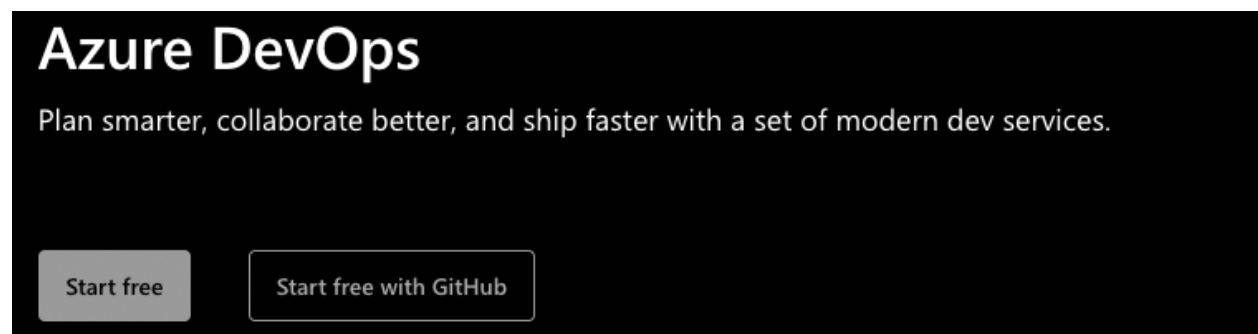
check if git is installed using the following command:

```
git --version
```

If it returned the current version, git is installed. If not run the following command to install git:

```
sudo apt install git
```

1. Go to <https://azure.microsoft.com/en-us/products/devops/> and click "Start free with GitHub"



If you have any issues with logging in with a GitHub account:

- use this URL to sign out of all MS services- <https://login.microsoftonline.com/logout.srf>
- close all the browser windows/tabs
- reopen the browser and try again

2. Using the Github account created for this workshop log into Azure DevOps

There will be a few pages of details to fill out e.g.:

- If possible, Use your AWS username as the Name of your Azure DevOps organization.

We need a few more details

Your name:

Full name is required.

We'll reach you at:

Email is required.

From:

☐ I would like to receive information, tips, and resources related to Microsoft developer tools and services, including Azure DevOps, Visual Studio, Visual Studio Subscriptions, and other Microsoft products and services.

Continue

To keep our lawyers happy:
By continuing, you agree to the [Terms of Service](#),
[Privacy Statement](#), and [Code of Conduct](#).

Azure DevOps

peterka60@gmail.com

Almost done...

Name your Azure DevOps organization

dev.azure.com/ grupa-x

We'll host your projects in

West Europe

Enter the characters you see

New Audio



DkYY65Y

Challenge validation failed, please try again

Continue

3. After you fill out all the details you will be taken to “Create a Project” page, do it.:

-Use “IoT-Workshop” for the Project Name

Create a project to get started

Project name *

IoT-Workshop

Visibility



Public

Anyone on the internet
can view the project.
Certain features like
TFVC are not supported.



Private

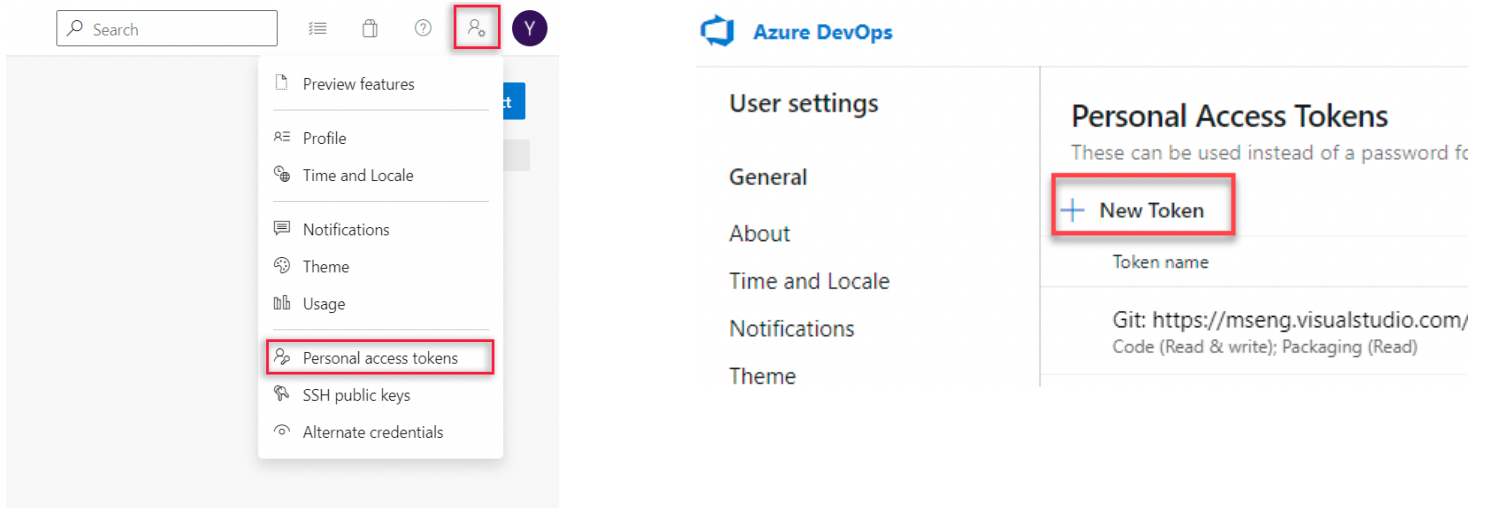
Only people you give
access to will be able to
view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

