ALASKA INBRE

IDeA Network of Biomedical Research Excellence

DATA SCIENCE CORE ALASKA INBRE

Nanopore Sequencing Workshop

Data Science Core

Alaska INBRE NIH IDeA (P20GM103395)

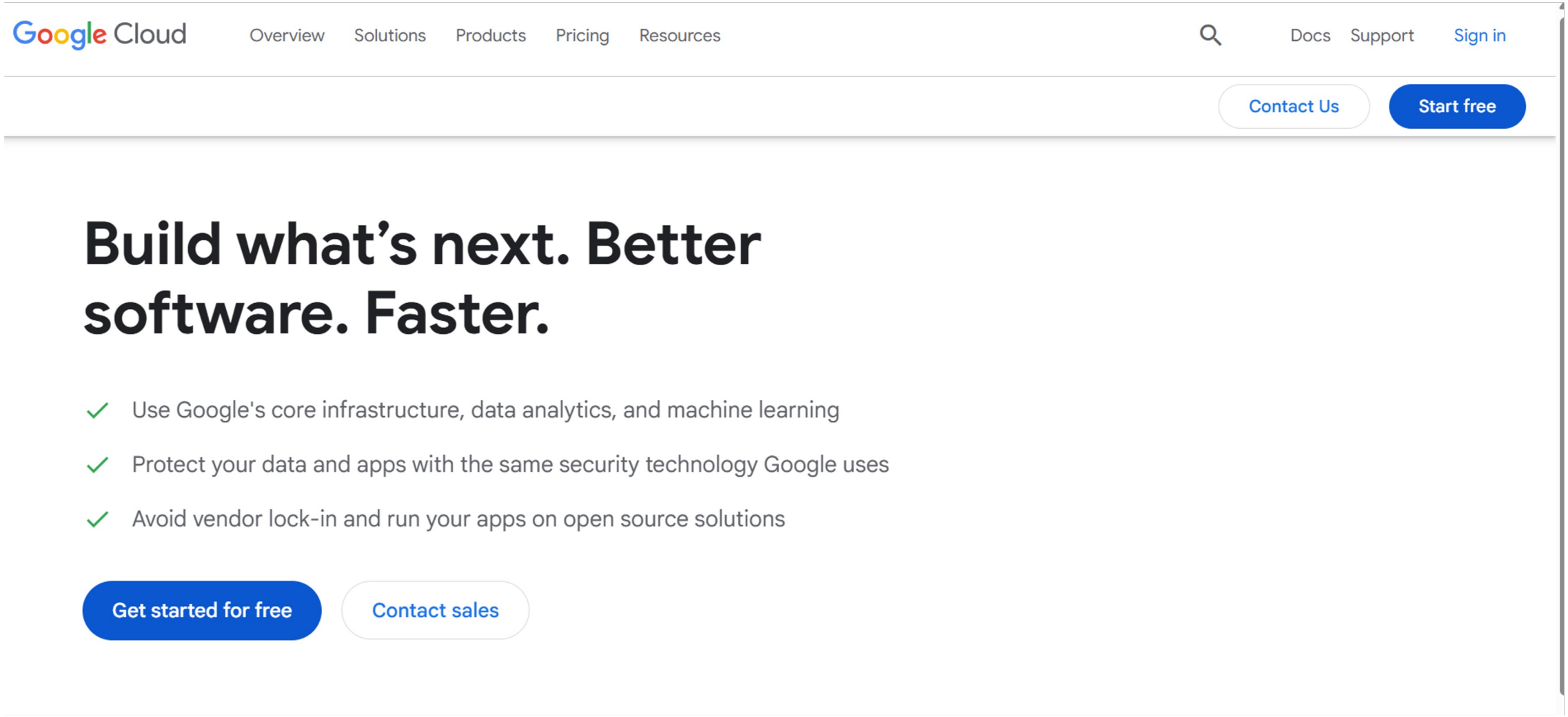
UAF UNIVERSITY OF ALASKA FAIRBANKS

Day 2 Agenda

Activity	Type	Start	End
Cloud Computing and HPC	Bioinformatics	Tue 9:00 AM	Tue 10:15 AM
Brain Break	Break	Tue 10:15 AM	Tue 10:30 AM
Introduction to the Command Line	Bioinformatics	Tue 10:30 AM	Tue 12:00 PM
Lunch	Meal	Tue 12:00 PM	Tue 1:00 PM
Google Colab	Bioinformatics	Tue 1:00 PM	Tue 2:00 PM
Nanopore Bioinformatics	Bioinformatics	Tue 2:00 PM	Tue 3:30 PM
Daily Wrap-up	Discussion	Tue 3:30 PM	Tue 4:00 PM

Cloud Computing and HPCs

Introduction to the Google Cloud Platform



What is Cloud Computing?

- On-demand availability of computing resources (such as storage and infrastructure), as services over the internet.

- Eliminates the need self-manage physical resources and only pay for what you use.

What are the types of cloud computing services?

Infrastructure as a service (IaaS) offers on-demand access to IT infrastructure services, including compute, storage, networking, and virtualization. It provides the highest level of control over your IT resources and most closely resembles traditional on-premises IT resources.

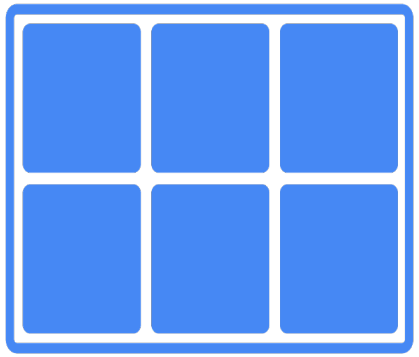
Platform as a service (PaaS) offers all the hardware and software resources needed for cloud application development. With PaaS, companies can focus fully on application development without the burden of managing and maintaining the underlying infrastructure.

Software as a service (SaaS) delivers a full application stack as a service, from underlying infrastructure to maintenance and updates to the app software itself. A SaaS solution is often an end-user application, where both the service and the infrastructure is managed and maintained by the cloud service provider.

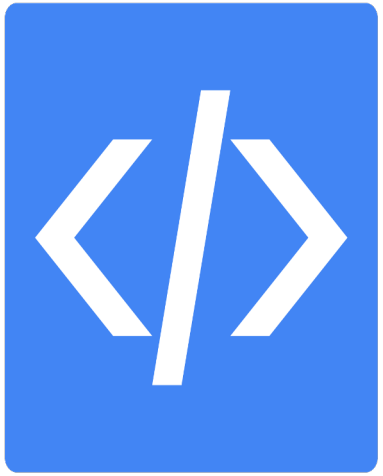
Computing

Virtual machines

Containers



Serverless



Virtual Machines

Compute Engine is used to launch virtual machines on demand. It includes high performance and general-purpose virtual machines offering a good balance of price and performance. Compute Engine VMs are suitable for a wide variety of common workloads including databases, development and testing environments, web applications, and mobile gaming.

