



Урок 12. Kubernetes и работа с данными. Использование JupyterHub в Kubernetes для тестирования oneAPI от Intel

Дмитрий Сивков
Инженер-консультант
Intel Russia

Александр Волынский
Technical Product Manager
ML Platform, VK Cloud Solutions

Александр Волынский



Technical Product Manager
ML Platform, VK Cloud Solutions

- Архитектор VK Cloud Solutions
- Специалист по Big Data
- Участвовал в создании хранилищ данных в «Платформе ОФД», Mail.ru Group, X5 Group и других компаниях
- Энтузиаст использования Kubernetes для построения Data Lake, DWH и ML-платформ в облаках

Дмитрий Сивков



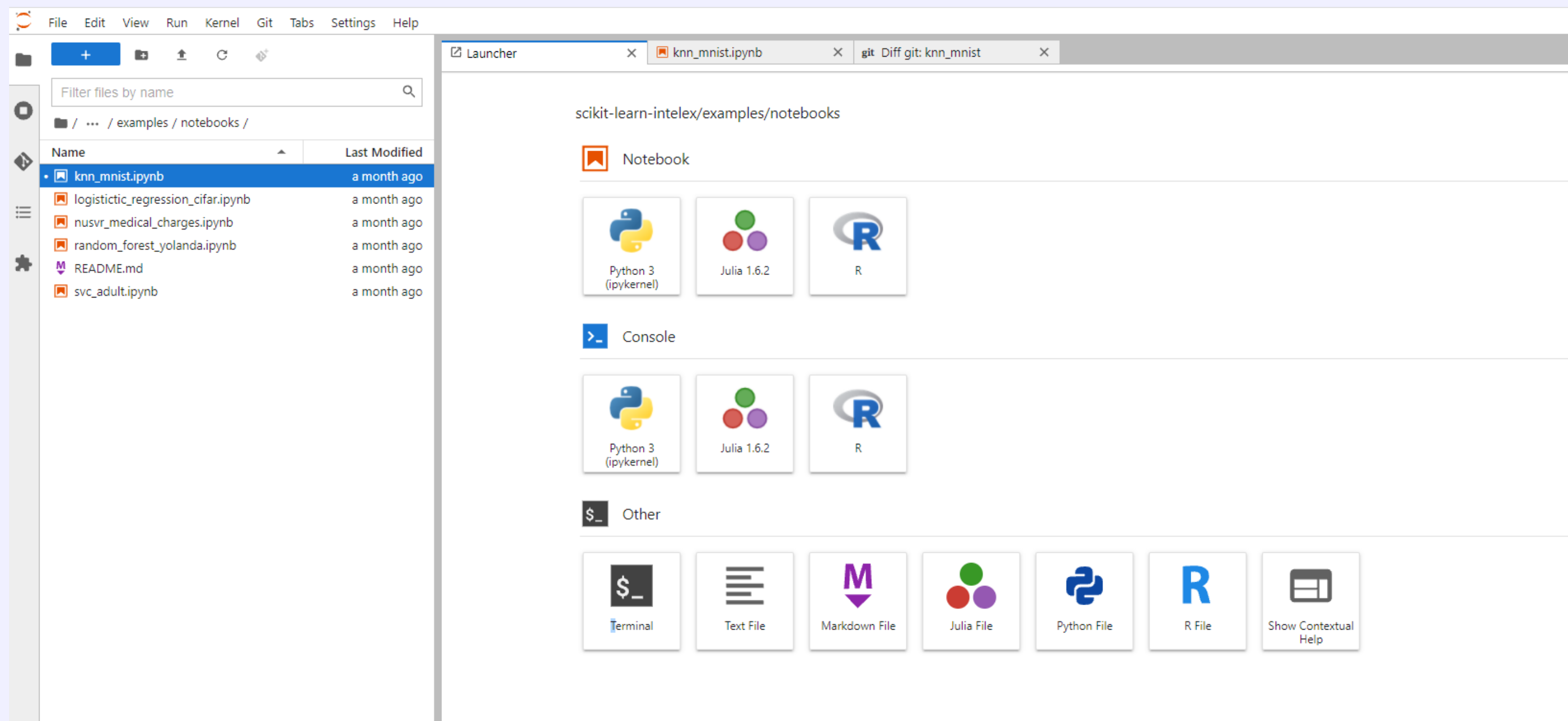
Инженер-консультант
Intel Russia

- Помогает партнёрам раскрывать возможности платформ Intel и эффективно их использовать
- Десятки успешных проектов по всему миру
- Доклады на многих конференциях, симпозиумах и митапах
- Практические занятия
- Просто хороший человек

