

EuroCC/Castiel oneAPI Workshop, 17-18 Feb 2022

Intel® DevCloud for oneAPI

Overview

Klaus-Dieter Oertel

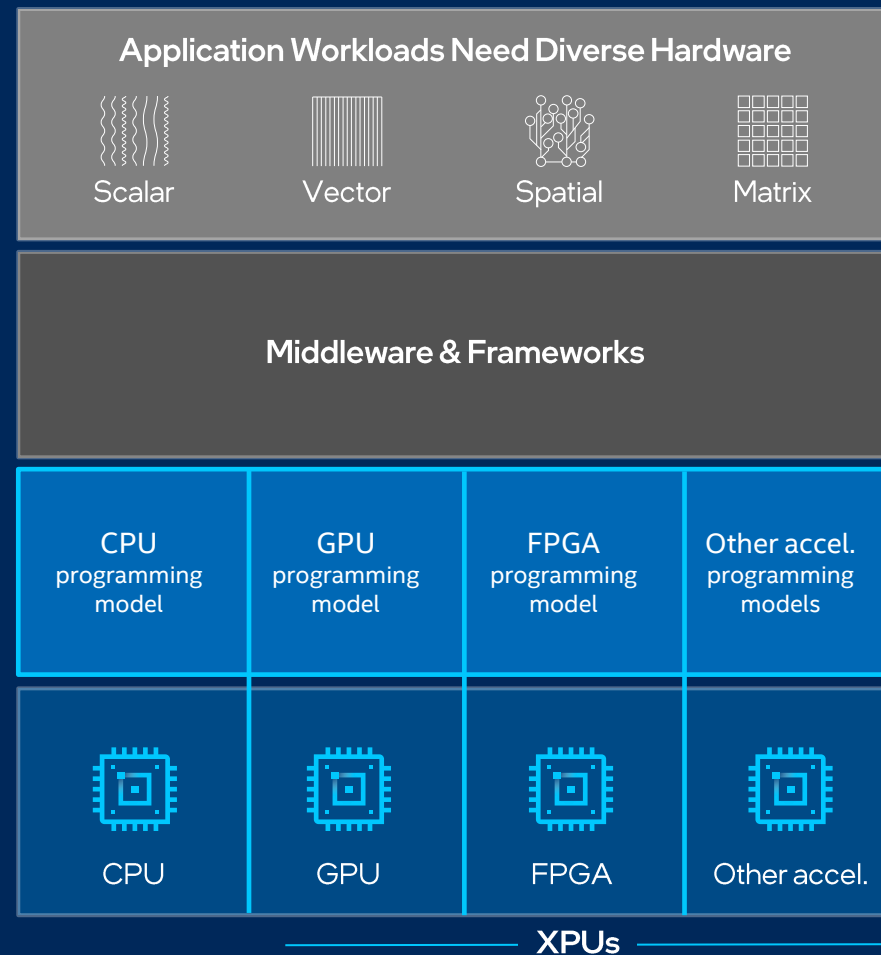
Programming Challenges for Multiple Architectures

