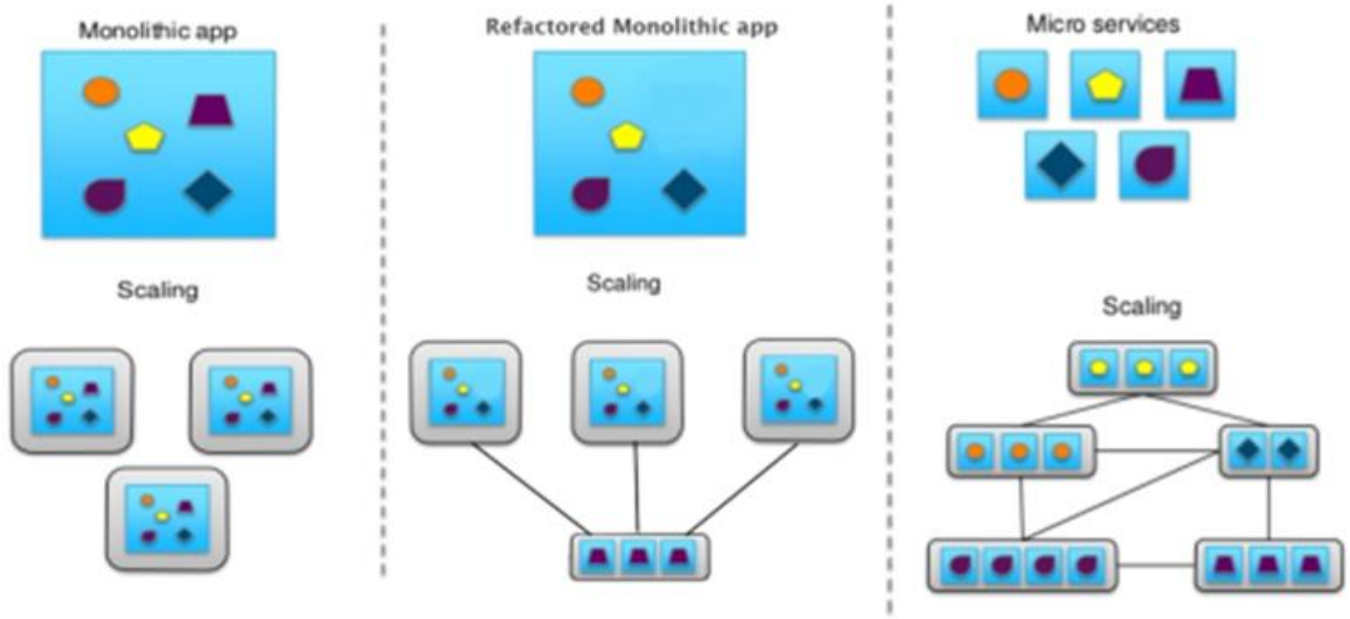
CI/CD of Containerized Apps in Azure

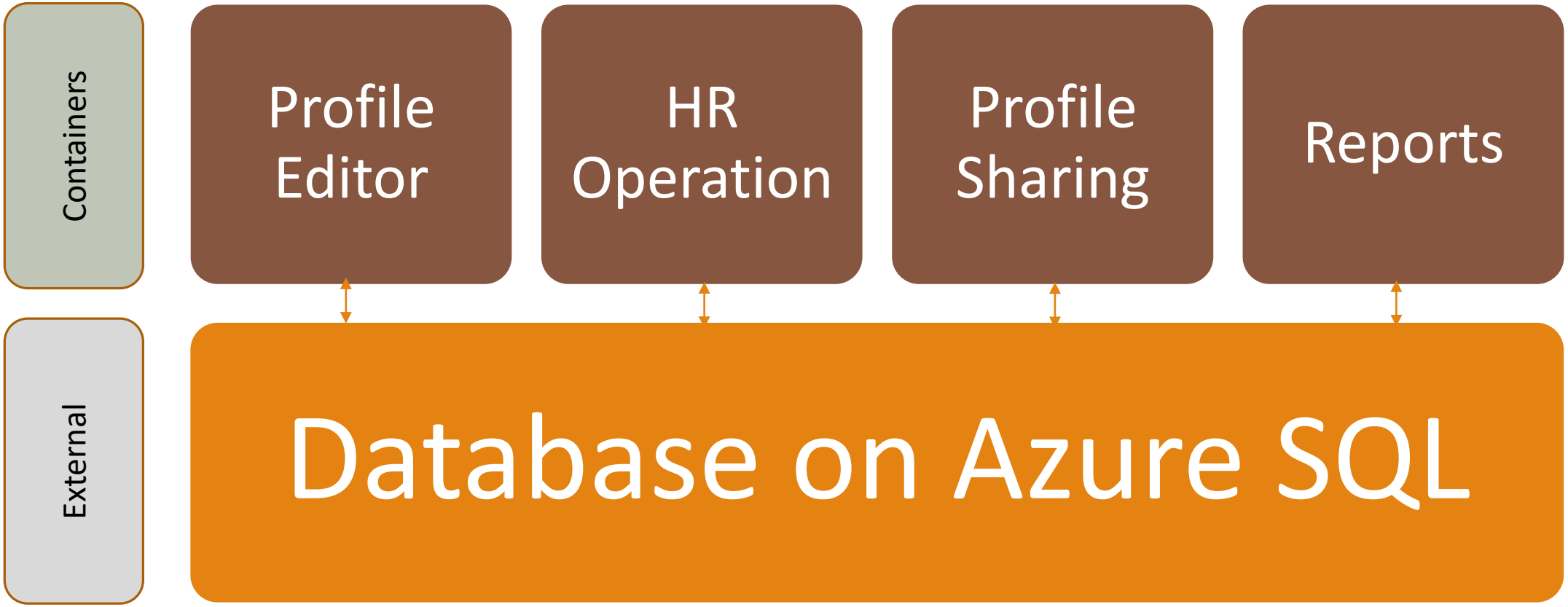
SUBODH SOHONI

Microservices

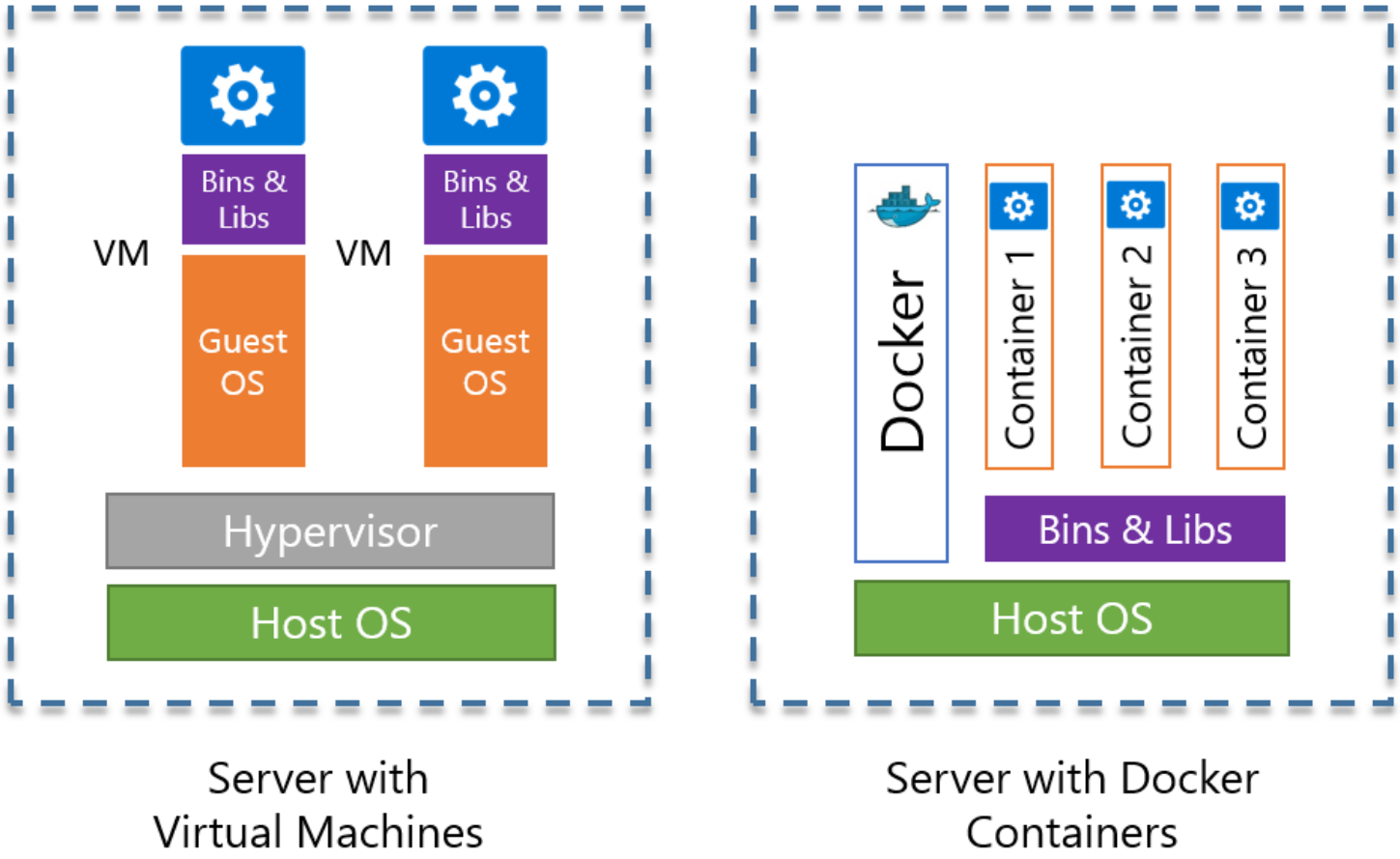
1. Small and Granular Building Blocks
2. Each Service can be Independently upgraded and maintained
3. No Single Point of Failure
4. Can Scale as per the Level of Demand
5. Much More Reusable



SSGS EMS (Profile Management) App Architecture

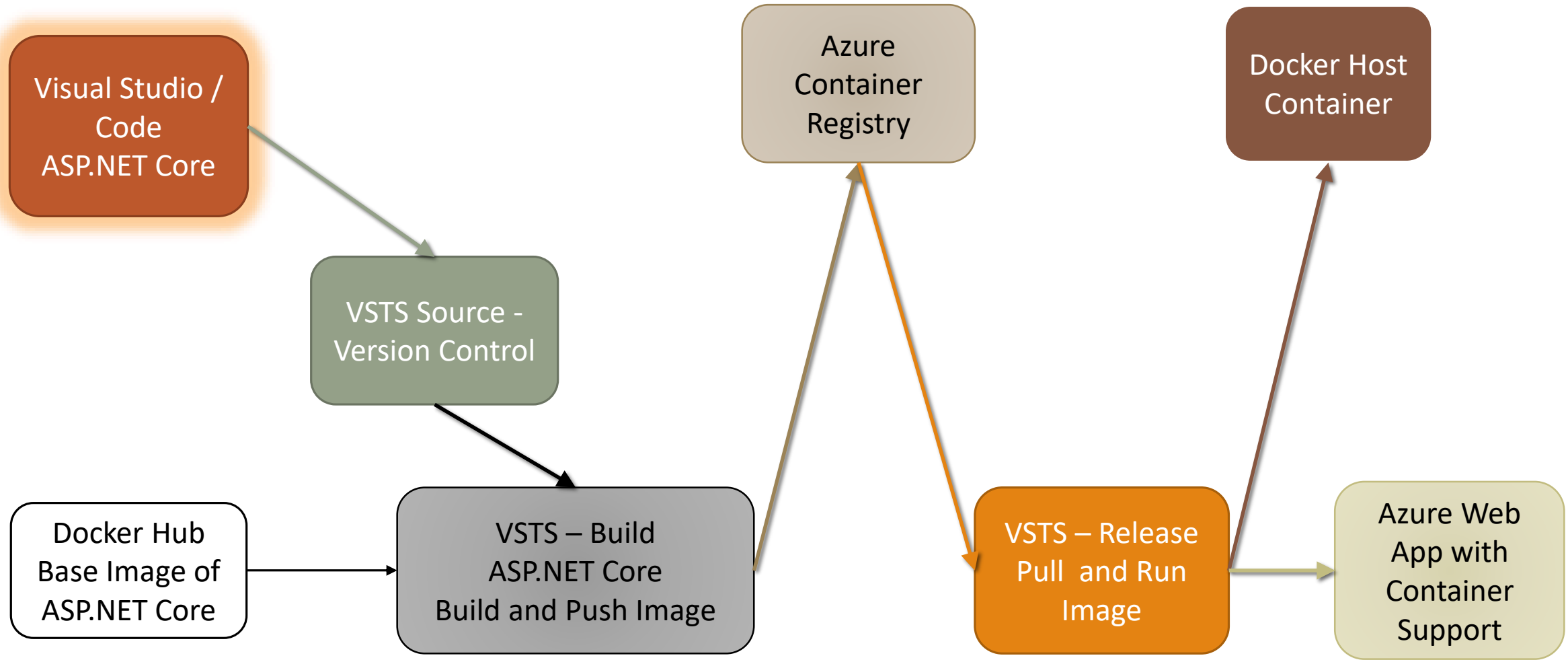


Containers - Docker



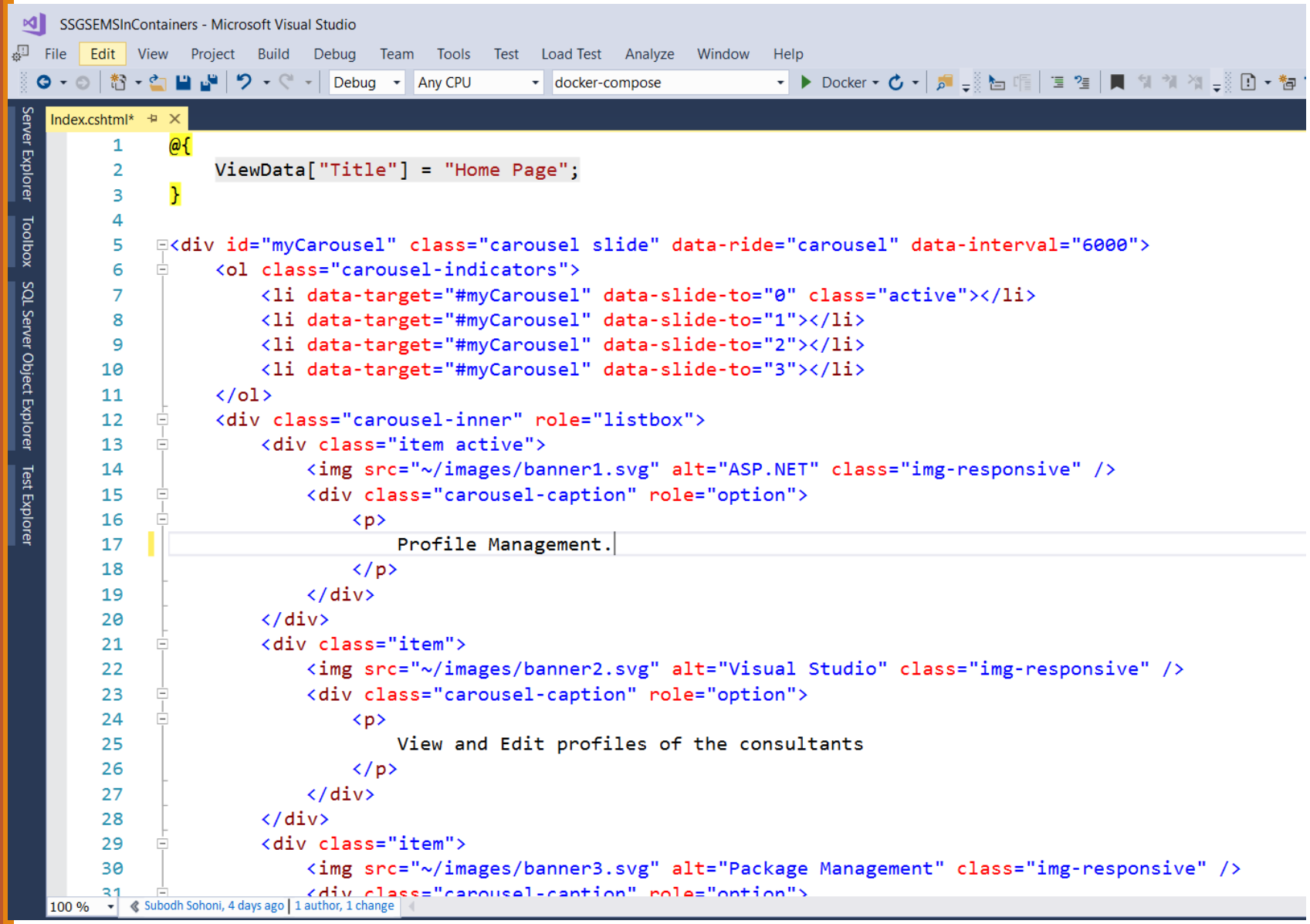
Server with Virtual Machines

Server with Docker Containers


Visual Studio / Code ASP.NET Core
VSTS Source - Version Control
Docker Hub Base Image of ASP.NET Core
VSTS – Build ASP.NET Core Build and Push Image
Azure Container Registry
VSTS – Release Pull and Run Image
Docker Host Container
Azure Web App with Container Support

