|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4UMGC Capstone | | | | |
|  |  | | |  |
| DevSecOps | | | | |
|  | | Runbook |  | |

TABLE OF CONTENTS

[Introduction 3](#_Toc54094999)

[1. Managing Github 4](#_Toc54095000)

[Process Overview 4](#_Toc54095001)

[Code Review Checklist 5](#_Toc54095002)

[2. Azure Pipelines 6](#_Toc54095003)

[*Creating a Pipeline* 6](#_Toc54095004)

# Introduction

This document outlines the standards, best practices, and processes that are to be adhered to throughout the course of the UMGC software engineering capstone project by DevSecOps with regards to managing github and running azure pipelines.

This remainder of this document will operate within the context of the UMGC software engineering capstone project context, referring to the business as the class as a whole and project teams as development teams. The purpose of this document is to inform, and thus the remainder of the document will be

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reasons for change | Version |
| Dustin Emerson | 10/10/20 | Initial Draft | V 0.1 |
| Glenn Goodlett | 10/11/20 | Peer Review | V 0.2 |
| Dustin Emerson | 10/14/20 | V 1.0 Release | V 1.0 |

# Managing Github

## Process Overview

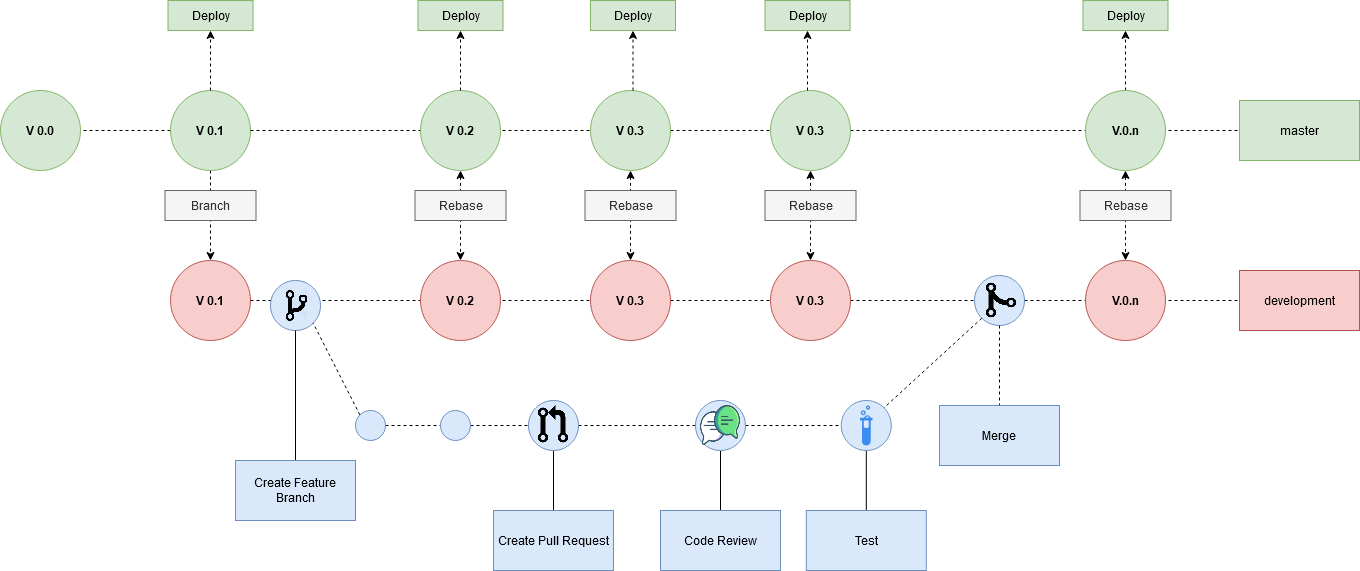


Figure: Version Control Process Overview

Projects repositories will be set up as such:

1. There is a mainline **master** branch that requires a member of DevSecOps to approve code being pushed to it. This branch will be pipelined as such where the project deploys after a build is successful.
2. There is a **development** branch forked off of master. This is the branch the dev teams will use when working. The team can decide how often they wish to move the development branch into master.