Growth in specialized workloads

Variety of data-centric hardware required

Requires separate programming models and toolchains for each architecture

Software development complexity limits freedom of architectural choice



oneAPI

One Programming Model for Multiple Architectures and Vendors



Freedom to Make Your Best Choice

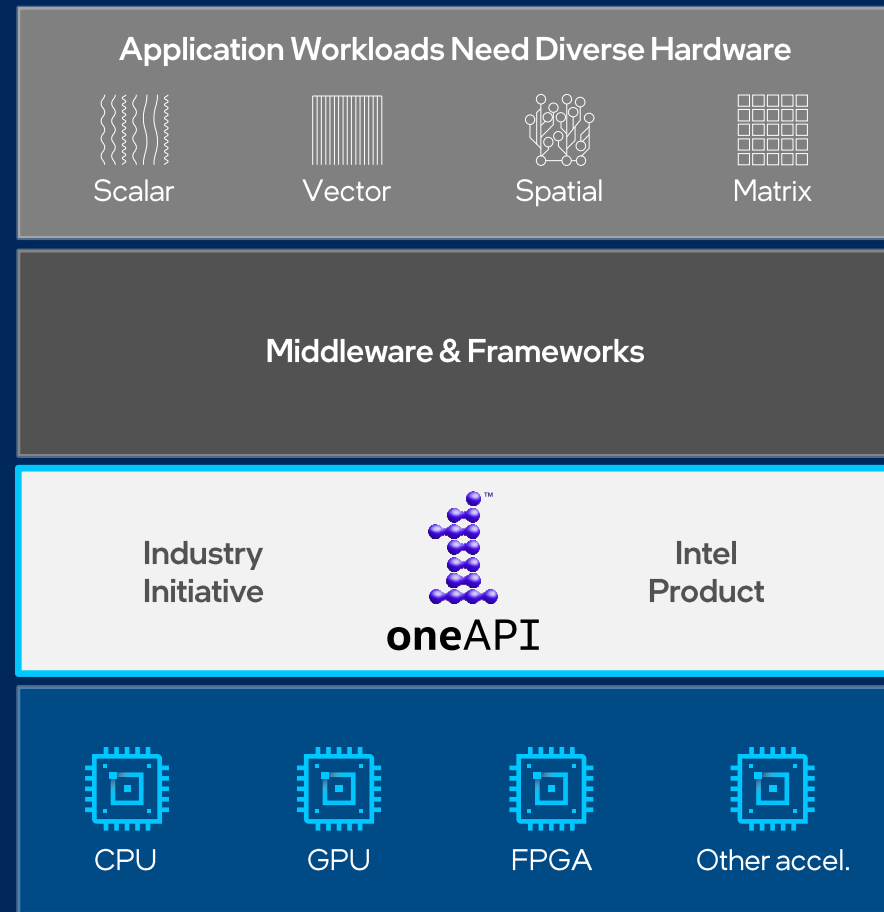
- Choose the best accelerated technology the software doesn't decide for you

Realize all the Hardware Value

- Performance across CPU, GPUs, FPGAs, and other accelerators

Develop & Deploy Software with Peace of Mind

- Open industry standards provide a safe, clear path to the future