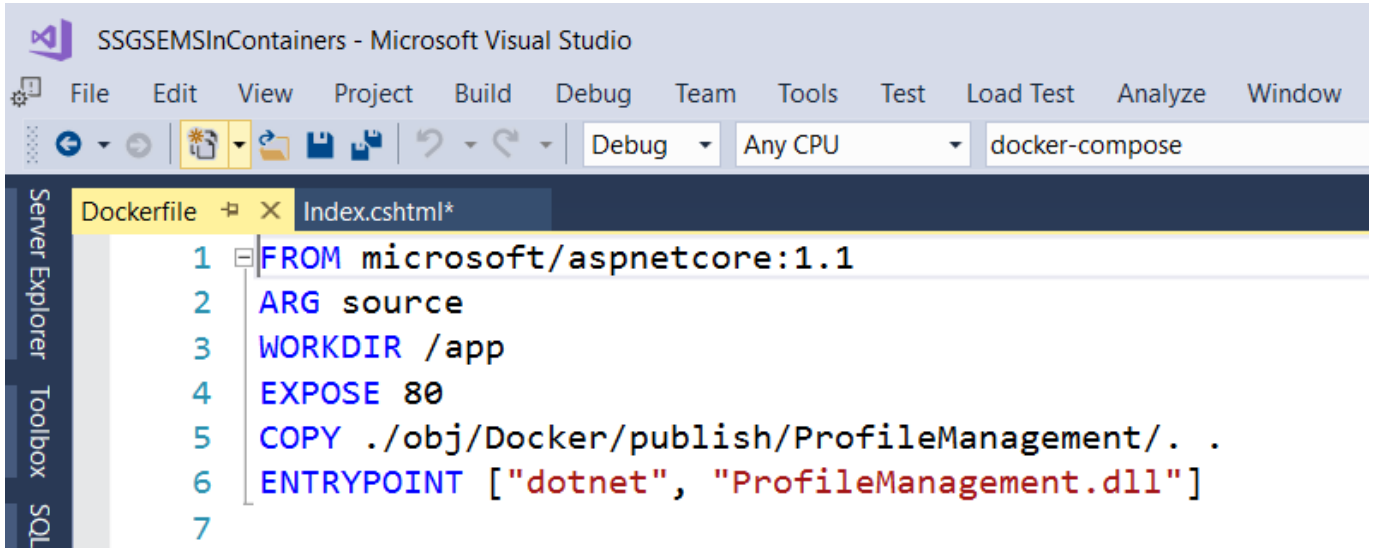
Code

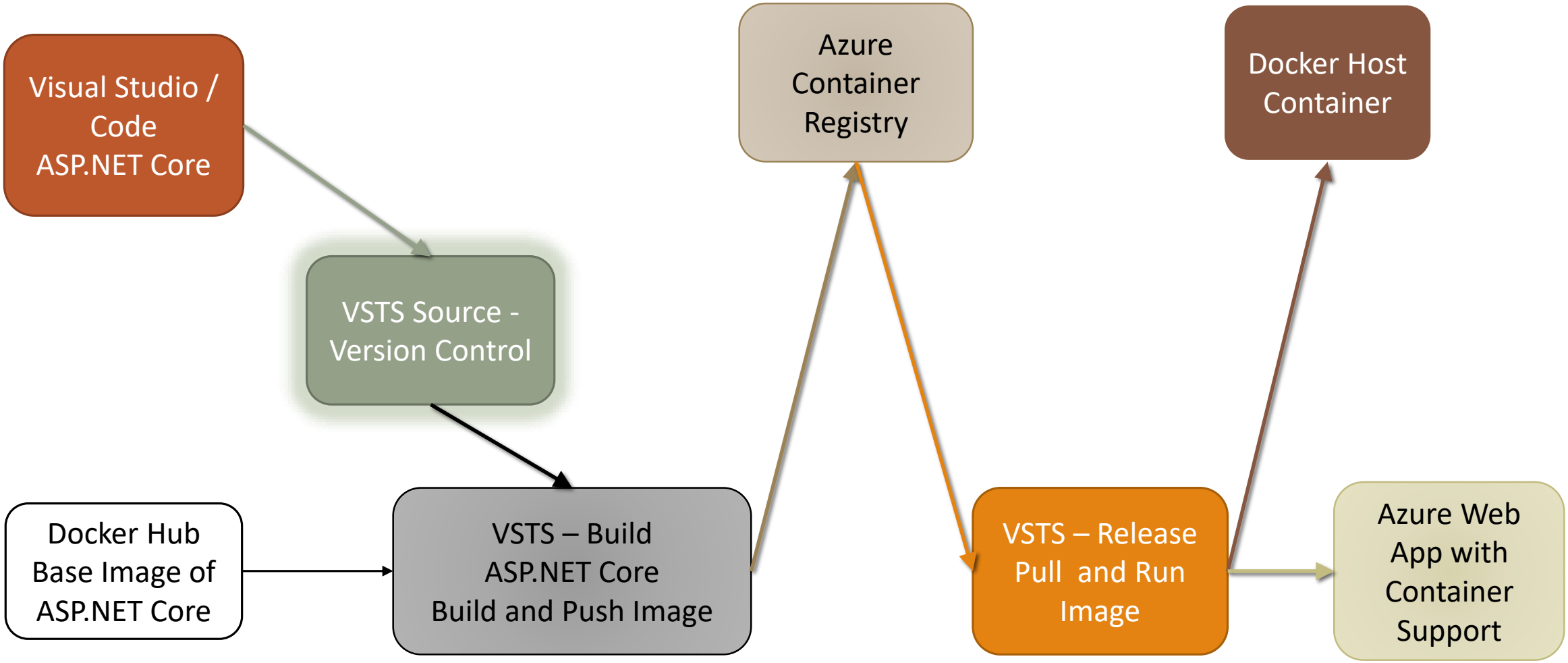
- ASP.NET Core
- Profile Editor service



Dockerfile

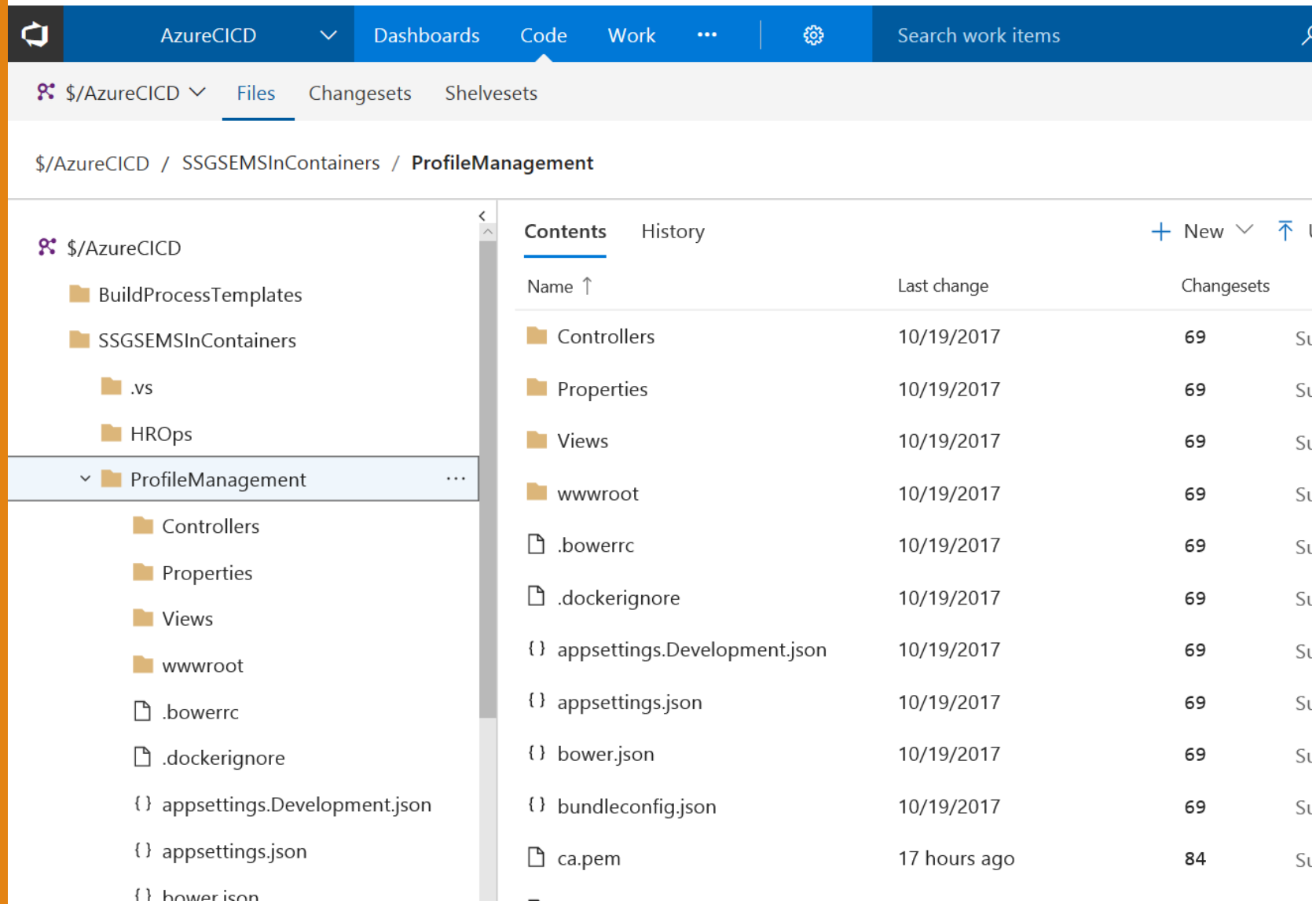
- FROM – Base image pulled from Docker Hub
- ARG – Arguments to be passed when image is built
- EXPOSE – Port on which the service is available from container
- COPY – File / Folder to be copied from Build Agent to Image
- ENTRYPOINT – Service to be started when the container is created (Run)

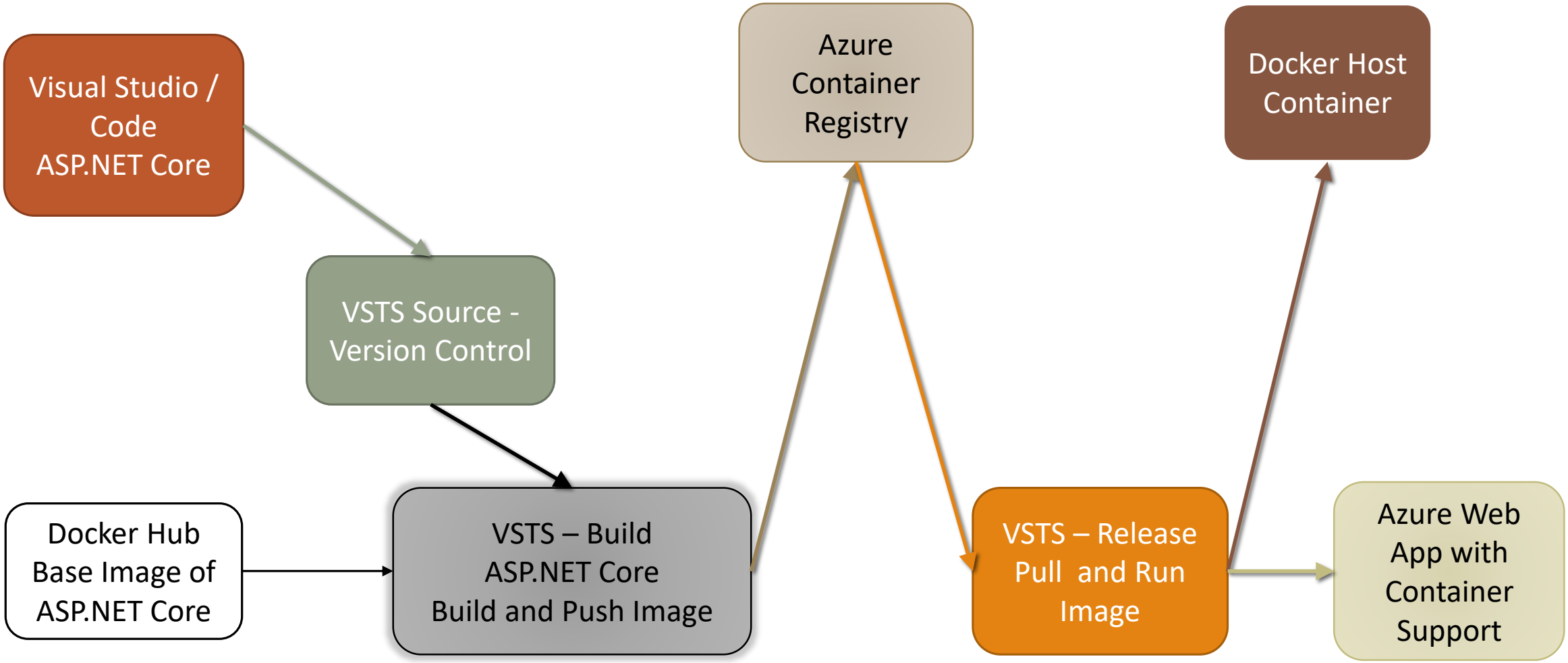



Visual Studio / Code ASP.NET Core
VSTS Source - Version Control
Docker Hub Base Image of ASP.NET Core
VSTS – Build ASP.NET Core Build and Push Image
Azure Container Registry
VSTS – Release Pull and Run Image
Docker Host Container
Azure Web App with Container Support

VSTS SCM

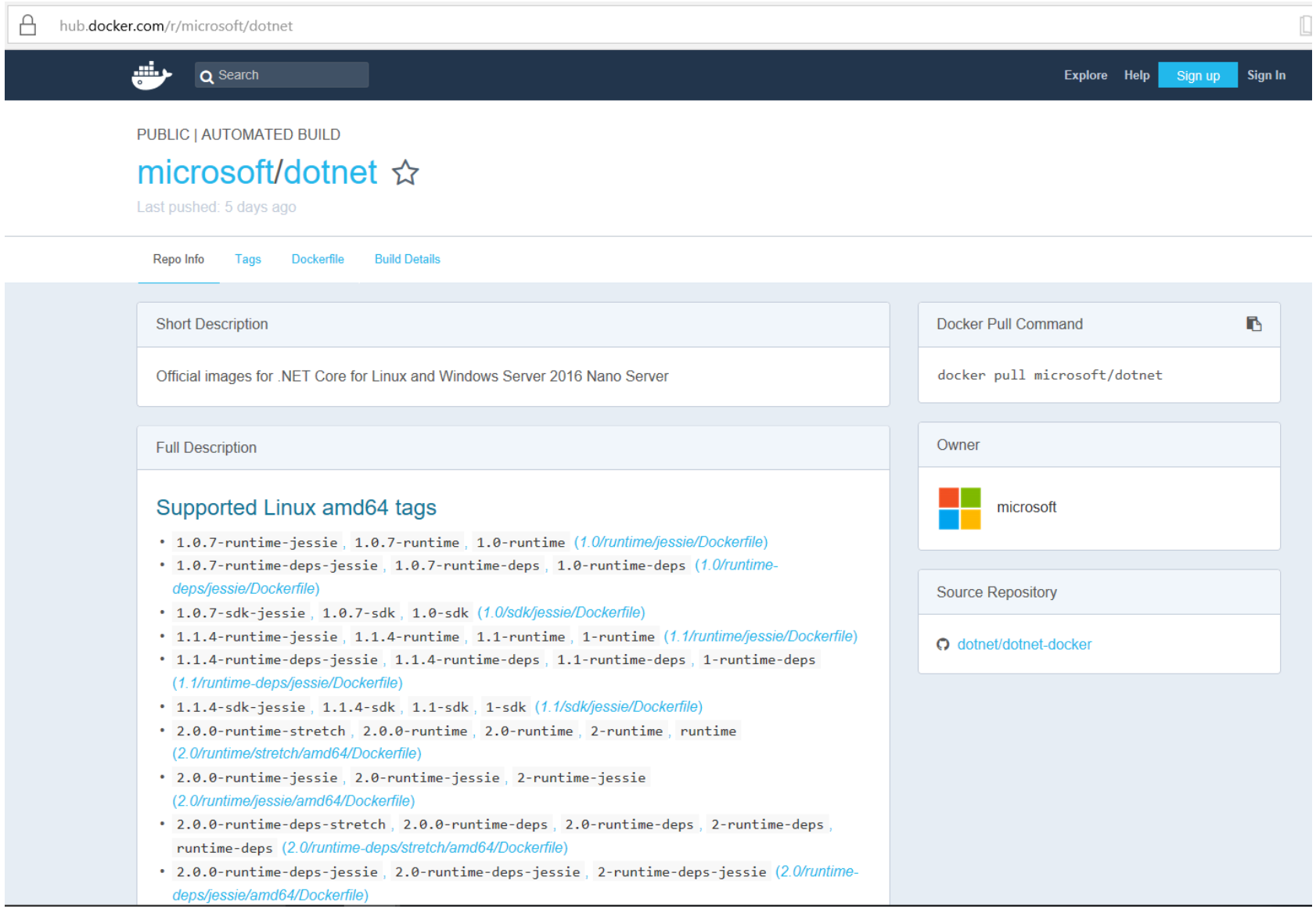
- TFVC
- GIT




Visual Studio / Code ASP.NET Core
VSTS Source - Version Control
Docker Hub Base Image of ASP.NET Core
VSTS – Build ASP.NET Core Build and Push Image
Azure Container Registry
VSTS – Release Pull and Run Image
Docker Host Container
Azure Web App with Container Support

