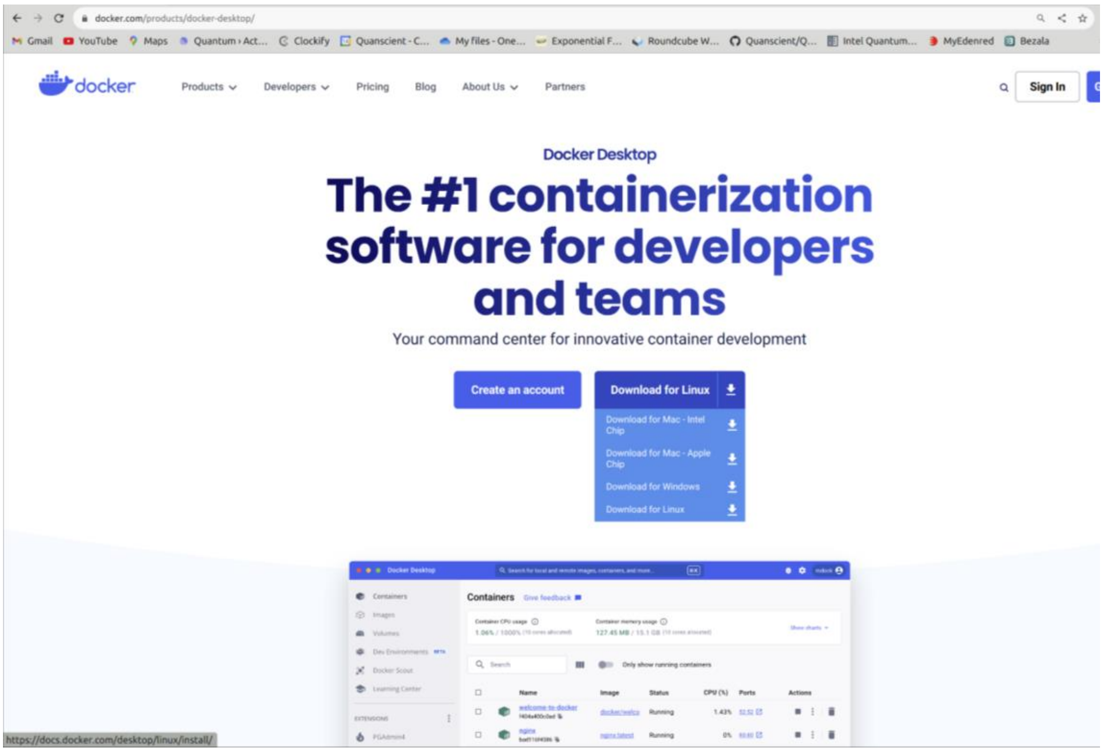
Using Intel Quantum SDK on your PC using docker desktop and visual studio code

Tejas Shinde

The following tutorial shows the steps on a Linux Machine. Similar steps can be followed on Windows and Mac.

1) Installing Docker Desktop -
   Download the setup for installing the docker desktop from the website - https://www.docker.com/products/docker-desktop/ .



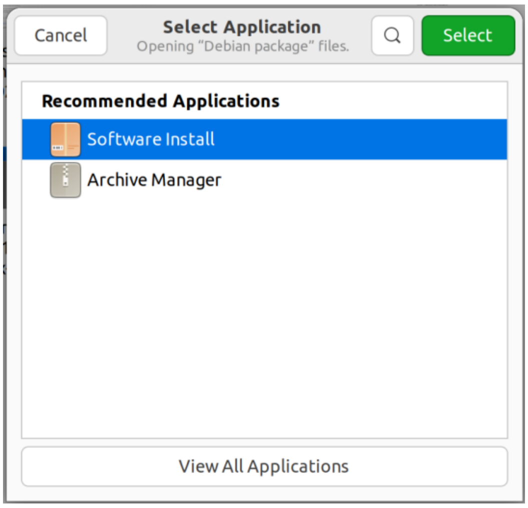
Docker Desktop

The #1 containerization software for developers and teams

Your command center for innovative container development

   The installation steps for different OS are given here - https://docs.docker.com/get-docker/

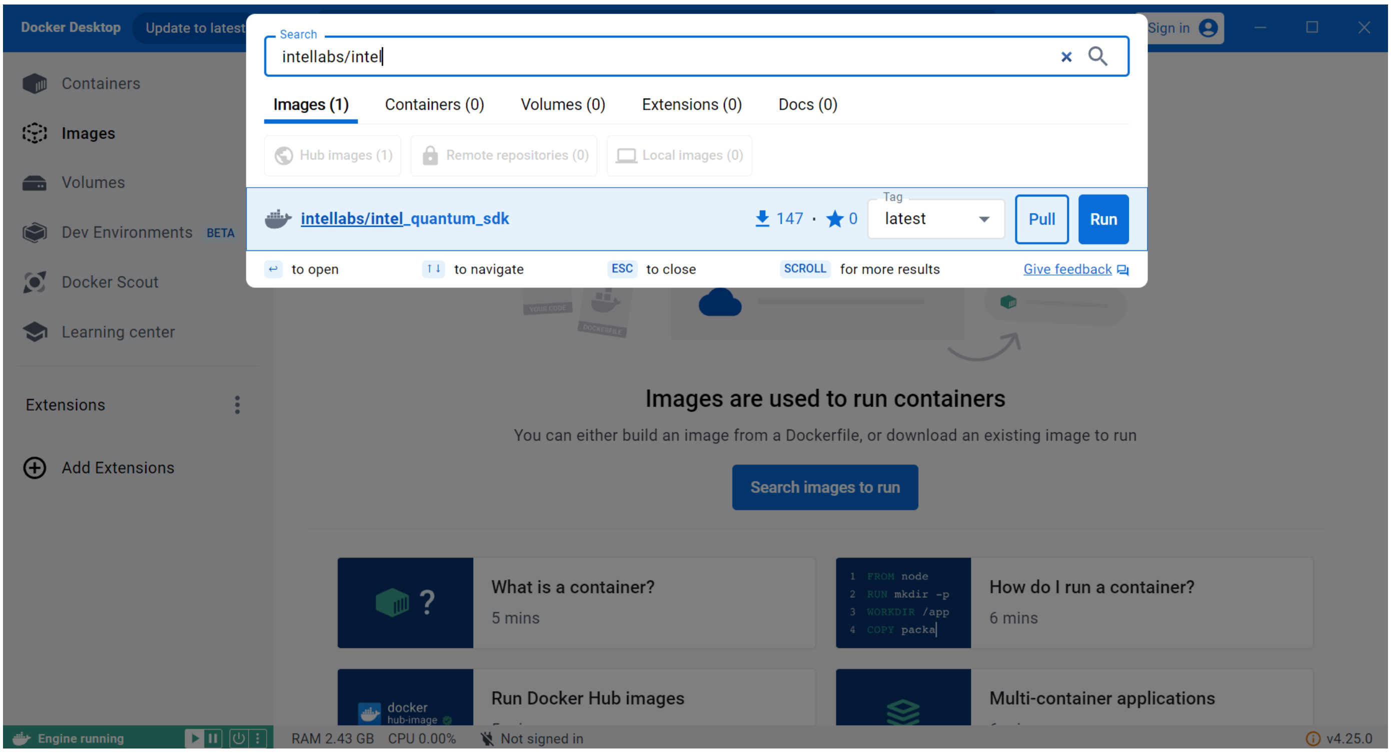
   For Linux - we have the .deb package.
   Installing .deb files - https://help.ubuntu.com/kubuntu/desktopguide/C/manual-install.html
   1) Right-click and select the software updater