## Code Review Checklist

By default DevSecOps is required to approve all pull requests merging to master. Below is the criteria required to fufill what should be an approval from DevSecOps in order of precedence:

1. Does the code pass the pipeline checks?
2. Does the code conform to style guildines? (if not instruct them to run mvn spotless::apply)
3. Is the code free of binaries?
4. Is the code free of any IDE or build artifacts? (if not instruct them on updating .gitignore)

Once that criteria is met, the pull request can be approved and the author is free to merge to master.

# Azure Pipelines

DevSecOps will be using Azure to automate the build process. Thanks to ADF, creating a pipleline in Azure is very easy.

Note: If a development project was not integrated with ADF, please consult the DevSecOps Programmers guide.

## *Creating a Pipeline*

1. Using the umgc.capstone.bot credentials provided to you by your professor or mentor, log into [dev.azure.com](https://dev.azure.com). Then navigate to the capstone project at [dev.azure.com/umgccapstone/Capstone](https://dev.azure.com/umgccapstone/Capstone).
2. In the left navigation pane `Pipelines`
3. In the Pipelines page, click the “New pipeline” button.
4. On the “Where is your code?” screen, select GitHub
5. On the select repository screen, type in the name of your repo and select it.
6. This will prompt you to “Approve and Install Azure Piplelines” on your repo. Leave everything preconfigured and select “Approve & Install”
7. Enter your password for GitHub
8. After logging into GitHub and back into Azure, at the “Configure your pipline” step select “Existing Azure Pipelines YAML file”
9. Select the path to the YAML file you created when integrating ADF to the project when following the programmers guide.
10. You should now see the YAML file your created. From here click the Ellipses Icon in the top right of the menu and select “Triggers”

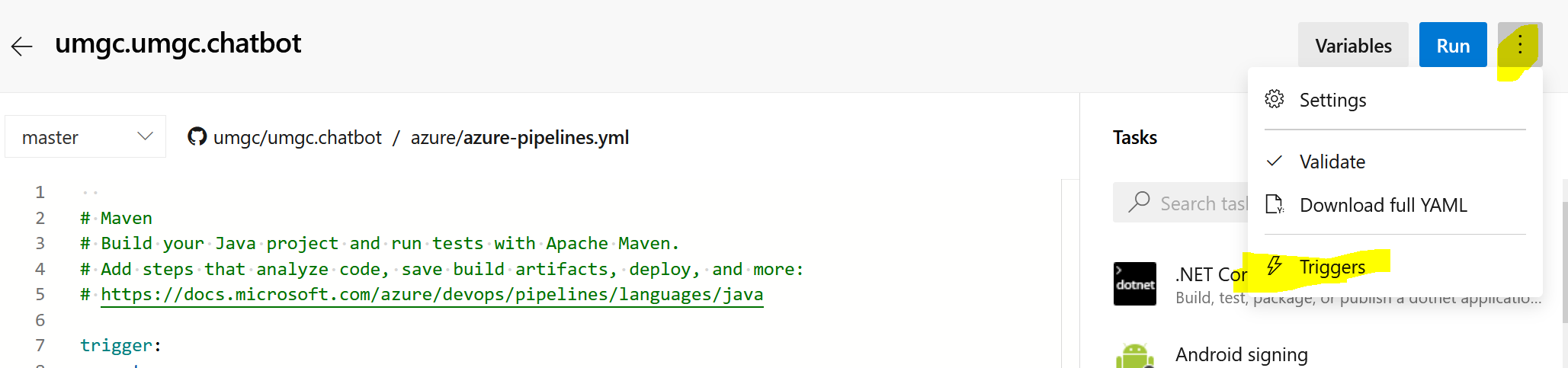


Figure: Project YAML file

1. From here select “Variables”
2. From the variable tab select “Variable Groups”
3. Select the DockerHub variable group and link the variable group

