NIH STRIDES

Accelerating biomedical research

Accelerating Bioinformatics Workflows with Nextflow

Zelaikha Yosufzai, MS [C]

Introduction to Cloud Lab

NIH STRIDES
Accelerating biomedical research

Opportunities and Challenges of the Cloud

Cloud is the new frontier of frontier of biomedical research: the potential benefits are considerable, but so are the challenges.

THE OPPORTUNITIES

Simplifies testing and iteration

Democratizes access to scientific data

Always available on demand

Easily scales up and down

Pay as you go for only what you use

Provides a rich set of tools & services

THE CHALLENGES

Acquisition is complex and time-consuming

Security protections are unclear

Costs are unpredictable

Training deficit is substantial

NIH STRIDES
Accelerating biomedical research

Gap Between Cloud Interest and Cloud Adoption

It's very common for researchers to face significant barriers in taking the first step toward cloud adoption. Some of the barriers we hear about most frequently are included below.

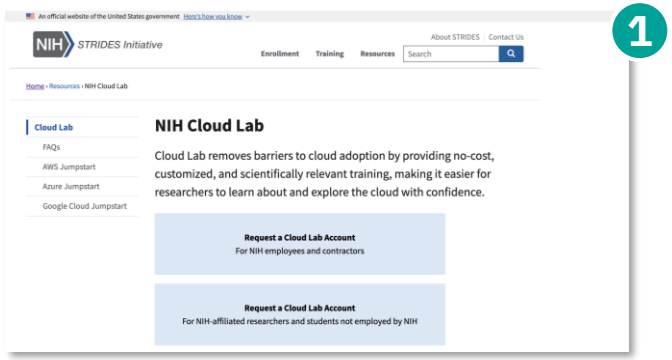
- "I don't know which cloud platform to use or how different services compare to one another within or across cloud platforms."
- "Coordinating internal funding at NIH (MOUs and DCCs) and/or establishing contractual agreements with resellers for extramural institutions seems complicated and time-consuming."
- "I'm not sure how much I need to budget for commercial cloud services and am concerned about overspending."
- "I'm not confident that I understand all of the roles and responsibilities necessary to support work in the cloud."
- "I'm uncertain about what the transition to cloud will entail and am hesitant to jump in with both feet."

NIH Cloud Lab | Experiment in the Cloud

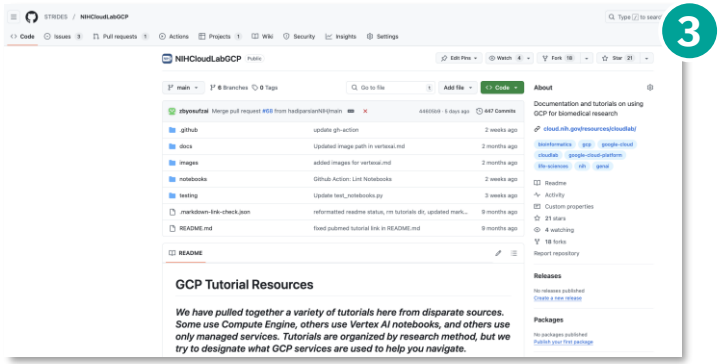
NIH Cloud Lab is a no-cost, 90-day program for NIH intra- and extramural researchers to try commercial cloud services in an NIH-approved environment. Cloud Lab provides training and guardrails to protect against financial and security risks.

How It Works

1. **Fill out interest form**
2. **Get account and $500 of credits***
3. **Access tailored cloud trainings**
4. **Practice and learn for 90 days**



NIH Cloud Lab Sign Up Page



NIH Cloud Lab GCP Tutorial Repository

Example of NIH Cloud Lab Use Case

*Step 2 will be provided to you by the Cloud Lab team.

NIH STRIDES
Accelerating biomedical research

NIH Use Cases

Evaluate Utility & Cost

Provides an easy route to evaluate the cloud's utility/cost for a project without major time or financial commitments

Develop New Tools

Allows experienced teams to prototype new architectures and evaluate software and hardware combinations

Share Ideas

Connects NIH'ers from across ICs to share ideas on how to conduct biomedical research in the cloud

Learn New Skills

Simplifies access to tools and cloud environments that participants can use for training purposes

Training Materials

We provide jumpstart pages, in-depth tutorials, and how-to documents that help users get ready for the cloud.