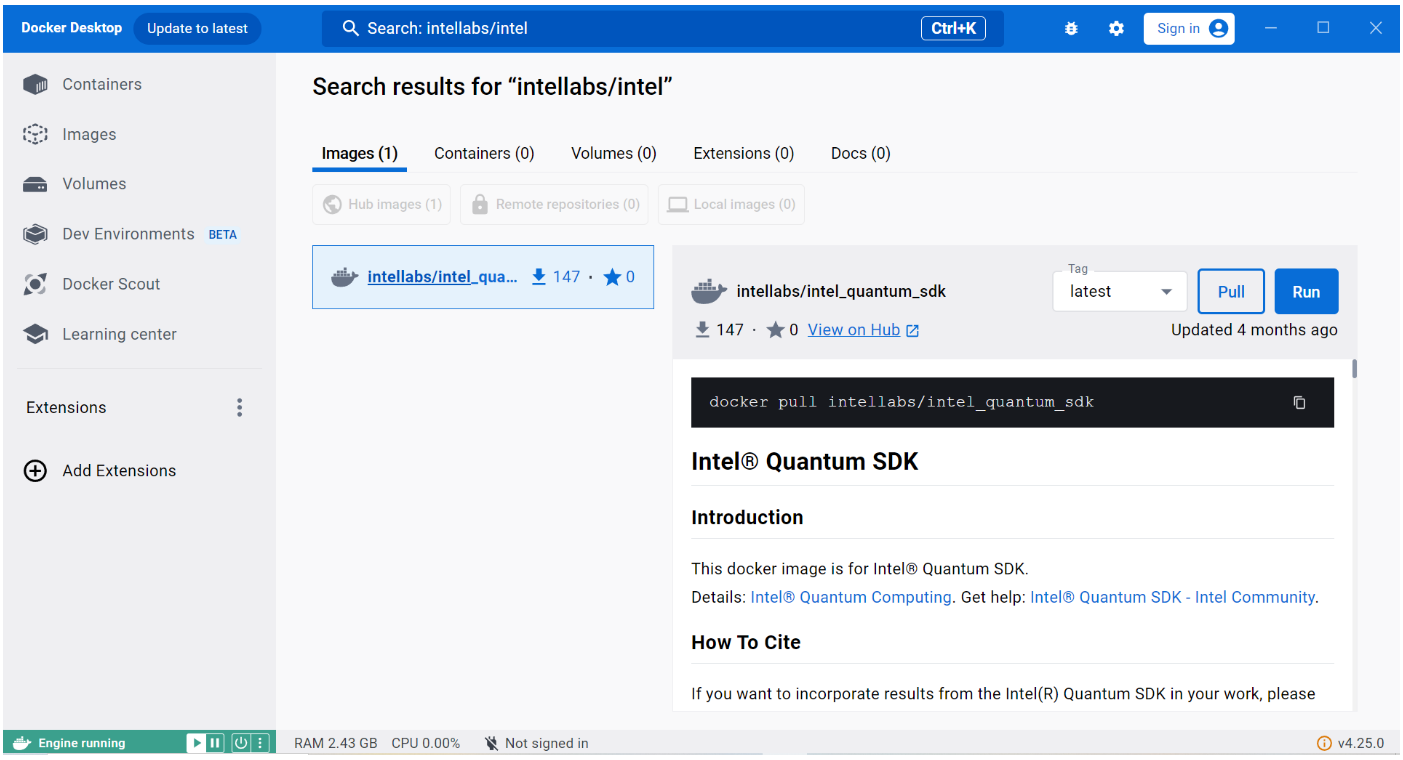
2) Open the directory where the docker installation .deb package is located in the terminal and type -
   sudo dpkg -i package_file_name.deb

Once installed - open the docker desktop app.

2) Retrieving the Intel Quantum SDK docker image -
   After opening the docker desktop app search for the intel quantum SDK docker container.
   intellabs/intel_quantum_sdk


Search: intellabs/intel

Clicking on the option - intellabs/intel_quantum_sdk opens the image for intel quantum SDK as shown below, now we can pull the image to save it locally.
Then click pull. It will take some time here.



Search results for "intellabs/intel"

intellabs/intel_quantum_sdk

147 · 0

Updated 4 months ago

```
docker pull intellabs/intel_quantum_sdk
```

Intel® Quantum SDK

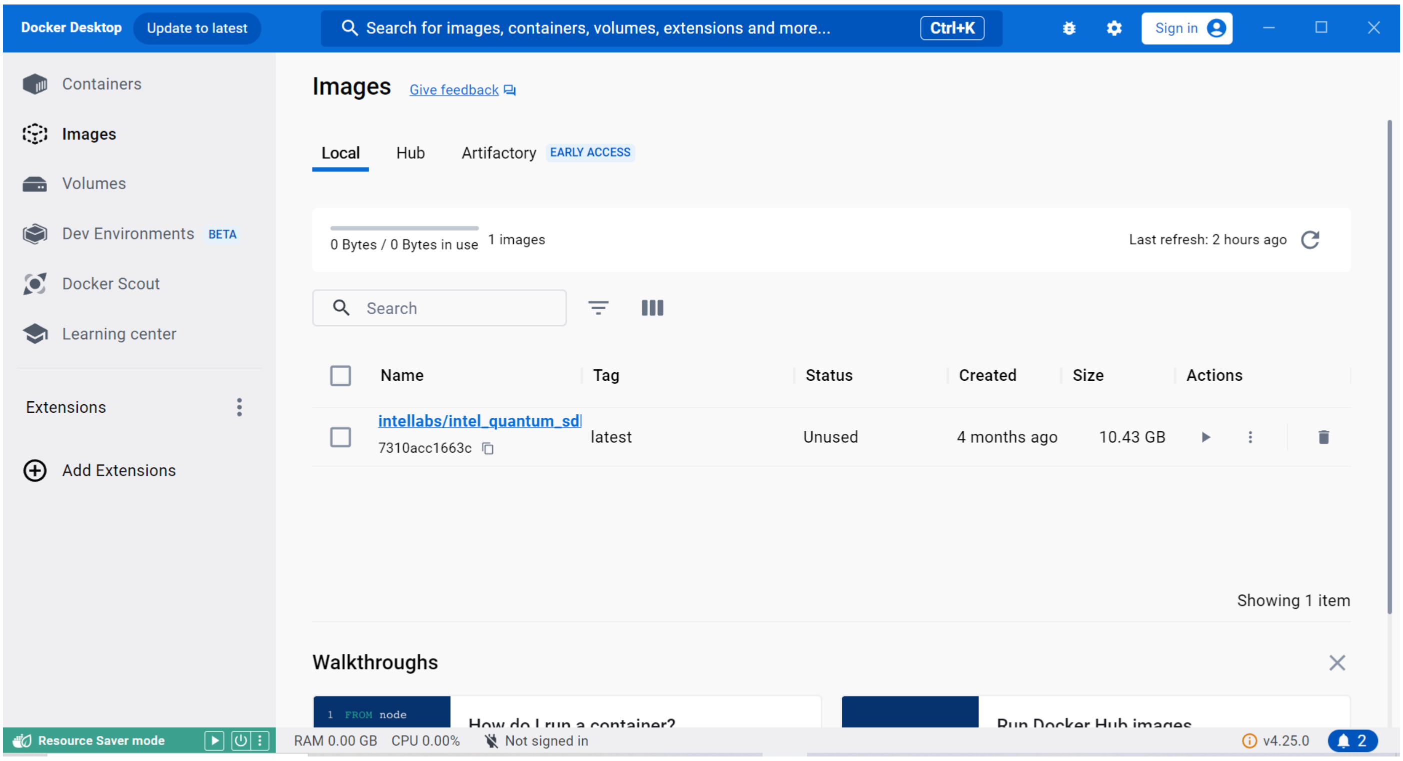
Introduction

This docker image is for Intel® Quantum SDK.
Details: Intel® Quantum Computing. Get help: Intel® Quantum SDK - Intel Community.