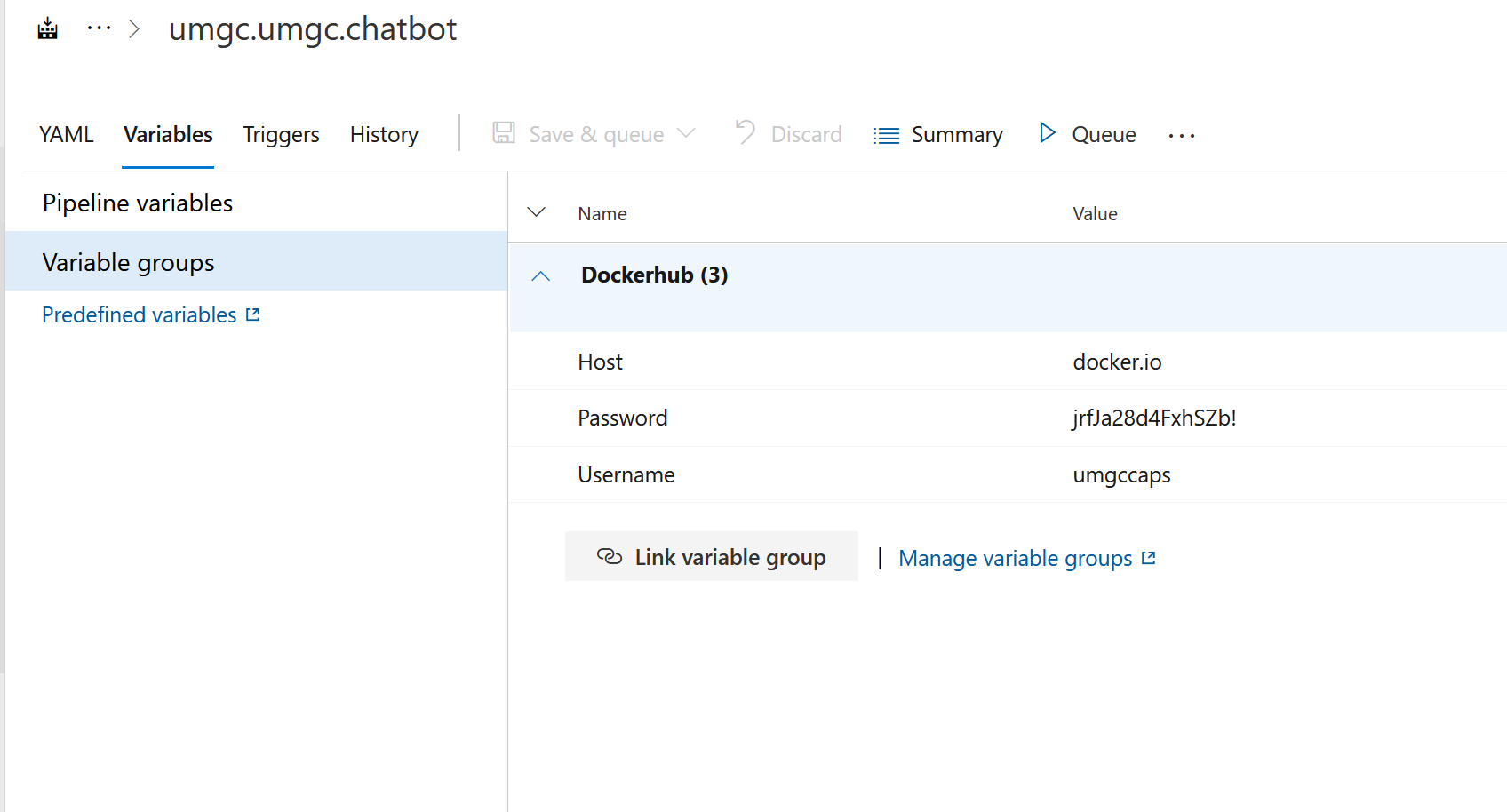


Figure: Linking Variable Groups

1. Click “Save and queue” which will return you to the YAML file.
2. Click Run to see the pipeline run.