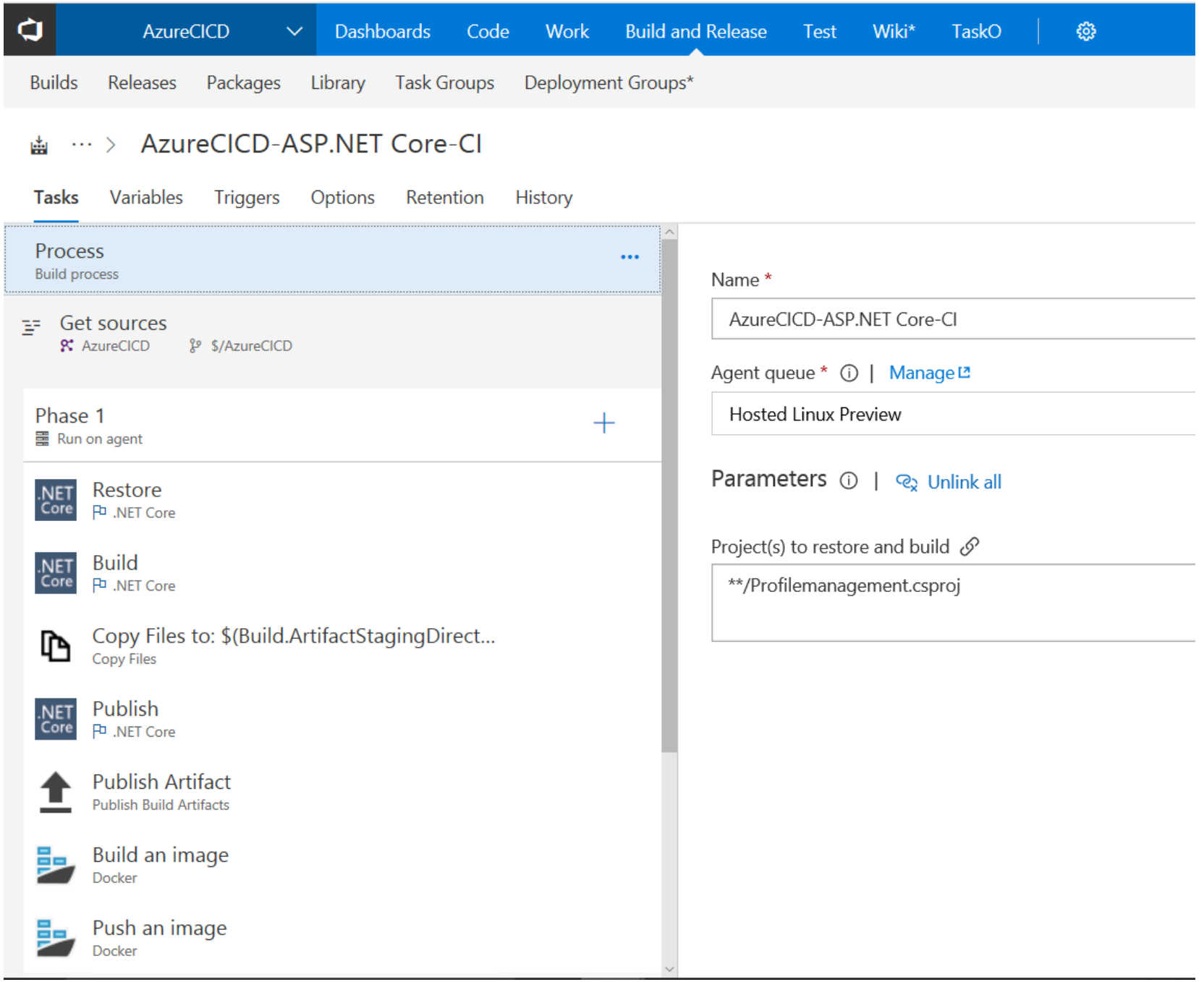
Docker Hub

- Shared Images

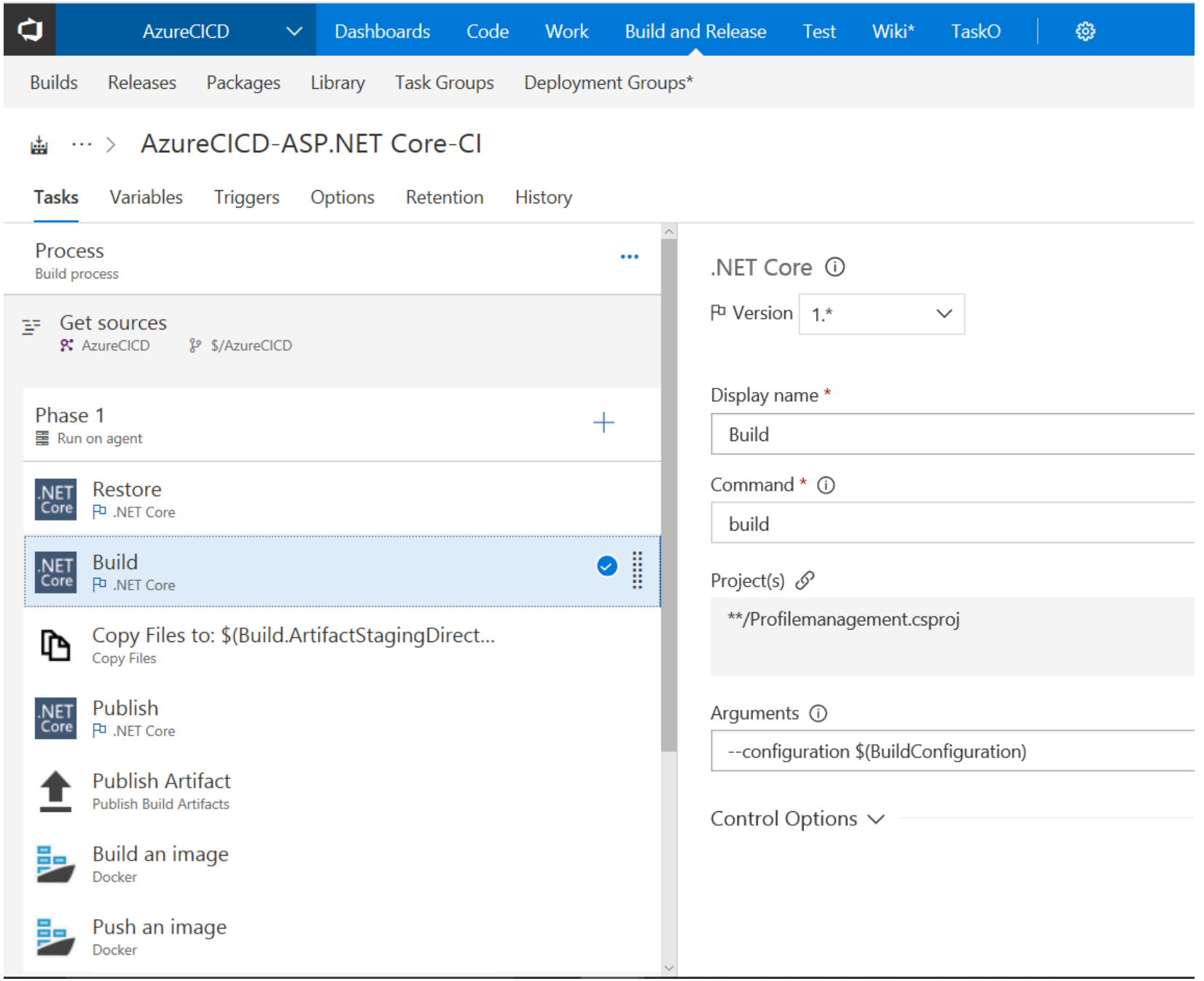


Build - Process

- Name
- Agent Queue – Hosted Linux
- Project – One Service to Build

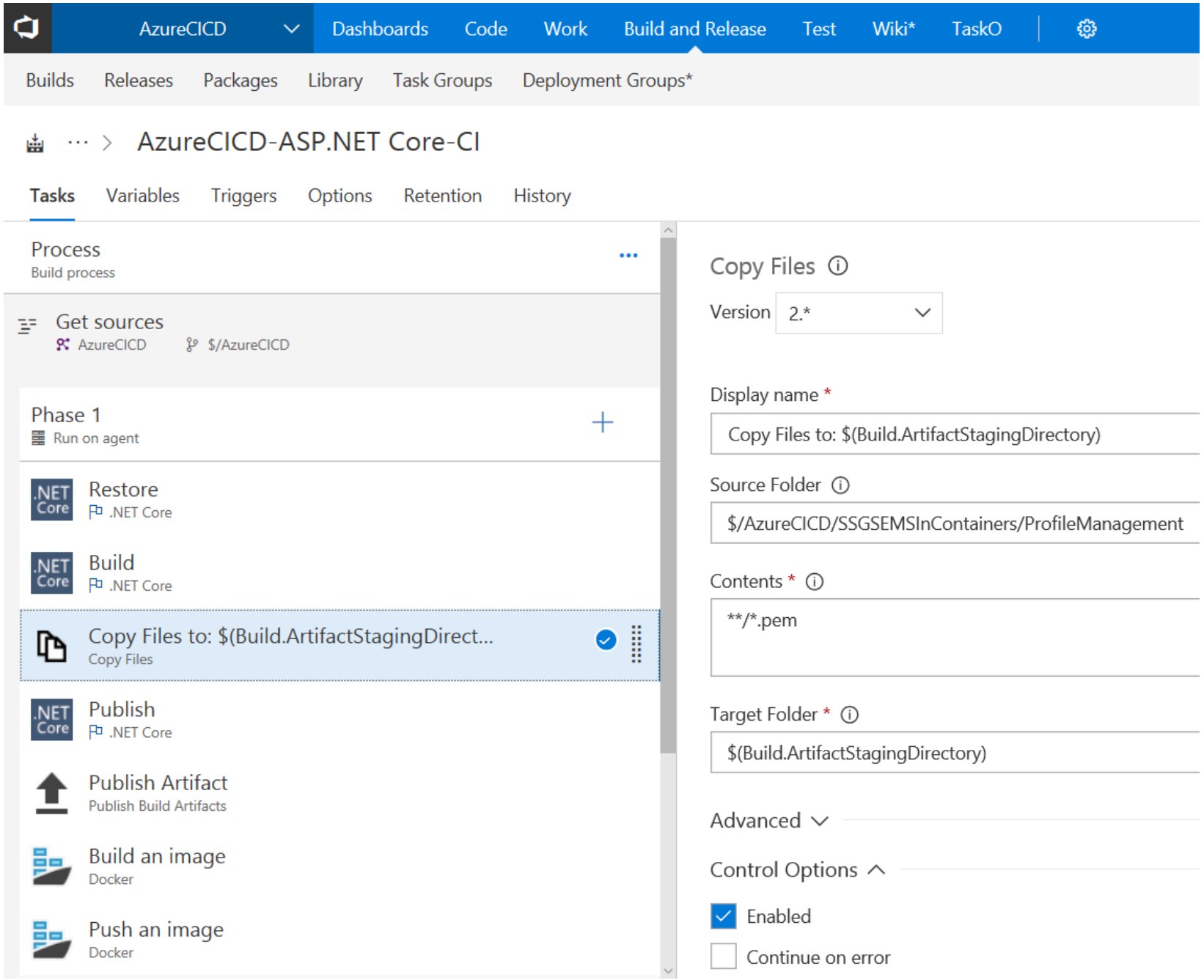


Build - Project



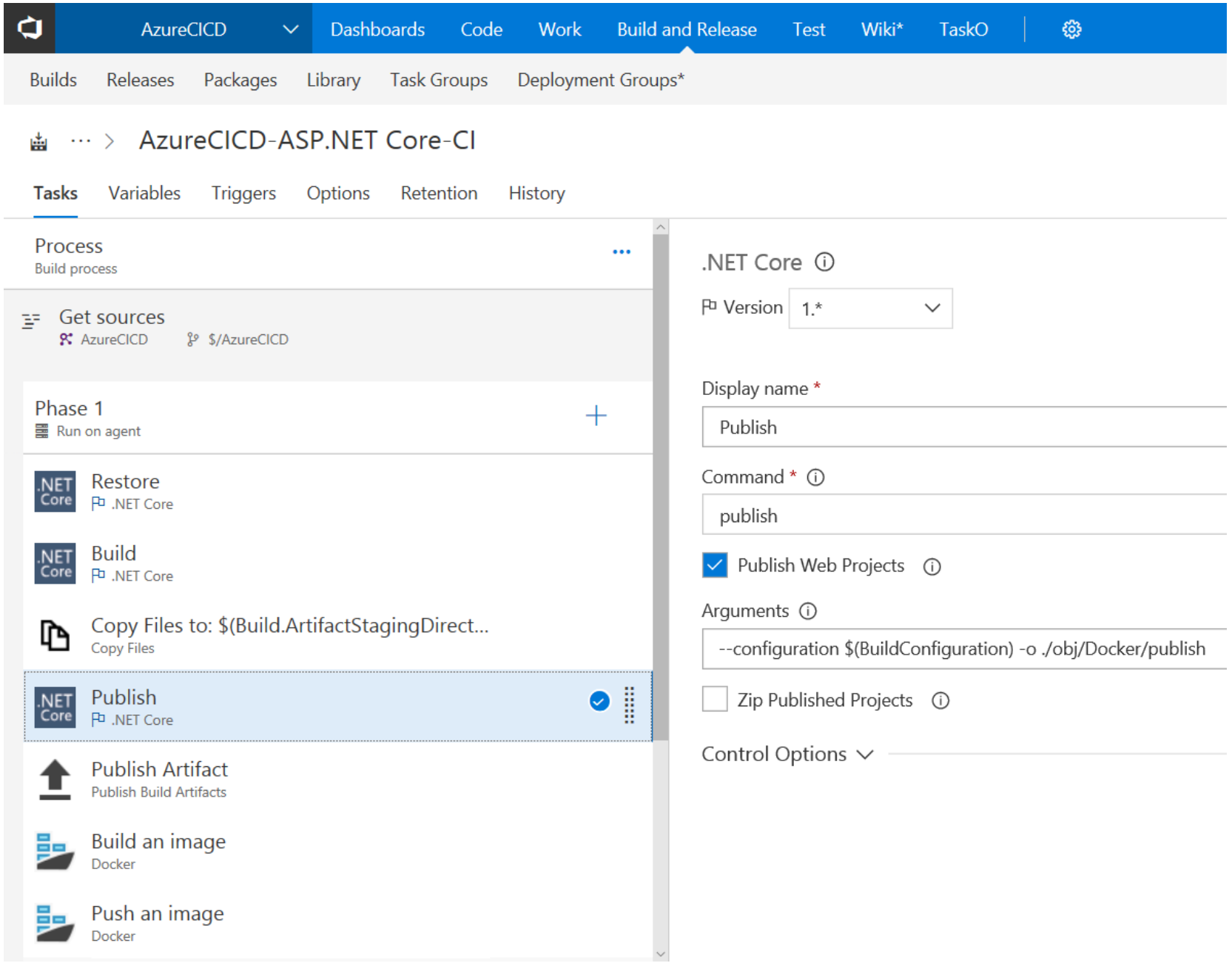
Build – Copy PEM Files

- Certificate for Docker Host

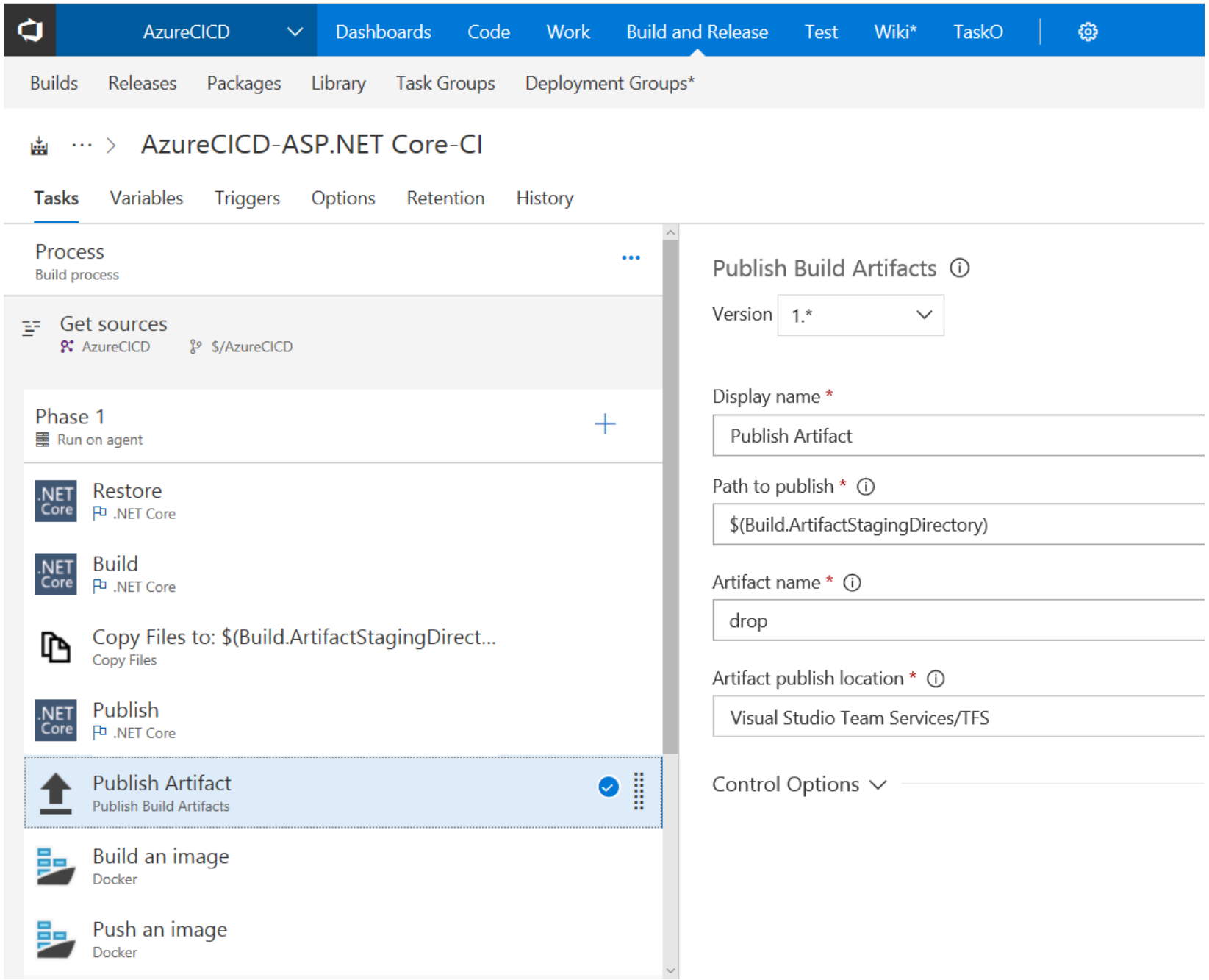


Build - Publish

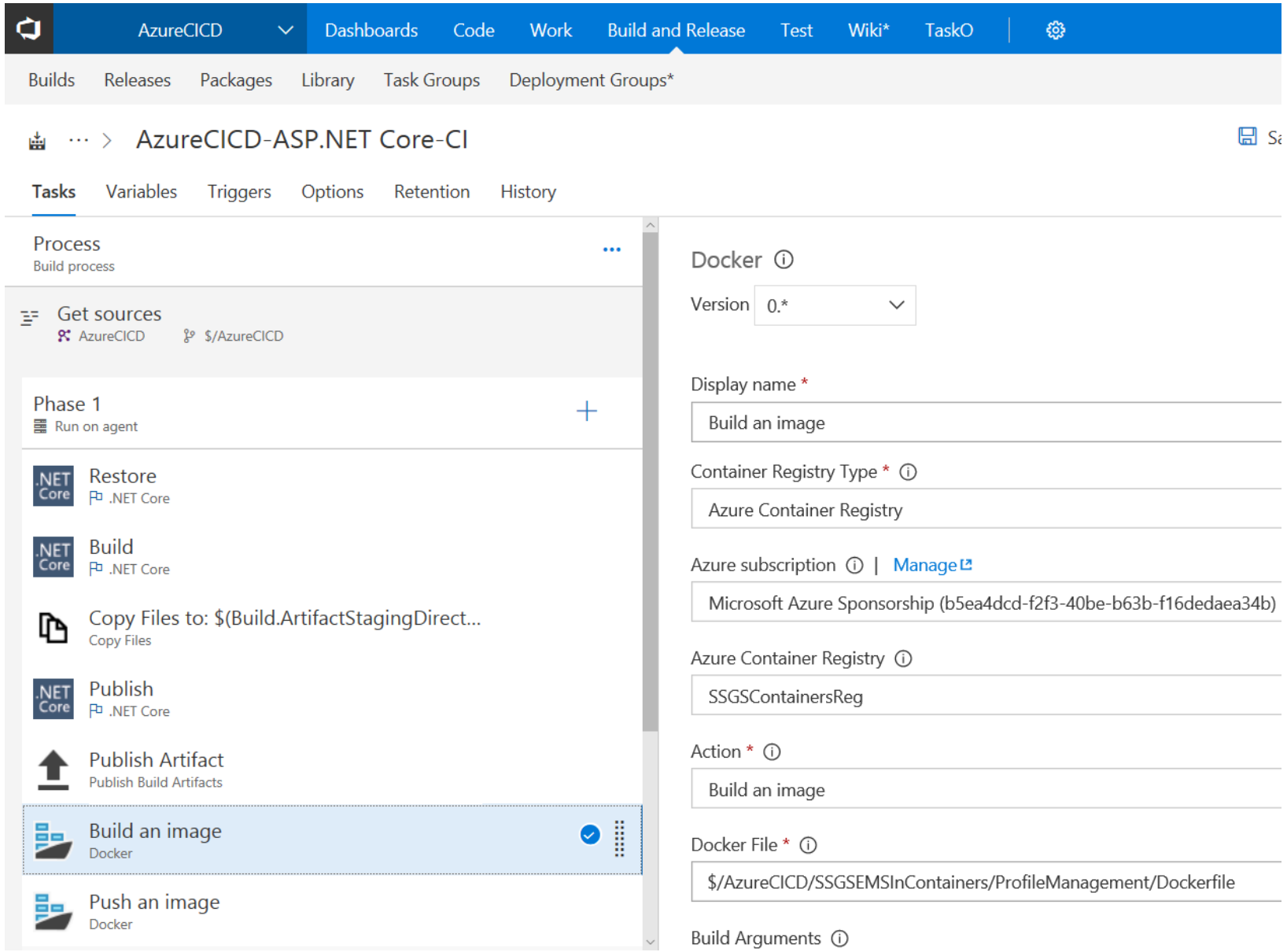
- Output Argument set to same as copy statement in Dockerfile