How To Cite

If you want to incorporate results from the Intel(R) Quantum SDK in your work, please

Then you can see the image in the Local directory for images.



Images

0 Bytes / 0 Bytes in use 1 images

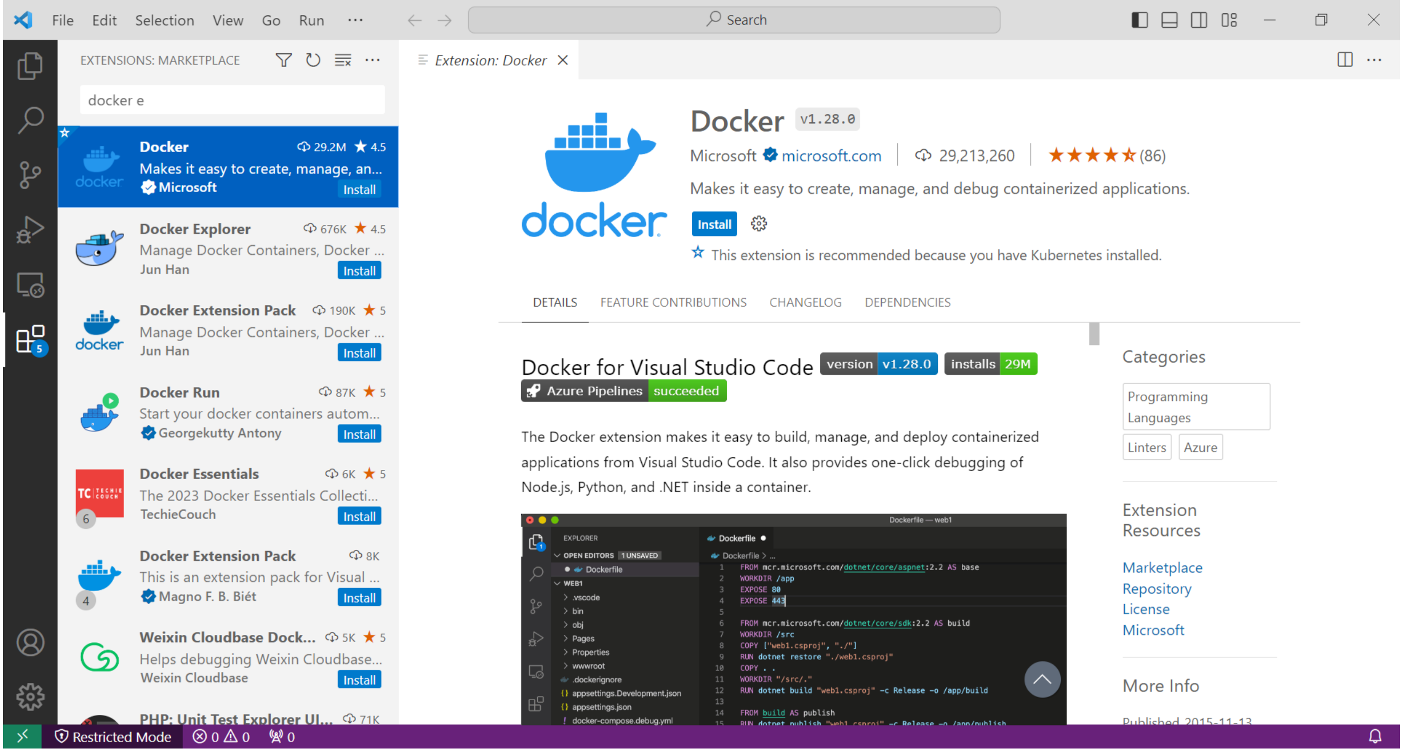
Last refresh: 2 hours ago

□	Name	Tag	Status	Created	Size	Actions
□	intellabs/intel_quantum_sd 7310acc1663c	latest	Unused	4 months ago	10.43 GB	

Showing 1 item

3) Installing visual studio code -
Download the setup from the following link to install Visual Studio Code
https://code.visualstudio.com/download

4) Installing the docker extension -
Search in extensions - for docker and install



Docker v1.28.0

Microsoft microsoft.com 29,213,260 ★★★★★ (86)

Makes it easy to create, manage, and debug containerized applications.

Install

☆ This extension is recommended because you have Kubernetes installed.

Docker for Visual Studio Code version v1.28.0 installs 29M

Azure Pipelines succeeded

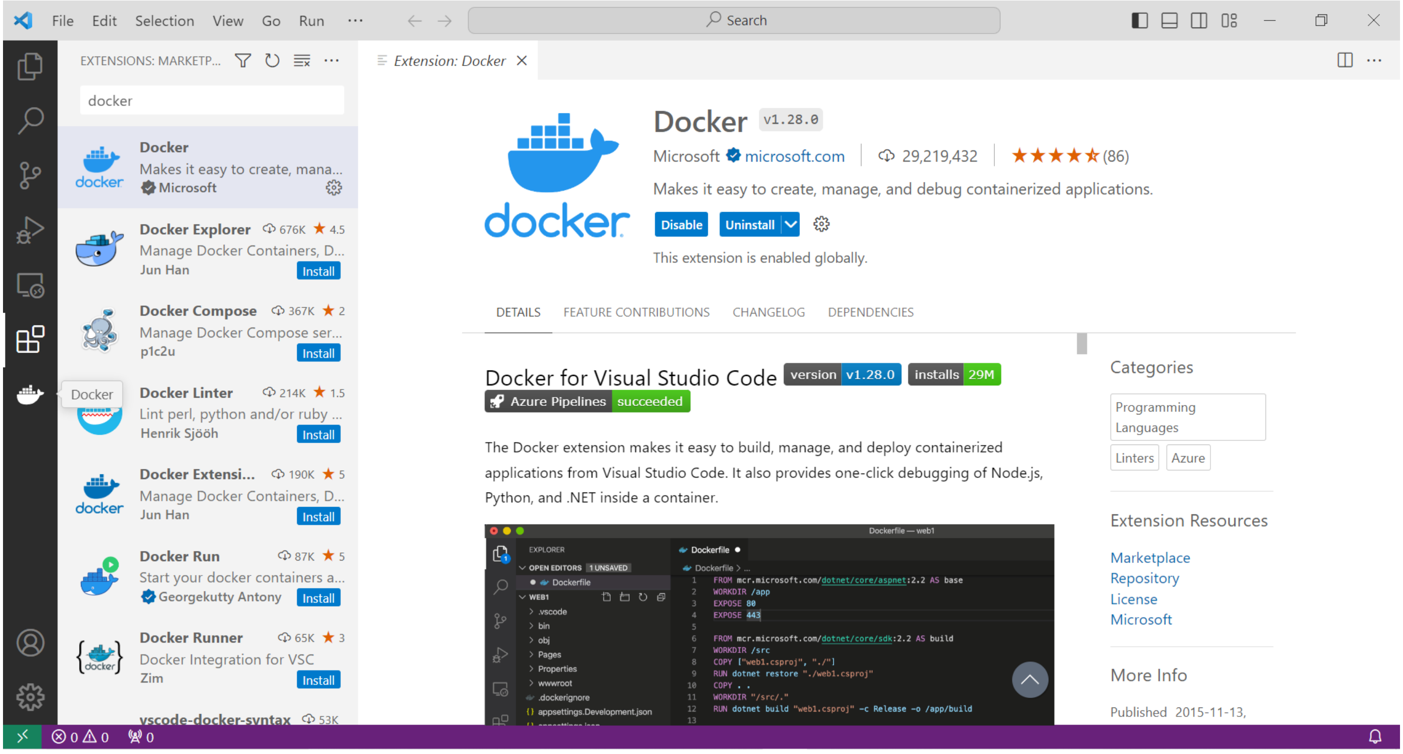
The Docker extension makes it easy to build, manage, and deploy containerized applications from Visual Studio Code. It also provides one-click debugging of Node.js, Python, and .NET inside a container.

```
FROM mcr.microsoft.com/dotnet/core/aspnet:2.2 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/core/sdk:2.2 AS build
WORKDIR /src
COPY ["web1.csproj", "./"]
RUN dotnet restore "./web1.csproj"
COPY . .
WORKDIR "/src/."
RUN dotnet build "web1.csproj" -c Release -o /app/build

FROM build AS publish
```