- Compatible with existing languages and programming models including C++, Python, SYCL, OpenMP, Fortran, and MPI



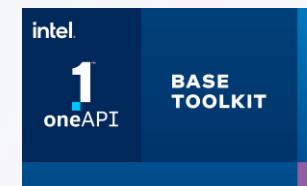
Intel® oneAPI Toolkits

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



Intel® oneAPI Base Toolkit

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

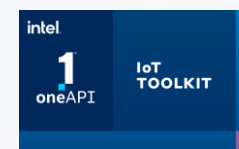


Add-on Domain-specific Toolkits



Intel® oneAPI Tools for HPC

Deliver fast Fortran, OpenMP & MPI applications that scale



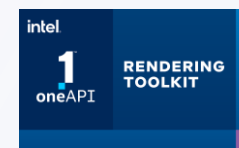
Intel® oneAPI Tools for IoT

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



Intel® Distribution of OpenVINO™ Toolkit

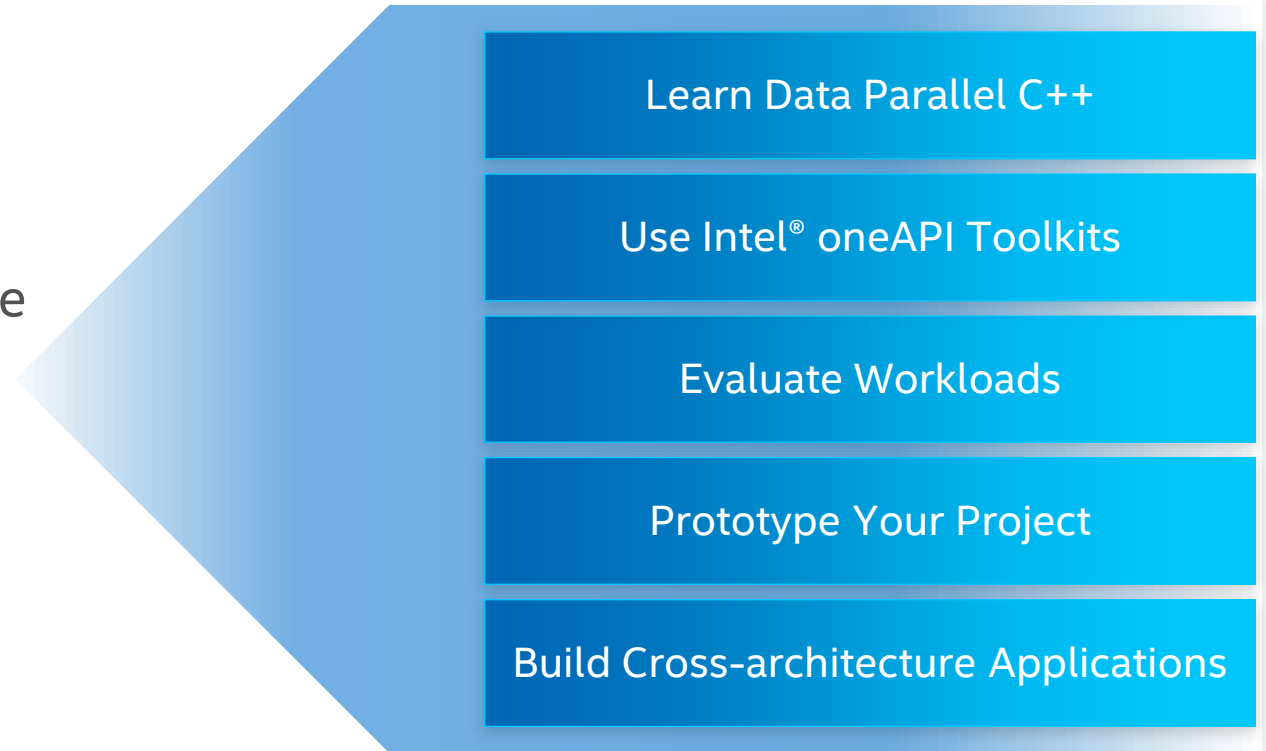
Deploy high performance inference & applications from edge to cloud

