

# Intel® oneAPI, News and Advances 2024

## Europar24-Madrid

CGS

July 30, 2024

- “Intel® oneAPI Programming Guide”,  
<https://www.intel.com/content/www/us/en/develop/documentation/oneapi-programming-guide/top.html>



# Outline

1 Introduction

2 Suite oneAPI

3 Intel Developer Cloud

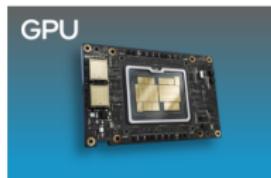
4 Other





# Introduction

- High-Performance Computing (HPC) should be an exclusive concern of large-scale science
- ... but it is becoming a fundamental feature in other domains such as AI, data analysis, content creation, or graphics
- **Proliferation of architectures:** GPUs, FPGAs, ASICs...
  - Programmers have to adapt their developments to more and more complex programming on different type of accelerators



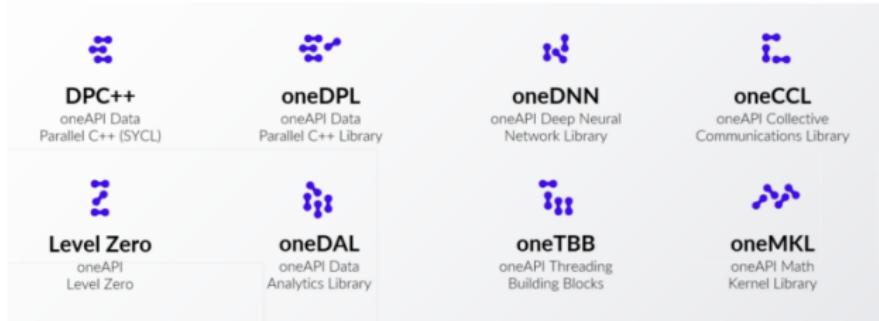
- Establishment of the [Unified Acceleration Foundation \(UXL\)](#) announced at Linux Foundation Open Source Summit Sept23
- Executive members: Arm, Fujitsu, Google, Imagination Technologies, Intel, Qualcomm Technologies Inc, and Samsung
- **Objective:** an open and standards-based programming model for all accelerators, promoting support for multiple architectures and multiple vendors

## UXL FOUNDATION

Unified Acceleration



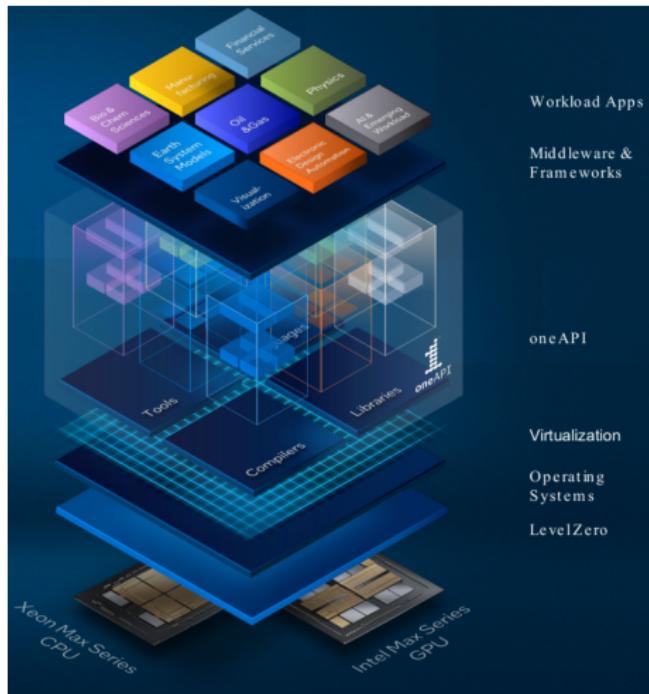
- **SYCL** as the open standard from Khronos and the **oneAPI** specification form the basis of UXL Foundation's efforts
- Collaboration with processor vendors and software developers, aligning with standards bodies such as Khronos and ISO C++



oneAPI elements Intel is donating to UXL

# oneAPI Intro

- Unified programming model: diverse architectures
- Optimized language and libraries
- Performance equivalent to high-level native language
- Based on industry standards and open specifications
- Compatible with existing HPC programming models