After installing the docker extension, you will see the docker option on the left-hand side as shown in the picture below



Docker v1.28.0

Microsoft microsoft.com 29,219,432 ★★★★★ (86)

Makes it easy to create, manage, and debug containerized applications.

Disable Uninstall

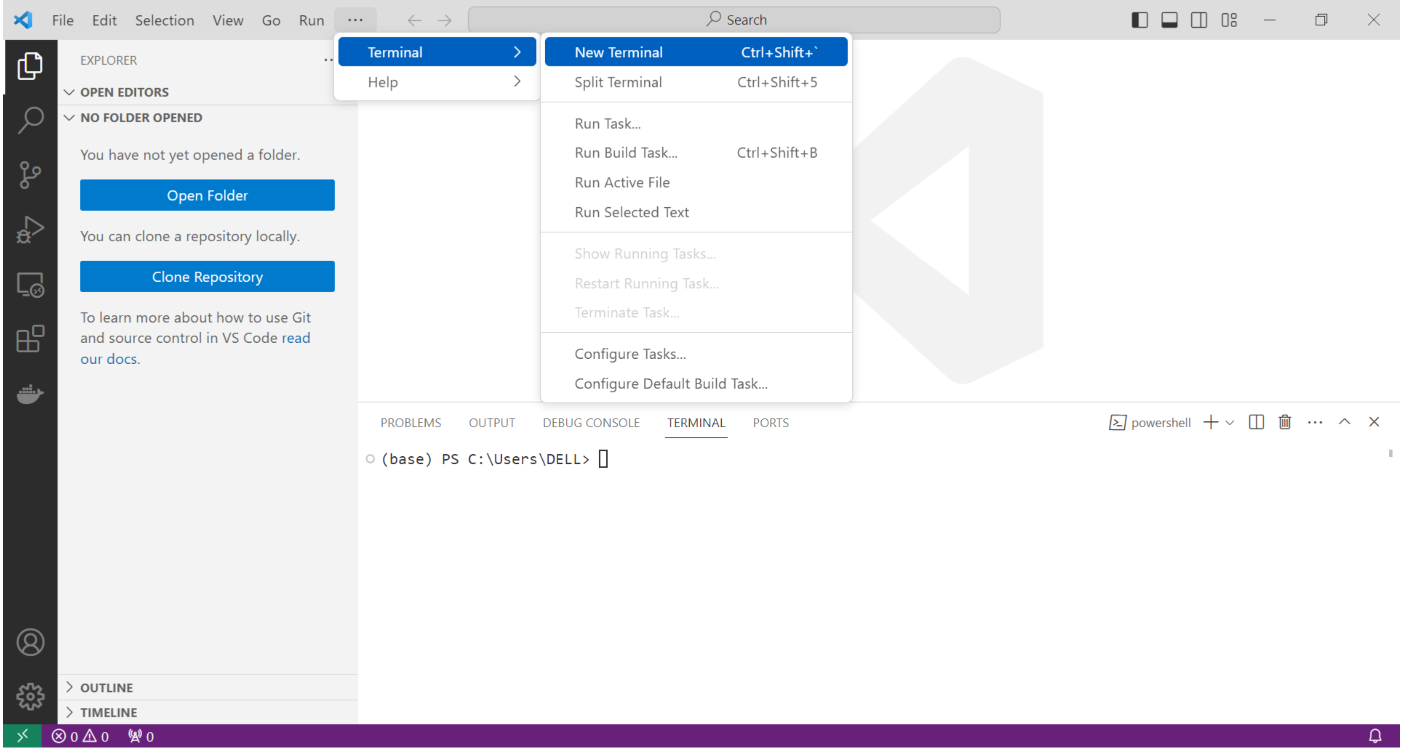
This extension is enabled globally.

Docker for Visual Studio Code version v1.28.0 installs 29M

Azure Pipelines succeeded

The Docker extension makes it easy to build, manage, and deploy containerized applications from Visual Studio Code. It also provides one-click debugging of Node.js, Python, and .NET inside a container.

Then we need to create a docker container, hence open a terminal in Visual Studio code as shown in the picture below



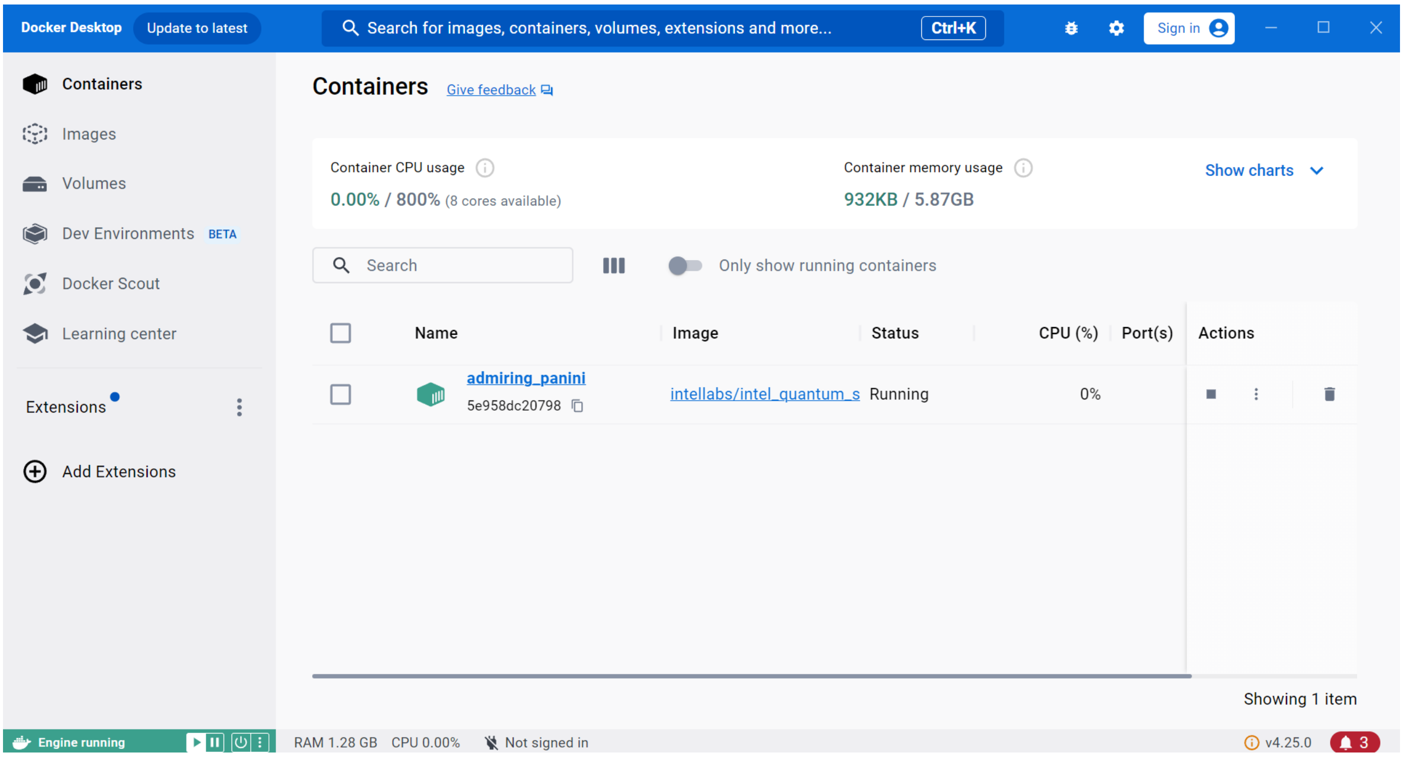
You have not yet opened a folder.