+ Create project

The basis of our CI/CD service is ready.

4. Now we have to add a self-hosted agent that will execute our code. First, we create a Personal Access Token - we will use it later to authenticate the agent in Azure DevOps
Create a token as indicated in the screenshots - remember to save it somewhere, if you lose it create a new one.



Create a new personal access token



Name

Agent

Organization

grupa-x

Expiration (UTC)

30 days

09/01/2023



Scopes

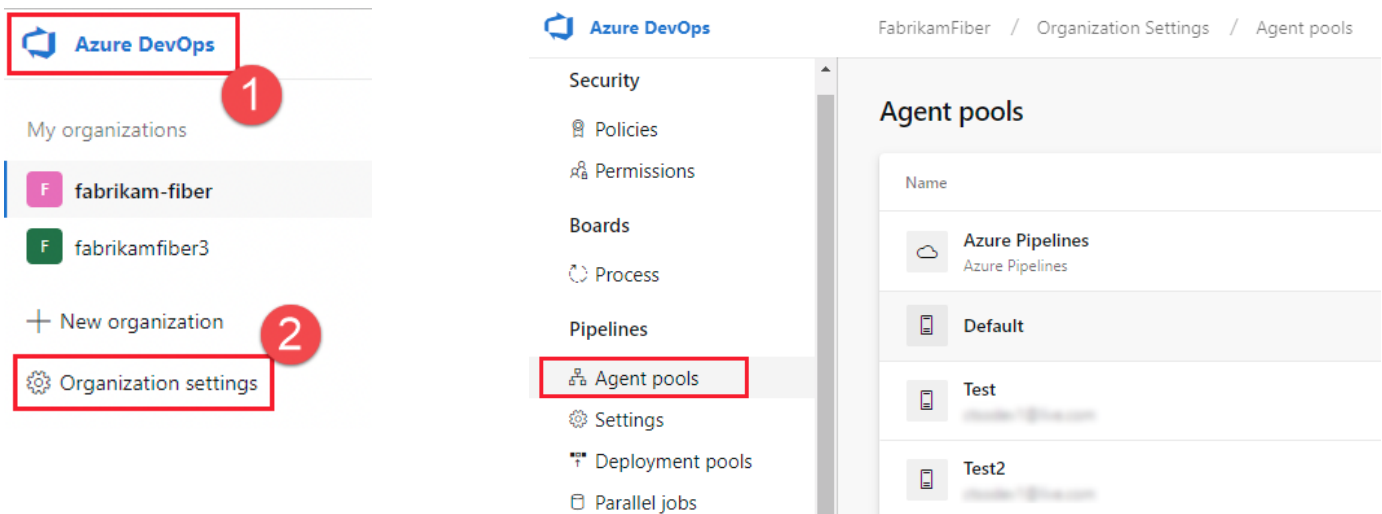
Authorize the scope of access associated with this token