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-based 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: Intel® Iris® Xe MAX GPU
Intel® Xeon® processors with Intel® Graphics Technology/GPU
FPGA: Intel® Stratix® 10 FPGAs
Intel® Arria® 10 FPGAs

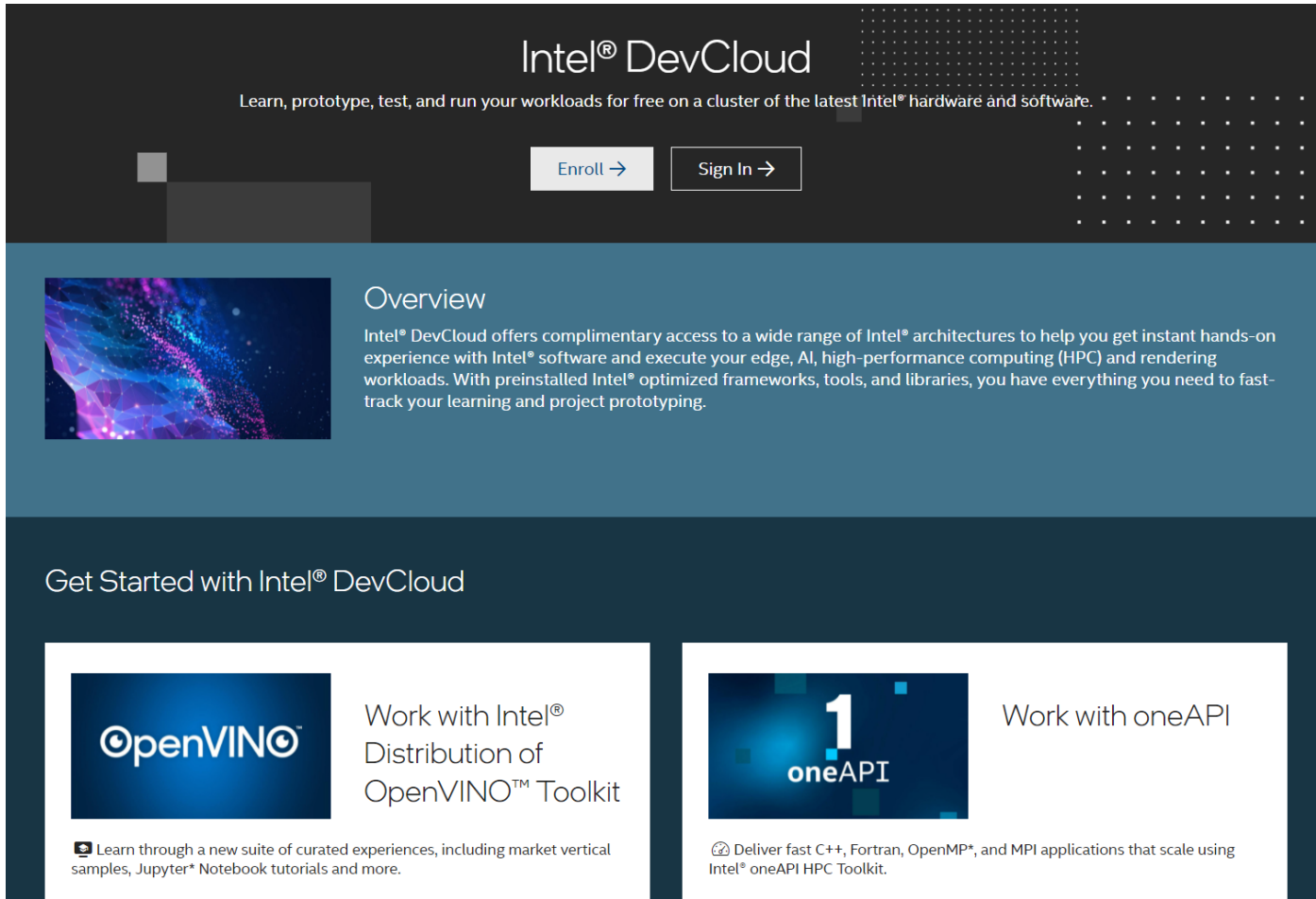
Featured Tools & Libraries

- 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...

The screenshot shows the Intel Developer Zone website. At the top, there's a blue header with the Intel logo, 'Developer Zone', and a search bar. Below the header, the main content area has a dark background with the text 'INTEL® DEVCLOUD Workloads' and 'A DEVELOPMENT SANDBOX FOR DATA CENTER TO EDGE WORKLOADS'. It describes the environment as a development sandbox for data center to edge workloads, allowing users to develop, test, and run workloads on a cluster of the latest Intel hardware and software. Below this, there are two sections: 'Try Out a Diverse Collection of Intel® Hardware' which lists various Intel processors (Atom, Core, Arria 10, Xeon Platinum, Xeon) and 'Develop with Intel Software Tools' which lists various software tools (oneAPI, Quartus Prime, OpenVINO).

Registration

<http://devcloud.intel.com>



The screenshot shows the Intel DevCloud website. At the top, the Intel DevCloud logo is displayed with the tagline "Learn, prototype, test, and run your workloads for free on a cluster of the latest Intel® hardware and software." Below this are "Enroll →" and "Sign In →" buttons. The main content area is divided into two sections. The "Overview" section features a colorful abstract image and text stating that Intel DevCloud offers complimentary access to a wide range of Intel architectures for hands-on experience with Intel software, AI, high-performance computing (HPC), and rendering workloads. The "Get Started with Intel DevCloud" section contains two cards: one for "OpenVINO™" titled "Work with Intel® Distribution of OpenVINO™ Toolkit" and another for "oneAPI" titled "Work with oneAPI". Each card includes a brief description of the toolkit and a link to learn more.

Intel® DevCloud


Learn, prototype, test, and run your workloads for free on a cluster of the latest Intel® hardware and software.

Enroll → Sign In →

Overview


Intel® DevCloud offers complimentary access to a wide range of Intel® architectures to help you get instant hands-on experience with Intel® software and execute your edge, AI, high-performance computing (HPC) and rendering workloads. With preinstalled Intel® optimized frameworks, tools, and libraries, you have everything you need to fast-track your learning and project prototyping.

Get Started with Intel® DevCloud



Work with Intel® Distribution of OpenVINO™ Toolkit

Learn through a new suite of curated experiences, including market vertical samples, Jupyter® Notebook tutorials and more.



Work with oneAPI

Deliver fast C++, Fortran, OpenMP®, and MPI applications that scale using Intel® oneAPI HPC Toolkit.


Create an Intel® DevCloud Account

Sign up for immediate access to the latest Intel technology without downloads or hardware setup.

[Intel Employee? Create account here](#)

All fields are required except any fields specifically marked as optional.