Flexible Machine Types: Offers various machine types (general-purpose, memory-optimized, compute-optimized, accelerator-optimized with GPUs/TPUs) to suit different performance and cost requirements.

Networking and Storage: You configure networking (VPC, firewalls) and attach persistent disks or local SSDs.

Cost Management: You pay for the VMs based on usage, with options for sustained-use discounts, committed-use discounts, and Spot VMs for fault-tolerant workloads.

Integration: Can be integrated with other Google Cloud services, including data analytics, storage, and networking.

Iaas vs Paas Instances

Compute Engine VM Instances

Infrastructure as a Service (IaaS): Compute Engine provides raw virtual machines (VMs) that give you complete control over the operating system, software, and configuration. You're responsible for everything from installing libraries to managing updates and scaling.

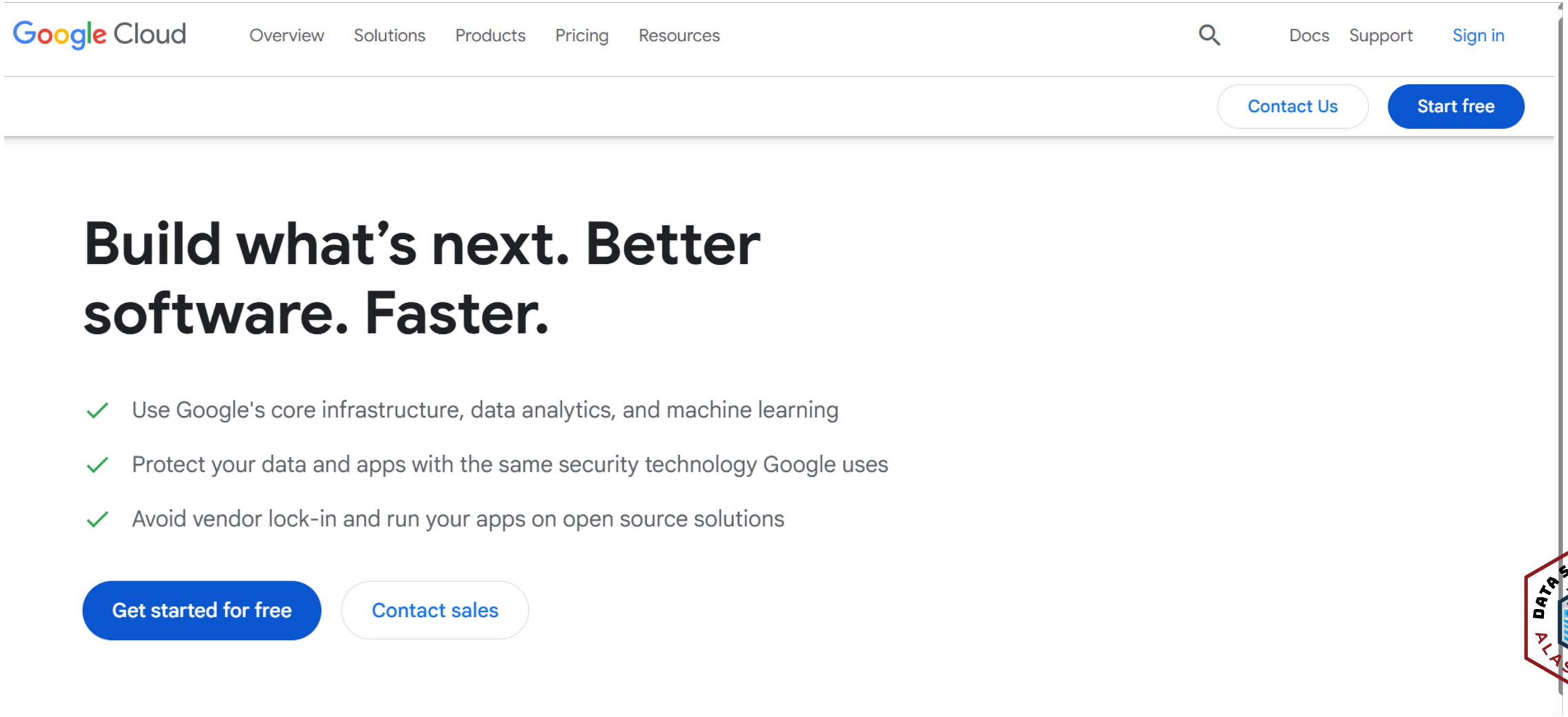
General Purpose: It's designed for a wide range of workloads that require customizability and fine-grained control over the compute environment.

Vertex AI Workbench instances

Platform as a Service (PaaS) for ML: It abstracts away many of the complexities of setting up and managing an ML development environment.

Managed Machine Learning Environment: Vertex AI Workbench instances are fully managed, Jupyter-notebook-based environments specifically designed for machine learning (ML) development. They are built on top of Compute Engine VMs, but Google manages much of the underlying infrastructure.

https://console.cloud.google.com/



On Google Cloud, via the Console

- Create a new Project
- Create a new Vertex AI VM Instance
- Turn on the Instance
- Connect with the Instance via
  - Jupyterlab Interface
  - SSH

On Google Cloud, via the Console

Google Cloud projects form the basis for creating, enabling, and using all Google Cloud services including managing APIs, enabling billing, adding and removing collaborators, and managing permissions for Google Cloud resources.

Project name: A human-readable name for your project.
The project name isn't used by any Google APIs. You can edit the project name at any time during or after project creation. Project names do not need to be unique.

Project ID: A globally unique identifier for your project.
A project ID is a unique string used to differentiate your project from all others in Google Cloud. After you enter a project name, the Google Cloud console generates a unique project ID that can be a combination of letters, numbers, and hyphens. We recommend you use the generated project ID, but you can edit it during project creation. After the project has been created, the project ID is permanent.

Project number: An automatically generated unique identifier for your project.

