LRZ oneAPI Workshop, 8-10 November 2022

Intel® DevCloud for oneAPI

Overview

Klaus-Dieter Oertel

Intel® oneAPI Toolkits

A complete set of proven developer tools expanded from CPU to Accelerators

oneAPI

Intel® oneAPI **Base** Toolkit

A core set of high-performance libraries and tools for building C++, SYCL and Python applications

Add-on **Domain-specific** Toolkits

Intel® oneAPI Tools for HPC

Deliver fast Fortran, OpenMP & MPI applications that scale

Intel® oneAPI Tools for IoT

Build efficient, reliable solutions that run at network's edge

Intel® oneAPI AI Analytics Toolkit

Accelerate machine learning & data science pipelines with optimized DL frameworks & high-performing Python libraries

Intel® oneAPI Rendering Toolkit

Create performant, high-fidelity visualization applications

Toolkit **powered by oneAPI**

OpenVINO™

Intel® Distribution of OpenVINO™ Toolkit

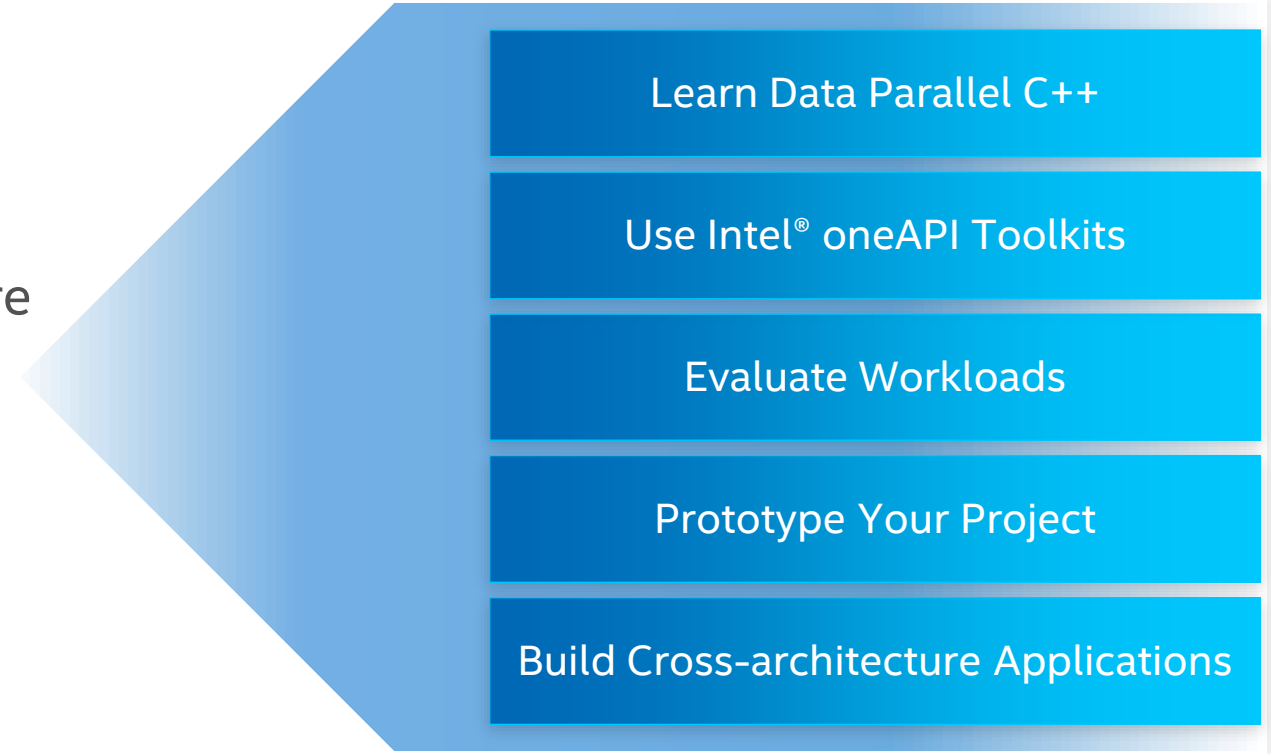
Deploy high performance inference & applications from edge to cloud

Intel® DevCloud for oneAPI

Free Access, A Fast Way to Start Coding

A development sandbox to develop, test and run workloads across a range of Intel® CPUs, GPUs, and FPGAs using Intel's oneAPI software

For customers focused on data-centric workloads on a variety of Intel® architecture



No Downloads | No Hardware Acquisition | No Installation | No Set-up & Configuration

Get Up & Running in Seconds!