Basic Contact Information



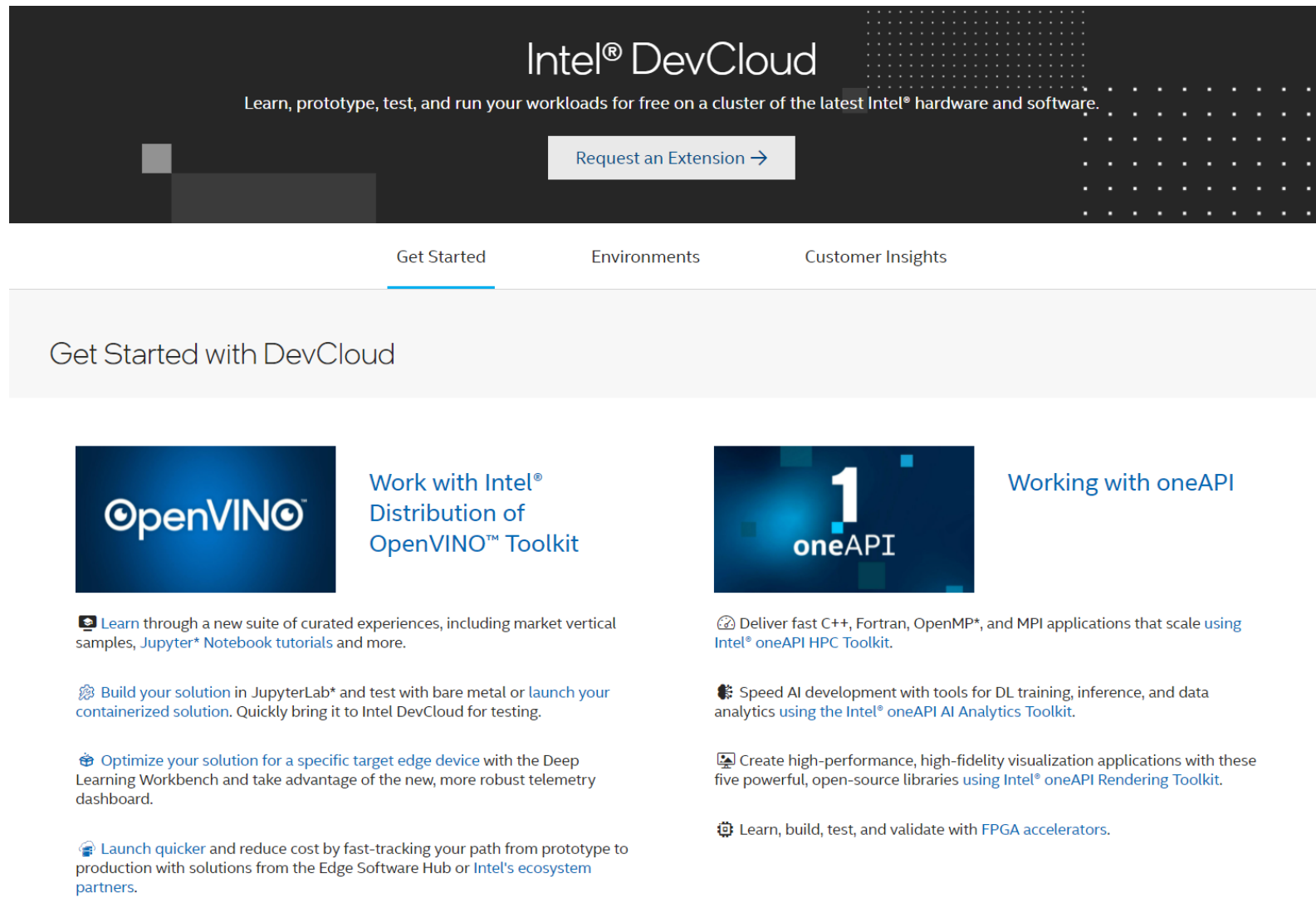
First Name	Last Name
Email Address	Username
Password	Confirm Password
Country/Region	

Next Step



Email with user account details

Login on Web Page



The screenshot shows the Intel DevCloud website. At the top, the Intel DevCloud logo is displayed with the tagline "Learn, prototype, test, and run your workloads for free on a cluster of the latest Intel® hardware and software." Below this is a "Request an Extension →" button. A navigation bar contains "Get Started", "Environments", and "Customer Insights". The "Get Started" section is active, showing a "Get Started with DevCloud" heading. Two main toolkits are featured: "Work with Intel® Distribution of OpenVINO™ Toolkit" and "Working with oneAPI". Each toolkit has a list of features and links to learn more.


Intel® DevCloud

Learn, prototype, test, and run your workloads for free on a cluster of the latest Intel® hardware and software.

[Request an Extension →](#)


[Get Started](#) [Environments](#) [Customer Insights](#)

Get Started with DevCloud



Work with Intel® Distribution of OpenVINO™ Toolkit

- [Learn](#) through a new suite of curated experiences, including market vertical samples, [Jupyter* Notebook tutorials](#) and more.
- [Build your solution](#) in JupyterLab* and test with bare metal or [launch your containerized solution](#). Quickly bring it to Intel DevCloud for testing.
- [Optimize your solution](#) for a specific target edge device with the Deep Learning Workbench and take advantage of the new, more robust telemetry dashboard.
- [Launch quicker](#) and reduce cost by fast-tracking your path from prototype to production with solutions from the Edge Software Hub or [Intel's ecosystem partners](#).