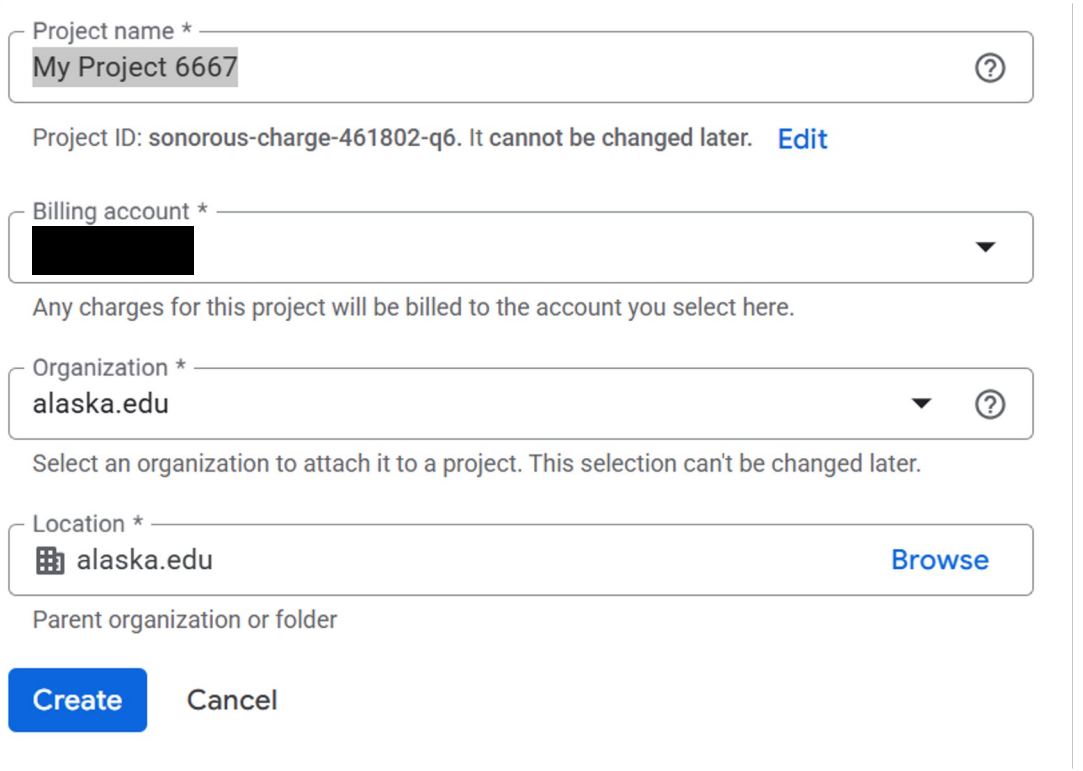
Don't include sensitive information such as personally identifiable information (PII) or security data in your project name, project ID, or other resource names. The project ID is used in the name of many other Google Cloud resources, and any reference to the project or related resources exposes the project ID and resource name.

Create a Project: https://console.cloud.google.com/cloud-resource-manager

Go to the **Manage resources** page in the Google Cloud console.

Go to Manage Resources

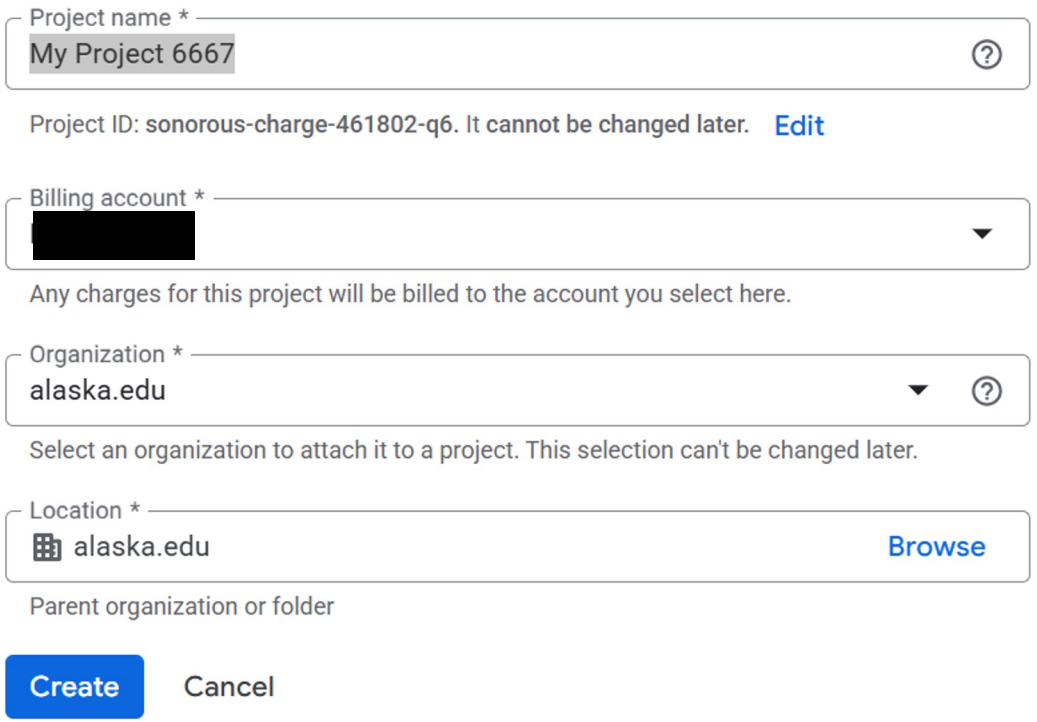
1. Click **Create Project**.
2. In the **New Project** window that appears, enter a project name and select a billing account as applicable. A project name can contain only letters, numbers, single quotes, hyphens, spaces, or exclamation points, and must be between 4 and 30 characters.
3. Enter the parent organization or folder resource in the **Location** box. That resource will be the hierarchical parent of the new project. If **No organization** is an option, you can select it to create your new project as the top level of its own resource hierarchy.
4. When you're finished entering new project details, click **Create**.