Jupyter и JupyterHub

- Jupyter – open source интерактивная web среда для разработки, проведения экспериментов, построения визуализаций
- Используется аналитиками, data science и data engineering командами
- Поддерживает различные языки, включая Python, Scala, R, Julia
- Есть поддержка расширений, можно настроить под различные задачи
- JupyterHub – multi-user версия Jupyter, решающая задачи аутентификации, предоставления индивидуальных окружений, масштабирования

Jupyter и JupyterHub: интерфейс



Jupyter и JupyterHub: возможности настройки UI

The screenshot displays the JupyterLab environment. On the left is a file browser showing a directory structure with files like `knn_mnist.ipynb`, `logistic_regression.ipynb`, `nusvr_medical_charges.ipynb`, `random_forest.ipynb`, `README.md`, and `svc_adult.ipynb`. The main area is a code editor for `knn_mnist.ipynb` running on a Python 3 (ipykernel) environment. The code includes imports for `sklearn` and `sklearnex`, data fetching, splitting, and KNN training/prediction. A message indicates that the Intel(R) Extension for Scikit-learn is enabled. The terminal on the right shows the user `jovyan` at the `jupyter-stockblog` environment.

```

[1]: from sklearn import metrics
    from sklearn.model_selection import train_test_split

[2]: from sklearn.datasets import fetch_openml
    x, y = fetch_openml(name='mnist_784', return_X_y=True)