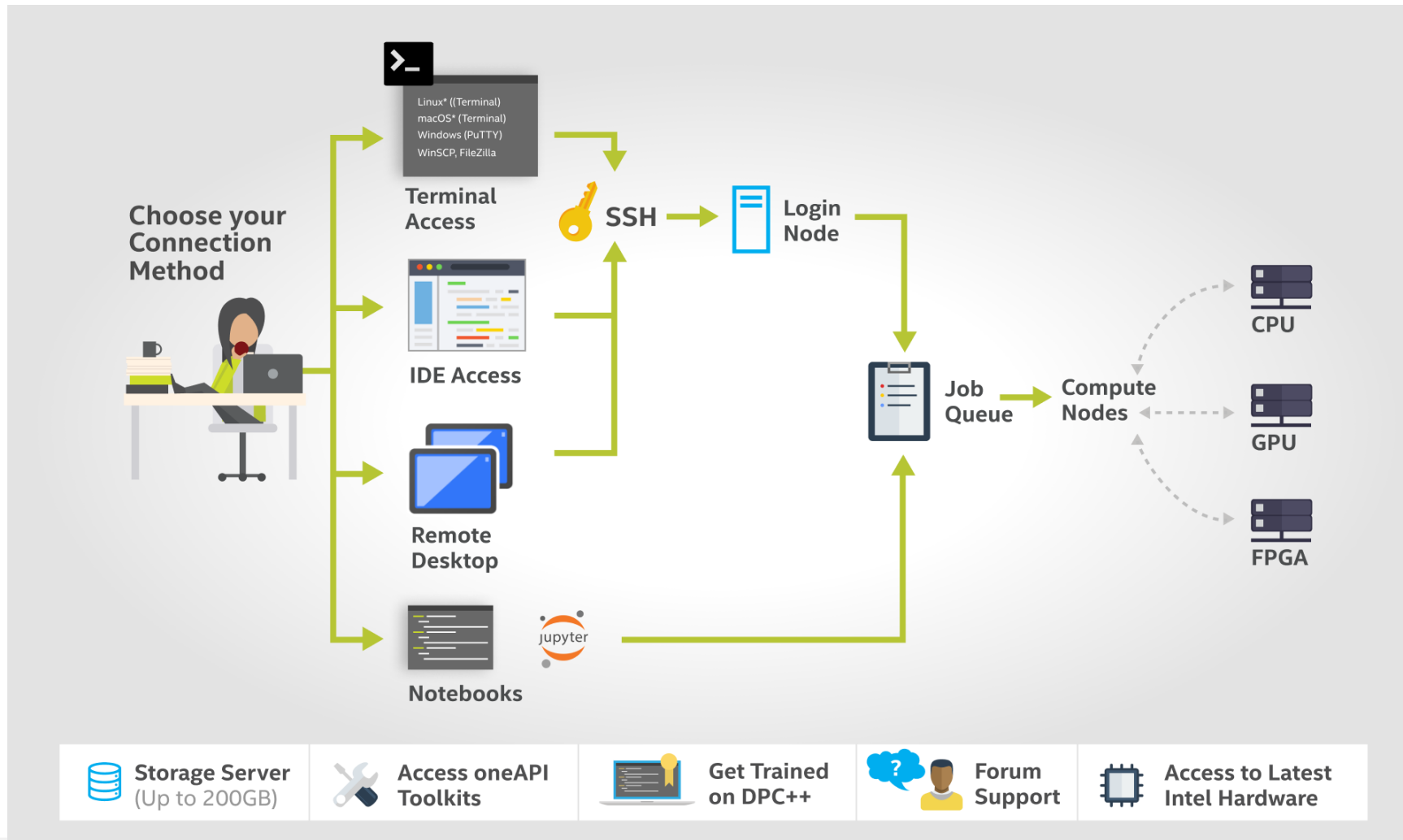


Working with oneAPI

- [Deliver fast C++, Fortran, OpenMP*, and MPI applications](#) that scale [using Intel® oneAPI HPC Toolkit](#).
- [Speed AI development](#) with tools for DL training, inference, and data analytics [using the Intel® oneAPI AI Analytics Toolkit](#).
- [Create high-performance, high-fidelity visualization applications](#) with these five powerful, open-source libraries [using Intel® oneAPI Rendering Toolkit](#).
- [Learn, build, test, and validate](#) with [FPGA accelerators](#).