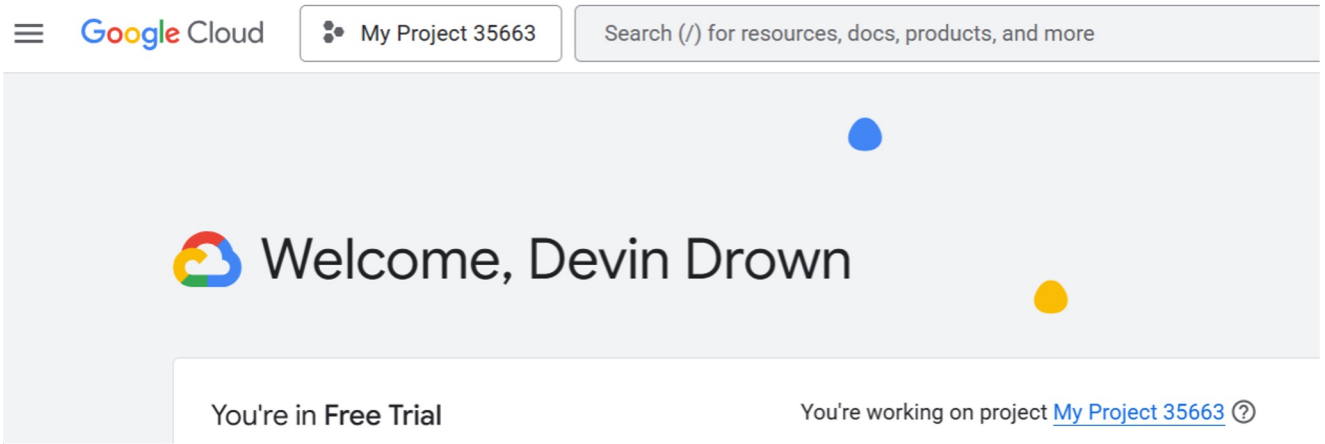


Create a Project: https://console.cloud.google.com/cloud-resource-manager

Go to the **Manage resources** page in the Google Cloud console.

Go to Manage Resources





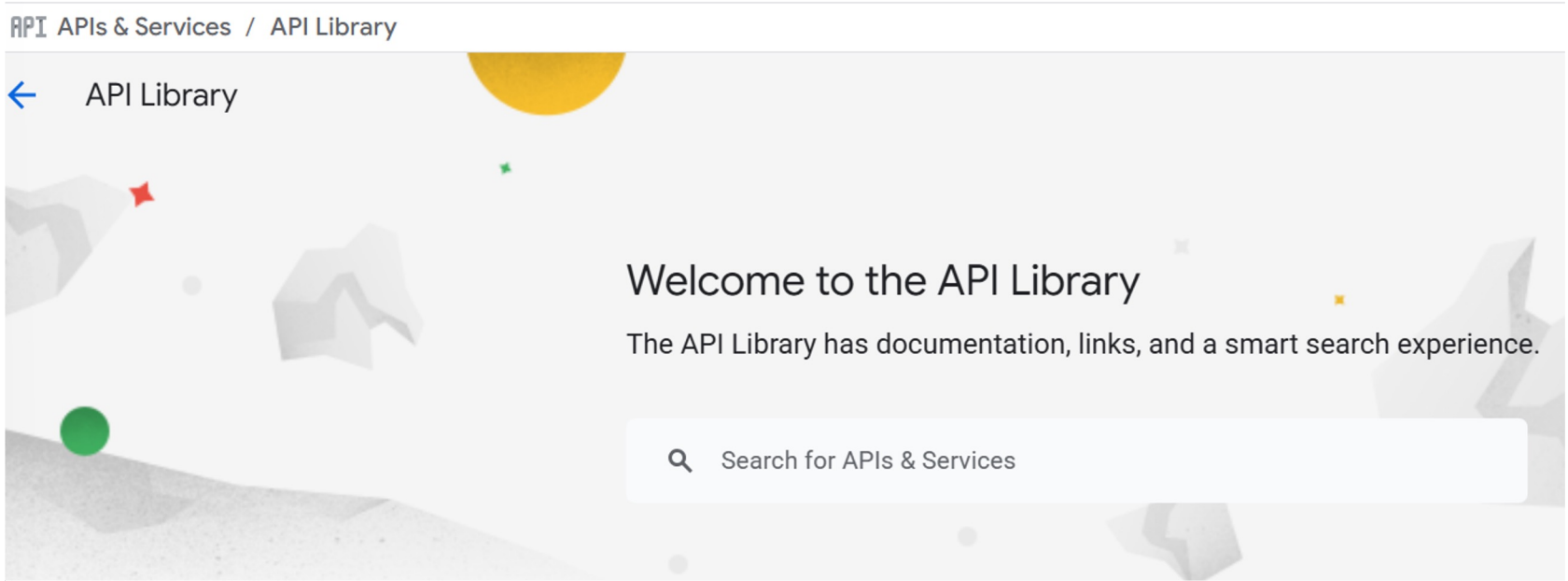
Create a new Vertex AI VM Instance

https://drownlab.github.io/dsc_workshop_2025/tutorials/vertex-ai-workbench/

Create a new Vertex AI VM Instance

https://drownlab.github.io/dsc_workshop_2025/tutorials/vertex-ai-workbench/

1. Enable required APIs (Console),
2. Left sidebar, select APIs & Services → Library
3. Search for and enable the following:
    - Vertex AI API
    - Notebooks API
    - Compute Engine API