software.intel.com/devcloud/oneapi

Getting Started

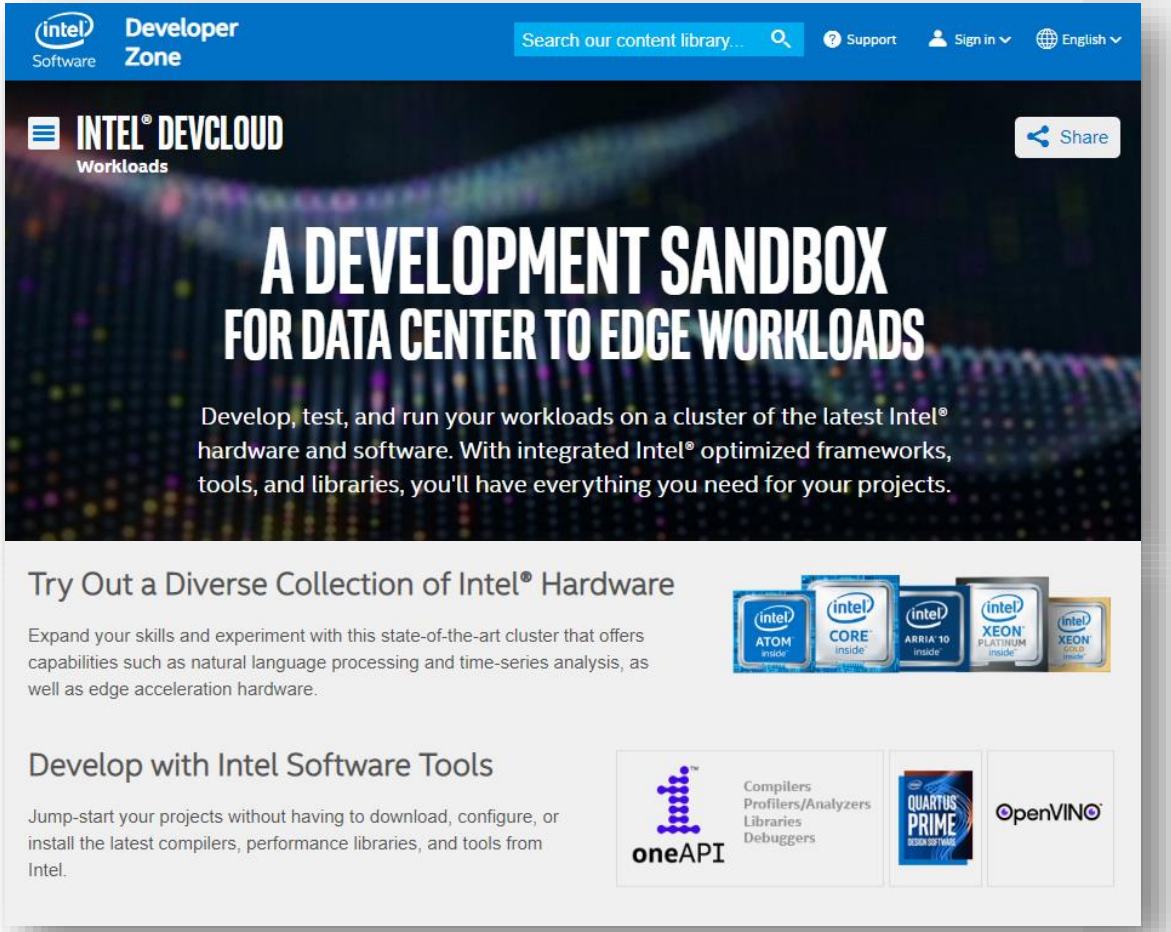
Intel® DevCloud for oneAPI

Users sign up to get an account, & can get started using the DevCloud environment immediately

Learn how to use Intel® oneAPI Toolkits, compilers, performance libraries & tools

Test code & workloads on current & emerging hardware & software

- An open, standards-bases cross-architecture language (Data Parallel C++)
- Advanced libraries for expressing parallelism
- Uncompromised native high-level language performance
- Develop using oneAPI tools on the latest hardware & mix of architectures



Included Toolkits

- Intel® oneAPI Base Toolkit
- Intel® oneAPI HPC Toolkit
- Intel® AI Analytics Toolkit
- Intel® oneAPI Rendering Toolkit
- Intel® oneAPI DL Framework Developer Toolkit
- Intel® Distribution of OpenVINO™ Toolkit
- + more

Hardware

CPU: Intel® Xeon® Scalable processors

GPU: Gen9 and Gen11 GPUs
Intel® Xeon® processors with Intel® Graphics Technology/GPU

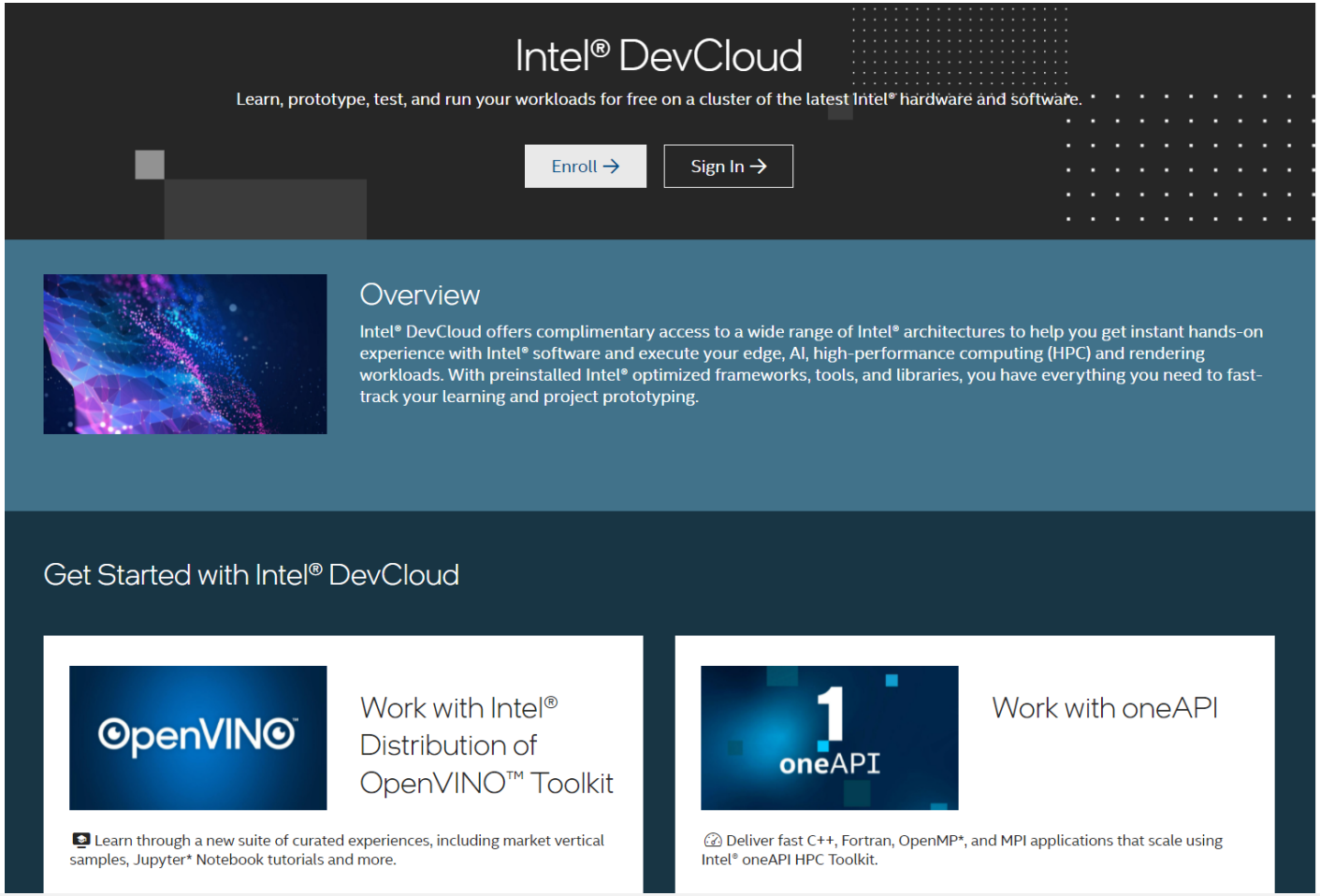
FPGA: Intel® Stratix® 10 FPGAs
Intel® Arria® 10 FPGAs