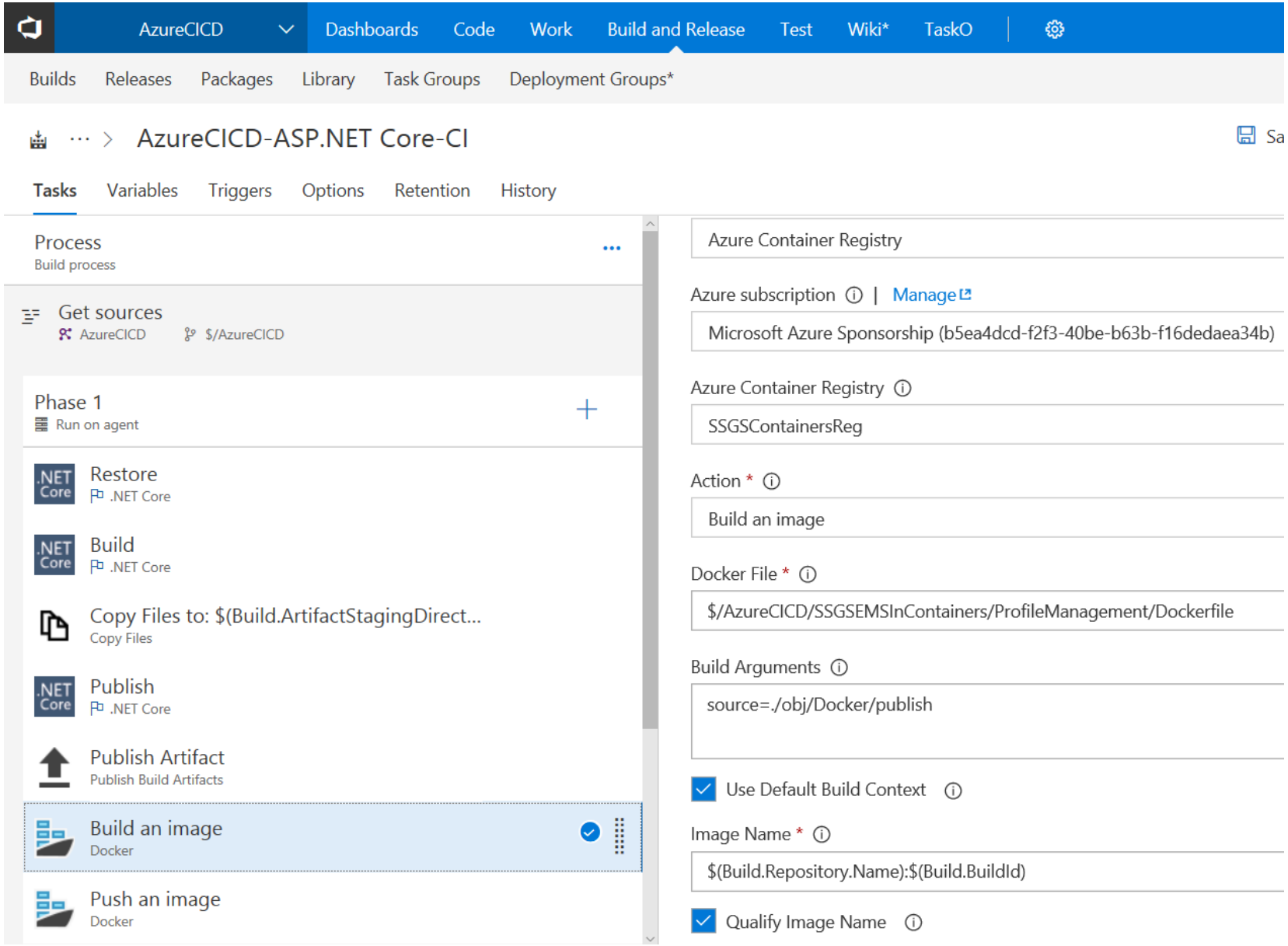
Build – Publish Artifact - PEM



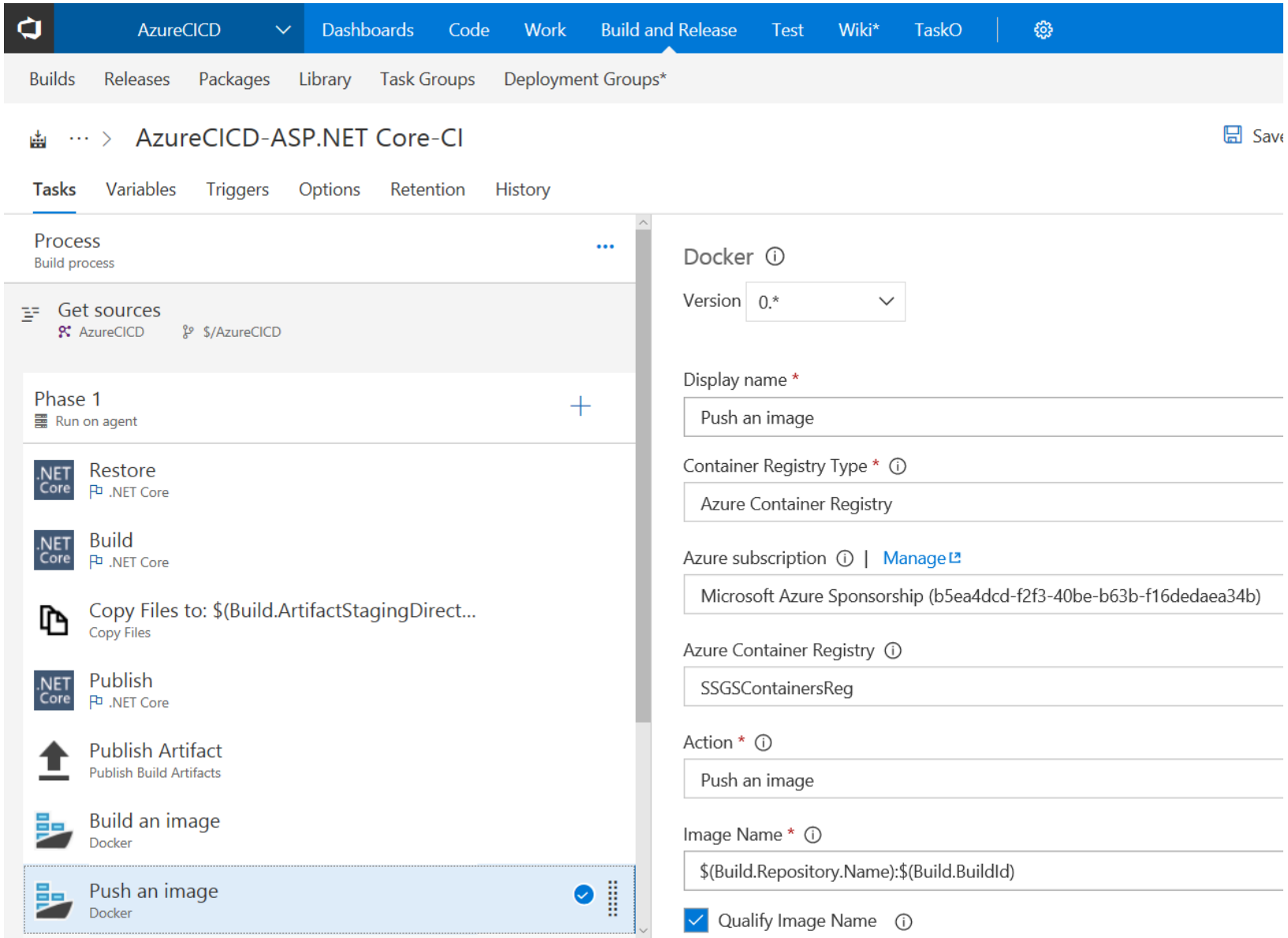
Build – Image - 1

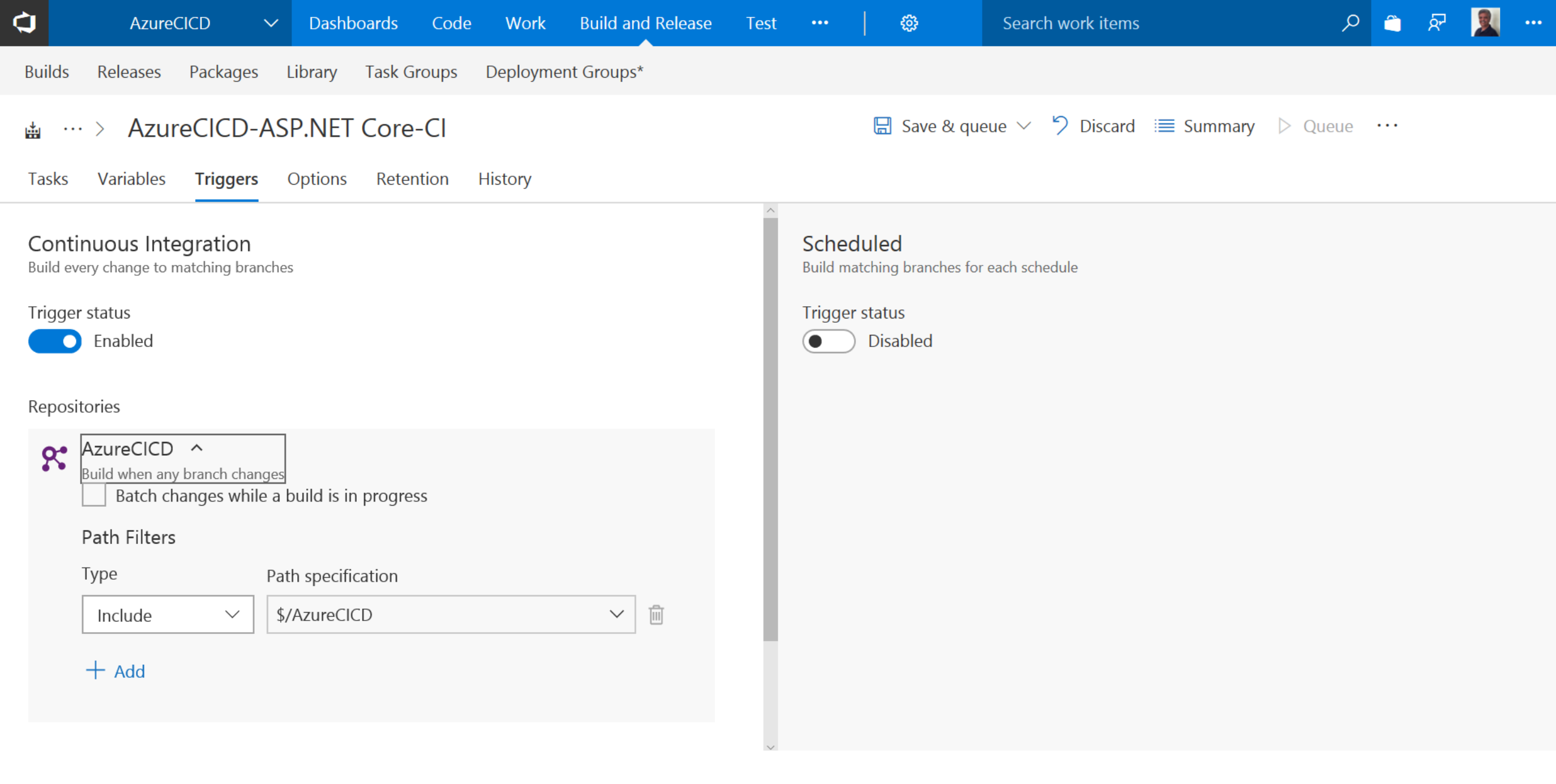


Build – Image - 2

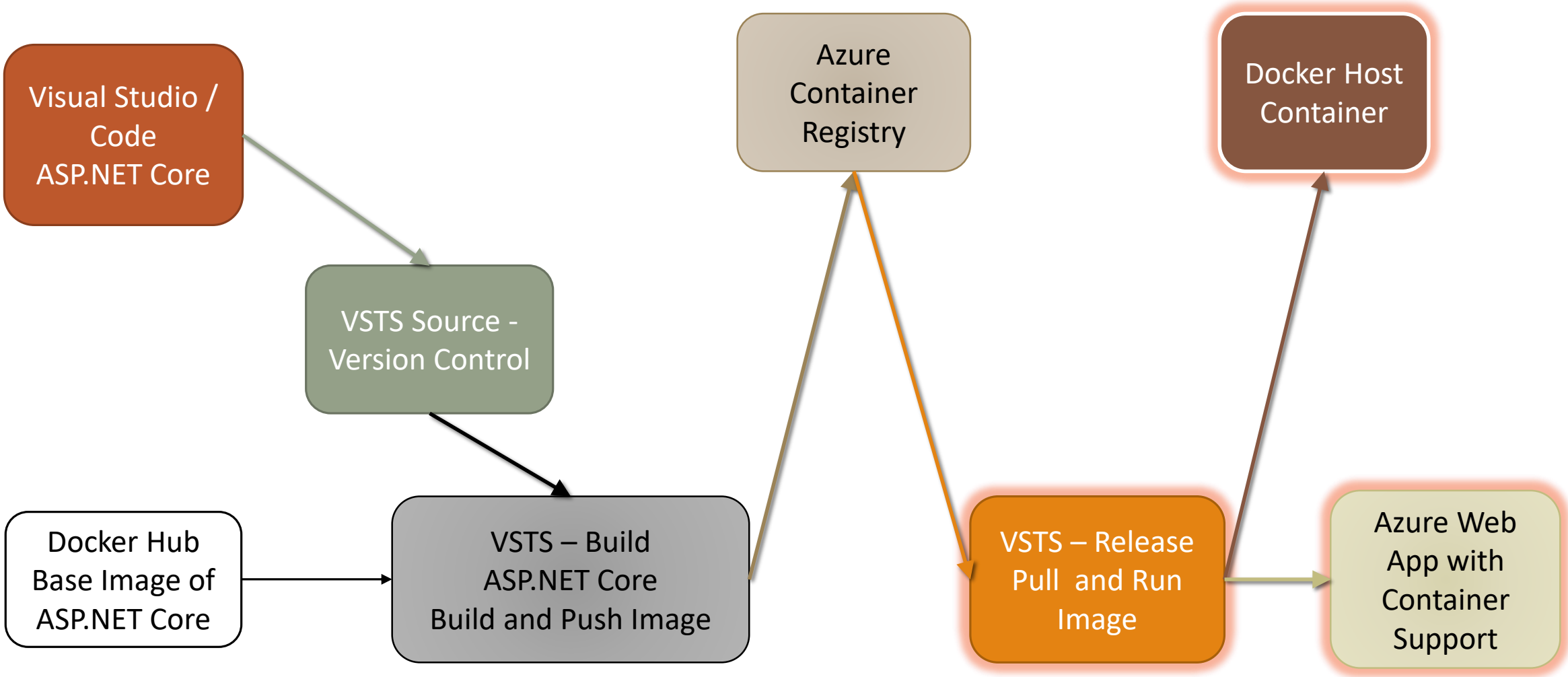


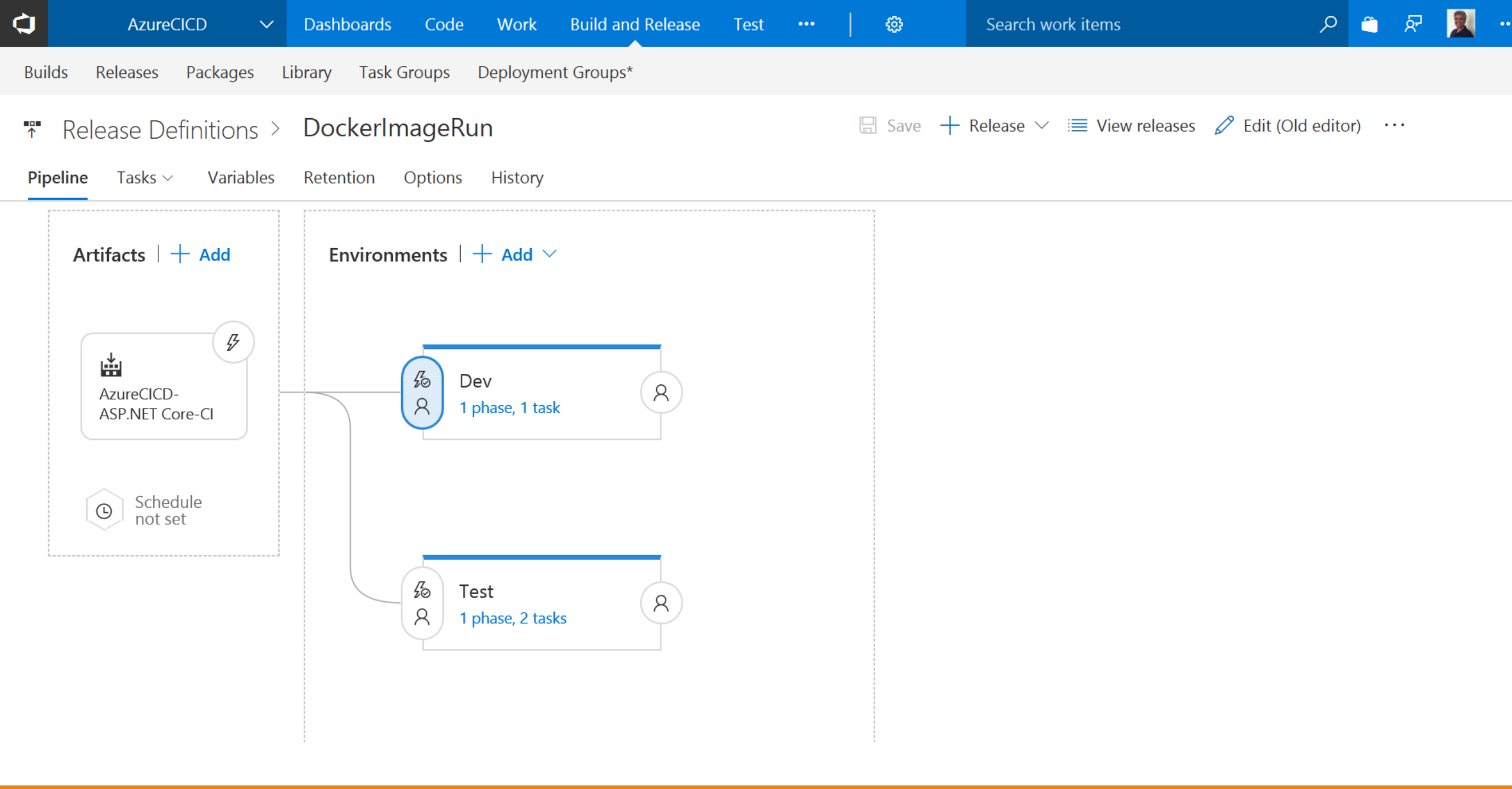
Build – Push Image



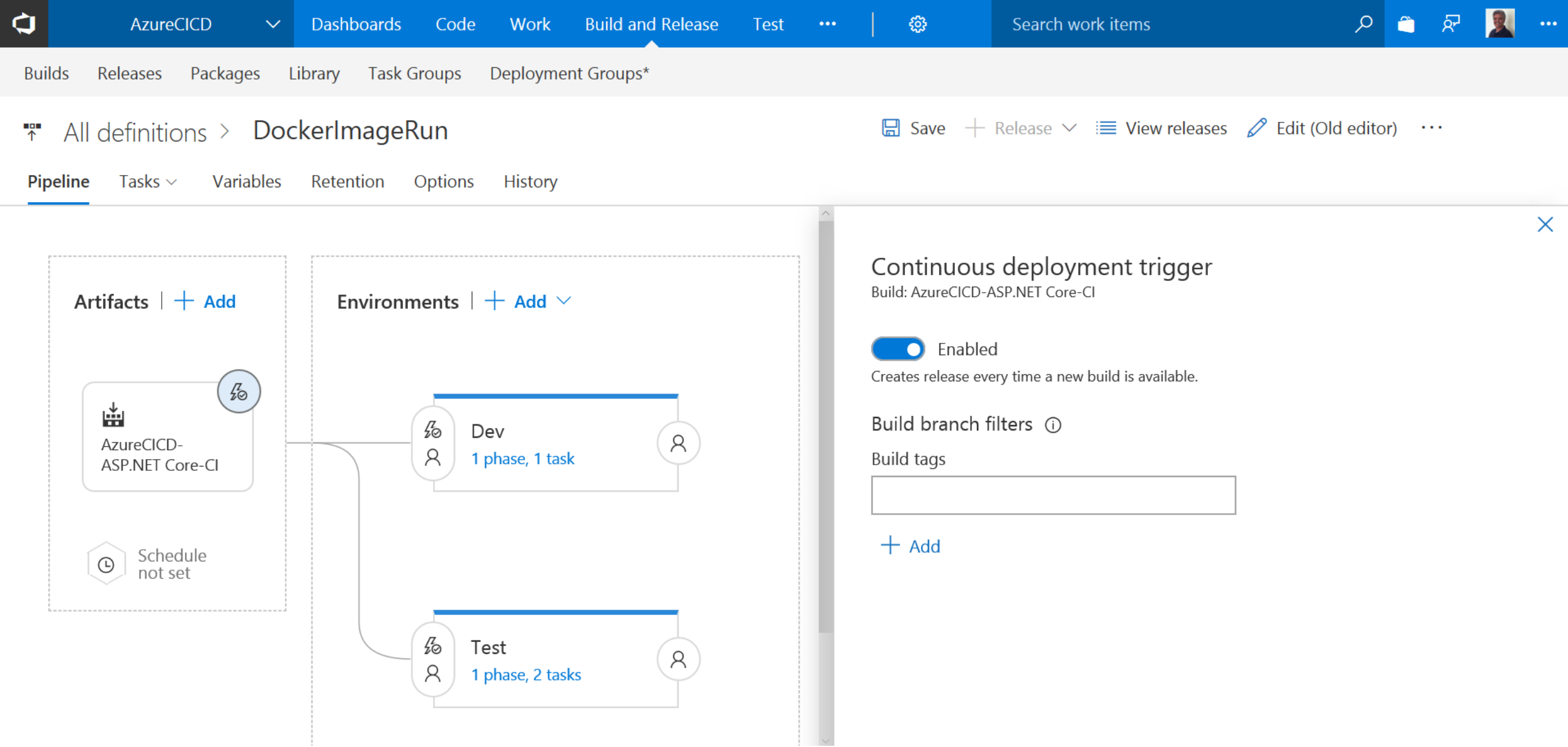


Continuous Integration Trigger

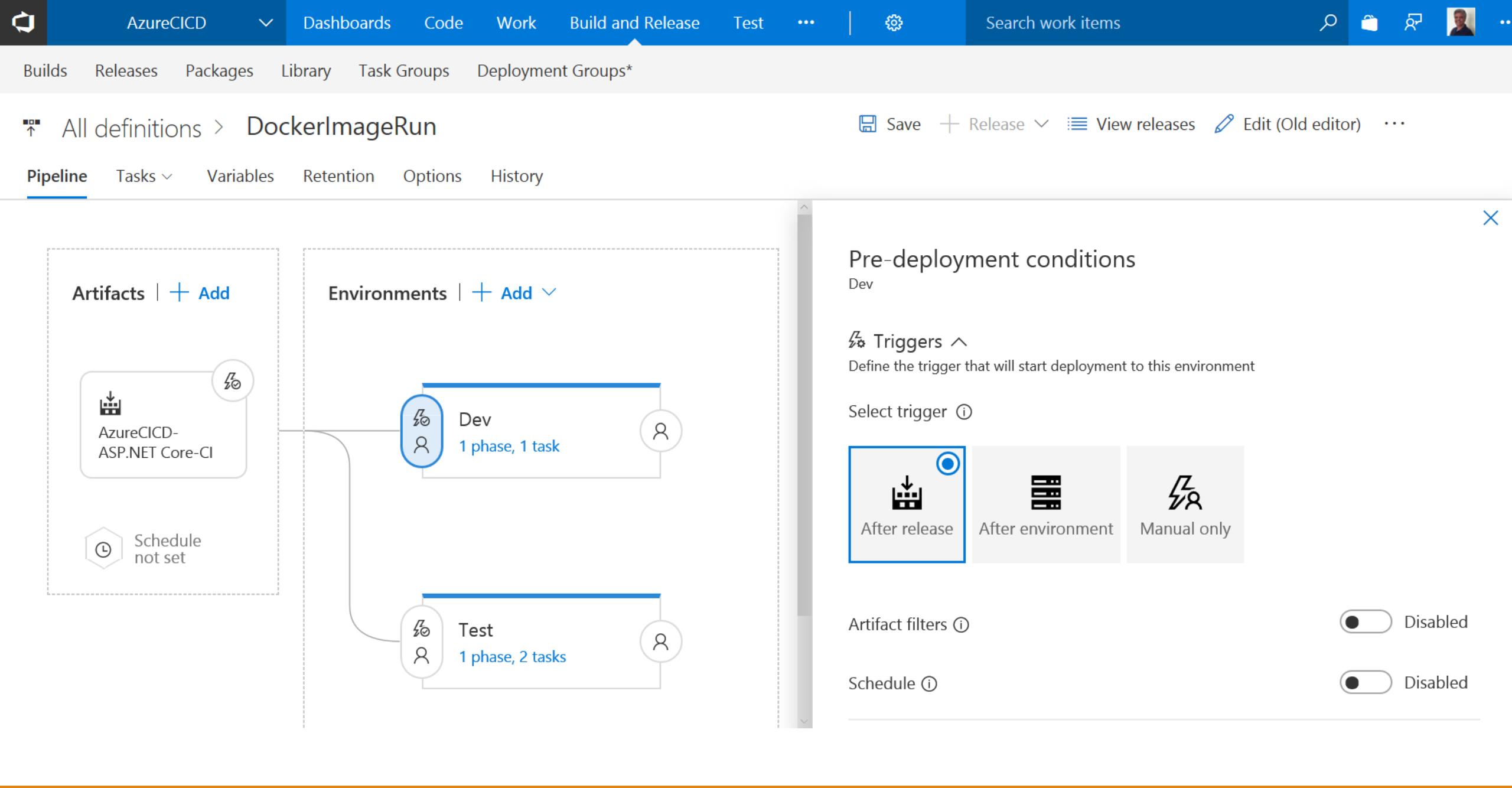

Visual Studio / Code ASP.NET Core
VSTS Source - Version Control
Docker Hub Base Image of ASP.NET Core
VSTS – Build ASP.NET Core Build and Push Image
Azure Container Registry
VSTS – Release Pull and Run Image
Docker Host Container
Azure Web App with Container Support