[3]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=72)

Intel Extension for Scikit-learn (previously known as daal4py) contains drop-in replacement functionality for the stock scikit-learn package. You can take advantage of the performance optimizations of Intel Extension for Scikit-learn by adding just two lines of code before the usual scikit-learn imports:

[4]: from sklearnex import patch_sklearn
    patch_sklearn()

Intel(R) Extension for Scikit-learn* enabled (https://github.com/intel/scikit-learn-intelx)

Intel(R) Extension for Scikit-learn patching affects performance of specific Scikit-learn functionality. Refer to the list of supported algorithms and parameters for details. In cases when unsupported parameters are used, the package fallbacks into original Scikit-learn. If the patching does not cover your scenarios, submit an issue on GitHub.

[6]: params = {
    'n_neighbors': 40,
    'weights': 'distance',
    'n_jobs': -1
}

Training and predict KNN algorithm with Intel(R) Extension for Scikit-learn for MNIST dataset

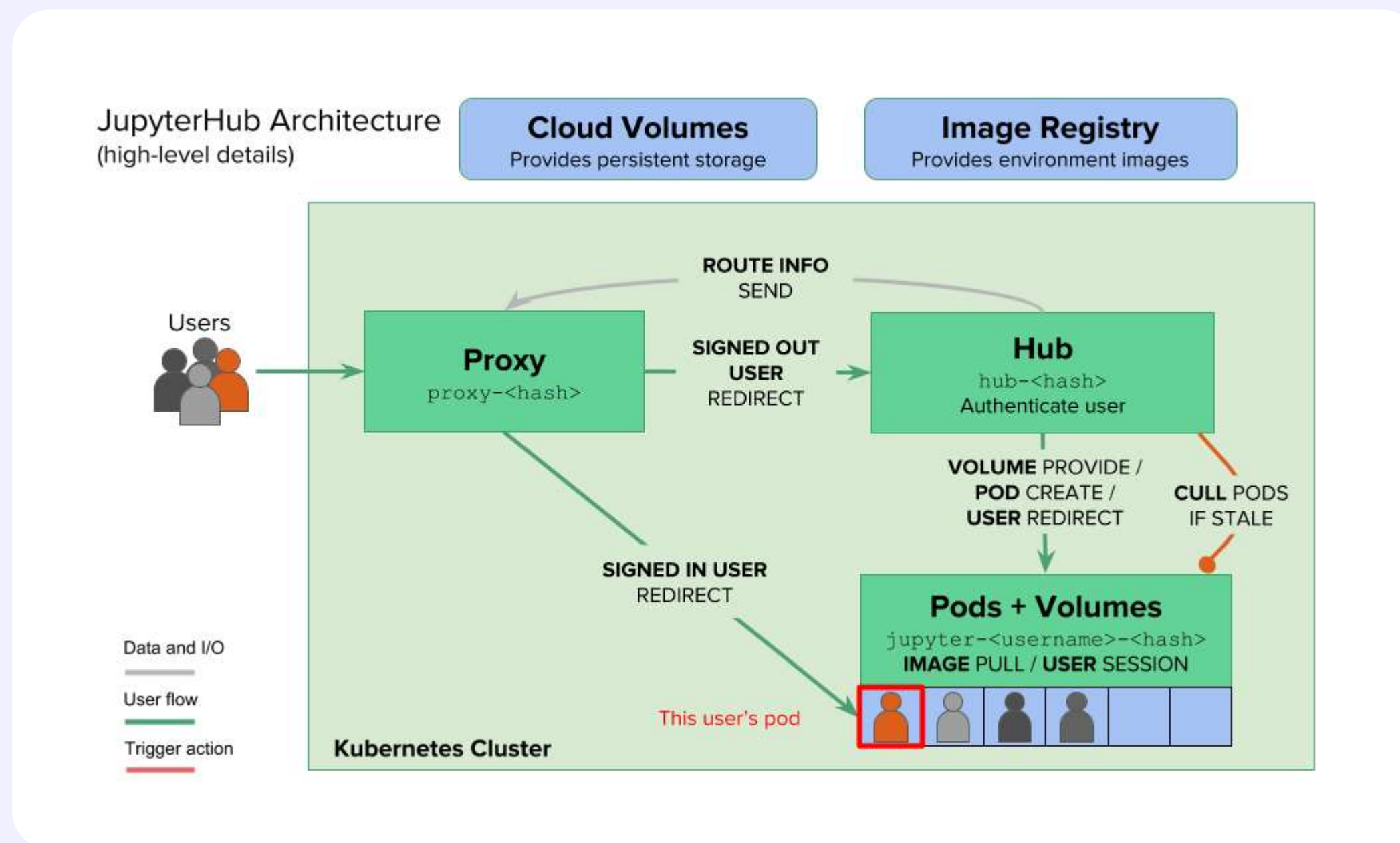
[7]: start = time()
    from sklearn.neighbors import KNeighborsClassifier
    knn = KNeighborsClassifier(**params).fit(x_train, y_train)
    predicted = knn.predict(x_test)
    f"Intel(R) extension for Scikit-learn time: {(time() - start):.2f} s"

[7]: 'Intel(R) extension for Scikit-learn time: 11.38 s'
  