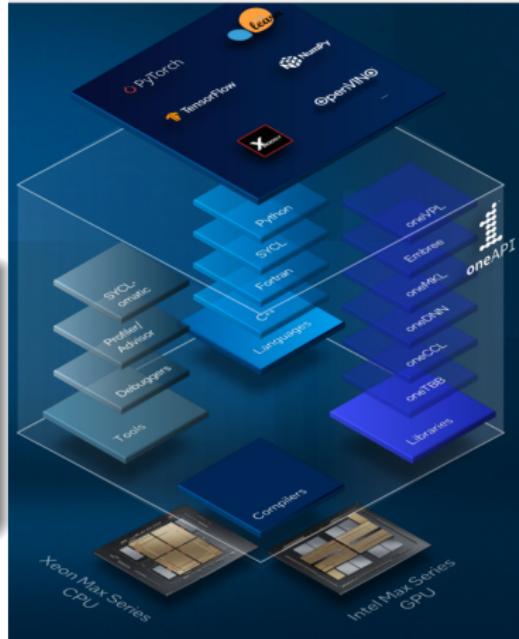


# oneAPI Intro

- Standards-based language: C++ and SYCL
- Powerful APIs to accelerate domain-specific functions

## Solution to single provider

- Open standard to promote community and industry support
- Allows code reuse across different architectures and providers





- A comprehensive set of development tools tested from CPU to XPU
- Available for installation in Intel® oneAPI Toolkits

Intel® oneAPI Base Toolkit	 The logo consists of the Intel 'i' icon followed by a large white '1' inside a blue square, with the text 'oneAPI' below it, all contained within a dark blue rectangular background.	A core set of high-performance libraries and tools for building C++, SYCL, C/OpenMP, and Python applications
Add-on Domain-specific Toolkits	 The logo features the Intel 'i' icon, a large white '1' in a blue square, and the text 'HPC TOOLKIT'.   The logo features the Intel 'i' icon, a large white '1' in a blue square, and the text 'IOT TOOLKIT'.   The logo features the Intel 'i' icon, a large white '1' in a blue square, and the text 'RENDERING TOOLKIT'.	<p>For HPC developers</p> <p>For Edge &amp; IoT developers</p> <p>For visual creators, scientists, and engineers</p> <p><b>Intel® oneAPI Tools for HPC</b> Deliver fast Fortran, OpenMP &amp; MPI applications that scale</p> <p><b>Intel® oneAPI Tools for IoT</b> Build efficient, reliable solutions that run at network's edge</p> <p><b>Intel® oneAPI Rendering Toolkit</b> Create performant, high-fidelity visualization applications</p>
Toolkits powered by oneAPI	 The logo features the Intel 'i' icon, a large white '1' in a blue square, and the text 'AI ANALYTICS TOOLKIT'.	<p>For AI developers and data scientists</p> <p><b>Intel® AI Analytics Toolkit</b> Accelerate machine learning &amp; data science pipelines end-to-end with optimized DL &amp; ML frameworks &amp; high-performing Python libraries</p> <p><b>OpenVINO™</b> For deep learning inference developers</p> <p><b>Intel® OpenVINO™ toolkit</b> Deploy high performance inference &amp; applications from edge to cloud</p>
<p>Download at <a href="https://intel.com/oneAPI">intel.com/oneAPI</a> Or run tools in the Intel® Developer Cloud</p>		