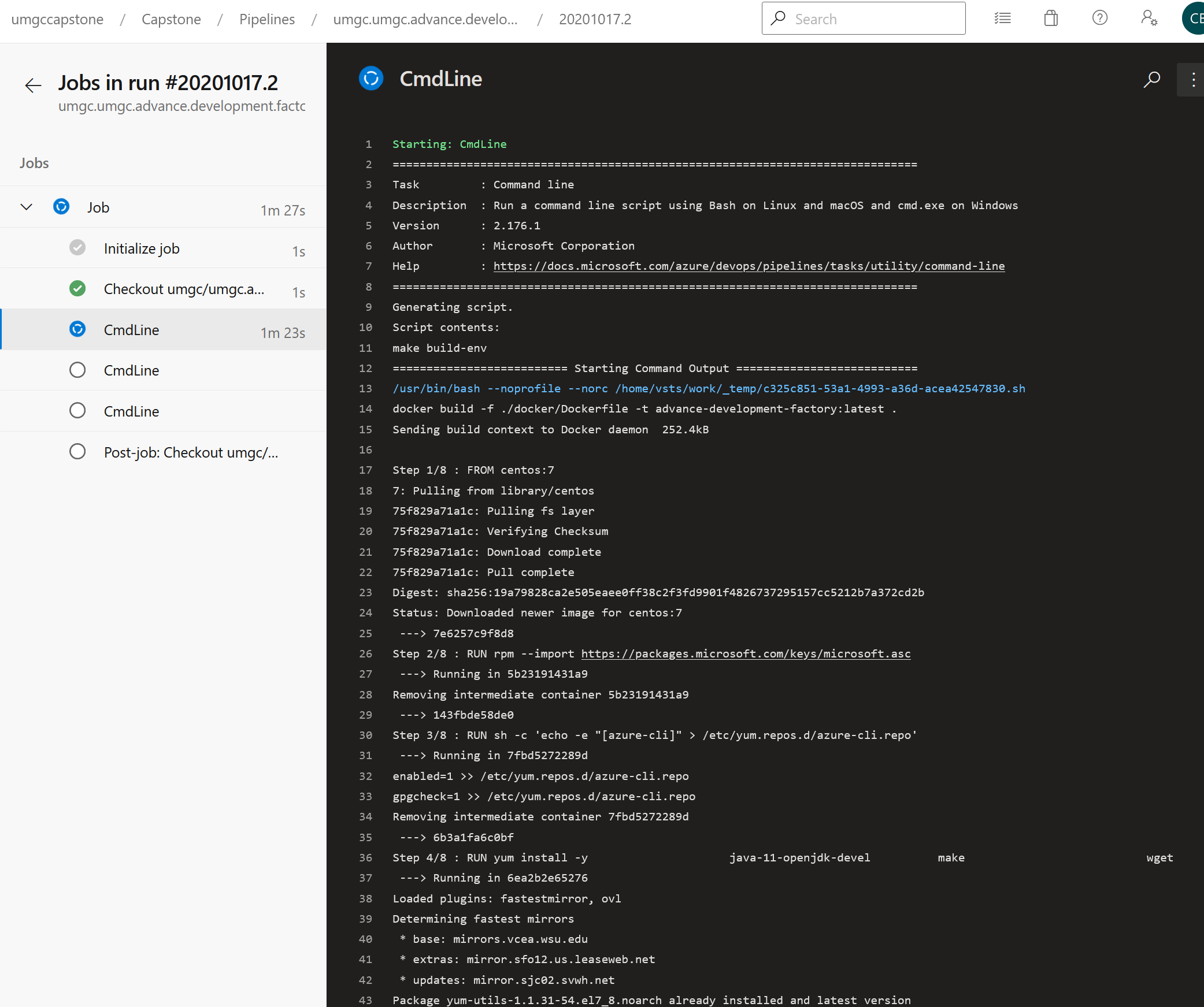


Figure: Running pipeline

The pipeline will henceforth automatically be tied into pull requests for that repo.