```

JupyterHub и Kubernetes

- Решает проблему с масштабированием и предоставлением ресурсов пользователям
- Интегрируется с системой автомасштабирования кластеров Kubernetes в облаке
- Позволяет создавать преднастроенные окружения под различные типы задач на основе Docker образов
- Есть Helm chart и подробная инструкция по установке
<https://zero-to-jupyterhub.readthedocs.io/>

Архитектура JupyterHub в Kubernetes



JupyterHub в Kubernetes и безопасность

- По умолчанию устанавливается с DummyAuthenticator и без https
- Интеграция с Let's Encrypt в несколько строк
- Возможность подключения своих сертификатов
- Аутентификация пользователей на базе OAuth2 с поддержкой GitHub, Google и других OAuth2 identity provider
- Также поддерживаются LDAP и AD, OpenID Connect, KeyCloak
- Есть возможность ограничить доступность сервиса только для определенных IP

<https://zero-to-jupyterhub.readthedocs.io/en/latest/administrator/authentication.html>

Пример настройки https

```

proxy:
  https:
    enabled: true
    hosts:
      - your-domain-name.com
    letsencrypt:
      contactEmail: YOUR_EMAIL
  service:
    loadBalancerIP: PLACE_EXTERNAL_IP_OF_LOADBALANCER
    loadBalancerSourceRanges:
      - PLACE_YOUR_IP
  
```

Пример настройки аутентификации с использованием GitHub

```
hub:
  config:
    Authenticator:
      admin_users:
        - YOUR_GITHUB_LOGIN
    GitHubOAuthenticator:
      client_id: YOUR_CLIENT_ID_GITHUB
      client_secret: YOUR_SECRET_FROM_GITHUB
      oauth_callback_url: https://your-domain-name.com/hub/oauth_callback
      allowed_organizations:
        - YOUR_ORG_NAME_FROM_GITHUB
      scope:
        - read:org
    JupyterHub:
      authenticator_class: github
```