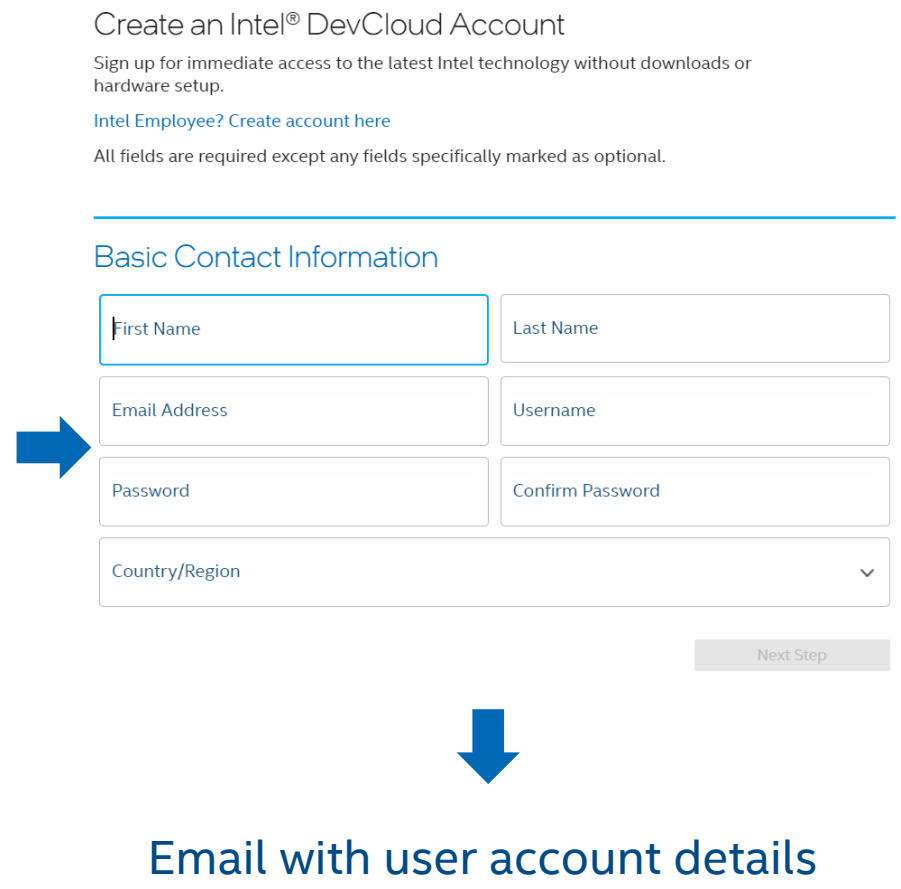
Featured Tools & Libraries

- Intel® OneAPI DPC++ Compiler & Library
- Intel® C++ & Intel® Fortran Compilers
- Intel® OneAPI Math Kernel Library
- Intel® OneAPI Data Analytics Library
- Intel® OneAPI DNN Library
- Intel® Distribution for Python
- Intel® VTune™ Profiler & Intel® Advisor
- Intel® FPGA Add-on for oneAPI Base Toolkit
- + many more...