You can clone a repository locally.

To learn more about how to use Git and source control in VS Code read our docs.

```
(base) PS C:\Users\DELL> 
```

Run the following command to create a docker container, it will create a container with a random name - docker run -it intellabs/intel_quantum_sdk bash



Containers

Container CPU usage: 0.00% / 800% (8 cores available)

Container memory usage: 932KB / 5.87GB

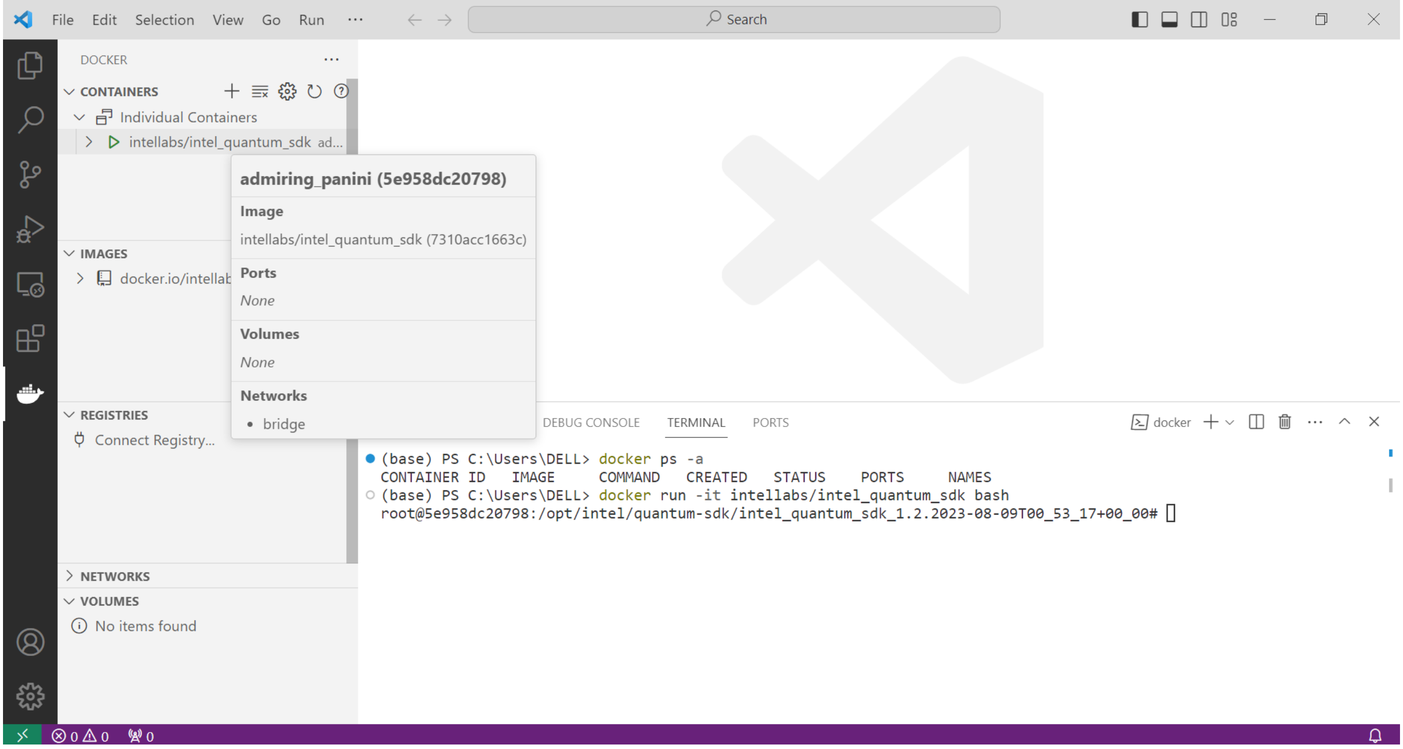
□	Name	Image	Status	CPU (%)	Port(s)	Actions
□	admiring_panini 5e958dc20798	intellabs/intel_quantum_s	Running	0%		■ ⋮ 🗑

Showing 1 item

RAM 1.28 GB CPU 0.00%

here, admiring_panini is a container created

Then open the docker option on the left side of Visual Studio code, and you will see the container we just created. (admiring_panini here)



admiring_panini (5e958dc20798)

Image
intellabs/intel_quantum_sdk (7310acc1663c)