Scopes ☒ Full access

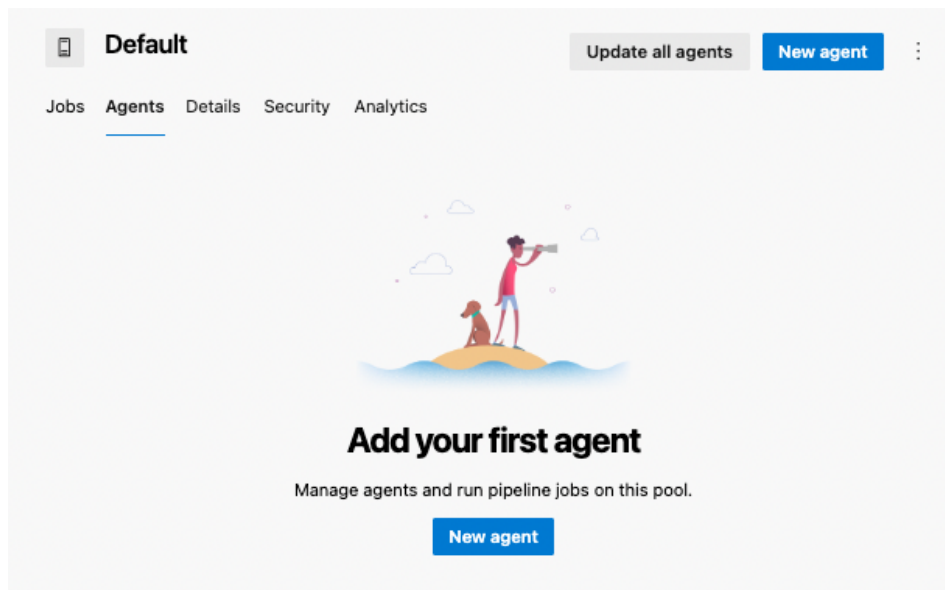
☐ Custom defined

5. Now we add a new agent to the agent pool in Azure DevOps.
(We do it with an inbound connection to Azure DevOps from our virtual machine.)

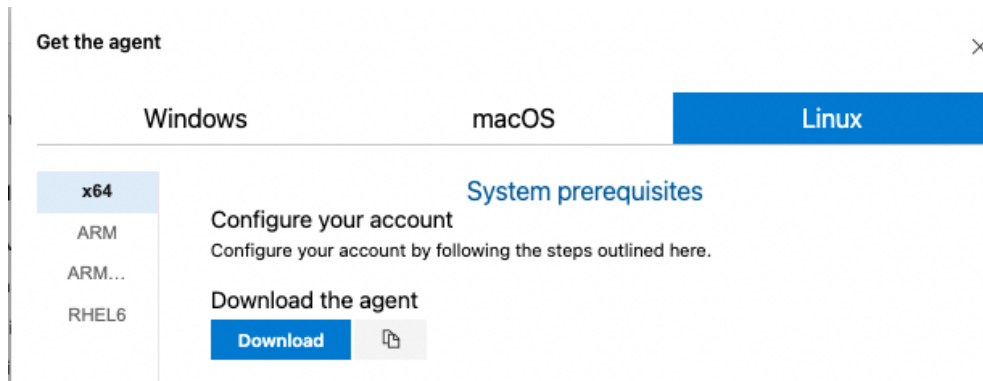
- Using screenshots as a guide, go to the Agent pools in and choose the Default pool



- Click New Agent



- Choose the "Linux" tab and click "Download"



- Open the terminal on the VM and run the following commands one by one, to prepare the agent for configuration:

```
mkdir myagent && cd myagent
tar zxvf ~/Downloads/vsts-agent-linux-x64-2.213.2.tar.gz
sudo ./bin/installdependencies.sh
```

- Run the following command in the terminal to begin the configuration:

```
./config.sh
```

There will invoke a script with an interactive prompt.

Refer to the block below for exact input and to the screenshot for the expected feedback.

Consecutive input for the config script:

- Y
- <https://dev.azure.com/{name of the organization}> (for me it was <https://dev.azure.com/grupa-x>; look at your Azure DevOps URL if you have any doubts)
- (enter)
- (the PAT that we created in step 3.)
- (enter)
- (enter)
- (enter)

```
piotr@piotr-VirtualBox: ~/myagent
piotr@piotr-VirtualBox:~/myagent$ ./config.sh
libcurl.so.4 => not found
Warning: on some platforms, libcurl3 is required.
It was not found.
Execute ./bin/installdependencies.sh to install missing dependencies.

  agent v2.213.2      (commit 4f90e68)

>> End User License Agreements:

Building sources from a TFVC repository requires accepting the Team Explorer Everywhere End User License Agreement. This step is
es.

A copy of the Team Explorer Everywhere license agreement can be found at:
/home/piotr/myagent/license.html

Enter (Y/N) Accept the Team Explorer Everywhere license agreement now? (press enter for N) > Y

>> Connect:

Enter server URL > https://dev.azure.com/grupa-x
Enter authentication type (press enter for PAT) >
Enter personal access token > *****
Connecting to server ...

>> Register Agent:

Enter agent pool (press enter for default) >
Enter agent name (press enter for piotr-VirtualBox) >
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
Enter work folder (press enter for _work) >
2022-12-11 17:43:20Z: Settings Saved.
```

6. Start the agent by running the following command in the terminal

-Run the following command in the terminal:
`./run.sh`

```
piotr@piotr-VirtualBox:~/myagent$ ./run.sh
Scanning for tool capabilities.
Connecting to the server.
2022-12-11 17:41:14Z: Listening for Jobs
```

This agent runs interactively (as long as the terminal window is not closed).
If you need to use the terminal open a new tab using the button in the top left corner.