Registration

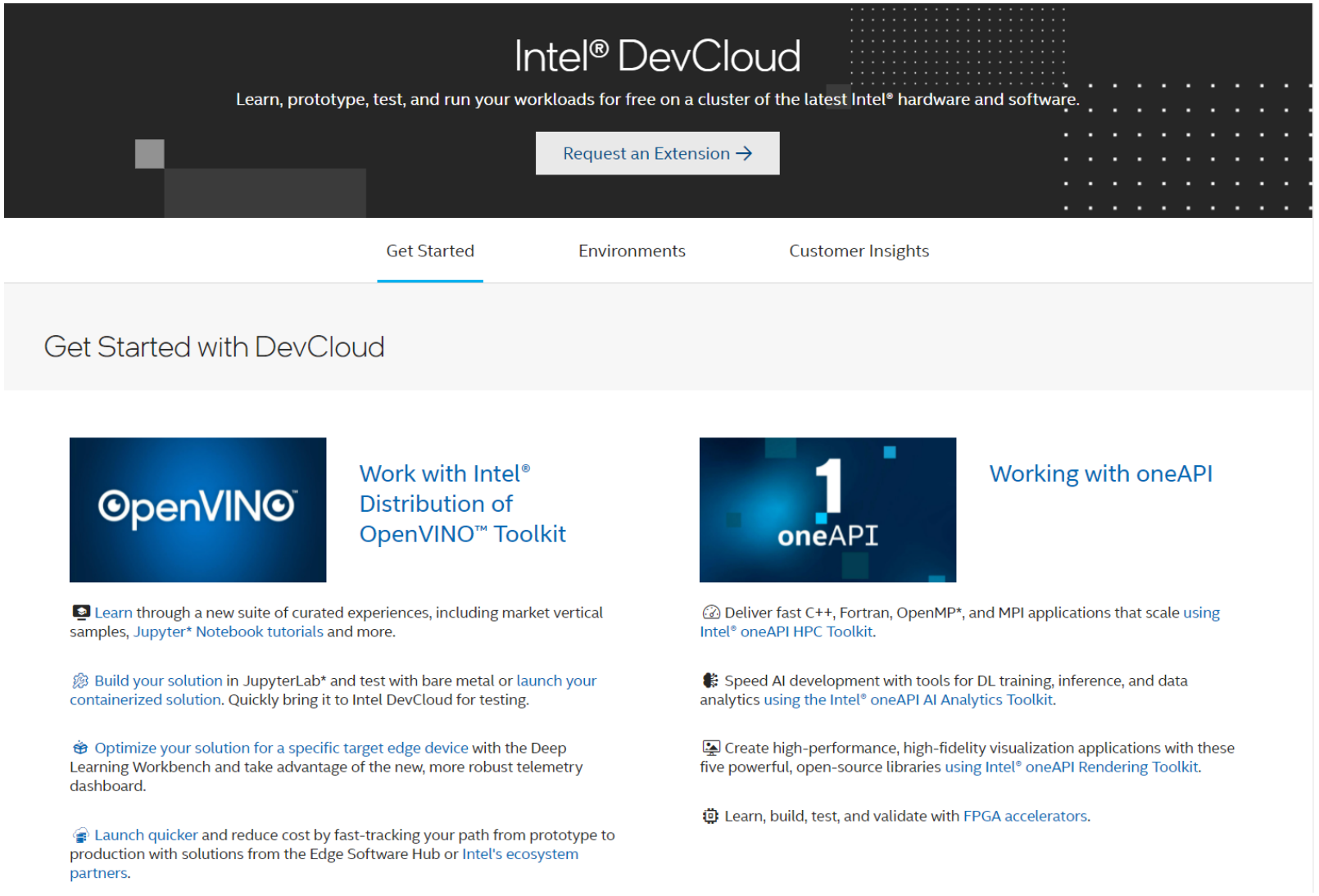
http://devcloud.intel.com





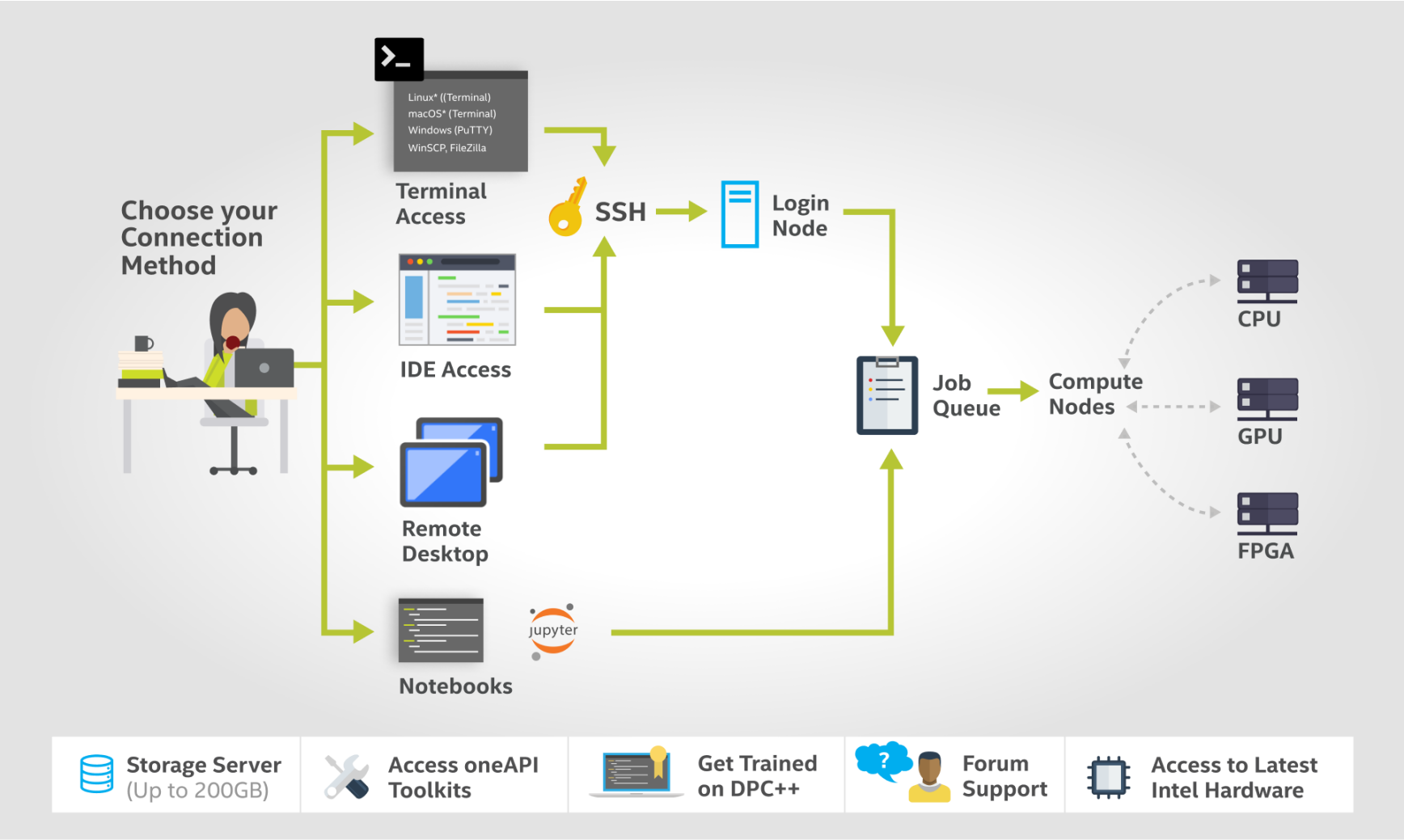
Email with user account details

Login on Web Page



- Access is provided for 120 days
- Extension upon request

Connection Methods



Secure Shell (ssh) Direct Connection

If you are running **Linux or a macOS** operating system you can access the cluster using the native Secure Shell (SSH) client, you will need to set up SSH tunneling as described below.