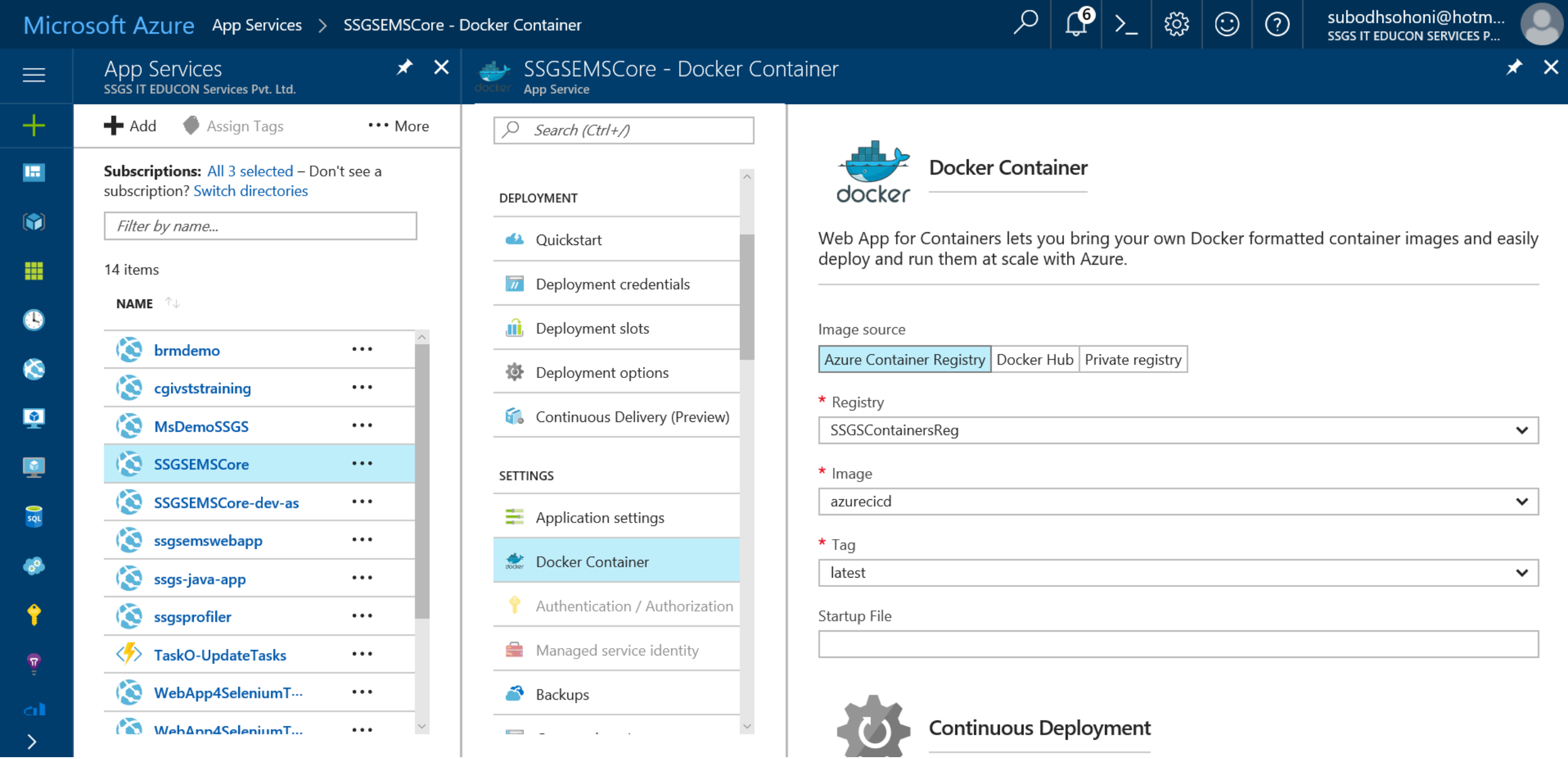
VSTS Release Definition Pipeline



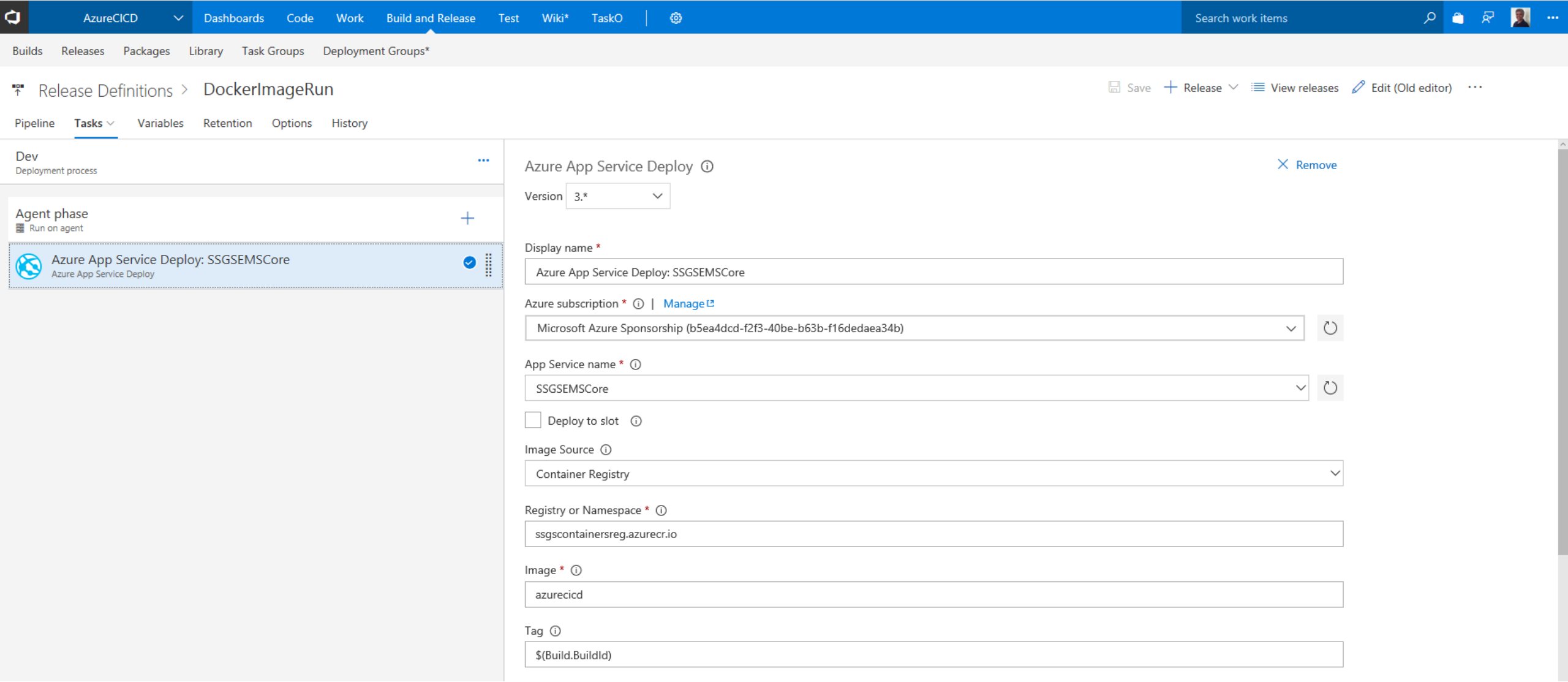
Continuous Deployment – Release Trigger



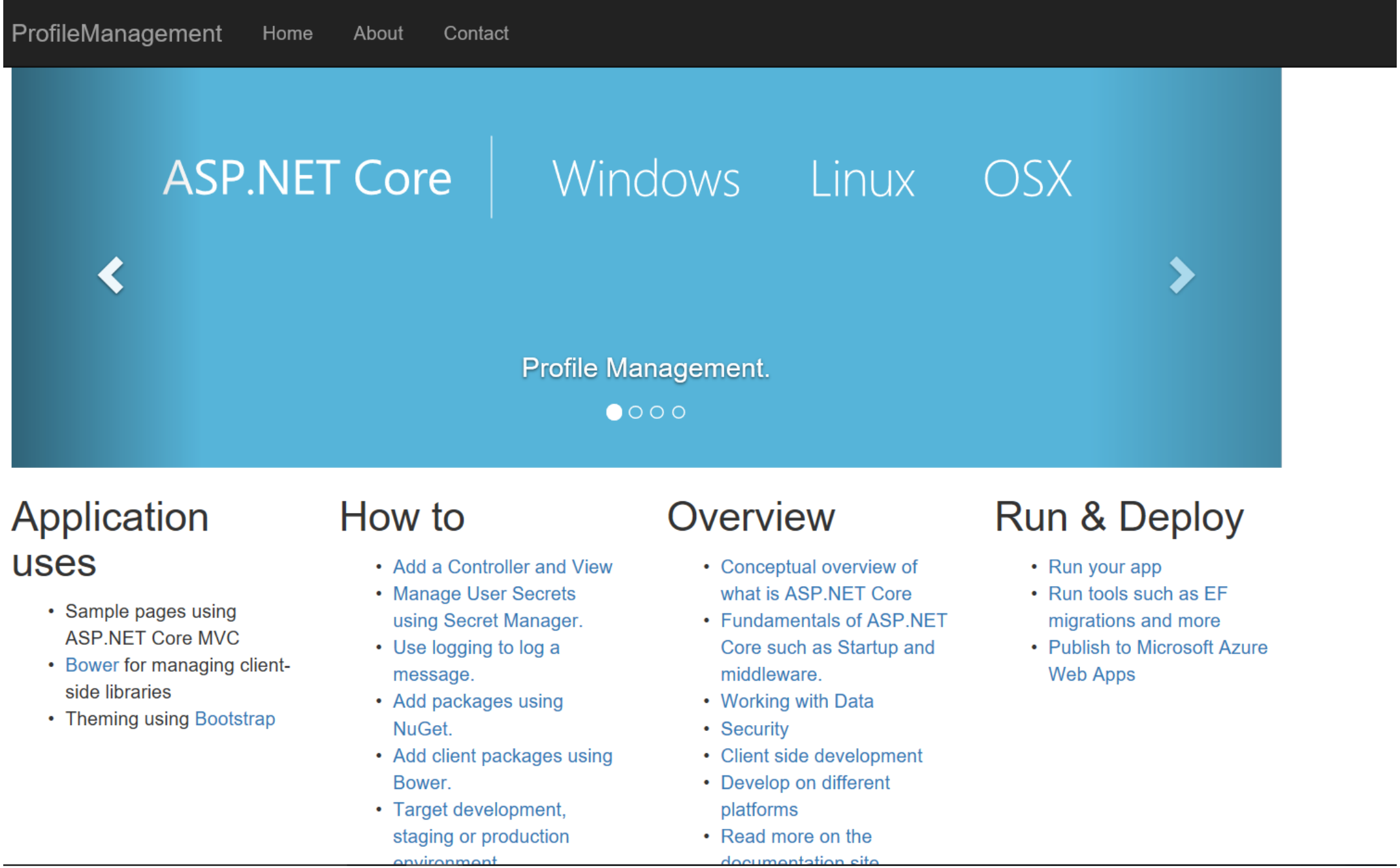
Continuous Deployment – For Dev Environment



Azure App Service Containerized Web App



Deployment of container using image from Azure Container Registry to Azure Web App



Application uses

- Sample pages using ASP.NET Core MVC
- Bower for managing client-side libraries
- Theming using Bootstrap

How to

- Add a Controller and View
- Manage User Secrets using Secret Manager.
- Use logging to log a message.
- Add packages using NuGet.
- Add client packages using Bower.
- Target development, staging or production environment

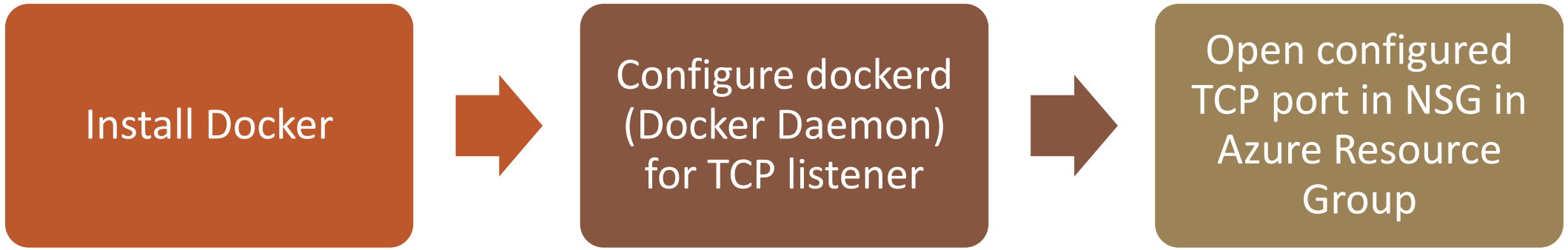
Overview

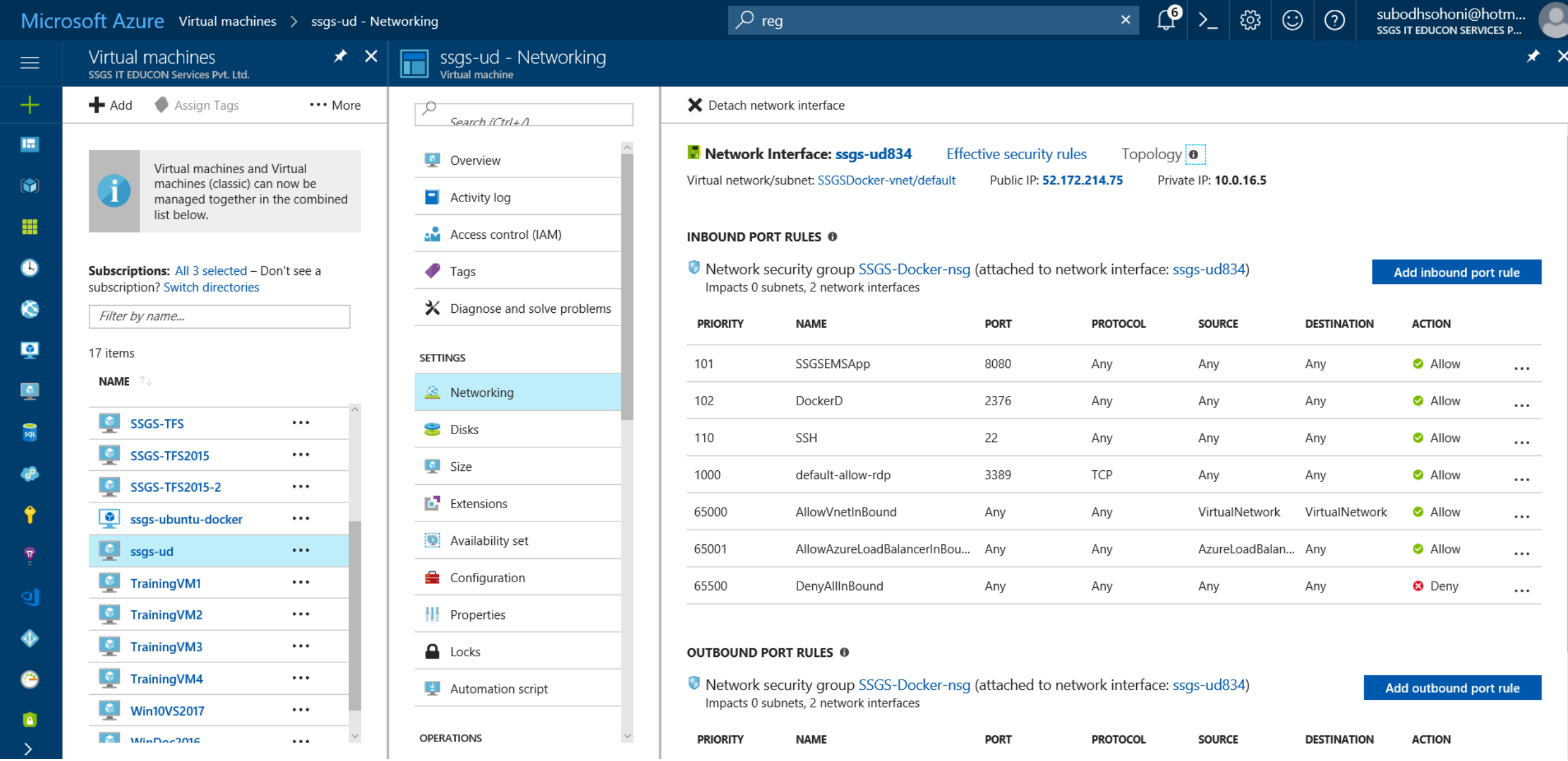
- Conceptual overview of what is ASP.NET Core