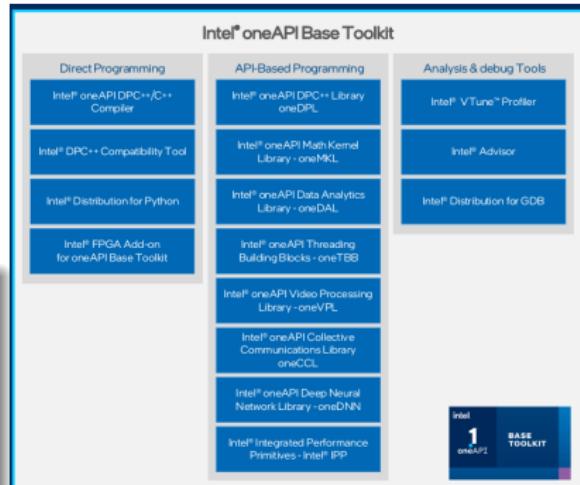


# oneAPI Base Toolkit

- Basic set of high-performance tools and libraries
- C++ compiler with SYCL support (heterogeneous computing)

## Features

- Data Parallel C++ Compiler
- Portability with SYCLomatic
- Python distribution (optimized libraries scikit-learn, NumPy)



# Updates in DPC++

- Improves support for accelerators with more features from SYCL2020 and OpenMP 5.0 and v5.1
- Enhances performance in CPU-GPU applications
  - SYCL/DPC++ offers equivalent performance to OpenMP on CPU
    - Example: HPCBench achieves 106% performance compared to OpenMP
    - Example: HPCBench achieves 138% performance compared to CUDA@A100

## What is DPC++?

- Compiler with SYCL support
- Based on standards such as LLVM



# Libraries

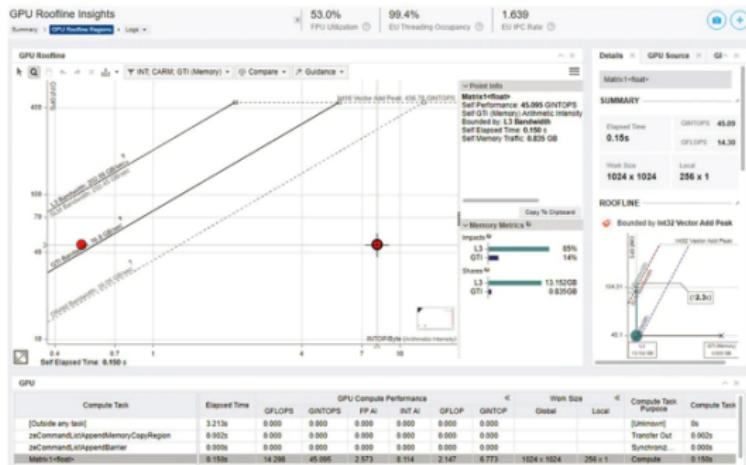
- Optimized libraries supported by compilers
  - oneDNN uses AMX, AVX-512, VNNI, and bfloat16 for acceleration in machine learning processes
- Optimized libraries
  - oneDNN supports AMX, AVX-512, VNNI, and bfloat16 to accelerate training in DL
  - oneVPL is compatible with AV1 hardware codec via Hardware (Intel® Data Center GPU Flex Series and Intel® Arc)
  - oneMKL improves portability and compatibility
    - OpenMP offload support and recent Intel® XMX repertoire support
    - Optimized operations in TF32, FP16, BF16, and INT8.
    - Interfaces for SYCL and C/Fortran OpenMP
    - Adds GPU synchronization support for BLAS



- Analysis tool
  - Supports C, C++, Fortran, SYCL, OpenMP, OpenCL, or Python code
- Facilitates efficient CPU code development
  - Monitors efficient threading, vectorization, and memory usage
- Possibility of efficient GPU offload suggestions
  - Identifies application parts that can be offloaded cost-effectively
  - Optimizes code for computation and memory.
- Flowchart design and analysis
  - Create, visualize, and analyze tasks and dependencies for heterogeneous algorithms



- Roofline model and performance on CPUs/GPUs
  - Practical advice for generating code on GPUs
  - Recommendations for CPU vs GPU usage (memory hierarchy)
  - User guide for GPUs