Option 1: Automated Configuration

The easiest method to set up SSH connection to is by downloading and running an automated installer. The installer will add SSH configuration entries to `~/.ssh/config` and create a private SSH key file inside `~/.ssh`. This method works best if you have only one account.

1. Download and save the automatic installer script customized for your account `u115975`:

   Download setup-devcloud-access-115975.txt

2. Execute this script in a terminal (you may need to adjust the command according to your download location and the downloaded file name):

```
[myname@myhomecomputer] $ | bash ~/Downloads/setup-devcloud-access-115975.txt
```
Bash Copy

3. Clean up for security:

```
[myname@myhomecomputer] $ | rm ~/Downloads/setup-devcloud-access-115975.txt
```
Bash Copy

Option 2: Manual Configuration

Alternatively, if you do not want a script to modify your SSH configuration, you can proceed with manual installation instructions below.

Direct SSH Connection

1. Download and save the SSH access key for Linux/macOS to the folder `~/Downloads/` on your computer

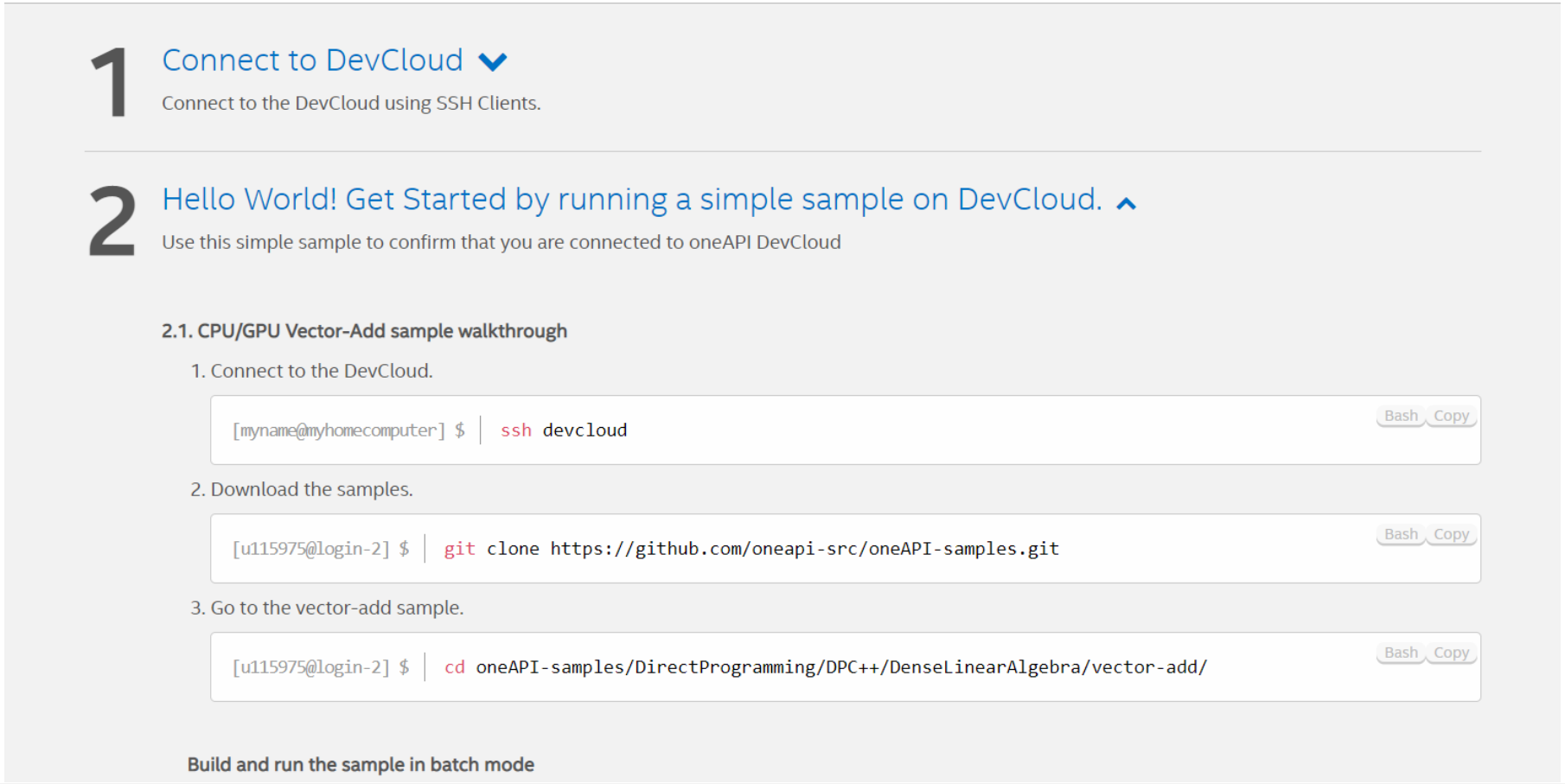
   SSH key for Linux/macOS/Cygwin

2. Create the directory `~/.ssh`, unless it already exists and move the private SSH key into permanent storage in `~/.ssh`:

- Connect with ssh
- SSH key is provided by the DevCloud
- Script available to adapt $HOME/.ssh/config

Get Started (ssh)

https://devcloud.intel.com/oneapi/get_started/baseToolkitSamples



1 Connect to DevCloud

Connect to the DevCloud using SSH Clients.

2 Hello World! Get Started by running a simple sample on DevCloud.

Use this simple sample to confirm that you are connected to oneAPI DevCloud

2.1. CPU/GPU Vector-Add sample walkthrough

1. Connect to the DevCloud.

```
[myname@myhomecomputer] $ | ssh devcloud
```

Bash Copy

2. Download the samples.

```
[u115975@login-2] $ | git clone https://github.com/oneapi-src/oneAPI-samples.git
```

Bash Copy

3. Go to the vector-add sample.

```
[u115975@login-2] $ | cd oneAPI-samples/DirectProgramming/DPC++/DenseLinearAlgebra/vector-add/
```

Bash Copy

Build and run the sample in batch mode

PBS Batch System

- DevCloud uses the PBS Batch System for node access
- Interactive jobs are possible (6 hours default)
- https://devcloud.intel.com/oneapi/documentation/job-submission

How to submit a batch job

```
[u115975@login-2] $ qsub -l nodes=1:gpu:ppn=2 -d . job.sh
```

Note: `-l nodes=1:gpu:ppn=2` (lower case L) is used to assign one full GPU node to the job.
Note: The `-d .` is used to configure the current folder as the working directory for the task.
Note: `job.sh` is the script that gets executed on the compute node.