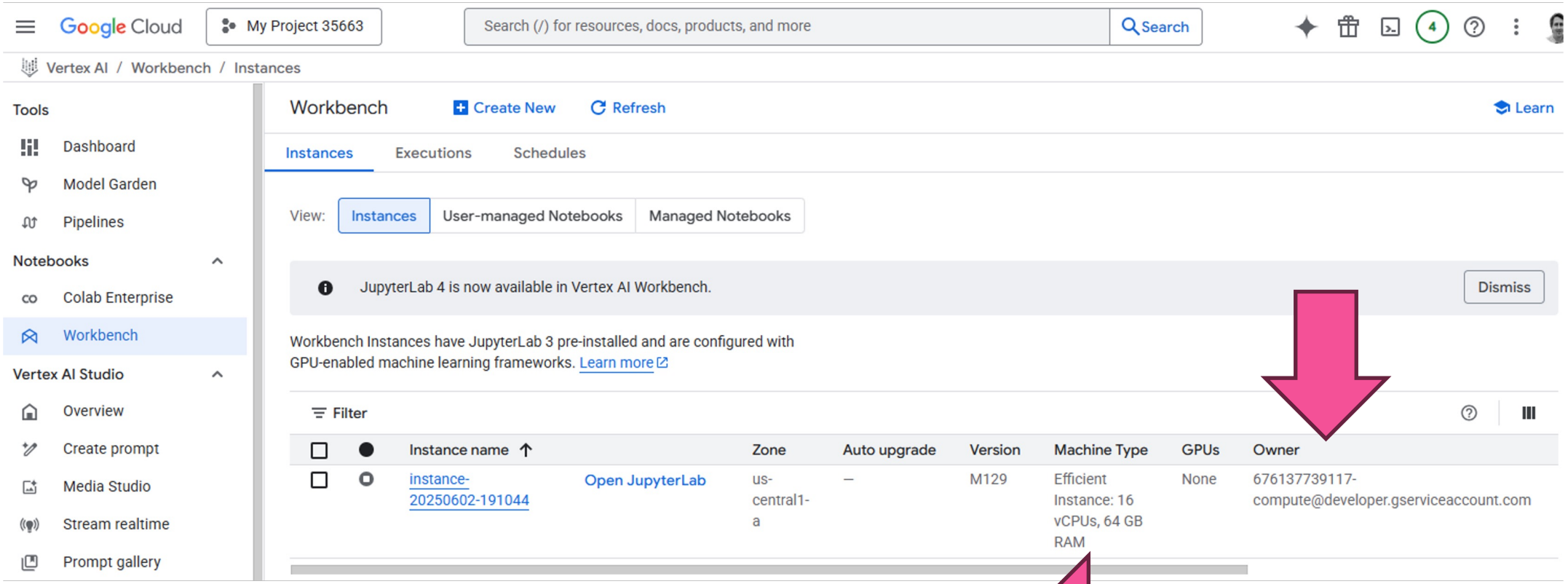
Create the Workbench instance

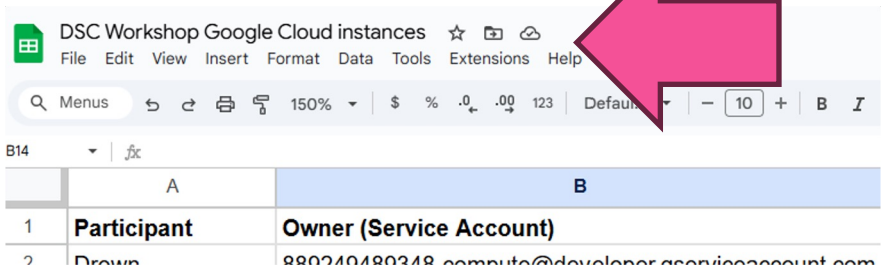
`https://drownlab.github.io/dsc_workshop_2025/tutorials/vertex-ai-workbench/`

1. Navigate to Vertex AI → Workbench.
2. Make sure you're in the Instances view.
3. Click **Create New**.

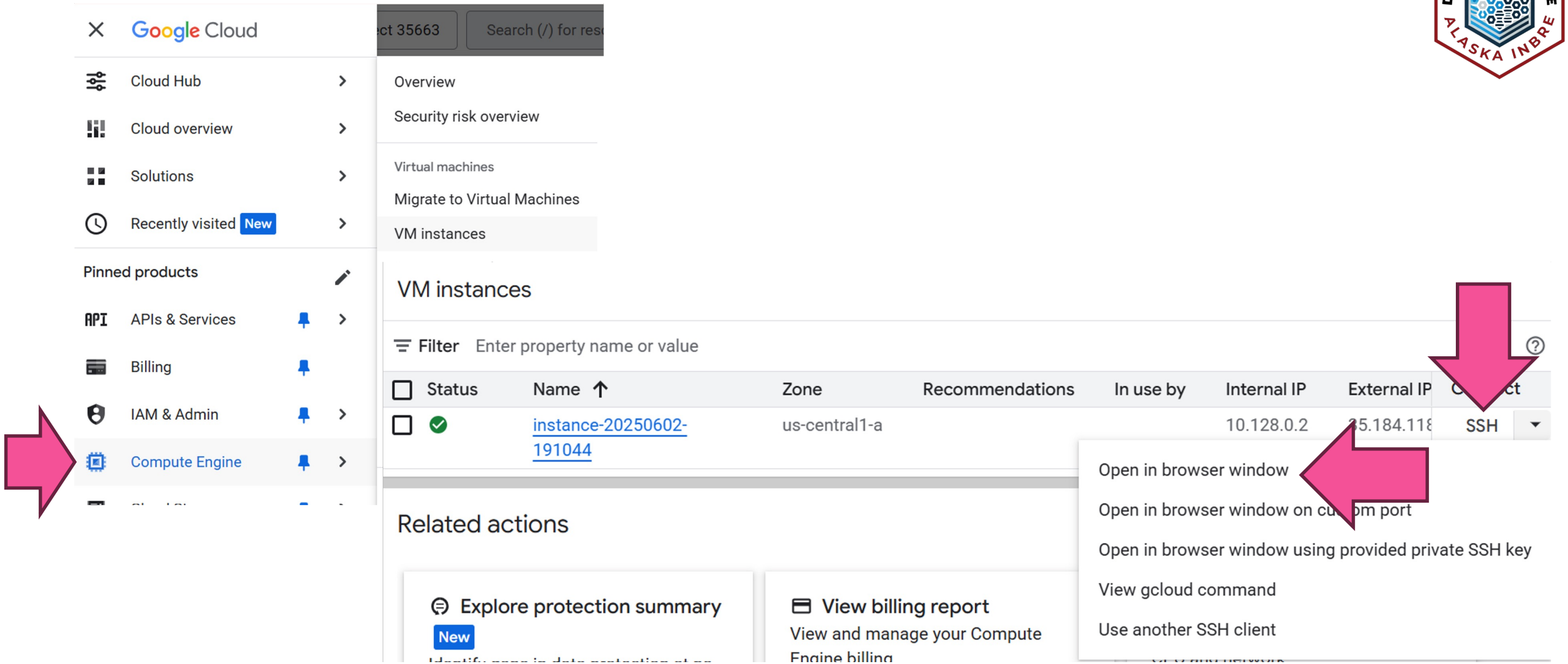
1. Follow along the rest of the instructions on the tutorial.