# Intel® VTune™ Profiler

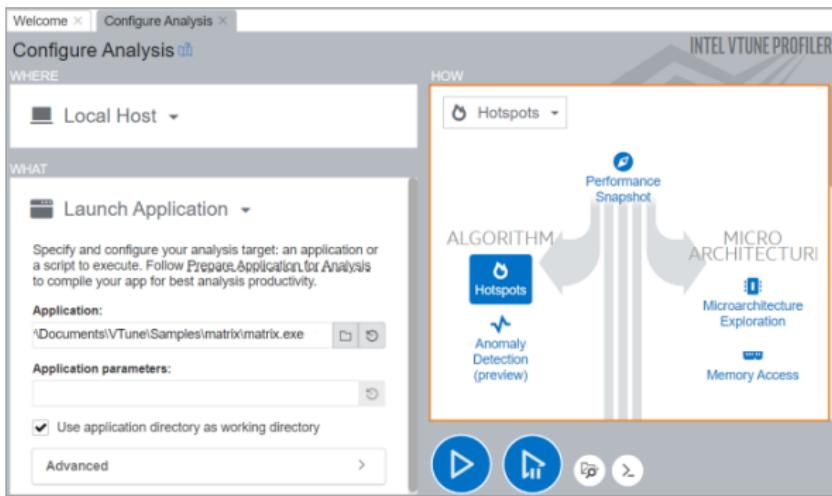
- Intel® VTune™ Profiler

- Tool for optimizing application performance, system performance, and system configuration for HPC, cloud, IoT, storage...
- CPU, GPU, and FPGA analysis
- Multi-language support: SYCL, C, C++, C#, Fortran, OpenCL, Python
- Tool to find out bottlenecks on several devices
- Several levels: system and/or application



# Intel® VTune™ Profiler

- Various supported analyses
  - Performance Snapshot or overall performance
  - HPC characterization and Parallelism
  - Accelerators: GPUs and CPU/FPGAs interaction
  - Microarchitecture and Memory
  - I/O Monitoring

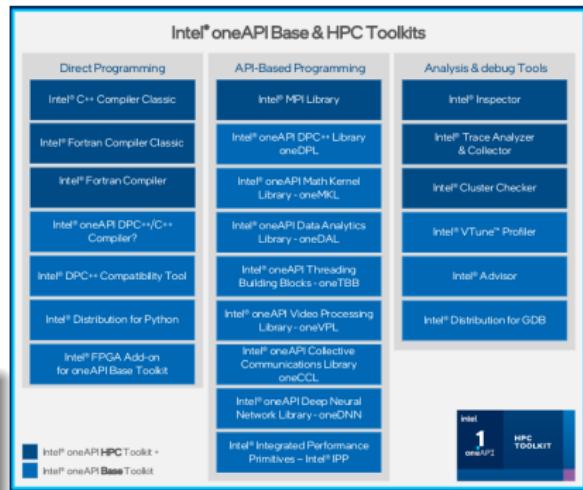


# oneAPI HPC Toolkit

- Toolkit complementing the Intel® oneAPI Base Toolkit
- Scalable and high-performance parallel code in C++, Fortran, SYCL, OpenMP, and MPI (from enterprise to the cloud)

## Features

- Acceleration and performance across the range of Intel Cores & Intel Accelerators
- Fast, scalable, and reliable parallel code with less effort based on industry standards



BASIC TOOLKIT

CGS

Intel® oneAPI, News and Advances 2024

Danysoft  
Software Elite Reseller

July 30, 2024

18 / 40



- Available [in the URL](#)



The logo features the Intel brand name in its signature blue font, with a small registered trademark symbol (®) to the right of the 'l'. Below the brand name, the words 'Developer Cloud' are written in a large, bold, sans-serif font.