How to request interactive mode

```
[u115975@login-2] $ qsub -I -l nodes=1:gpu:ppn=2 -d .
```

Note: `-I` (upper case i) is the argument used to request an interactive session.

Basic PBS Queries

- Query available nodes

```
> pbsnodes | grep '^s' | sort
s001-n001
…
s042-n009
```

- Check node characteristics

```
> pbsnodes | grep properties | sort -u
     properties = core,tgl,i9-11900kb,ram32gb,netgbe,gpu,gen11
     properties = xeon,cfl,e-2176g,ram64gb,net1gbe,gpu,gen9
     properties = xeon,clx,ram192gb,net1gbe,batch,extended,fpga,stratix10,fpga_runtime
     properties = xeon,icx,gold6348,ramgb,netgbe,jupyter,batch
     properties = xeon,icx,plat8358,ram256gb,net1gbe,batch
     properties = xeon,icx,plat8380,ram2tb,net1gbe,batch
     properties = xeon,skl,gold6128,ram192gb,net1gbe,fpga_runtime,fpga,arria10
     properties = xeon,skl,gold6128,ram192gb,net1gbe,jupyter,batch
     properties = xeon,skl,gold6128,ram192gb,net1gbe,jupyter,batch,fpga_compile
     properties = xeon,skl,ram384gb,net1gbe,renderkit
```

Basic PBS Queries – NDA Access

- Acquire NDA access for DevCloud account by mail request to Intel
- Query available nodes

```
> export PBS_DEFAULT=v-qsvr-nda
> pbsnodes | grep '^s' | sort
S001-n136
…
s101-n001
```

- Check node characteristics

```
> pbsnodes | grep properties | sort -u
     properties = core,cfl,i9-10920x,ram32gb,net1gbe,batch,xe_hp,gpu,quad_gpu
     properties = core,icx,gold6336y,ram128gb,net1gbe,batch,gpu,single_gpu
     properties = core,icx,plat8380,ram2048gb,net1gbe,batch
     properties = xeon,cnl,e-2146g,ram64gb,net1gbe,experimental,fpga,arria10
     properties = xeon,icx,gold6336y,ram128gb,net1gbe,batch,ats,gpu,dual_gpu
     properties = xeon,icx,gold6336y,ram128gb,net1gbe,batch,ats,gpu,single_gpu
     …
     properties = xeon,spr,ram1024gb,netgbe,nda,dnp50,gpu,quad_gpu
```

Basic oneAPI Queries

- oneAPI environment on node

```
> source /opt/intel/oneapi/setvars.sh
> which dpcpp icx sycl-ls
/glob/development-tools/versions/oneapi/2022.3/oneapi/compiler/2022.2.0/linux/bin/dpcpp
/glob/development-tools/versions/oneapi/2022.3/oneapi/compiler/2022.2.0/linux/bin/icx
/glob/development-tools/versions/oneapi/2022.3/oneapi/compiler/2022.2.0/linux/bin/sycl-ls
```

- Check GPU characteristics

```
> sycl-ls --verbose
...
Platform [#4]:
    Version  : 1.3
    Devices  : 1
        Device [#0]:
        Type        : gpu
        Version     : 1.3
        Name        : Intel(R) UHD Graphics [0x9a60]
        Vendor      : Intel(R) Corporation
        Driver      : 1.3.23405
```

Code Samples

- OpenMP offload

MandelbrotOMP sample
This sample demonstrates how to accelerate program performance with SIMD and parallelization using OpenMP*, in the context of calculating the Mandelbrot set.

View code on GitHub*

openMP Reduction Sample
The openmp_reduction code sample is a simple program that calculates pi. This program is implemented using C++ and openMP for Intel CPU and accelerators.

View code on GitHub*

ISO3DFD Open MP Offload Sample
The ISO3DFD sample refers to Three-Dimensional Finite-Difference Wave Propagation in Isotropic Media. It is a three-dimensional stencil to simulate a wave propagating in a 3D isotropic medium and shows some of the more common challenges and techniques when targeting OMP Offload devices (GPU) in more complex applications to achieve good performance.

View code on GitHub*

- DPC++

Direct Programming/DPC++

Vector-Add
This simple vector-add program in Data Parallel C++ (DPC++) supports FPGAs, GPUs, and CPUs.

View code on GitHub*

Mandelbrot Sample
Mandelbrot is an infinitely complex fractal patterning that is derived from a simple formula. It demonstrates using DPC++ for offloading computations to a GPU (or other devices) and shows how processing time can be optimized and improved with parallelism.