Azure

- Jumpstart page: https://cloud.nih.gov/resources/cloudlab/azure-jumpstart/
- GitHub repo: https://github.com/STRIDES/NIHCloudLabAzure

Google Cloud

- Jumpstart page: https://cloud.nih.gov/resources/cloudlab/google-cloud-jumpstart/
- GitHub repo: https://github.com/STRIDES/NIHCloudLabGCP

AWS

- Jumpstart page: https://cloud.nih.gov/resources/cloudlab/aws-jumpstart/
- GitHub repo: https://github.com/STRIDES/NIHCloudLabAWS

NIH STRIDES
Accelerating biomedical research

Publicly Available Interactive Tutorials Across All Three CSPs!

Cloud Lab offers a **suite of interactive tutorials** designed to help participants perform viable research in the cloud. Open to all who wish to gain skills, not just Cloud Lab participants!

Scientific

- Accessing Open Datasets
- AI/ML
- BLAST
- FHIR
- Genomic Variant Calling
- GWAS
- Long Read Genomics
- Medical Image Segmentation
- Proteomics
- RNAseq
- scRNAseq
- Radiogenomics
- SRA Data Interaction
- VCF Query with SQL
- ...and more

GenAI

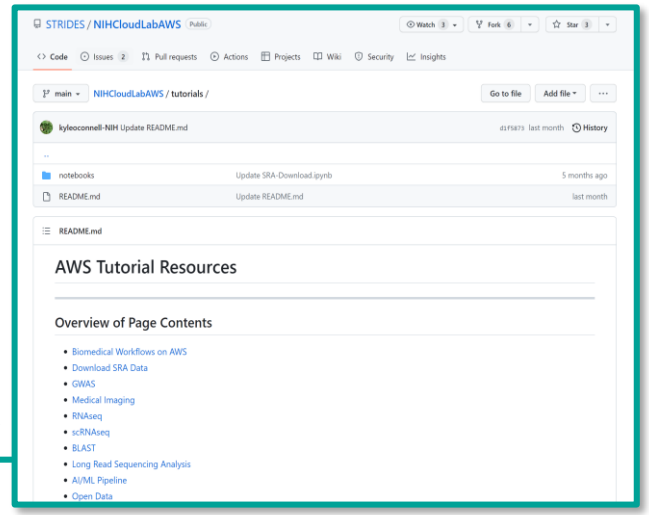
- Model Training
- Guidance & Best Practices
- Introduction to tools across all platforms
- Leverage Large Language Models (LLMs)
- Chatbot Implementation
- Retrieval- Augmented Generation (RAG)

General Cloud Ops

- Access Marketplace Offerings
- Access Public Datasets
- Command Line Tools / SDKs
- Disk Images
- Ingest and Store Data
- Introduction to Cloud
- Jupyter Notebooks
- Serverless Functionality
- Virtual Machines

Computing & Code

- Conda Environments
- Container Registries
- Git Repos
- HPC Clusters
- Kubeflow
- Serverless Functionality



NIH Cloud Lab AWS Tutorial Repository

NIH STRIDES
Accelerating biomedical research

Introduction to Nextflow

NIH STRIDES
Accelerating biomedical research

What is Nextflow?

Nextflow is a powerful workflow management system designed specifically for bioinformatics and data science that can be run locally or in the cloud.

Imagine you have a complex analysis task that involves many different programs and steps, potentially running on different computers or cloud services. Nextflow lets you define this entire process as a program, specifying the order of operations, how data flows between steps, and how to handle errors or resource limitations.

Nextflow consists of two main pieces:

Main file: This is the core of your workflow. It's written in the Nextflow DSL (Domain-Specific Language) and defines the processes, data channels, and execution logic of your analysis. Files are created using the .nf extension

- You can also break out processes into modules and sub workflows

Config file: Allows you to customize various aspects of your workflow's execution without modifying the main pipeline script improving reproducibility. Here you can set parameters like input/output paths, resource allocation (CPU, memory), execution environments (e.g., Docker images, Singularity containers), and other settings affecting the pipeline's behavior across different runs and environments. Config files are created with the .config extension.