- Multiple configurations catering to various workloads
  - From AI training and inference
  - ... prototyping and evaluating the latest hardware using the environment that best suits your business needs ... to FPGA applications
- Learn with practical tutorials
  - Experiment with real-world code examples
  - Evaluate performance and acceleration with multiple hardware configurations.
  - Create heterogeneous applications



# Available Hardware

- Test and evaluate a variety of virtual machines
  - Bare metal systems
  - Edge devices
  - Platforms for AI training
- Development environments
  - Containers
  - JupyterLabs
  - Direct SSH connection



# Access Instructions

- Documentation and updates available at  
<https://tinyurl.com/ReadmeIDC> or in the `Readme.md`

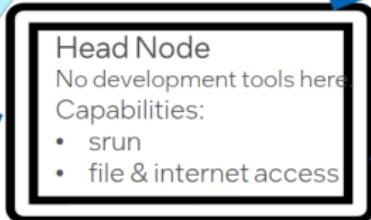
Picture it this way

nodes in queues  
pvc-shared  
and  
pvc

are identically configured  
same CPUs, same four PC cards  
(single tile PVCs – but four of them!)



`ssh`

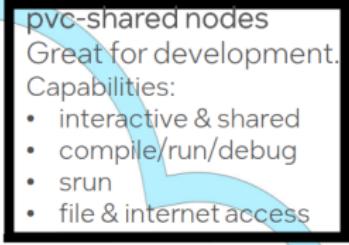


Intel  
Developer  
Cloud  
(IDC)

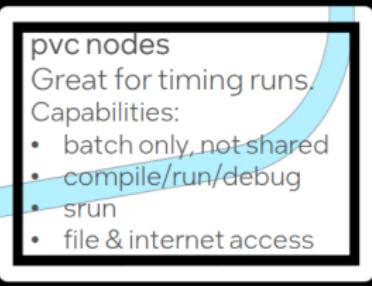
`srun`



`srun`



`srun`



# Register

- To have an account on **Intel® Developer Cloud** follow the link  
<http://cloud.intel.com>
- Follow the steps in the registration process:



# Register

- ① Select **usuario Standard**
- ② Create an account

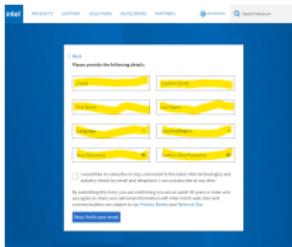
**Standard - Free**

Explore and evaluate the latest Intel® AI products plus:

- Develop AI skills.
- Access cutting edge learning resources.
- Get support from the Intel community.

[Subscribe](#)

- ③ Enter personal information



- ④ Verify email by sending an email with a **code**

Accept [terms and conditions of use](#) of the Intel® Developer Cloud

Danysort  
intel software elite reseller

# Access

- To access the account on Intel® Developer Cloud
  - Click at **Already a Member? Sign In** and enter your user/password
- Or follow the “shorcut” [https://console.cloud.intel.com/](https://console.cloud.intel.com)

The screenshot shows the Intel Developer Cloud Console Home page. At the top, there's a navigation bar with the Intel logo, the URL 'console.cloud.intel.com', and a dropdown for 'us-region-1'. On the left, a vertical sidebar contains icons for Home, Projects, API, Keys, and Help. The main content area has a dark background with several sections: 'Quick Start' (Hardware Catalog, Software Catalog, Training and Workshops, Cloud Credits), 'Learning and Support' (Getting started, Tutorials, What's new), and 'Notifications' (No notifications yet). Below these, there's a 'Gen AI Essentials' section featuring three cards: 'Text-to-Image with Stable Diffusion' (A Creative Playground for Artists, Writers, and Engineers, Launch button), 'Image-to-Image Generation with Stable Diffusion' (Perfect for artists and engineers who want to see their images transform in creative and unexpected ways), and 'Simple LLM Inference: Playing with Language Models' (A hands-on experience on language models and text generation, no technical background needed).

