# Exercise

# Build the application using your agent

Now that your build agent is running and ready to receive build jobs, let's see it in action. In this unit, you modify a basic build configuration that we provide to build the Space Game website by using your agent and not the Microsoft-hosted agent.

At the end of this unit, as an optional step, you can remove the agent pool from your Microsoft Azure DevOps organization.

## Fetch the branch from GitHub

In this section, you fetch the build-agent branch from GitHub and check out, or switch to, that branch.

This branch contains the Space Game project that you worked with in previous modules and an Azure Pipelines configuration to start with.

1. In Visual Studio Code, open the integrated terminal.
2. To download a branch named build-agent from the Microsoft repository and switch to that branch, run the following git fetch and git checkout commands:

**Bash**

git fetch upstream build-agent

git checkout -b build-agent upstream/build-agent

Recall that upstream refers to the Microsoft GitHub repository. Your project's Git configuration understands the upstream remote, because you set up that relationship when you forked the project from the Microsoft repository and cloned it locally.

Shortly, you'll push this branch up to your GitHub repository, known as origin.

1. Optionally, in Visual Studio Code, open the azure-pipelines.yml file, and familiarize yourself with the initial configuration.

The configuration resembles the basic one that you created in the Create a build pipeline with Azure Pipelines  module. It builds only the application's release configuration.

## Modify the build configuration

In this section, you modify the build configuration to switch from using a Microsoft-hosted agent to using the agent from your build pool.

1. In Visual Studio Code, open the azure-pipelines.yml file, and then look for the pool section.

**yml**

pool:

vmImage: 'ubuntu-18.04'

demands:

- npm

1. Modify the pool section, as shown here:

**yml**

pool:

name: 'MyAgentPool'

demands:

- npm

This version uses name to specify your agent pool, MyAgentPool. It maintains the demands section to specify that the build agent must have npm, the Node.js package manager, installed.

1. In the integrated terminal, add azure-pipelines.yml to the index, commit the changes, and push the branch up to GitHub.

**Bash**

**git add azure-pipelines.yml**

**git commit -m "Use private agent pool"**

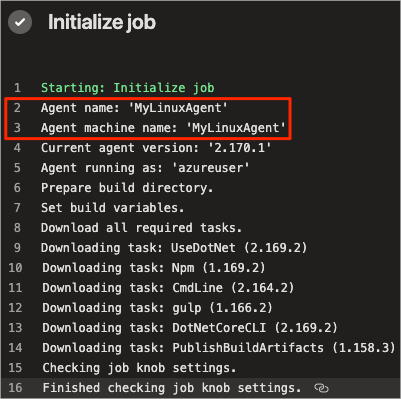
**git push origin build-agent**

## Watch Azure Pipelines use your build agent

Watch the build run in the pipeline by using your build agent.

1. In Azure DevOps, go to the **Space Game - web - Agent** project.
2. On the project page or in the left pane, select **Pipelines**.
3. Select **Builds**, and then select the running build.
4. Trace the build through each of the steps.

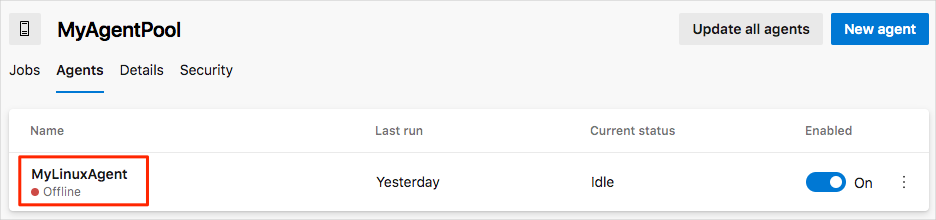
From the **Initialize job** task, you see that the build uses your build agent.



## Optional: Remove your build pool

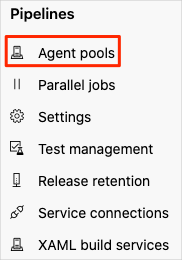
For future reference, you can keep the build pool configuration in your Azure DevOps organization. But keep in mind that the VM we provide will no longer be available to you after your sandbox session ends.

In fact, Azure DevOps will detect that the agent is offline. Azure Pipelines will check for an available agent the next time a build is queued by using the MyAgentPool pool.



As an optional step, you can remove the build pool configuration from Azure DevOps. Here's how:

1. In Azure DevOps, go to the **Space Game - web - Agent** project.
2. Select **Project settings**.
3. Under **Pipelines**, select **Agent pools**.



1. Under **MyAgentPool**, select the trash can icon, and then select **Delete**.