- Access is provided for 120 days
- Extension upon request

Connection Methods



Secure Shell (ssh) Direct Connection

^ Connect to the DevCloud

Download & Configure Third Party Dependencies

Connect with Cygwin

Connect with VSCode

Connect with
Linux/macOS SSH

v How to use the DevCloud

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'
s001-n001
...
s012-n005
```

■ Check node characteristics

```
> pbsnodes | grep properties | sort -u
properties = core,cfl,i9-10920x,ram32gb,net1gbe,gpu,iris_xe_max,dual_gpu
properties = core,cfl,i9-10920x,ram32gb,net1gbe,gpu,iris_xe_max,quad_gpu
properties = xeon,cfl,e-2176g,ram64gb,net1gbe,gpu,gen9
properties = xeon,clx,ram192gb,net1gbe,fpga_runtime,fpga,stratix10
properties = xeon,icx,gold6348,ramgb,netgbe,jupyter,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 oneAPI Queries

■ oneAPI environment on node

```
> source /opt/intel/oneapi/setvars.sh
> which dpcpp icx sycl-ls
/glob/development-tools/versions/oneapi/2022.1.2/oneapi/compiler/2022.0.2/linux/bin/dpcpp
/glob/development-tools/versions/oneapi/2022.1.2/oneapi/compiler/2022.0.2/linux/bin/icx
/glob/development-tools/versions/oneapi/2022.1.2/oneapi/compiler/2022.0.2/linux/bin/sycl-ls
```

■ Check GPU characteristics

```
> sycl-ls --verbose
...
Platform [#4]:
  Version   : 1.2
  Devices   : 1
    Device [#0]:
      Type      : GPU
      Version    : 1.2
      Name       : Intel(R) Iris(R) Xe MAX Graphics [0x4905]
      Vendor     : Intel(R) Corporation
      Driver     : 1.2.21786
```

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*](#)

Connect with Jupyter* Lab



Connect with Jupyter* Notebook

Use Jupyter Notebook to learn about how oneAPI can solve the challenges of programming in a heterogeneous world and understand the Data Parallel C++ (DPC++) language and programming model.

[Launch 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.

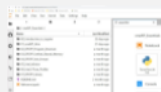
[Connect to JupyterLab](#)

[Connect to Container Playground](#)

Basic Training Modules in JupyterLab*

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

Learn the Essentials of Data Parallel C++



Module 0 Introduction to JupyterLab* and Notebooks.

Learn to use Jupyter notebooks to modify and run code as part of learning exercises.

[Try it in Jupyter](#)



Module 1 Introduction to DPC++

- Articulate how oneAPI can help to solve the challenges of programming in a heterogeneous world.
- Use oneAPI solutions to enable your workflows.
- Understand the DPC++ language and programming model.
- Become familiar with using Jupyter notebooks for training throughout the course.

[Try it in Jupyter](#)