Share the Workbench instance Service Account

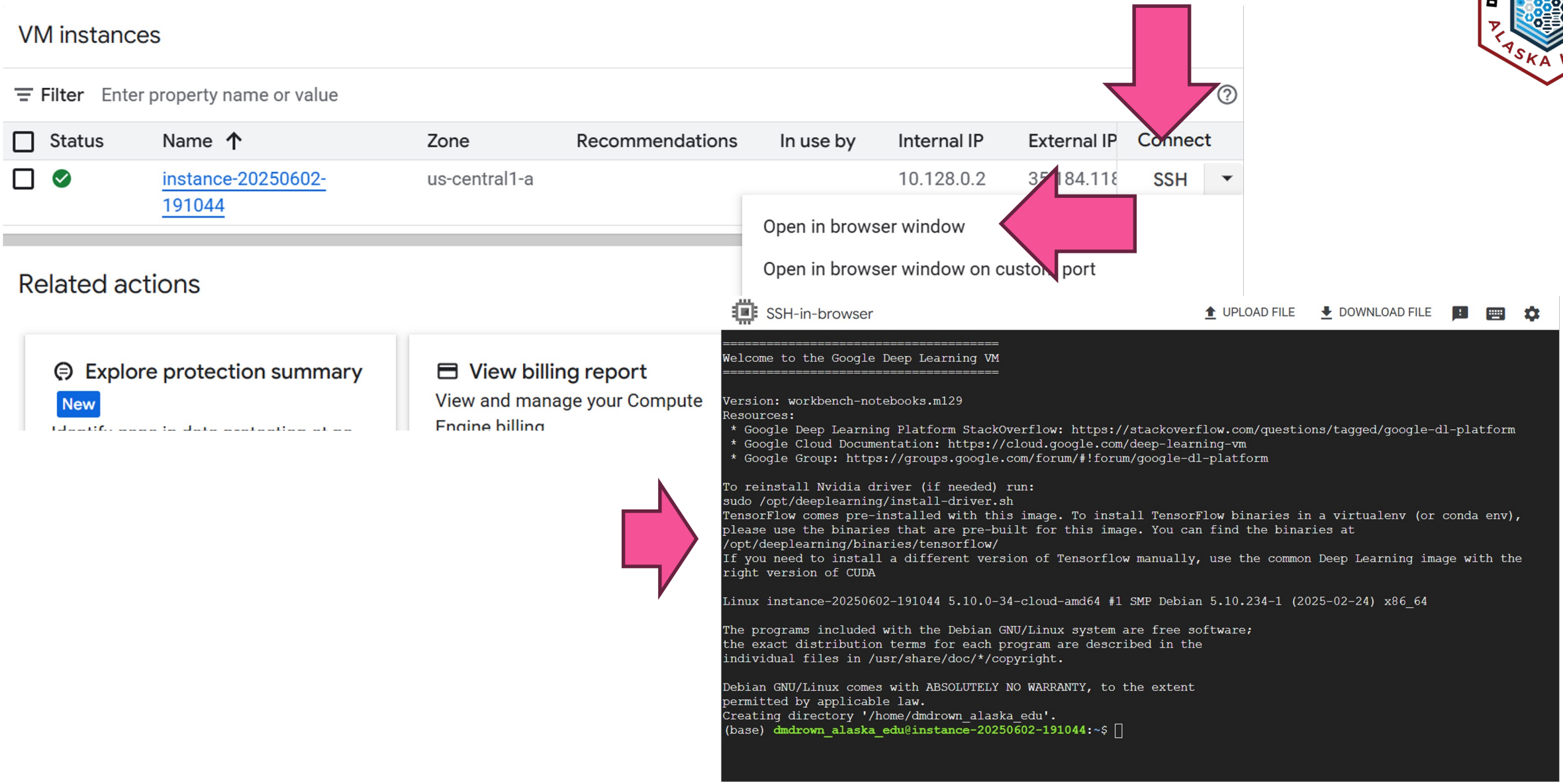




Connecting to your Workbench Instance via SSH



Connecting to your Workbench Instance via SSH



Introduction to the Command Line

Data Carpentry: Introduction to the Command Line for Genomics

Setup instructions

https://drownlab.github.io/dsc_workshop_2025/tutorials/cli-genomics-setup/

Data Carpentry: Introduction to the Command Line for Genomics

https://datacarpentry.github.io/shell-genomics/index.html

1. Introducing the Shell
2. Navigating Files and Directories
3. Working with Files and Directories
4. Redirection
5. Writing Scripts and Working with Data
6. Project Organization

Google Colab

Getting Started with Google Colab

What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier.