JupyterHub в Kubernetes и масштабирование

- Включаем continuous image puller, чтобы заранее подтянуть образы окружений
- Включаем user scheduler – пользователи будут сгруппированы по нодам, что позволит масштабироваться в меньшую сторону при падении нагрузки
- Выделяем в облаке одну или несколько node pools с нужными taint. На этих нодах будут жить только пользователи
- Настраиваем requests и limits для ноутбуков пользователей. Рекомендуется начать с пропорции 1 к 2
- Включаем функцию приостановки работы неактивных ноутбуков пользователей

<https://zero-to-jupyterhub.readthedocs.io/en/latest/jupyterhub/customizing/user-resources.html>

<https://zero-to-jupyterhub.readthedocs.io/en/latest/administrator/optimization.html>

JupyterHub в Kubernetes и индивидуальные окружения

- Создаем разные Docker образы под различные задачи на основе базовых образов Jupyter
- Включаем возможность выбора пользователем окружения с индивидуальными настройками
- Задаем переменные окружения для пользователей
- Настраиваем создание Persistent Volumes нужного типа и объема

<https://zero-to-jupyterhub.readthedocs.io/en/latest/jupyterhub/customizing/user-environment.html>

JupyterHub в Kubernetes и индивидуальные окружения

Server Options

<input checked="" type="radio"/>	Minimal environment To avoid too much bells and whistles: Python.
<input type="radio"/>	Tensorflow If you want the additional bells and whistles: Python, R, and Julia.
<input type="radio"/>	Spark environment The Jupyter Stacks spark image!
<input type="radio"/>	JupyterLab with git The Jupyter with git
<input type="radio"/>	JupyterLab with Intel libraries Use some Intel optimizations
<input type="radio"/>	Learning Data Science Datascience Environment with Sample Notebooks

Start



oneAPI: Industry Initiative & Intel Products

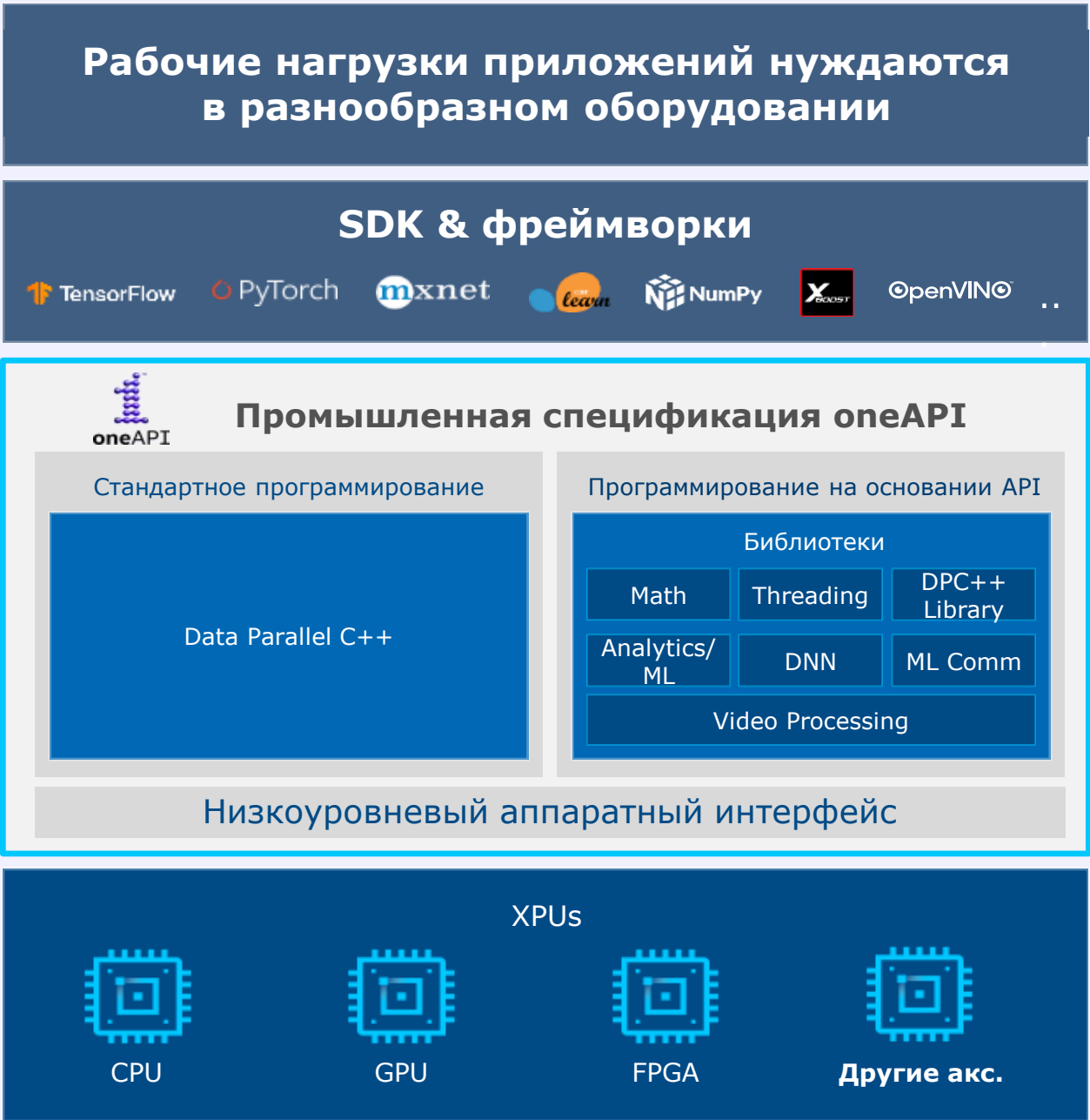
One Intel Software & Architecture group
Intel Architecture, Graphics & Software
November 2021