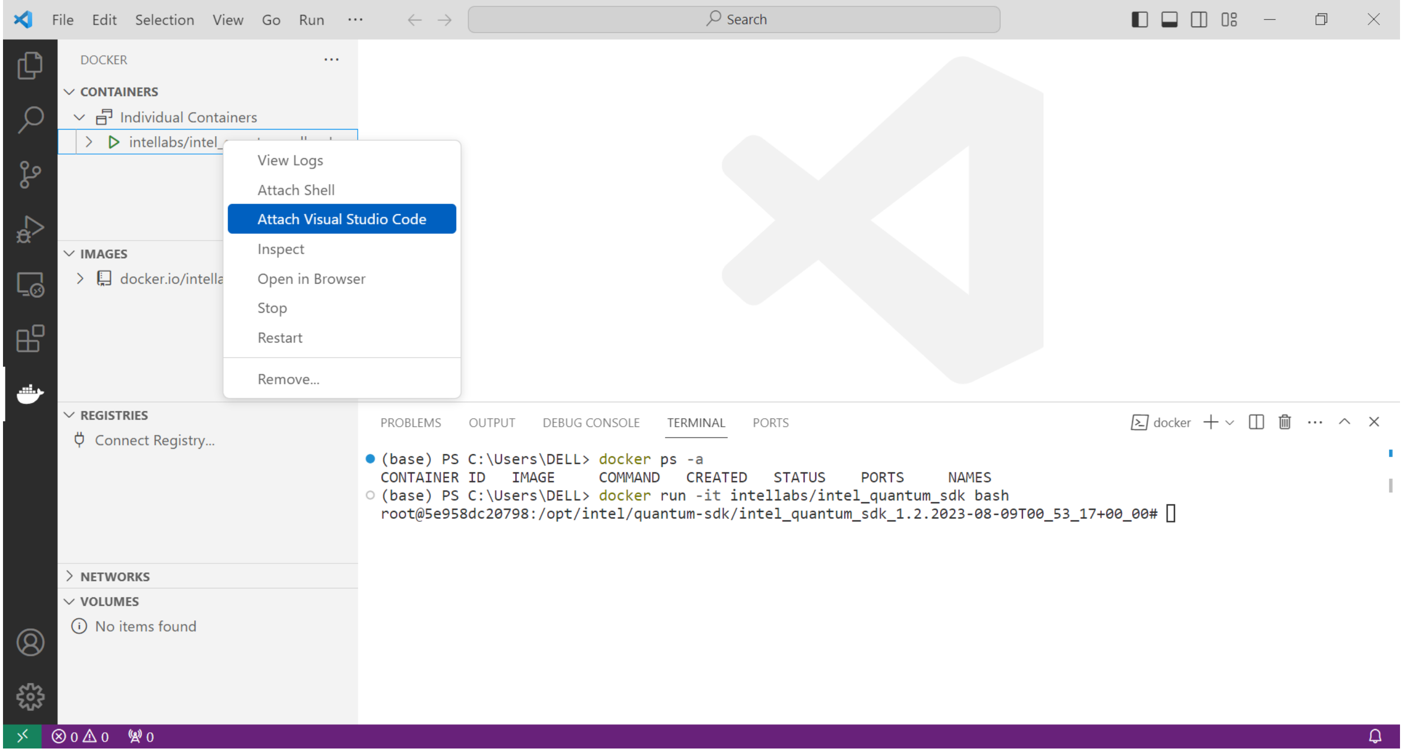
Ports
None

Volumes
None

Networks
- bridge

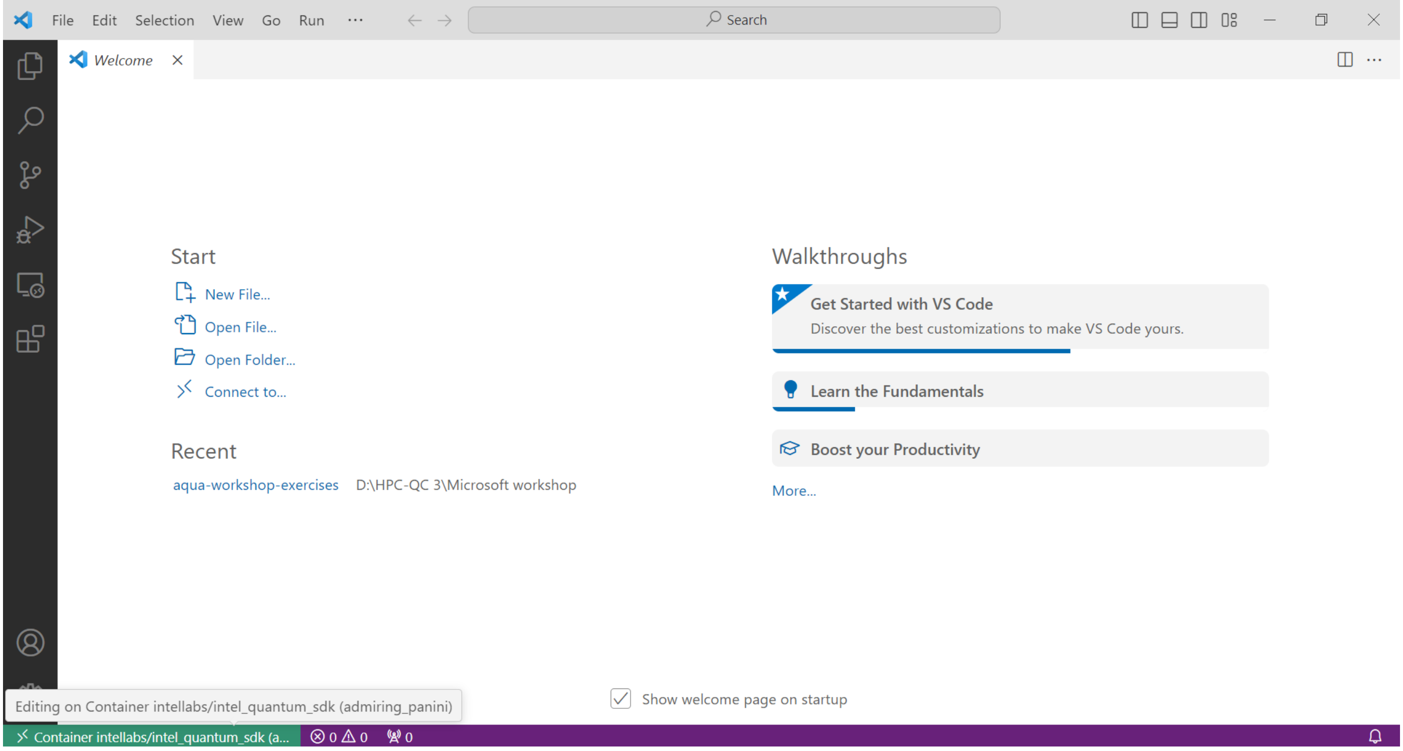
```
(base) PS C:\Users\DELL> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(base) PS C:\Users\DELL> docker run -it intellabs/intel_quantum_sdk bash
root@5e958dc20798:/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00# 
```

Right-click on the container option and in the drop-down list you will find - Attach Visual Studio code. Click - Attach Visual Studio Code



```
(base) PS C:\Users\DELL> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(base) PS C:\Users\DELL> docker run -it intellabs/intel_quantum_sdk bash
root@5e958dc20798:/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00# 
```

It will open a new Visual Studio code window and connect to the intel quantum sdk container (admiring_panini) we created. As shown on the bottom left side in the picture below.



aqua-workshop-exercises D:\HPC-QC 3\Microsoft workshop

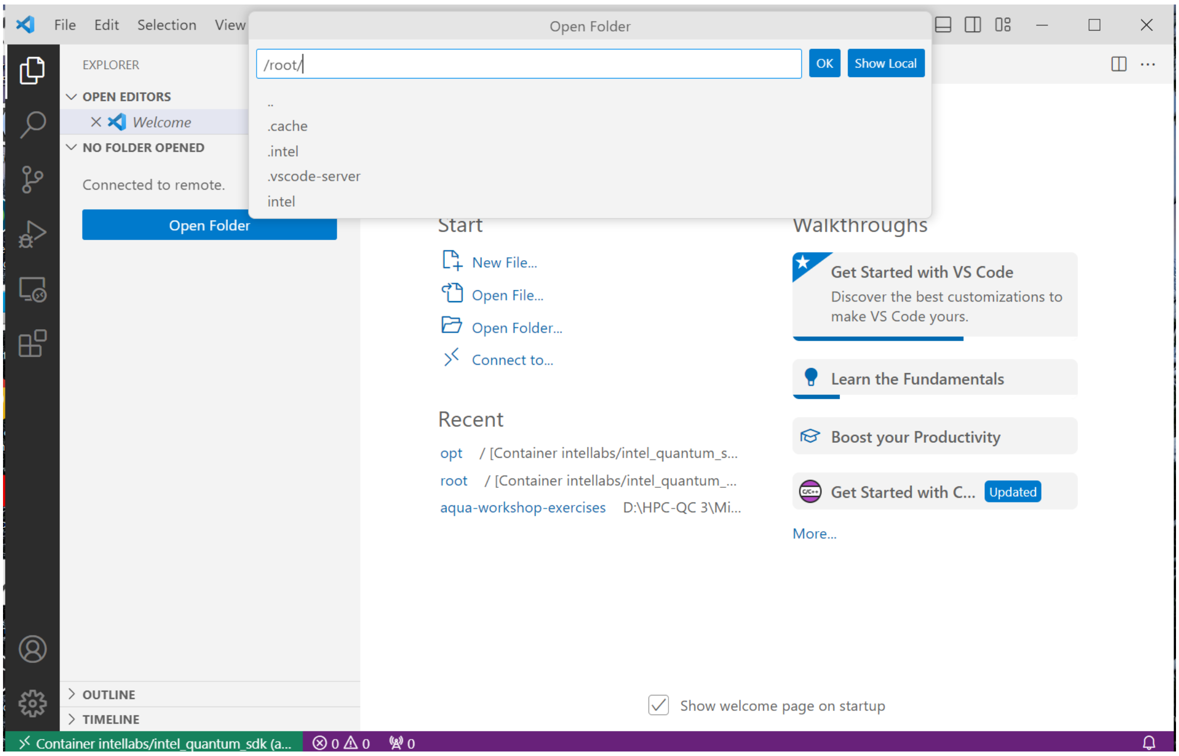
Discover the best customizations to make VS Code yours.