Getting Started with Google Colab

</> Generate Code

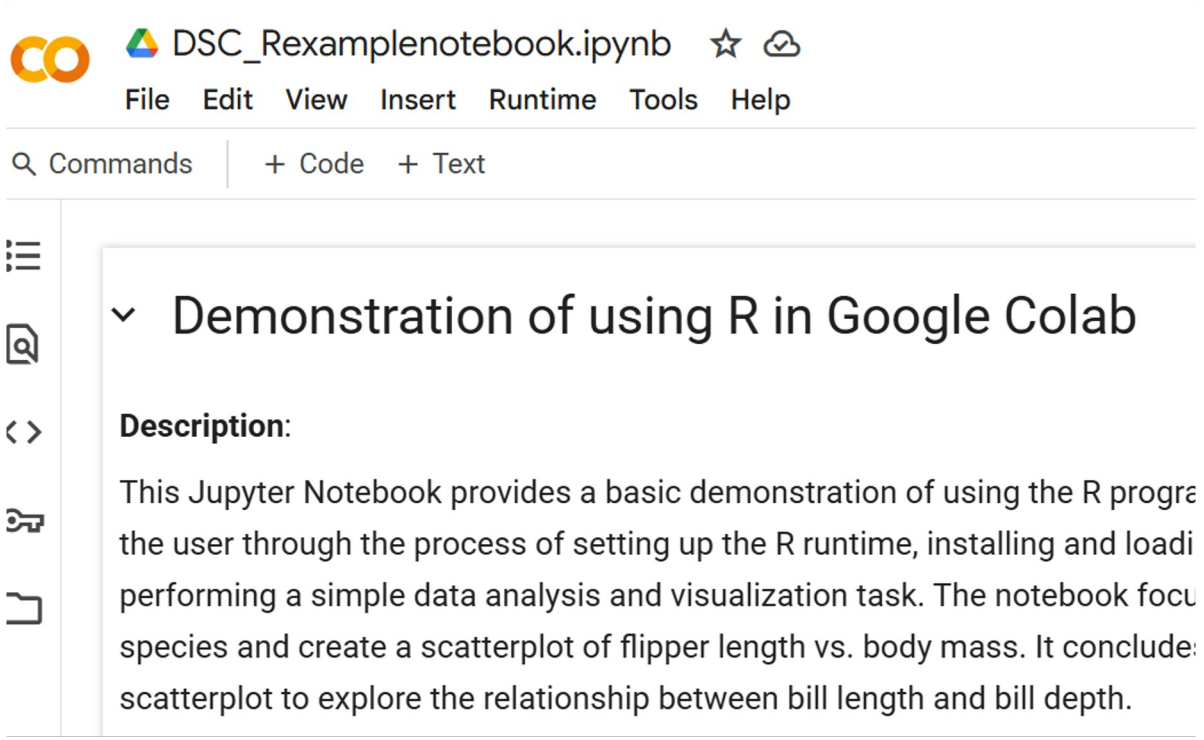
ⓘ Explain Error

Gemini Chat

Getting Started with Google Colab

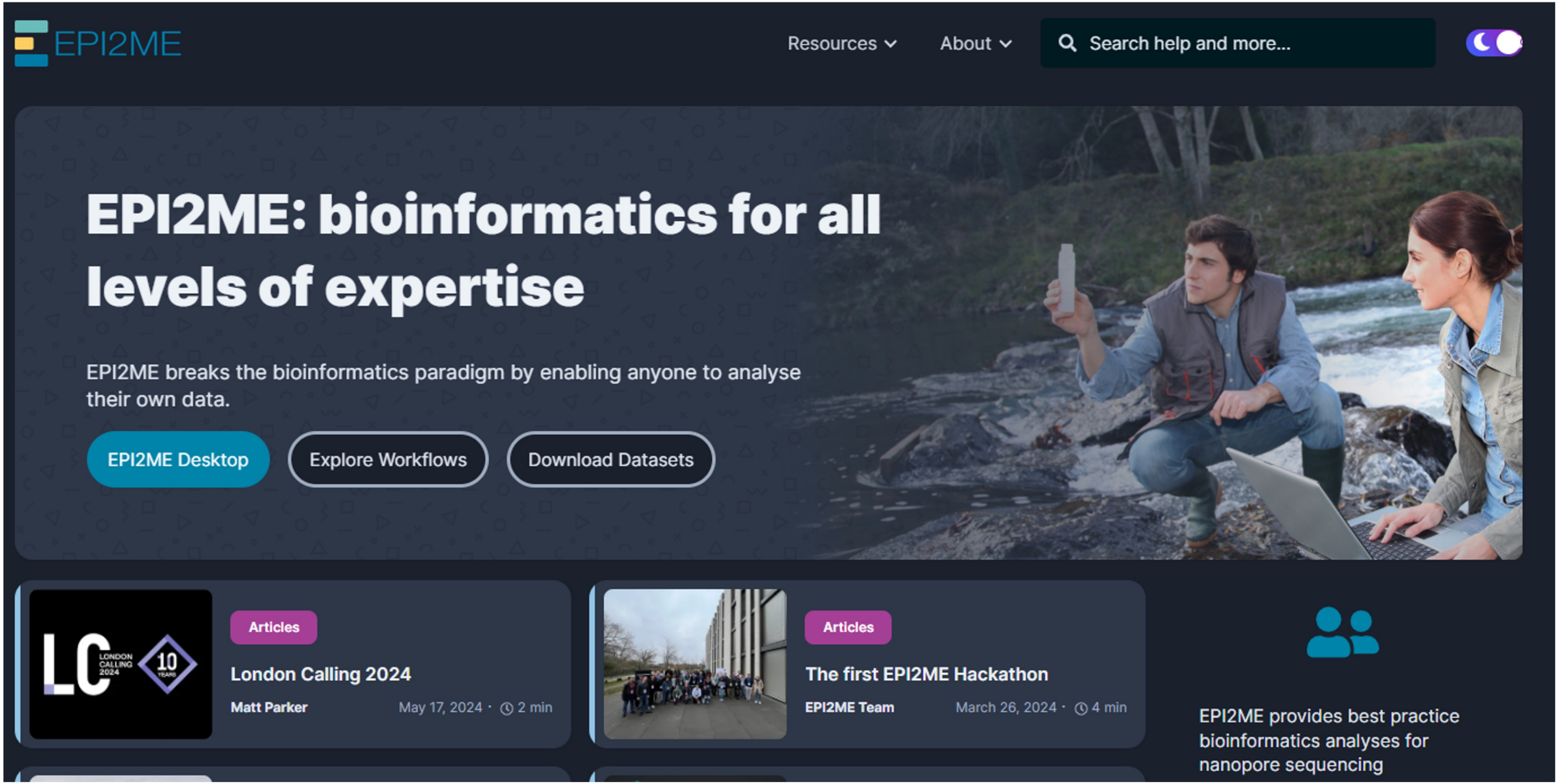
https://colab.research.google.com/

- R Example: DSC_Rexamplenotebook.ipynb
- Building a RAG AI:
  LLM for metagenomic discovery.ipynb



Nanopore Bioinformatics

Real Time Analysis



EPI2ME Workflows

EPI2ME Labs maintains a collection of Nextflow bioinformatics workflows tailored to Oxford Nanopore Technologies long-read sequencing data. They are curated and actively maintained by experts in long-read sequence analysis.

https://epi2me.nanoporetech.com/wfindex/

Introduction to *wf-bacterial-genomes*

What is it? A robust workflow from Oxford Nanopore's EPI2ME Labs for analyzing bacterial genomes.

Primary Goal: To assemble bacterial genomes from Nanopore reads and provide detailed information on features within these assemblies.

Key Capabilities:

- *De novo* (or reference-based) genome assembly.
- Annotation of assembled genomes.
- Optional advanced characterization for bacterial isolates (e.g., AMR, MLST).

Core Pipeline Steps - Data In & Assembly

Input Processing:

- **Purpose:** Prepare raw sequencing data for analysis.
- **Tools:** fastcat (concatenates multi-file samples), bamstats (generates per-read statistics like average read lengths and qualities).
- **Input:** FASTQ or BAM files (single files, directories, or nested directories).

Genome Assembly:

- **Purpose:** Construct the complete bacterial genome sequence.
- **Method:** Performs *de novo* assembly with Flye, meaning it builds the genome from scratch without a pre-existing reference. This is critical for novel strains or species.

Downstream Analysis - Annotation & Isolate Characterization

Genome Annotation:

- **Purpose:** Identify and label regions of interest within the assembled genome.
- **Tool:** Prokka (rapid prokaryotic genome annotator).
- **Output:** Identifies protein-coding genes, rRNA, tRNA, and other features.

Isolates Mode (Optional - use --isolates flag):

- **Purpose:** Provide in-depth characterization of bacterial isolates.
- **Components:**
  - **Multi-locus sequence typing (MLST):** Characterizes isolates using allelic variations in housekeeping genes (e.g., PubMLST schemes).
  - **Antimicrobial Resistance (AMR) Calling:** Identifies genes and SNVs associated with AMR using tools like ResFinder.
  - ***Salmonella* Serotyping:** Predicts serotype and antigenic profile for *Salmonella* samples using SeqSero2.

Key Outputs & Considerations

Primary Outputs:

- Assembled bacterial genome sequences.
- Detailed genome annotations.
- Taxonomic classifications (from MLST).
- Antimicrobial resistance profiles.
- *Salmonella* serotyping results (if applicable).

Benefits: Comprehensive, automated, reproducible, and scalable for bacterial genomics research.