# Training

- JupyterLabs: in the menu **Training and Workshops**
  - Click at **LaunchJupyterLab**

The screenshot shows the 'Training and Workshops' section of the Intel Developer Cloud. On the left, there's a vertical sidebar with icons for AI, C++, SYCL, and oneAPI. The main area is titled 'Training and Workshops' and is divided into two sections: 'AI' and 'C++ SYCL'. In the 'AI' section, there are three cards: 'AI Kit XGBoost Predictive Modeling', 'Heterogeneous Programming Using Data Parallel Extension for Numba® for AI and HPC', and 'Machine Learning Using oneAPI'. Each card has a 'Launch' button. In the 'C++ SYCL' section, there are three cards: 'Essentials of SYCL', 'Performance, Portability and Productivity', and 'Introduction to GPU Optimization'. Each card also has a 'Launch' button. The top right of the page has a 'Launch JupyterLab' button.

Developer Cloud

us-region-1

## Training and Workshops

### AI

- AI Kit XGBoost Predictive Modeling
- Heterogeneous Programming Using Data Parallel Extension for Numba® for AI and HPC
- Machine Learning Using oneAPI

### C++ SYCL

- Essentials of SYCL
- Performance, Portability and Productivity
- Introduction to GPU Optimization

Launch JupyterLab



# Training

- SSH: You can connect to the IDC via SSH by previously adding the *Account Keys*

The screenshot shows the 'Upload key' page of the Intel Developer Cloud. The URL in the browser is `console.cloud.intel.com/security/publickeys/import`. On the left, there's a vertical sidebar with icons for Home, Projects, Key Management (which is selected and highlighted in blue), and Help. The main content area has a header 'Upload key' with a warning message: '⚠ Warning Never share your private keys with anyone. Never create a SSH Private key without a passphrase'. Below this is a section titled 'SSH key details' with a 'Key Name:' field containing 'Key Name'. A link 'How to create a SSH key' is provided. Under 'Key contents', there's a text area for pasting key contents, with a note 'Paste your key contents: \*'. At the bottom are 'Upload' and 'Cancel' buttons.



# SSH config

- Modify the file *.ssh/config*

```
Host myidc #+YOU CAN CALL IT ANYTHING
Hostname idcbetabatch.eqlb.intel.com
User uXXXXXX #+ Request "scheduled access" at https://scheduler.cloud.intel.com
    /#/systems" to get your user identifier.
#ProxyCommand /usr/bin/nc -x YourProxy:XXXX %h %p # Uncomment if necessary
ServerAliveInterval 60
ServerAliveCountMax 10
StrictHostKeyChecking no # Frequent changes in the setup are taking place now,
    this will help reduce the known hosts errors.
UserKnownHostsFile=/dev/null
```



# ssh connection

- On the *front-end* or *head* node, it is Ubuntu 22.04LTS.
- Use the node in interactive mode: `srun --pty bash`
  - Node with Intel Data Center GPU Max 1100

```
user@localhost:~$ ssh myidc
uXXXX@idc-beta-batch-head-node:~$ srun --pty bash
uXXXX@idc-beta-batch-pvc-node-03:~$ source /opt/intel/oneapi/setvars.sh
```

```
:: initializing oneAPI environment ...
bash: BASH_VERSION = 5.1.16(1)-release
args: Using "$@" for setvars.sh arguments:
```

```
uXXXX@idc-beta-batch-pvc-node-03:~$ sycl-ls
Warning: ONEAPI_DEVICE_SELECTOR environment variable is set to opencl:cpu;opencl
:fpga;level_zero:3.
To see the correct device id, please unset ONEAPI_DEVICE_SELECTOR.
```

```
[opencl:cpu:0] Intel(R) OpenCL, Intel(R) Xeon(R) Platinum 8480L 3.0
[2023.16.7.0.21_160000]
[opencl:acc:1] Intel(R) FPGA Emulation Platform for OpenCL(TM), Intel(R) FPGA
Emulation Device 1.2 [2023.16.7.0.21_160000]
[opencl:cpu:2] Intel(R) OpenCL, Intel(R) Xeon(R) Platinum 8480L 3.0
[2023.16.7.0.21_160000]
```