View code on GitHub*

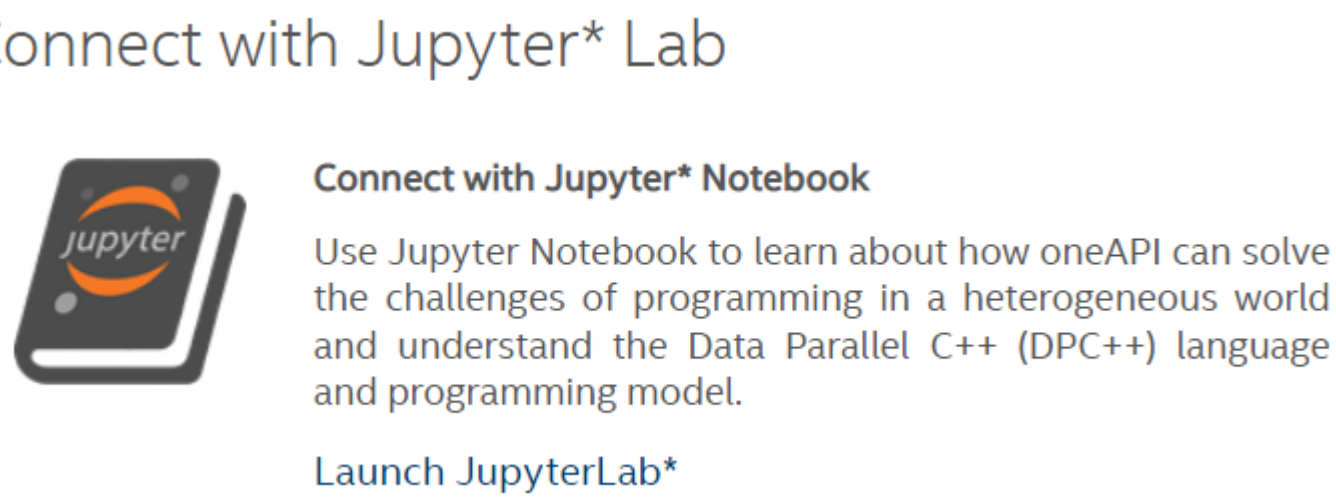
Complex Multiplication Sample
Complex multiplication is a program that multiplies two large vectors of Complex numbers in parallel and verifies the results. It also implements a custom device selector to target a specific vendor device. This program is implemented using C++ and DPC++ language for Intel CPU and accelerators. The Complex class is a custom class, and this program shows how we can use custom types of classes in a DPC++ program.

Sepia Filter
A program that converts an image to sepia tone.

View code on GitHub*

Connection with Jupyter* Notebook

- JupyterLab*

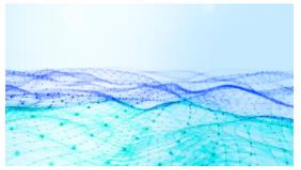


- JupyterLabs* for AI

AI Sample Applications

Find sample applications for your specific market needs with examples of how to optimize, tune, and accelerate your applications.

Learn More



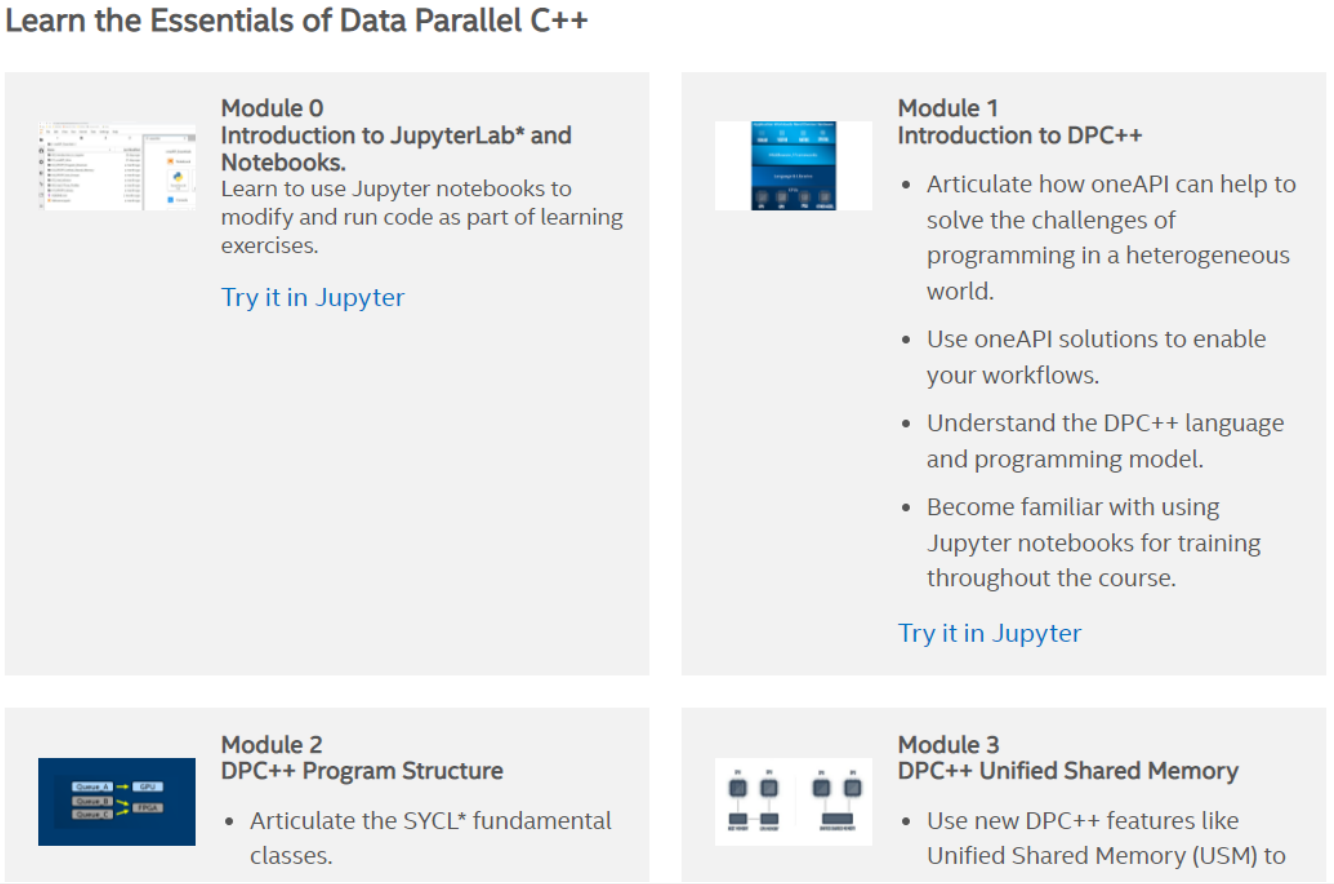
Connect and Create

Develop your own machine learning solutions using Jupyter* Notebooks or a containerized launch environment. Benchmark your code and optimize it for Intel® hardware.