- Fundamentals of ASP.NET Core such as Startup and middleware.
- Working with Data
- Security
- Client side development
- Develop on different platforms
- Read more on the documentation site

Run & Deploy

- Run your app
- Run tools such as EF migrations and more
- Publish to Microsoft Azure Web Apps

Prepare Linux Azure VM as Docker Host



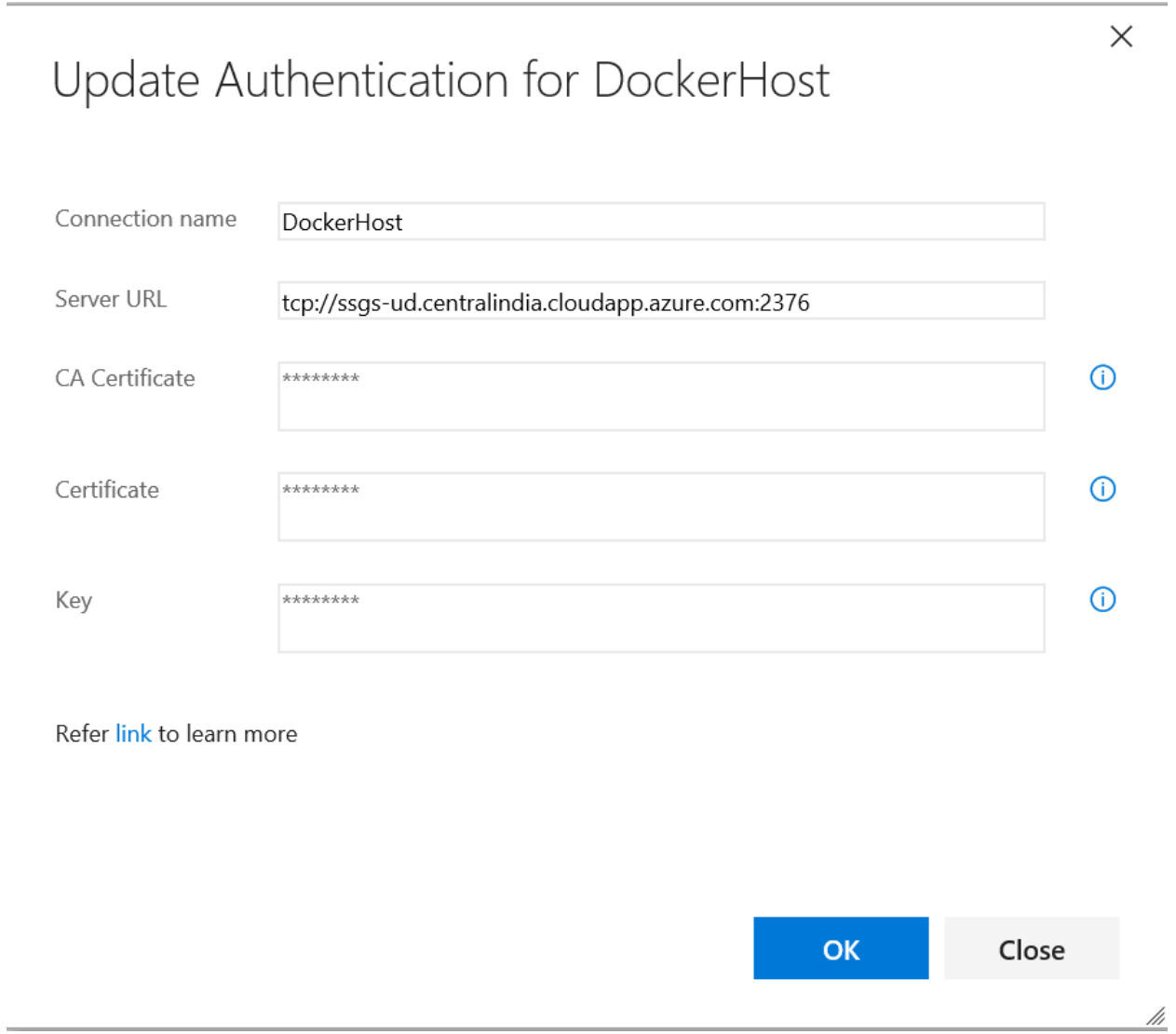


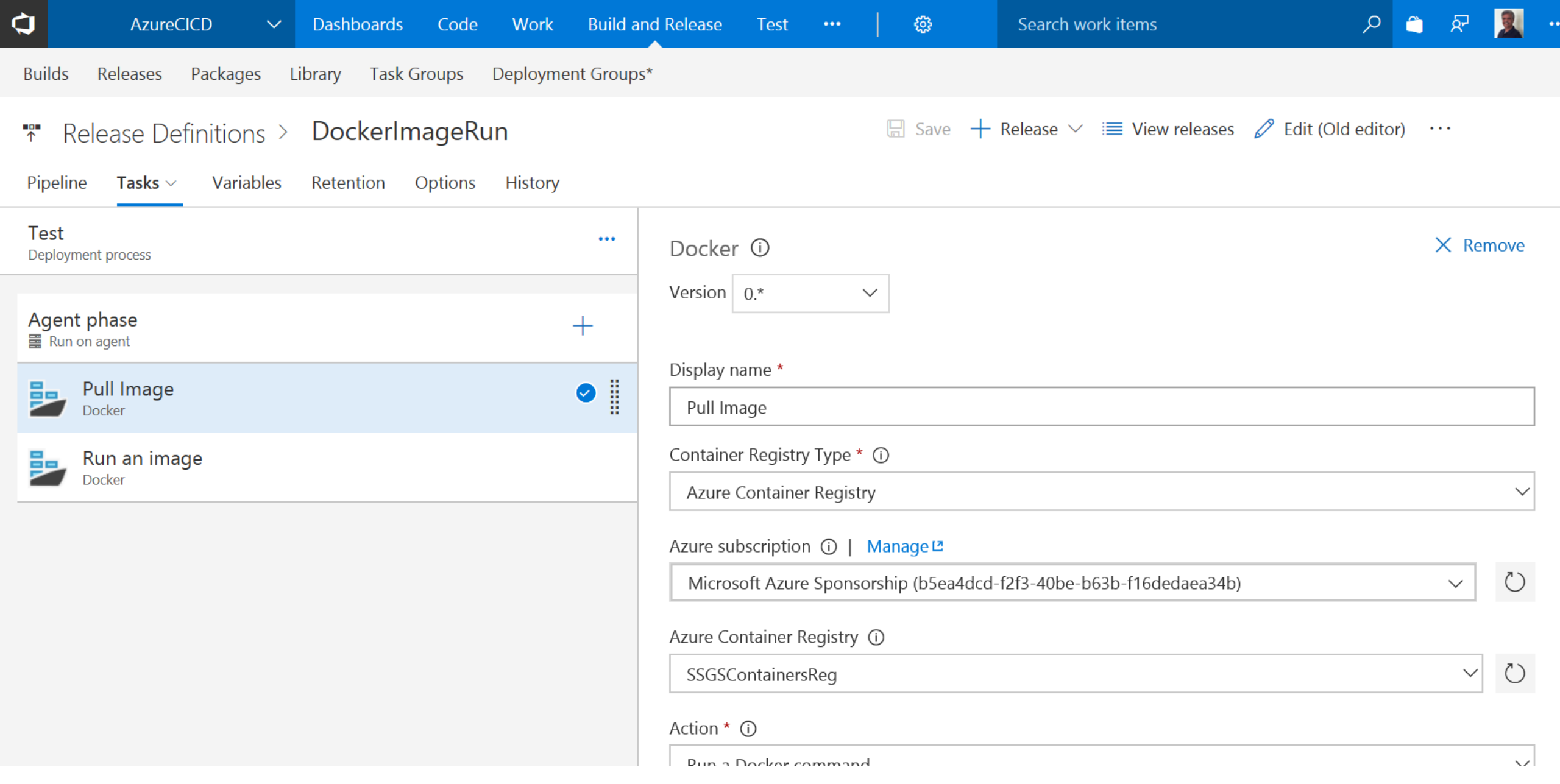
Linux (Ubuntu 16.04 LTS VM in Azure

Docker Host Endpoint

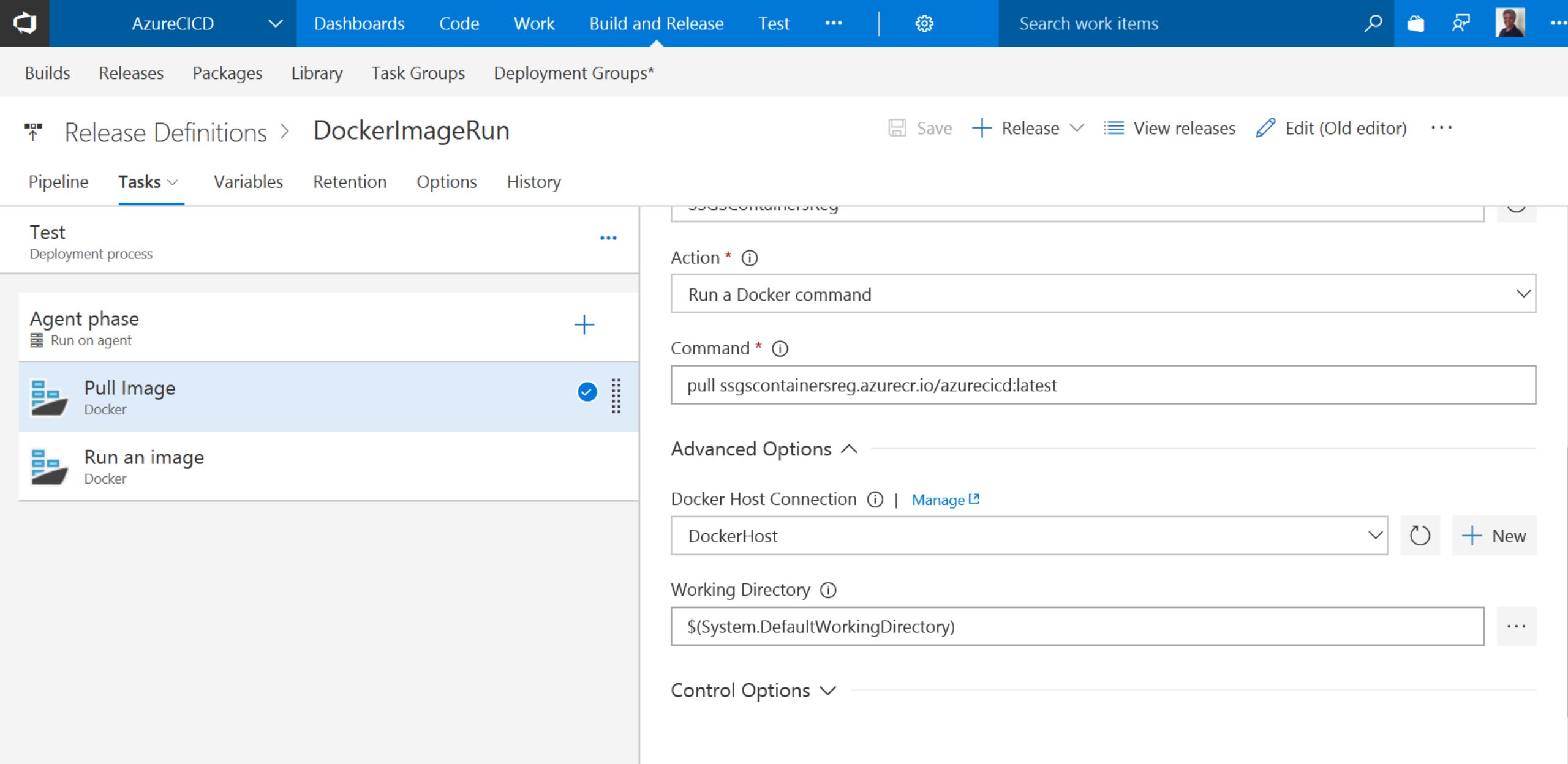
- Server URL: TCP protocol, DNS of Docker Host VM, Port on which dockerd is listening
- To get certificates use guidance provided in: https://docs.docker.com/engine/security/https/
- Mention same TLS details while starting dockerd

```
$ dockerd --tlsverify
          --tlscacert=ca.pem
          --tlscert=cert.pem
          --tlskey=key.pem
          -H=tcp://0.0.0.0:2376
```