Отраслевая инициатива oneAPI

Разорвите привязку к вендору

- Кросс-архитектурный язык, основанный на стандартах C++ и SYCL
- Эффективные библиотеки, предназначенные для ускорения предметно-ориентированных функций
- Уровень абстракции низкоуровневого аппаратного обеспечения
- **Открытый стандарт для использования сообществом и промышленностью**
- **Обеспечивает возможность адаптации кода для различных архитектур и вендоров**



Продуктивный и умный путь к освобождению ускоренных вычислений от экономического и технического бремени проприетарных моделей программирования. Посетите сайт oneapi.com, чтобы узнать больше деталей.

Экосистема oneAPI-2020



Перечисленные организации поддерживают концепцию инициативы OneAPI для единой, унифицированной кросс-архитектурной модели программирования. Это не подразумевает соглашения о покупке или использовании продукции Intel. *Прочие названия и бренды могут являться собственностью других компаний.

Экосистема oneAPI-2021



These organizations support the oneAPI initiative 'concept' for a single, unified programming model for cross-architecture development. It does not indicate any agreement to purchase or use of Intel's products. *Other names and brands may be claimed as the property of others.

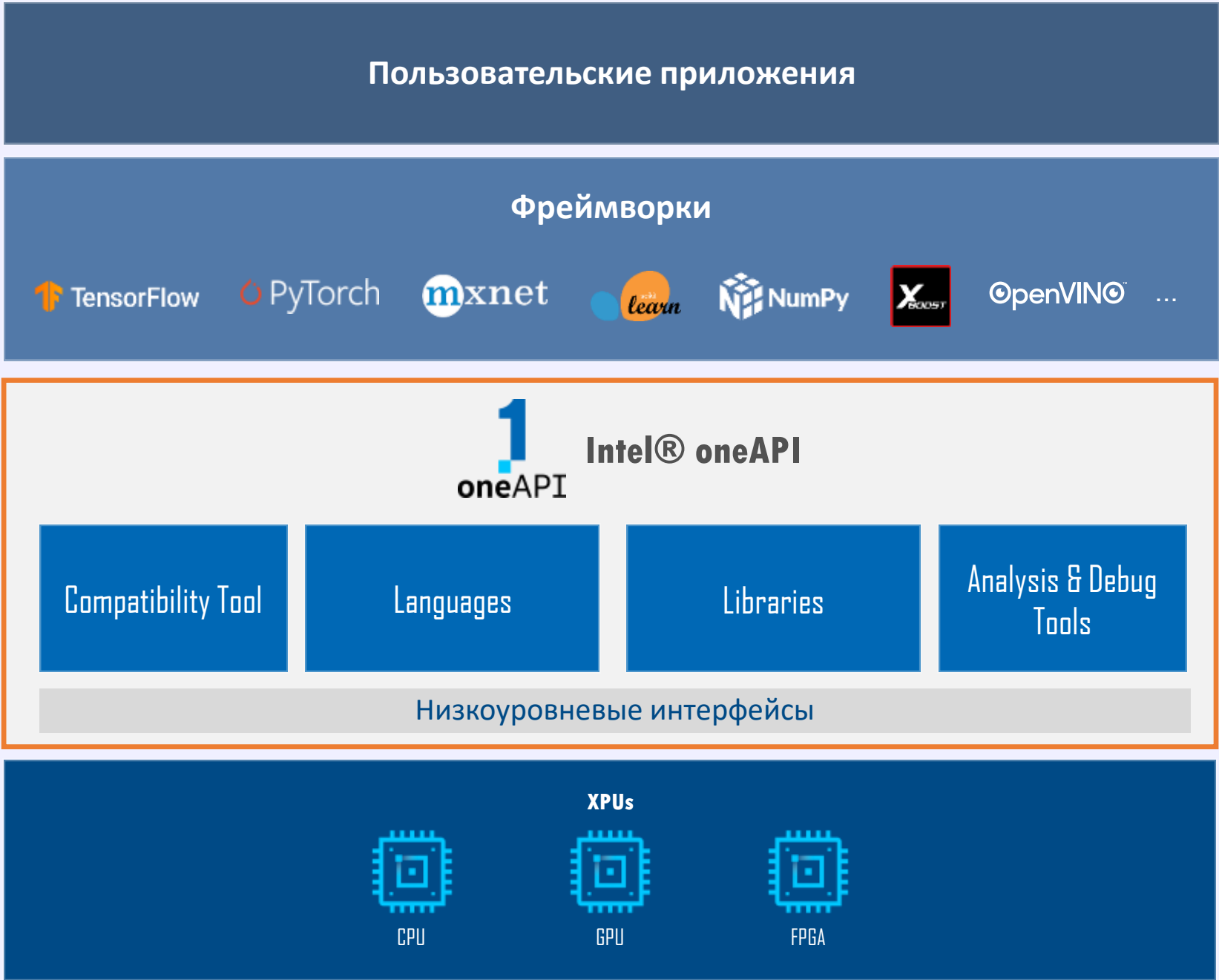
Intel® oneAPI как продукт

построен на зарекомендовавших себя инструментах Intel для CPU

Полный набор компиляторов, библиотек, инструментов портирования, анализа и отладки

- Ускорение вычислений новейшими аппаратными решениями
- Совместимость с существующими программными моделями (C++, Fortran, Python, OpenMP, etc.)
- Упрощает переход на новые системы и ускорители

[Загрузить](#)



Visit software.intel.com/oneapi for more details
Some capabilities may differ per architecture and custom-tuning will still be required.
Other accelerators to be supported in the future.

Intel® oneAPI AI Analytics

Ускорение процессов AI и анализа данных на платформах Intel

Преимущества

- Ускорение глубокого обучения с оптимизированными Intel фреймворками
- Прозрачное ускорение анализа данных и машинного обучения в Python



Intel® oneAPI AI Analytics Toolkit

Deep Learning	Data Analytics & Machine Learning		
Intel® Extension for TensorFlow	Accelerated Data Frames		
Intel® Extension for PyTorch	Intel® Distribution of Modin	OmniSci Backend	
Intel® Neural Compressor	Intel® Distribution for Python		
Model Zoo for Intel® Architecture	XGBoost	Scikit-learn	Data Parallel Python
	NumPy	SciPy	Pandas

Samples & End2End Workloads



CPU



GPU

Hardware support varies by individual tool. Architecture support will be expanded over time.