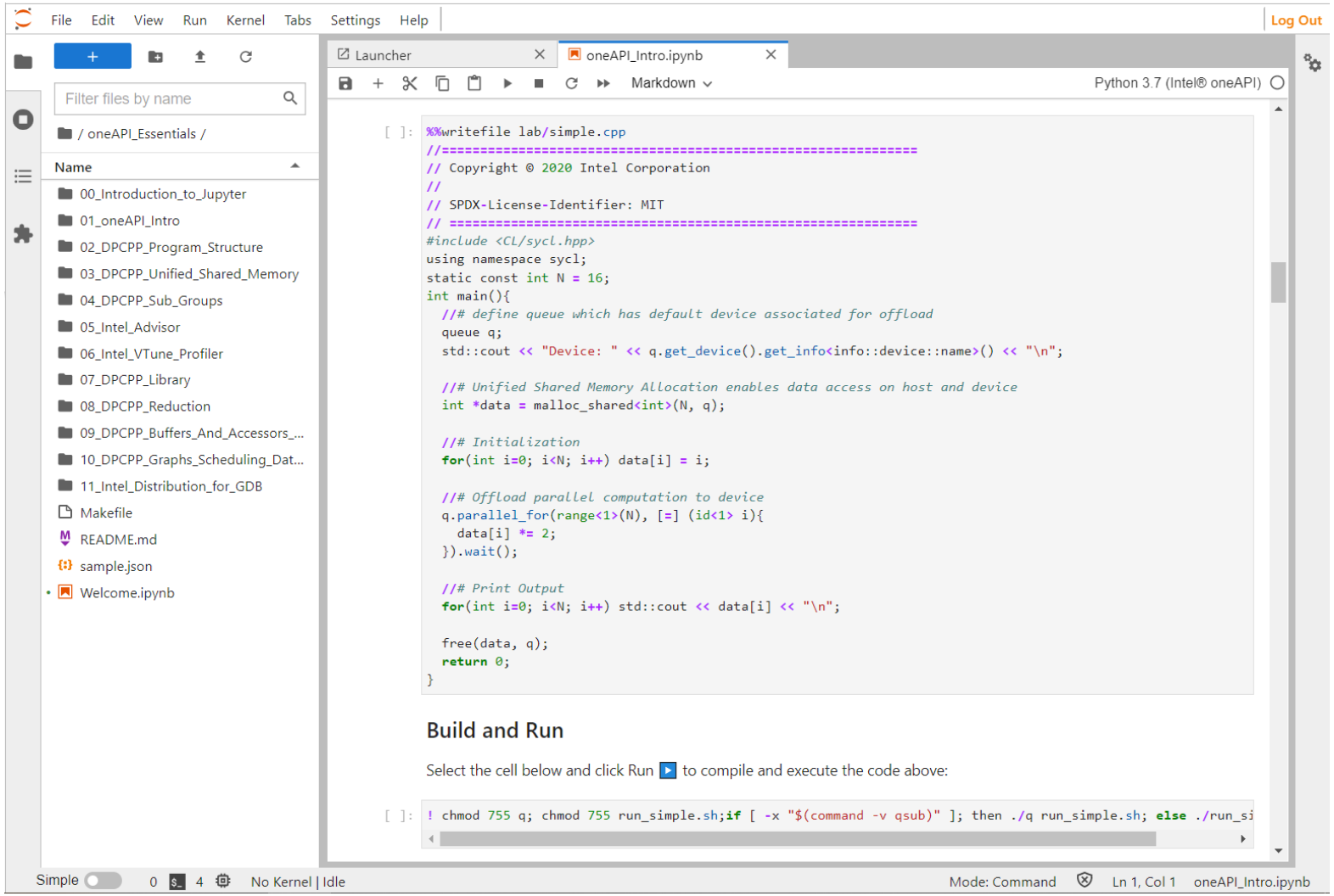
Connect to JupyterLab
Connect to Container Playground

Basic Training Modules in JupyterLab*

- https://devcloud.intel.com/oneapi/get_started/baseTrainingModules



oneAPI Essentials in JupyterLab*



Connection with Visual Studio Code*

Connect to DevCloud with Visual Studio Code

NOTE: Windows users must first download and install Cygwin before proceeding. Once it has been installed, return to this page to configure your connection.

Requirements:

- Windows users install Cygwin from the installation page
- VS Code
- VS Code SSH extension
- VS Code DevCloud Connector extension

Cygwin Installation

The Cygwin* environment offers a convenient way of connecting to the Intel® DevCloud from a local machine running Windows*, whether you have a direct connection or find yourself behind a proxy. If you already have Cygwin installed, please skip to the SSH connection instructions.

NOTE: Your Cygwin installation requires the openssh (ssh), nc and nano packages.

The following instructions will help you install a minimal version of Cygwin for accessing Intel DevCloud. For your convenience we're providing a simple script that automates the installation of Cygwin.

Download install_cygwin.bat from the installation page. It can be run from anywhere on your disk, either by executing it from the terminal or by double clicking on it.

The script uses curl to download the Cygwin setup file. When asked to provide proxy details, you can do so by entering proxy:port when asked, or by simply hitting enter to continue without a proxy.

The default installation path is c:\cygwin64. The script will prompt you to change this if you wish to install elsewhere.

Several Cygwin packages are downloaded during the installation. The script is configured to use mirrors.kernel.org as the default download site. A full list of Cygwin mirror sites can be found on the Cygwin homepage https://www.cygwin.com/.

Notices & Disclaimers

Texas Advanced Computing Center (TACC) Frontera references
Article: *HPCWire: Visualization & Filesystem Use Cases Show Value of Large Memory Fat Notes on Frontera*.
www.intel.com/content/dam/support/us/en/documents/memory-and-storage/data-center-persistent-mem/Intel-Optane-DC-Persistent-Memory-Quick-Start-Guide.pdf
software.intel.com/content/www/us/en/develop/articles/introduction-to-programming-with-persistent-memory-from-intel.html
wreda.github.io/papers/assise-osdi20.pdf

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.


intel