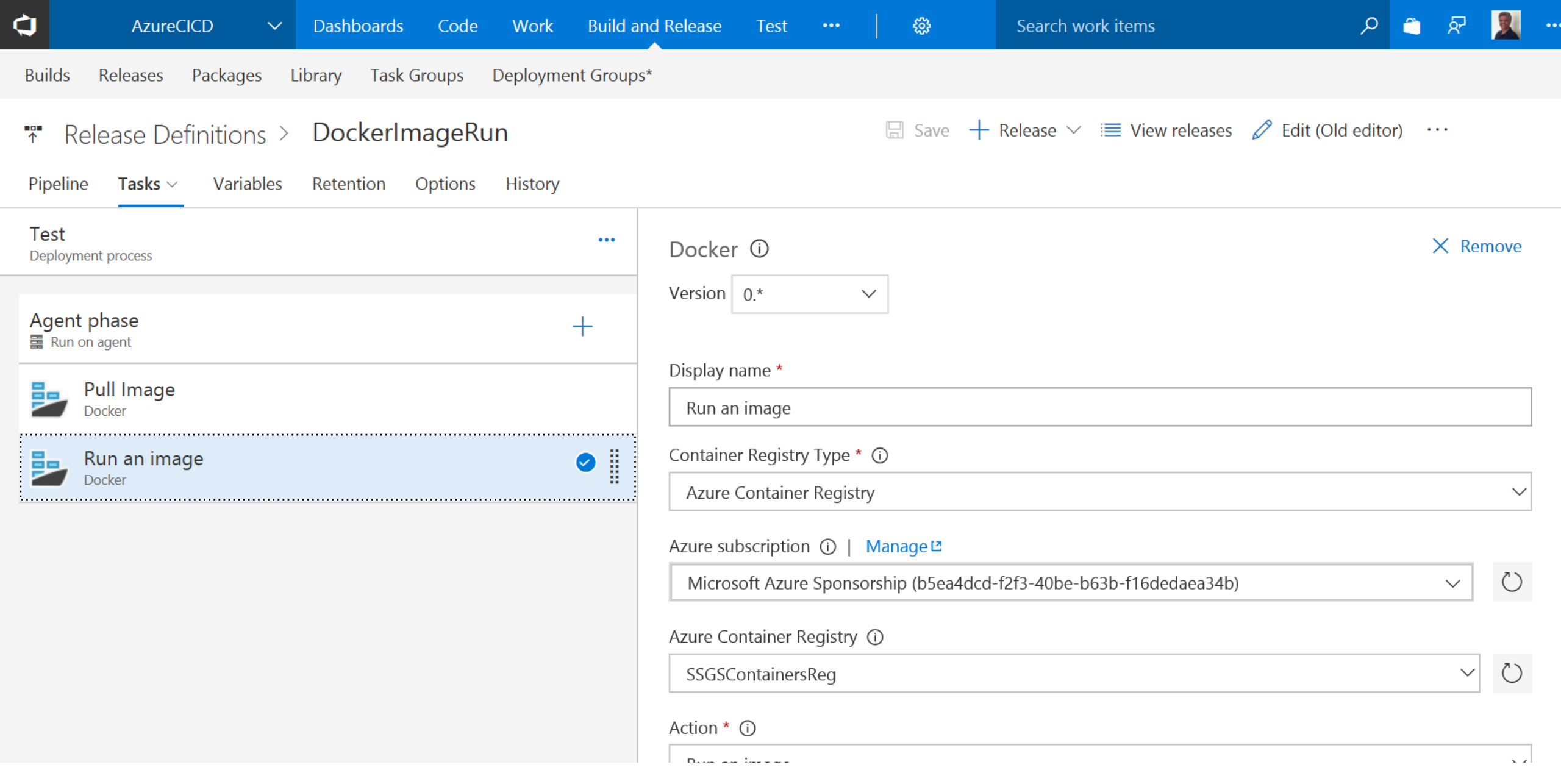




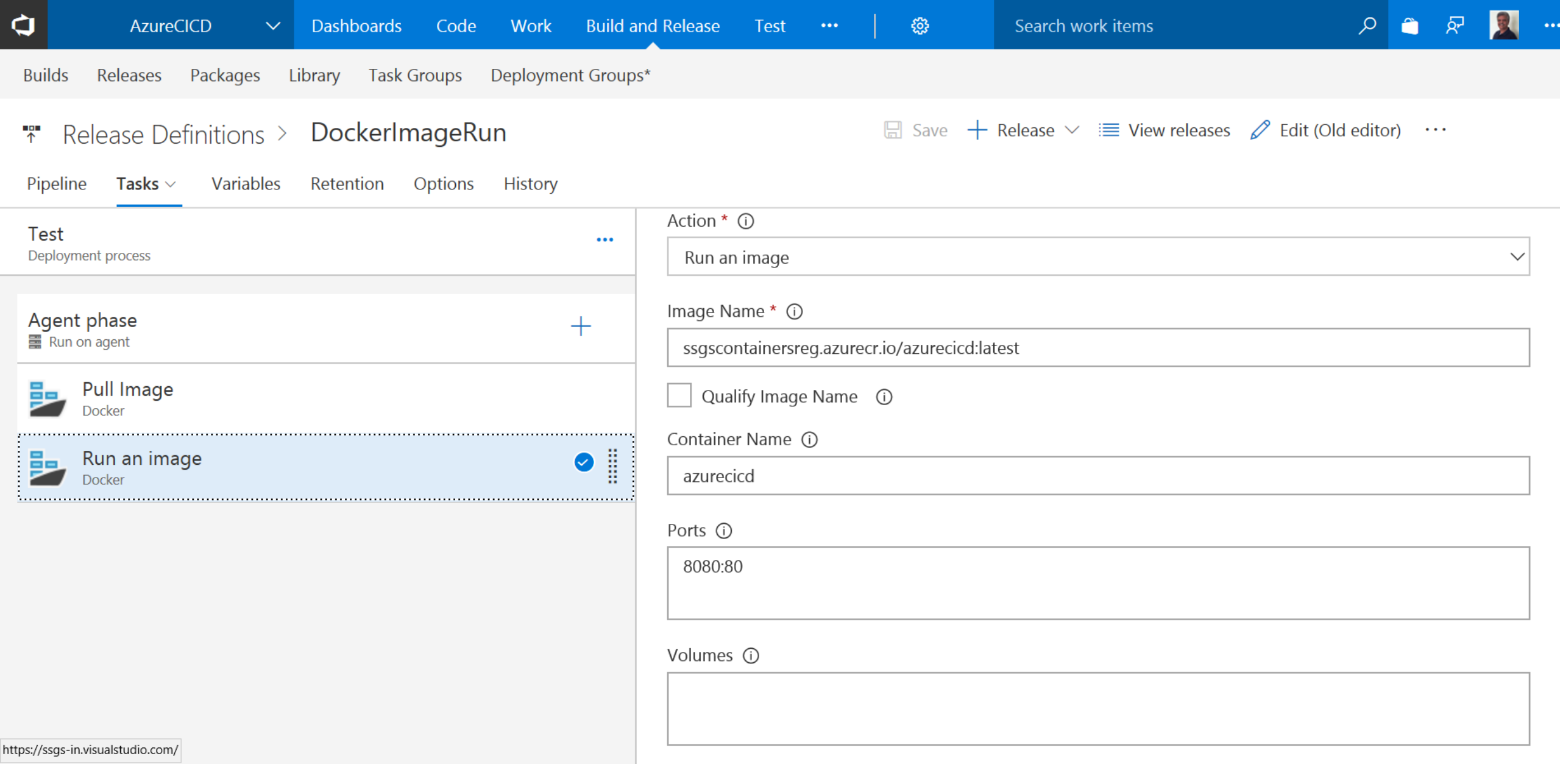
Pull image from Azure Container Registry to Docker Host



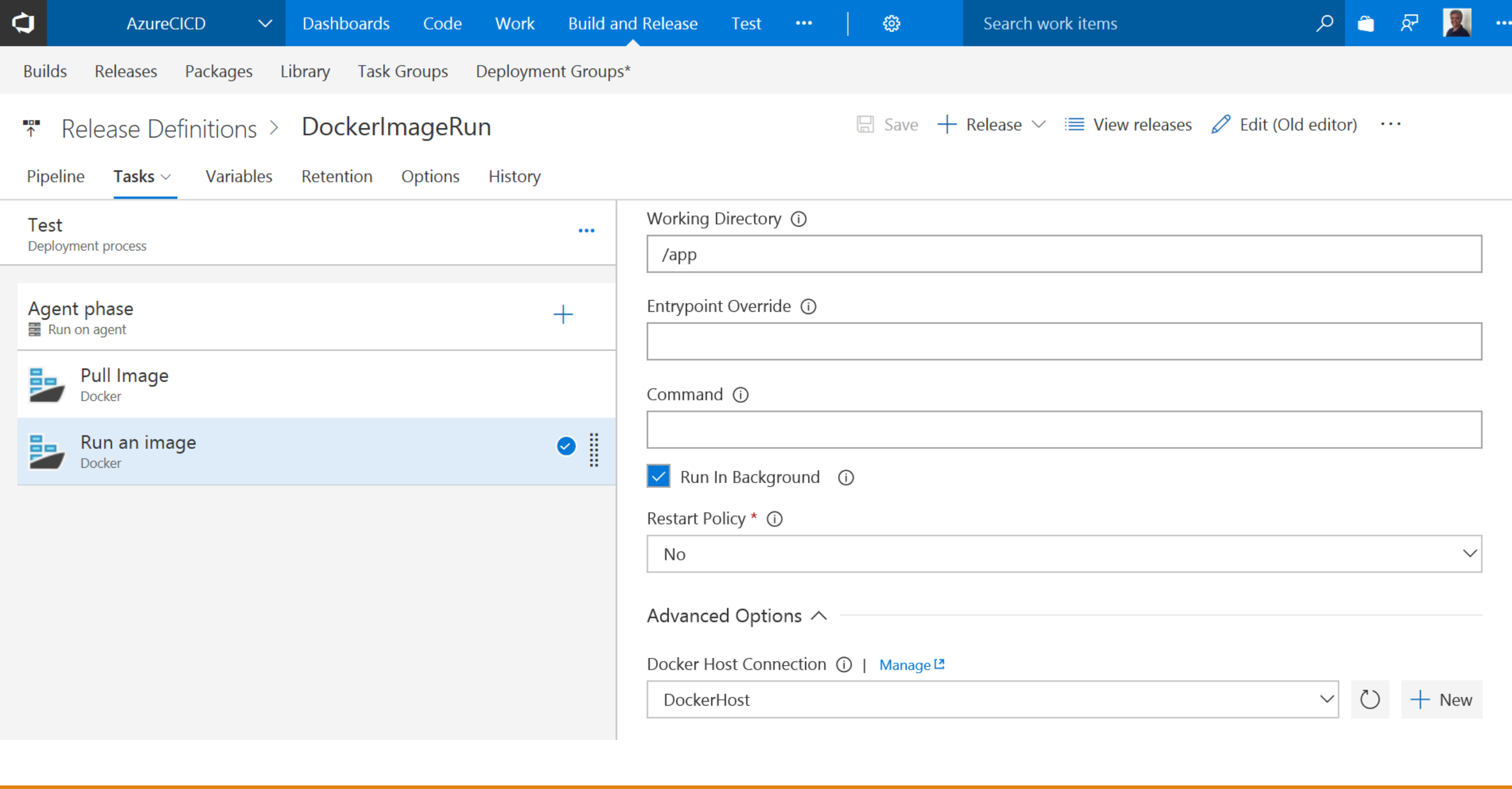
Pull image from Azure Container Registry to Docker Host



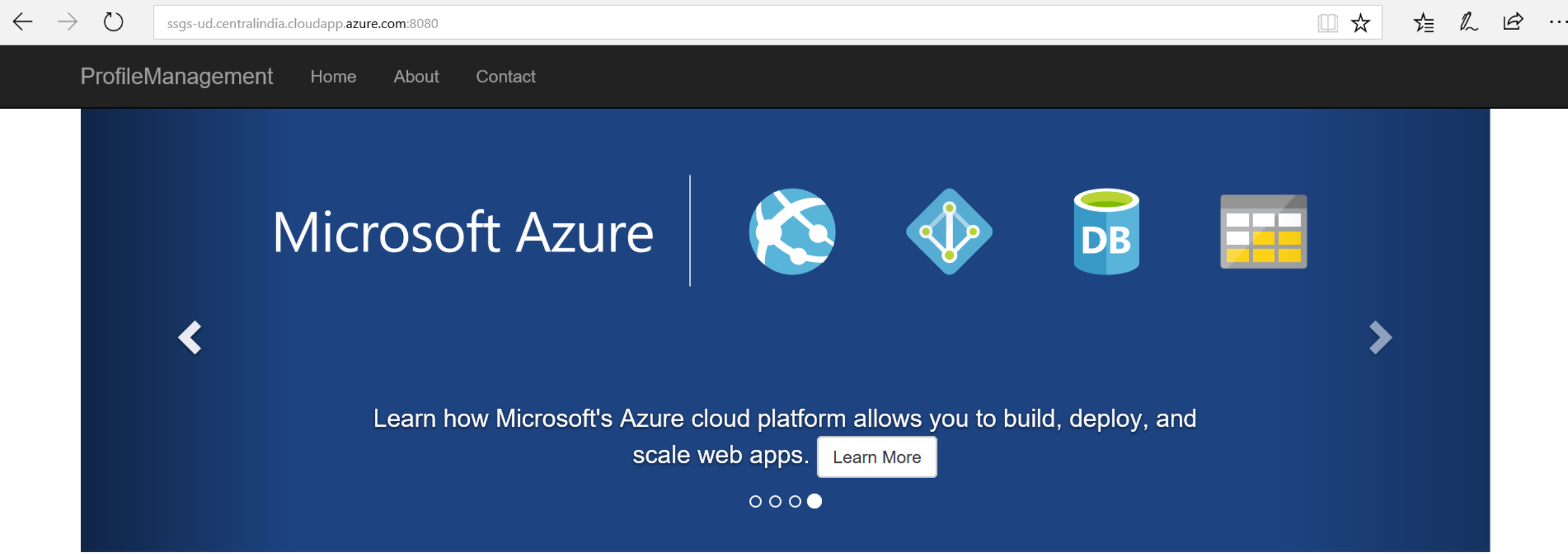
Run image on Docker Host VM



Run image on Docker Host VM



Run image on Docker Host VM



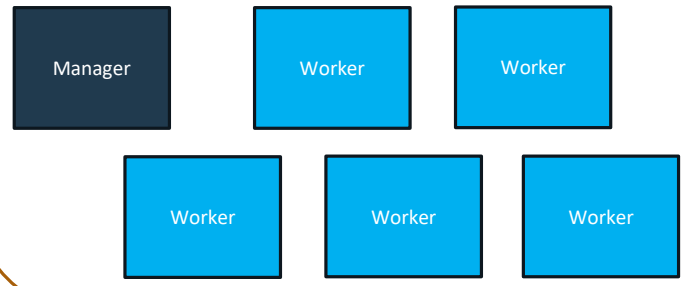
Run image on Docker Host VM

Multi-Container Application on Docker Swarm

Azure Container Service

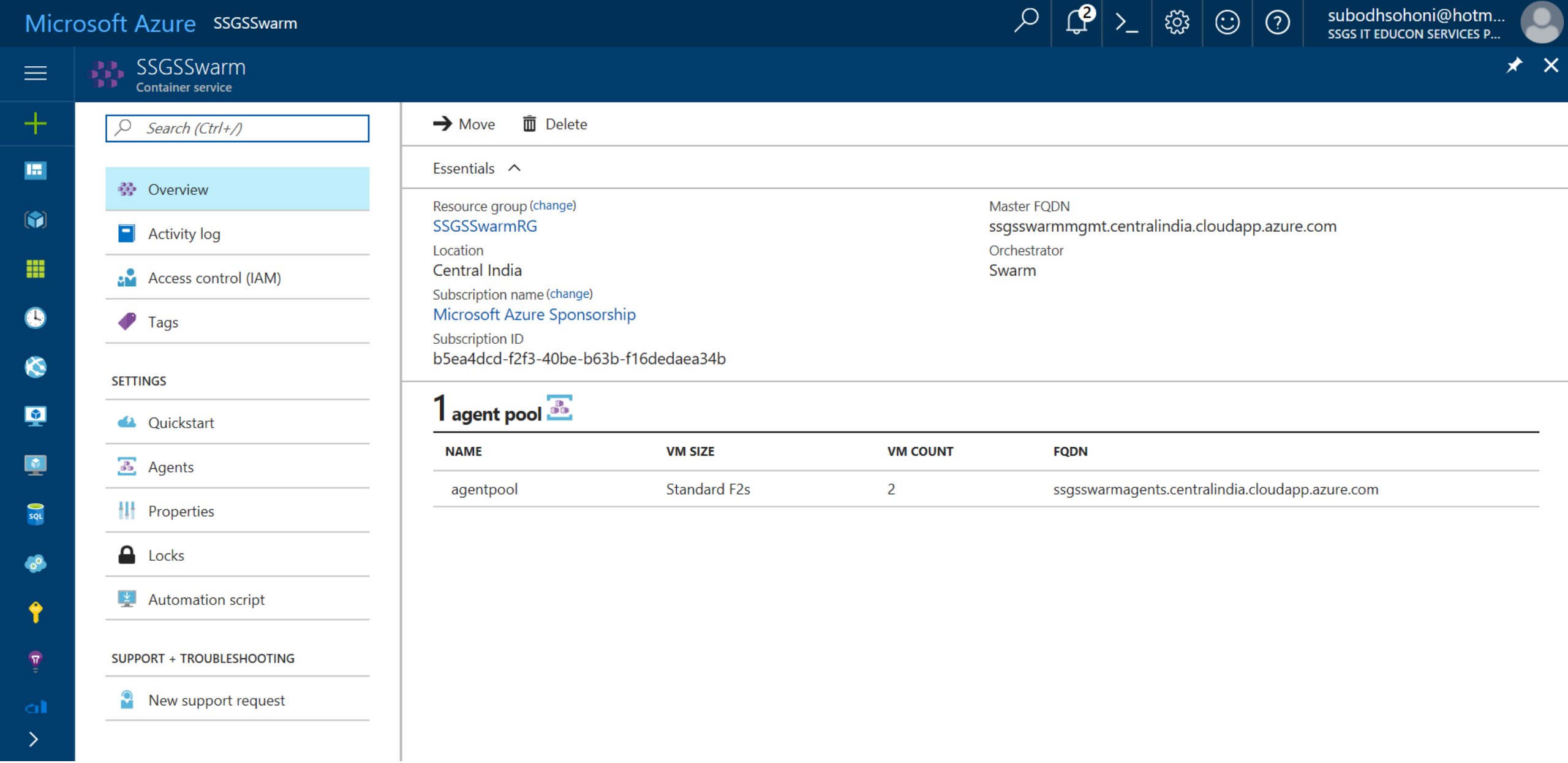
Orchestrated by
Docker Swarm
Or
DC/OS
Or
Kubernetes