Get the Toolkit [HERE](#) or via these locations

[Intel Installer](#)

[Docker](#)

[Apt, Yum](#)

[Conda](#)

[Intel® DevCloud](#)

Learn More: software.intel.com/oneapi/ai-kit

Оптимизированный Intel scikit-learn

```
from sklearn.svm import SVC
X, Y = get_dataset()

clf = SVC().fit(X, y)
res = clf.predict(X)
```

Вызовы Scikit-learn

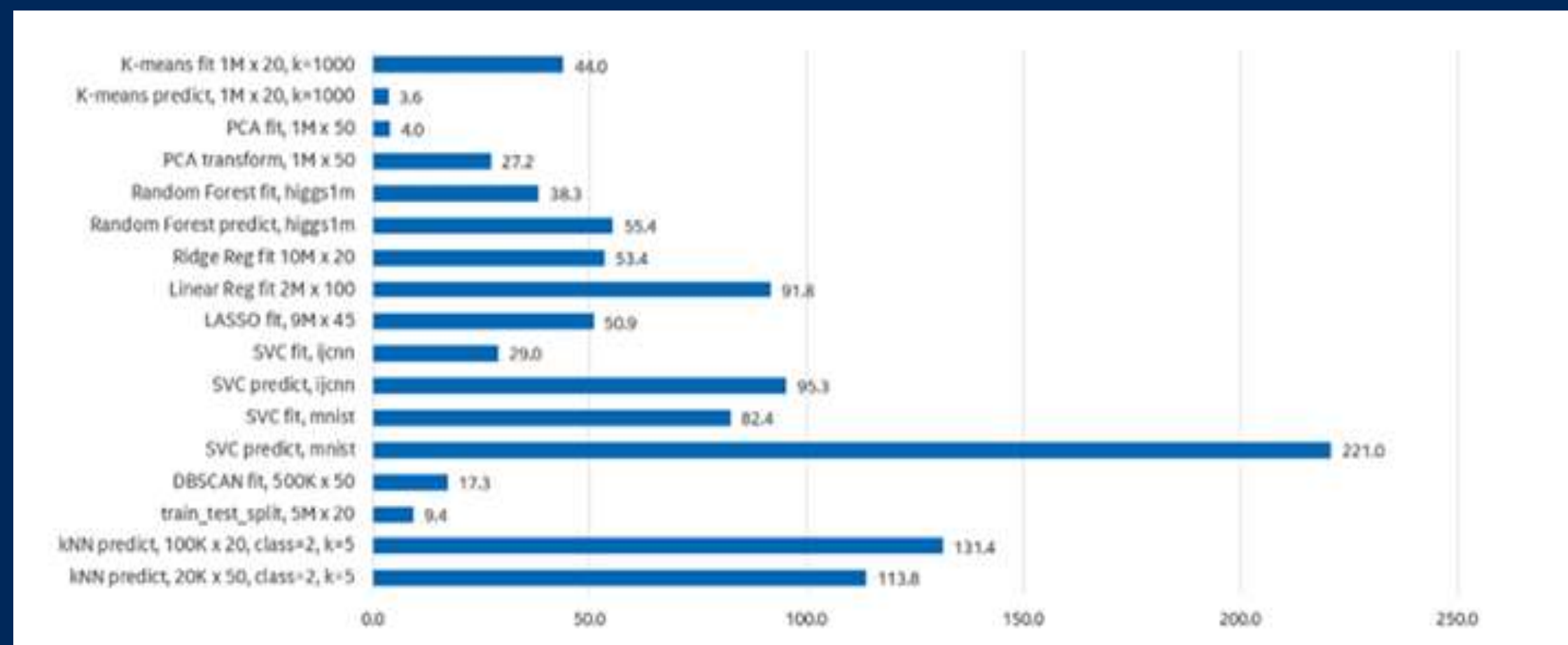
```
import daal4py as d4p
d4p.patch_sklearn()

from sklearn.svm import SVC
X, Y = get_dataset()

clf = SVC().fit(X, y)
res = clf.predict(X)
```

Оптимизированный Scikit-learn для Intel CPU

Обычный scikit-learn vs Intel-optimized scikit-learn



Один код — одни результаты

Именно *Scikit-learn*, а не как *scikit-learn*

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.
See backup for configuration details.

Notices & Disclaimers

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



Репозиторий с инструкцией и необходимыми шаблонами

https://github.com/stockblog/jupyterhub_k8s_mcs_slurm_intel

СЛЕПМ при поддержке



VK Cloud Solutions



intel.



Спасибо!



SRE: внедряем DevOps от Google

Четвёртый интенсив для команд, внедряющих SRE. Вы будете разбирать горящие проблемы своими руками и решать практические кейсы на боевом приложении.

3 — 5 декабря 2021