☑ Show welcome page on startup

Editing on Container intellabs/intel_quantum_sdk (admiring_panini)

Container intellabs/intel_quantum_sdk (a... ⊗ 0 ⚠ 0 0

The next step is to open the folder - and at first, it will show - /root/,



Connected to remote.

Open Folder

/root/

..
.cache
.intel
.vscode-server
intel

opt / [Container intellabs/intel_quantum_s...
root / [Container intellabs/intel_quantum_...
aqua-workshop-exercises D:\HPC-QC 3\Mi...

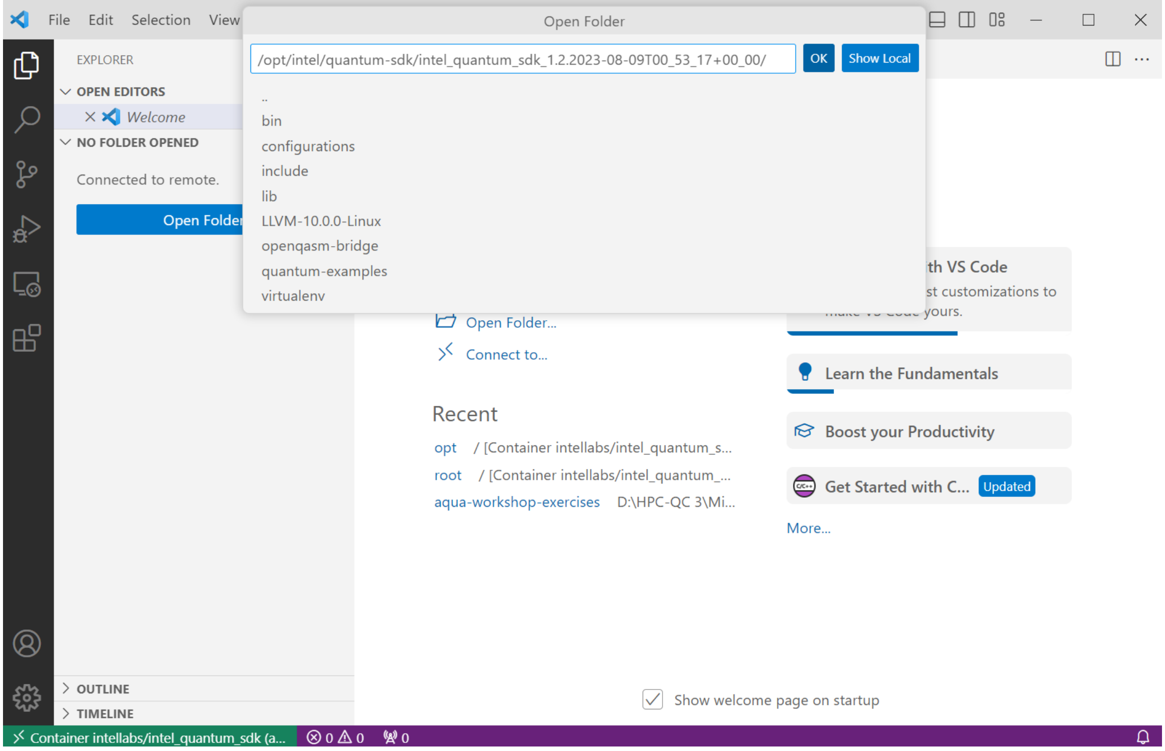
Discover the best customizations to make VS Code yours.

☑ Show welcome page on startup

Container intellabs/intel_quantum_sdk (a... ⊗ 0 ⚠ 0 0

But the location of intel quantum SDK is in the folder - /opt/intel/quantum_sdk

Hence go to the folder as shown in the pictures below. And click OK



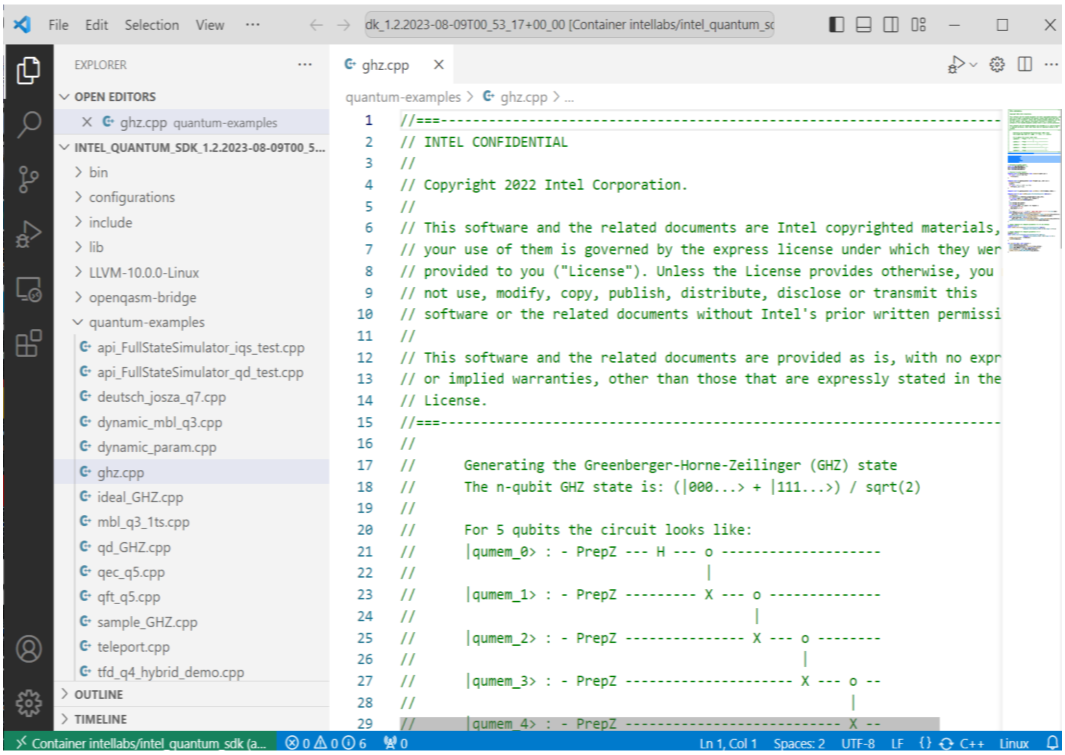
/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00/

```
..
bin
configurations
include
lib
LLVM-10.0.0-Linux
openqasm-bridge
quantum-examples
virtualenv
```

Recent

- opt / [Container intellabs/intel_quantum_s...
- root / [Container intellabs/intel_quantum_...
- aqua-workshop-exercises D:\HPC-QC 3\Mi...