Module 2 DPC++ Program Structure

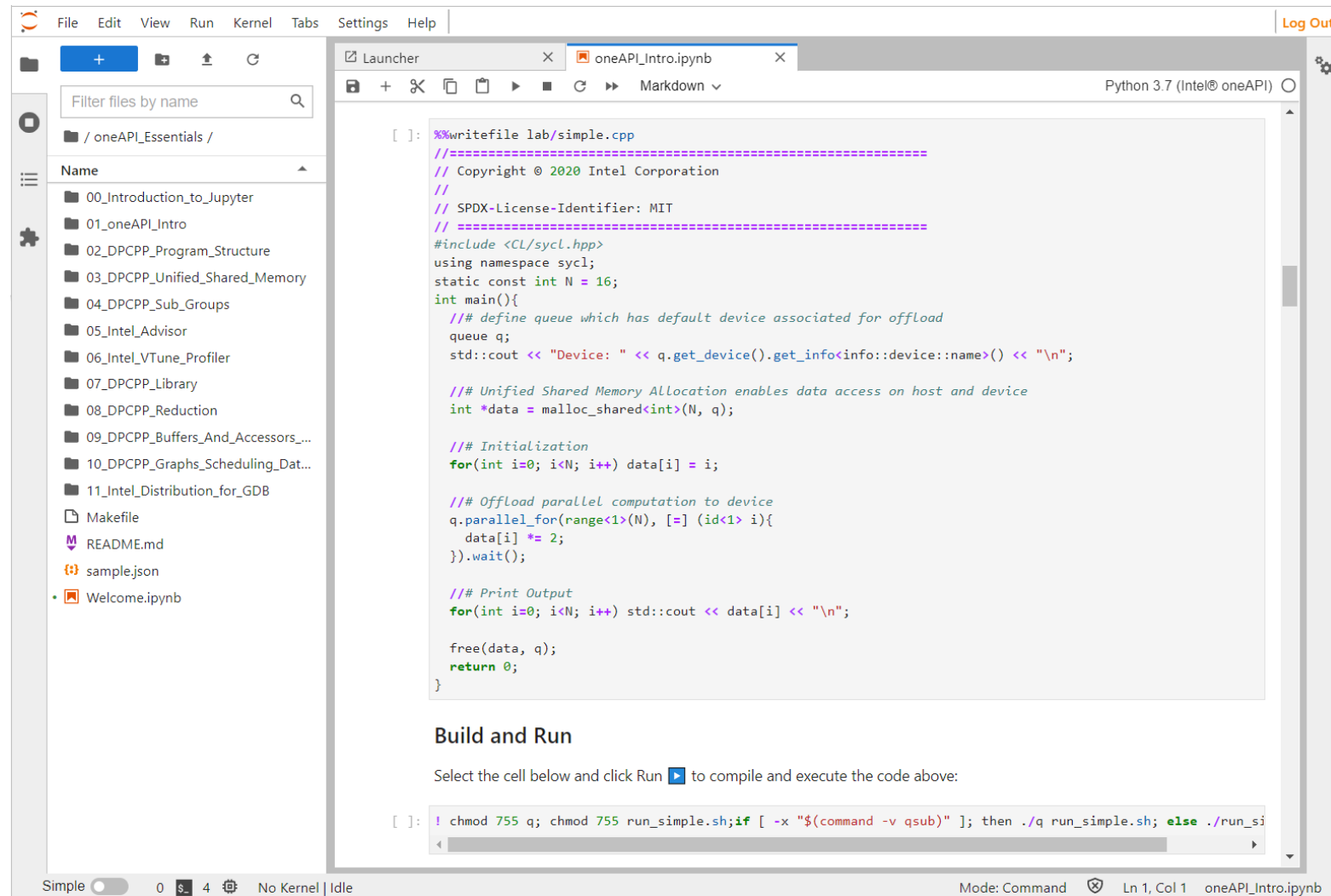
- Articulate the SYCL* fundamental classes.



Module 3 DPC++ Unified Shared Memory

- Use new DPC++ features like Unified Shared Memory (USM) to

oneAPI Essentials in JupyterLab*



The screenshot shows the JupyterLab interface. On the left is a file explorer with a search bar and a list of files and folders. The main area is a code editor with a tab for 'oneAPI_Intro.ipynb'. The code is a C++ program that demonstrates offload parallel computation to a device. Below the code, there is a 'Build and Run' section with instructions and a terminal output area.

```
[ ]: %%writefile lab/simple.cpp
// =====
// Copyright © 2020 Intel Corporation
//
// SPDX-License-Identifier: MIT
// =====
#include <CL/sycl.hpp>
using namespace sycl;
static const int N = 16;
int main(){
    // define queue which has default device associated for offload
    queue q;
    std::cout << "Device: " << q.get_device().get_info<info::device::name>() << "\n";

    // Unified Shared Memory Allocation enables data access on host and device
    int *data = malloc_shared<int>(N, q);


    // Initialization
    for(int i=0; i<N; i++) data[i] = i;

    // Offload parallel computation to device
    q.parallel_for(range<1>(N), [=] (id<1> i){
        data[i] *= 2;
    }).wait();



    // Print Output
    for(int i=0; i<N; i++) std::cout << data[i] << "\n";

    free(data, q);
    return 0;
}
```

Build and Run

Select the cell below and click Run  to compile and execute the code above:

```
[ ]: ! chmod 755 q; chmod 755 run_simple.sh;if [ -x "$(command -v qsub)" ]; then ./q run_simple.sh; else ./run_si
```

Simple ☐ 0  4  No Kernel | Idle Mode: Command Ln 1, Col 1 oneAPI_Intro.ipynb

Connection with Visual Studio Code*

▲ Connect to the DevCloud

Download & Configure Third Party Dependencies

Connect with Cygwin

Connect with VSCode

Using the Code Sample
Browser for Intel®
oneAPI Toolkit
Extension on DevCloud

Connect with
Linux/macOS SSH

▼ How to use the DevCloud

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 Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