# Slurm Commands

- *sinfo*: get information about available nodes
- *squeue*: view queued jobs
- *sbatch -p {PARTITION-NAME} {SCRIPT-NAME}*: enqueue a job in the queues
- *scancel {JOB-ID}*: cancel a job
- *srun –pty bash*: launch a job interactively



# Example Job Launch (I)

- ① Let's take an example SYCL code that returns the selected device, in this case, a GPU.

```
#include <sycl/sycl.hpp>
using namespace sycl;
int main() {
    // Create a device queue with device selector
    queue q(gpu_selector_v);
    // Print the device name
    std::cout << "Device: " << q.get_device().get_info<info::device::name>() << "\n";
    return 0;
}
```

- ② Compile it with the **icpx** compiler

```
uXXXX@idc-beta-batch-pvc-node-03:~$ icpx -o ex exampleSYCL.cpp -fsycl
```



# Example Job Launch (II)

## ③ Launch the job using Slurm

```
#!/bin/bash
#SBATCH --job-name=gpu_run
#SBATCH --partition=pvc-shared
#SBATCH --error=job.%J.err
#SBATCH --output=job.%J.out
#SBATCH --mail-type=ALL
#SBATCH --mail-user=your@email.com

icpx -o ex exampleSYCL.cpp -fsycl
./ex
```

## ④ Submit the job and view the output

```
uXXXX@idc-beta-batch-head-node:~$ sbatch job.sh
uXXXX@idc-beta-batch-head-node:~$ more job.42151.out
Device: Intel(R) Data Center GPU Max 1100
```



# Launch Interactive JupyterLab (I)

- ① Connect via SSH to the *head-node*: `ssh myidc`
- ② Start an interactive session: `srun --pty bash`
- ③ Activate the conda environment to launch JupyterLab: `conda activate pytorch_xpu`
- ④ Launch JupyterLab
  - Important: Find the listening address for JupyterLab, something like  
`http://10.10.10.X:8888/lab?token=9d83e1d8a0eb3ffed84fa3428aae01e592cab170a4119130`

```
user@localhost:~$ ssh myidc
uXXXX@idc-beta-batch-head-node:~$ srun --pty bash
uXXXX@idc-beta-batch-pvc-node-03:~$ source /opt/intel/oneapi/setvars.sh
uXXXX@idc-beta-batch-pvc-node-03:~$ conda activate pytorch-gpu
(pytorch-gpu) uXXXX@idc-beta-batch-pvc-node-04:~$ jupyter-lab --ip $(hostname -i
)
...
[I 2023-11-11 15:01:56.875 ServerApp] http://10.10.10.8:8888/lab?token=
f25e4d3977b7b059d5ff4b376085562dbbaa3f9dffed4fde
```



# Launch Interactive JupyterLab (II)

- ⑤ From another **local terminal** create an SSH tunnel to the address displayed in the launched console: **10.10.10.X**

```
user@localhost:~$ ssh myidc -L 8888:10.10.10.8:8888
```

- ⑥ Connect with a browser to the local address: **https://localhost:8888** enter the **token** from the example; in this case,  
*f25e4d3977b7b059d5ff4b376085562dbbaa3f9dffed4fde*



# IDC Instances

- Core compute
  - Based on Xeon 4th gen processor
    - VMs with 8, 16, 32 cores
    - Bare Metal 112 cores, 256GB, and 2TB disk
- Intel Max GPU
  - 4xGPUs 1100 + 2xsockets Xeon 4th gen
- Gaudi Deep Learning Server
  - 8x Gaudi HL + Xeon Platinum 3rd gen

The screenshot shows the Intel Developer Cloud Hardware Catalog interface. The left sidebar has icons for Home, Catalog, GPU, Bare Metal, Processor, and AI. The main area is titled "Hardware Catalog" and "Available platforms".  
**Core compute:** Shows "4th Generation Intel® Xeon® Scalable processors" with options to "Select" or "Released".  
**GPU:** Shows "Intel® Max Series GPU" with options to "Select" or "Released".  
**AI:** Shows "Gaudi™ Deep Learning Server" with options to "Select" or "Released".  
Filters on the left include "Filter by: Released", "Category: Released", "Type: Virtual Machine, Bare Metal", "Processor: CPU, GPU, AI processors", and "RECOMMENDED USE CASE". Logos for Intel oneAPI and Danysoft are visible at the bottom.