Typical Runtime: Approximately 20-40 minutes per sample with ~50x coverage (using minimum requirements).

Execution: Run via Nextflow on the command line (e.g., in a JupyterLab terminal).

Introduction to *wf-metagenomics*

What is it? A versatile workflow from Oxford Nanopore's EPI2ME Labs for analyzing metagenomic sequencing data.

Primary Goal: To unveil the taxonomic composition of microbial communities and identify key genetic features within them.

Key Capabilities:

- Taxonomic classification of reads (e.g., bacteria, archaea, fungi, viruses).
- Identification of antimicrobial resistance (AMR) genes.
- Generation of comprehensive reports and visualizations of microbial profiles

Pipeline Steps - Data Processing & Analysis

Input Data:

- **Purpose:** Provide raw sequencing data from the microbial community.
- **Input:** FASTQ or BAM files.

Taxonomic Classification:

- **Purpose:** Determine the "who is there" in your sample by assigning taxonomic labels to reads.
- **Methods:**
  - **Kraken 2:** A k-mer based classifier, fast and efficient for broad taxonomic assignment.
  - **Minimap2:** Aligns reads to a reference database for more detailed taxonomic identification.
- **Databases:** Utilizes built-in or custom databases (e.g., NCBI 16S/18S, comprehensive genomic databases).

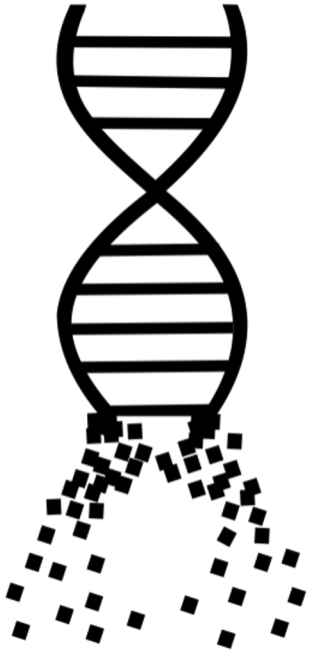
Bioinformatics pipeline

Basecalling → Dorado → ATGCATCG

Quality control → Length Filter → Quality Filter → QC Reads

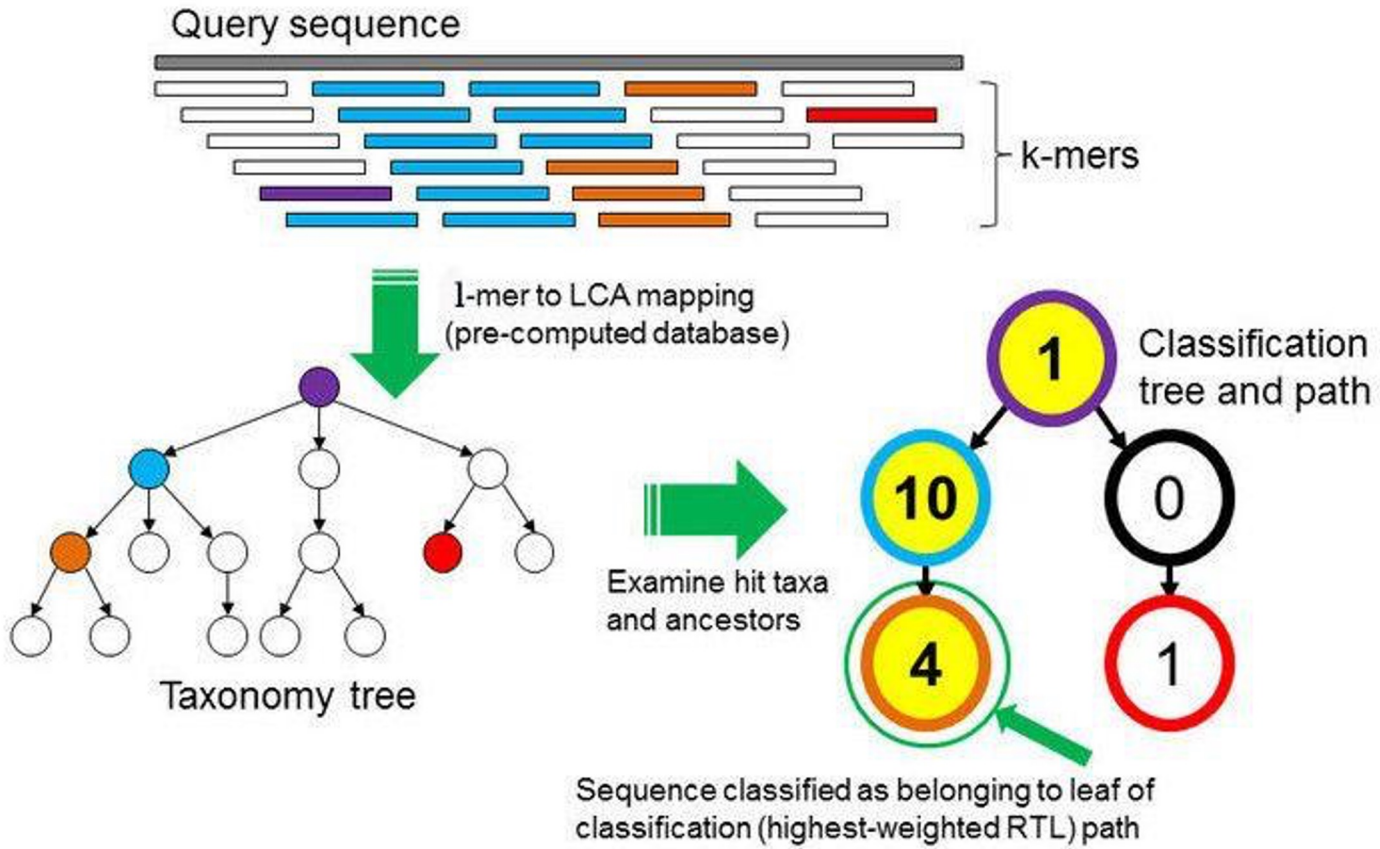
Taxonomic Classification → First pass classification → Classification Report

Refeq archaea, bacteria, viral, plasmid, human,



Taxonomic Classification with Kraken2

- Kraken2 uses exact-match database queries of k-mers, rather than inexact alignment of sequences.
- Sequences are classified by querying the database for each k-mer in a sequence, and then using the resulting set of lowest common ancestor (LCA) taxa to determine an appropriate label for the sequence.



Use your Vertex AI Instances and run the Workflows

https://drownlab.github.io/dsc_workshop_2025/tutorials/vertex-ai-EPI2ME-workbench/