Hence you have reached the intel quantum sdk folder and connected the docker container with Visual Studio code and you can write and compile code in Visual Studio code.


File
Edit
Selection
View
dk_1.2.2023-08-09T00_53_17+00_00 [Container intellabs/intel_quantum_sc
EXPLORER
OPEN EDITORS
ghz.cpp quantum-examples
INTEL_QUANTUM_SDK_1.2.2023-08-09T00_5...
ghz.cpp
quantum-examples > ghz.cpp > ...
OUTLINE
TIMELINE
Container intellabs/intel_quantum_sdk (a...
⊗ 0 ⚠ 0 ⓘ 6
0
Ln 1, Col 1
Spaces: 2
UTF-8
LF
{} C++
Linux

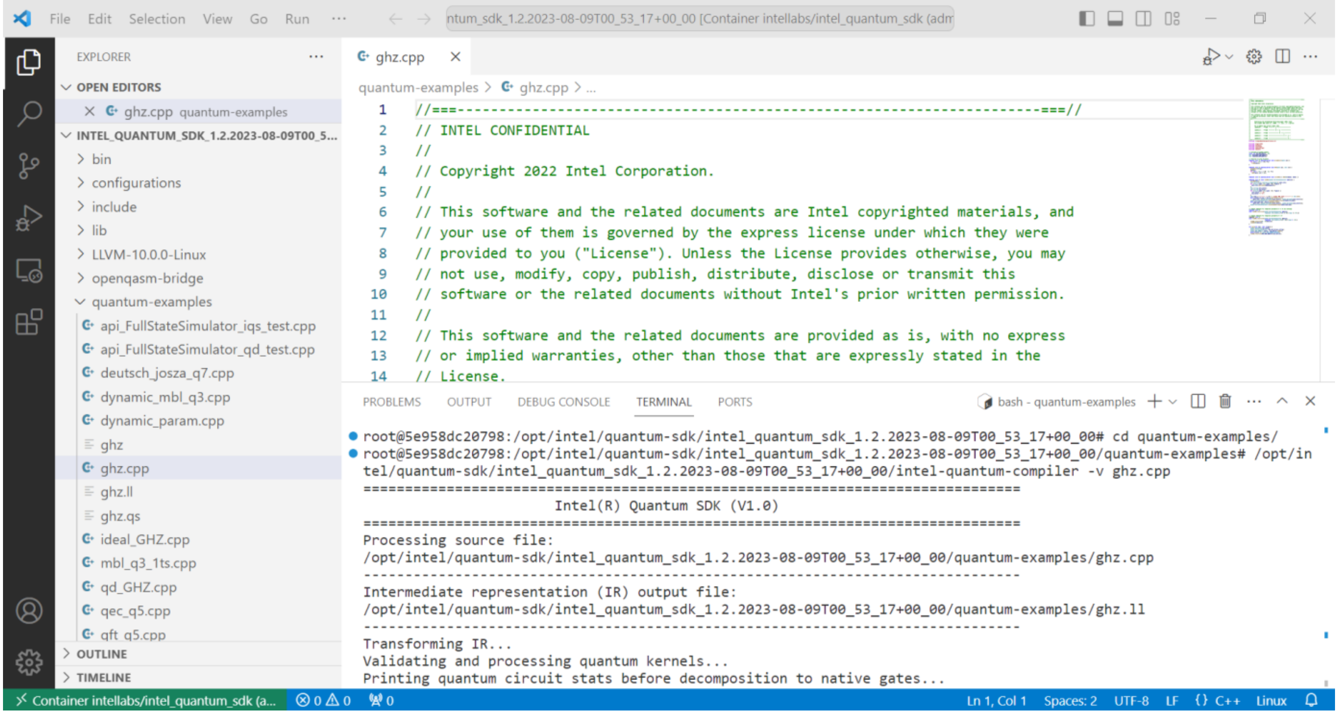
```
bin
configurations
include
lib
LLVM-10.0.0-Linux
openqasm-bridge
quantum-examples
  api_FullStateSimulator_iqs_test.cpp
  api_FullStateSimulator_qd_test.cpp
  deutsch_josza_q7.cpp
  dynamic_mbl_q3.cpp
  dynamic_param.cpp
  ghz.cpp
  ideal_GHZ.cpp
  mbl_q3_1ts.cpp
  qd_GHZ.cpp
  qec_q5.cpp
  qft_q5.cpp
  sample_GHZ.cpp
  teleport.cpp
  tfd_q4_hybrid_demo.cpp
```

```cpp
//===----------------------------------------------------------------------
// INTEL CONFIDENTIAL
//
// Copyright 2022 Intel Corporation.
//
// This software and the related documents are Intel copyrighted materials,
// your use of them is governed by the express license under which they wer
// provided to you ("License"). Unless the License provides otherwise, you
// not use, modify, copy, publish, distribute, disclose or transmit this
// software or the related documents without Intel's prior written permissi
//
// This software and the related documents are provided as is, with no expr
// or implied warranties, other than those that are expressly stated in the
// License.
//===----------------------------------------------------------------------
//
//      Generating the Greenberger-Horne-Zeilinger (GHZ) state
//      The n-qubit GHZ state is: (|000...> + |111...>) / sqrt(2)
//
//      For 5 qubits the circuit looks like:
//      |qumem_0> : - PrepZ --- H --- o --------------------
//                                    |
//      |qumem_1> : - PrepZ --------- X --- o --------------
//                                          |
//      |qumem_2> : - PrepZ --------------- X --- o --------
//                                                |
//      |qumem_3> : - PrepZ --------------------- X --- o --
//                                                      |
//      |qumem_4> : - PrepZ --------------------------- X --
```

5) Compiling file -

There are a few examples given in the quantum-examples folder, to compile and run the example, open the terminal in VSC and move to the quantum examples folder and execute the following commands - here I am compiling and executing the ghz.cpp example.

/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00/intel-quantum-compiler -v ghz.cpp
./ghz



```
//===----------------------------------------------------------------------===//
// INTEL CONFIDENTIAL
//
// Copyright 2022 Intel Corporation.
//
// This software and the related documents are Intel copyrighted materials, and
// your use of them is governed by the express license under which they were
// provided to you ("License"). Unless the License provides otherwise, you may
// not use, modify, copy, publish, distribute, disclose or transmit this
// software or the related documents without Intel's prior written permission.
//
// This software and the related documents are provided as is, with no express
// or implied warranties, other than those that are expressly stated in the
// License.
```

```
root@5e958dc20798:/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00# cd quantum-examples/
root@5e958dc20798:/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00/quantum-examples# /opt/in
tel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00/intel-quantum-compiler -v ghz.cpp
================================================================================
                       Intel(R) Quantum SDK (V1.0)
================================================================================
Processing source file:
/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00/quantum-examples/ghz.cpp
--------------------------------------------------------------------------------
Intermediate representation (IR) output file:
/opt/intel/quantum-sdk/intel_quantum_sdk_1.2.2023-08-09T00_53_17+00_00/quantum-examples/ghz.ll
--------------------------------------------------------------------------------
Transforming IR...
Validating and processing quantum kernels...
Printing quantum circuit stats before decomposition to native gates...
```

Hence, now you can create new folders and files write codes and compile and execute programs in VSC.