# Launch Instance

- 1 Launch an instance in the [console->Hardware Catalog](#)
- 2 Get from the [reservations section](#), for example, through an SSH connection.

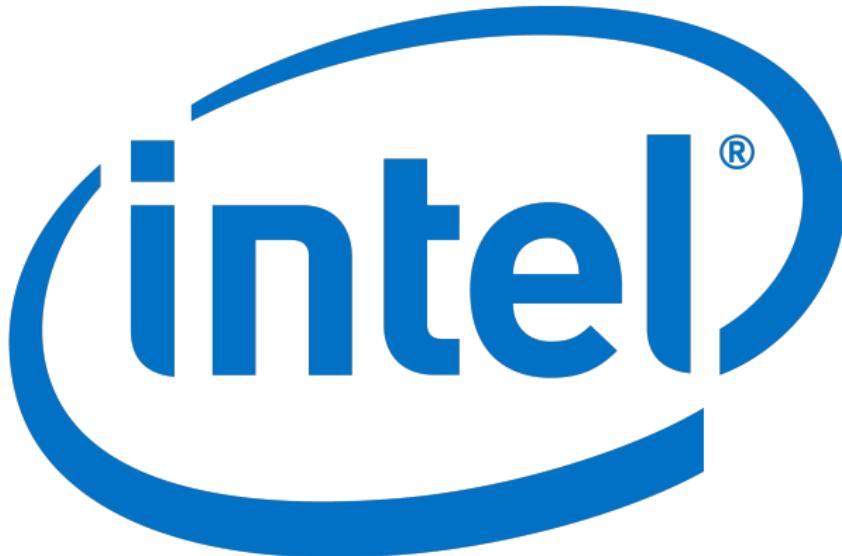
The screenshot shows the Intel Compute Console interface. On the left, there's a sidebar with icons for Home, API, Instances, and Support. The main area is titled 'Compute Console' and shows 'My Instances'. A search bar is at the top of the instances list. Below it, there's a table with columns for 'Instance Name' (4gen) and 'Ip' (100.82.61.22). To the right of the table, a modal window titled 'How to connect to your instance' is open. It has a heading 'Select your OS:' with radio buttons for Windows (selected), Linux, and macOS. Below this, instructions for using an SSH client are provided: '1. Open an SSH client', '2. Locate your public key file (my-key.ssh). The wizard automatically detects the key you used to launch the instance.', and '3. Your key must not be publicly visible for SSH to work. Use this command:'. A command line input field contains 'ssh -o StrictHostKeyChecking=no -i my-key.ssh guest@146.152.232.8' with a 'Copy' button next to it. Another section titled '4. SSH command to connect to instance' also contains the same command with a 'Copy' button. At the bottom of the modal, a note says 'Please note that in most cases the username above will be correct, however please ensure that you read your instance usage instructions to ensure that the instance owner has not changed the default instance username.' The bottom right corner of the modal has a 'Close' button.



# Available Resources

- Initiative [oneAPI](#)
- Intel oneAPI Base & HPC Toolkit
- Instructions for the [Intel Developer Cloud access](#)





# Software



# Thanks for your attention!!!



Avda. de la industria 4, edif. 1  
28108 Alcobendas | Madrid | España



[+34] 91 663 8683



Email  
[info@danysoft.com](mailto:info@danysoft.com)



Website  
[www.danysoft.com/intel](http://www.danysoft.com/intel)