Docker Swarm

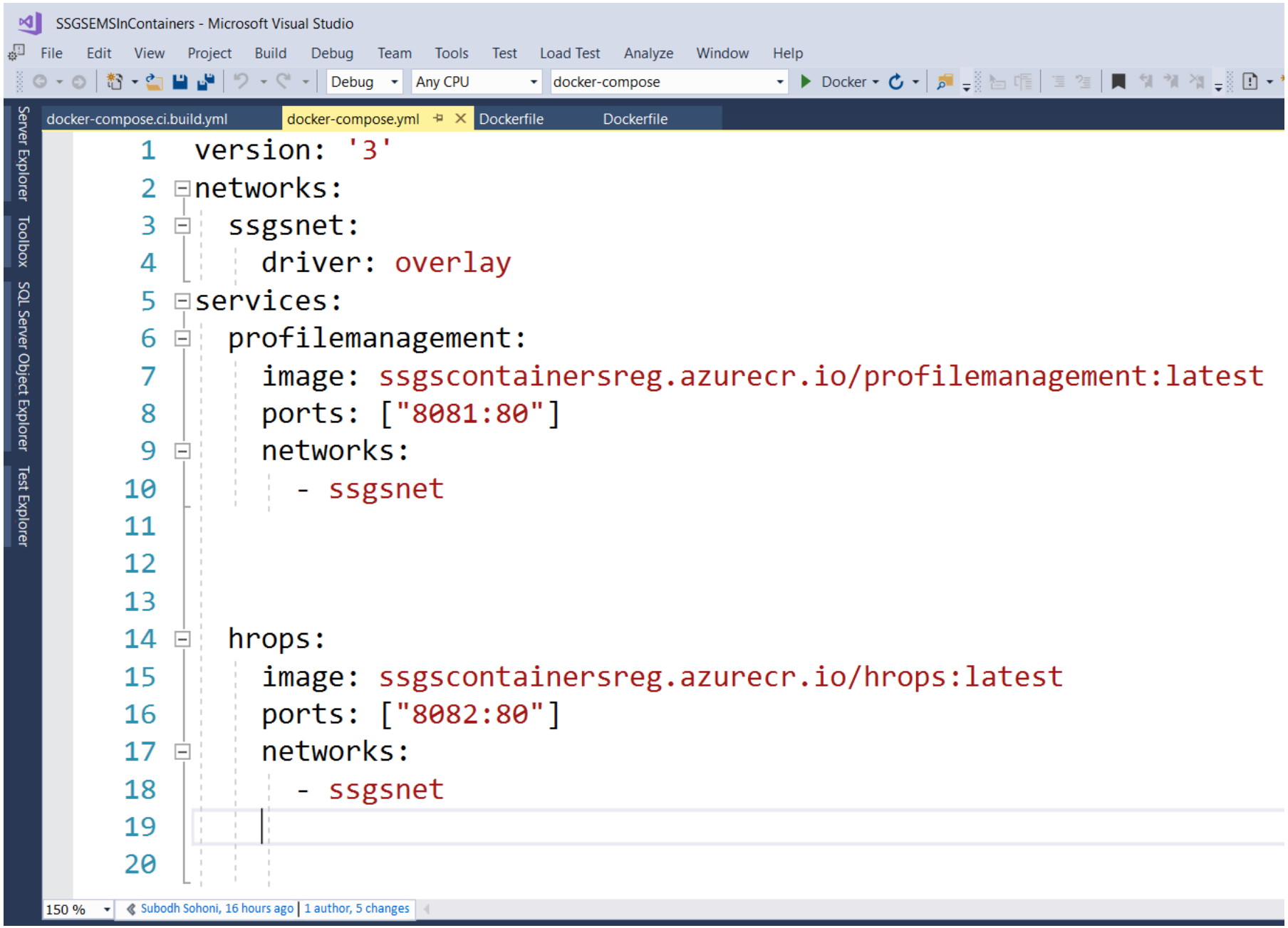


VSTS – Release Management

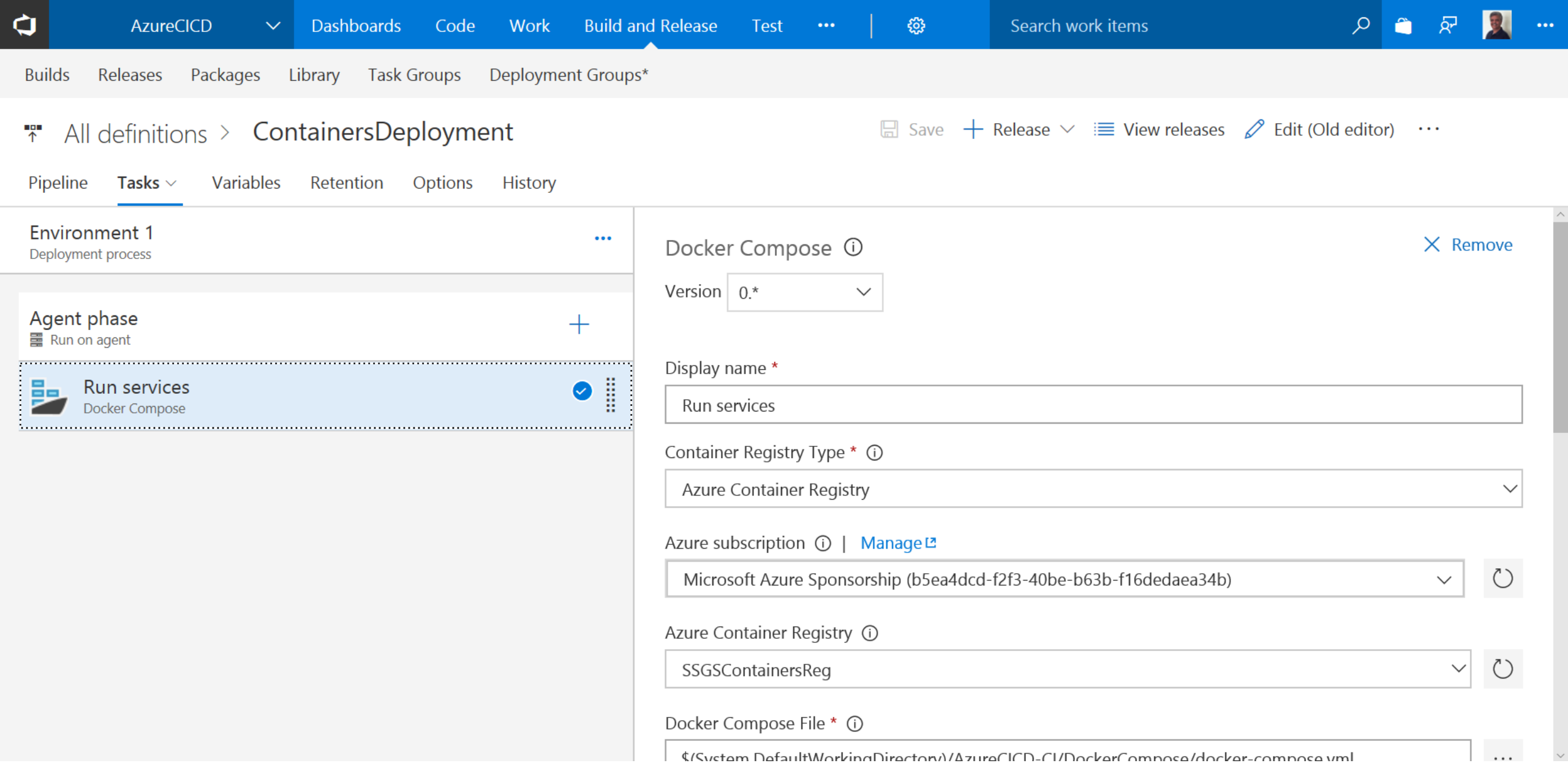
- Connect to Docker Swarm
- Use docker-compose.yml to provide deployment details
- Pull Images from Azure Container Service
- Run Images to create Multiple Containers
- Expose ports to map to Docker Hosts in Swarm
- Use Port 2377 to Run Container on Swarm



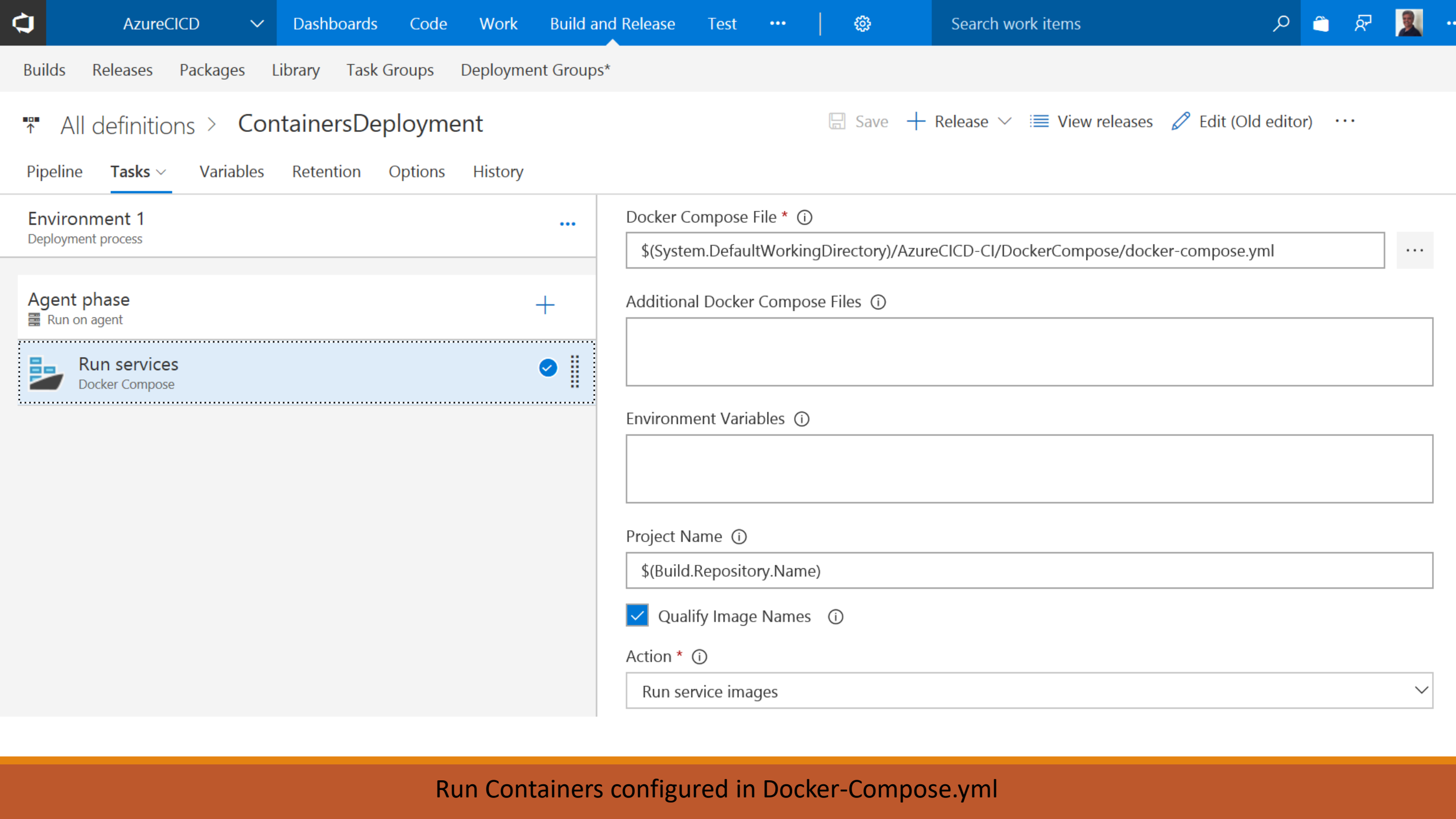
Azure Container Service for Docker Cluster Orchestrated by Docker Swarm



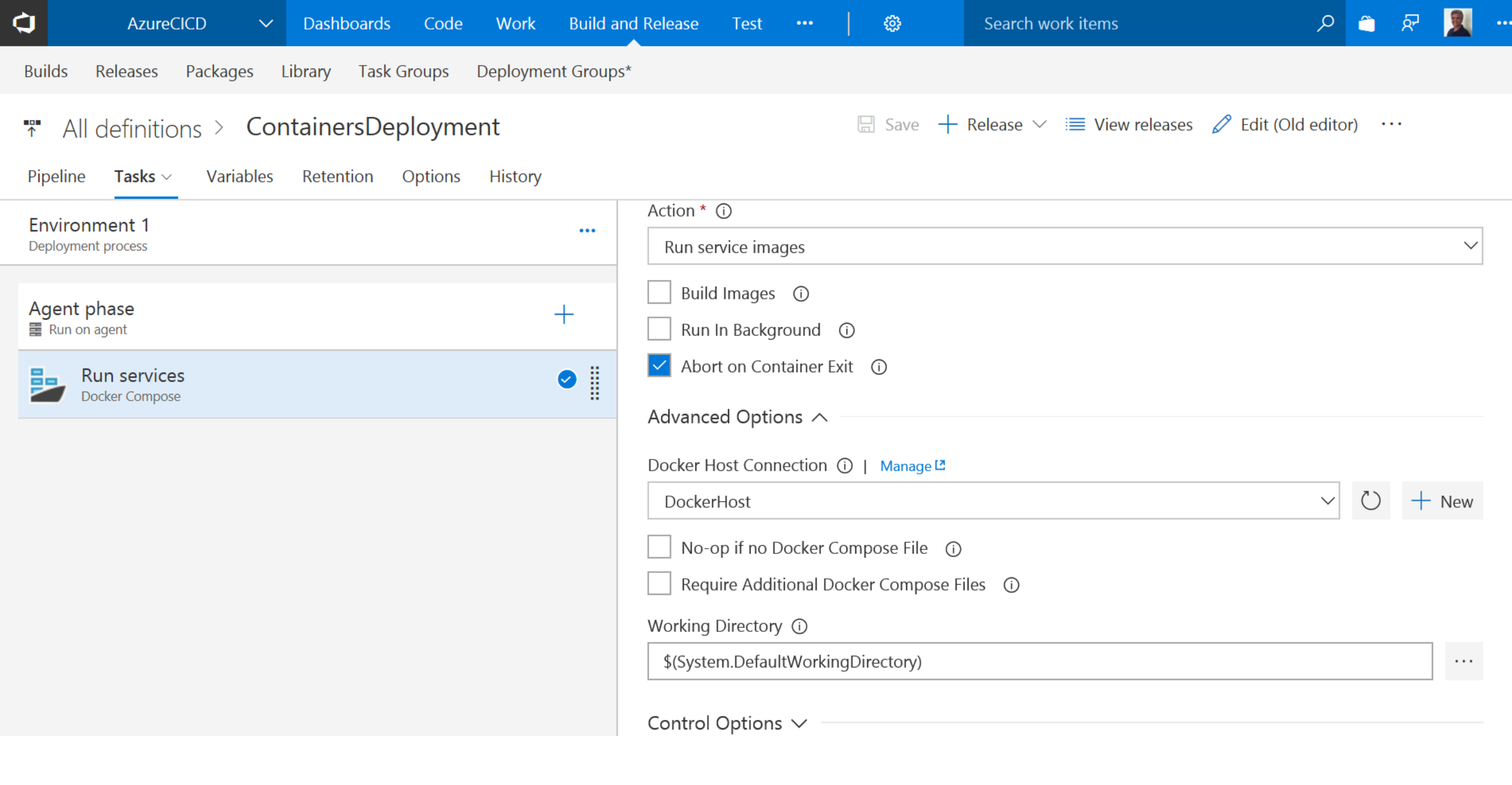
Docker-Compose.yml



Run Containers configured in Docker-Compose.yml



Run Containers configured in Docker-Compose.yml



Run Containers configured in Docker-Compose.yml