Scripting in Nextflow

A Nextflow script, typically with a .nf extension, is composed of several key parts working together to define your data processing workflow:

1. **`params` block:** Defines parameters that control the workflow's behavior.
    - These can be things like **input file paths, output directories, specific program options, or thresholds.** You can set default values, allowing users to run the workflow with minimal configuration, or override them at runtime. This allows for flexibility and reproducibility across different runs.
2. **`processes` block:** This is the heart of the script. It defines individual processing steps, each encapsulated as a "process." Each process consists of:
    1. **name:** A unique identifier for the process.
    2. **script:** The commands to execute (often shell commands). This is where you specify the software and its options.
    3. **input:** Specifies the data the process requires. This is often defined using channels, which manage the flow of data between processes.
    4. **output:** Specifies the data the process produces. Again, often uses channels.

```
#!/usr/bin/env nextflow
nextflow.enable.dsl = 2

Channel
    .fromPath("${params.input}", type: 'file', checkIfExists: true)
    .ifEmpty { exit 1, 'params.input_paths was empty - no input files supplied' }
    .map { file -> tuple(file.baseName, file) }
    .set{ read_ch }

process fastqc {
    tag "$name"

    label 'process_medium'

    conda "bioconda::fastqc=0.11.9"
    container "${ workflow.containerEngine == 'singularity' && !task.ext.singularity_pull_docker_container
        'https://depot.galaxyproject.org/singularity/fastqc:0.11.9--0' :
        'quay.io/biocontainers/fastqc:0.11.9--0' }"

    publishDir "${params.outdir}/fastqc", mode: "copy", overwrite: true,
        saveAs: { filename ->
                      filename.indexOf('.zip') > 0 ? "zips/$filename" : "$filename"
        }

    input:
    tuple val(name), path(reads)

    output:
    tuple val(name), path('*_fastqc.{zip,html}'), emit ch_fastqc_results_for_multiqc
    script:
    """
    fastqc --quiet --threads $params.cpus $reads
    """
}

workflow {

    fastqc(read_ch)
    multiqc(ch_fastqc_results_for_multiqc)
}
```

Scripting in Nextflow continued

3. **`channels`** (implicit or explicit): Channels are the pipelines' arteries. They manage the flow of data between processes. Data is passed between processes via channels. While not always explicitly defined, they are the fundamental mechanism for data transfer. If not explicitly named, Nextflow will infer channels based on input and output statements within processes.

4. **`workflow`** block: This section assembles the individual processes into a cohesive workflow. It defines the execution order and how the output of one process becomes the input of another. This is often achieved by connecting processes through channels.

5. **`publishDir`** (optional): This parameter specifies a directory where the final results of the workflow will be saved.

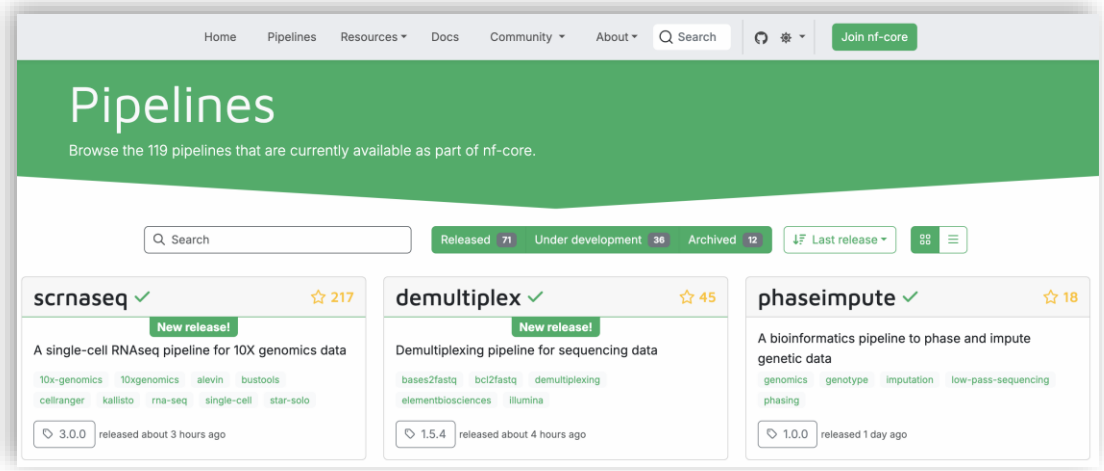
6. **`workDir`** (optional): This parameter specifies a directory where intermediate files will be stored during the workflow's execution.

Using Nf-Core

Let's not reinvent the wheel! One of the great advantages to using Nextflow is that it is integrated with a community driven effort named nf-core.

Nf-core's goal is to curate and collect sets of analysis workflows or pipelines built using Nextflow. These workflows/pipelines are bioinformatic focused and allow the user to run well known analyses like RNAseq, Methylseq, etc.

Running any of the nf-core workflows automatically pulls the workflow scripts from GitHub. As depicted from the image to the right Nextflow print outs the Nextflow version and that it is pulling the Methylseq scripts from the nf-core GitHub.



NIH STRIDES
Accelerating biomedical research

Installing Nextflow and Using nf-core

The code below shows how to install Nextflow and run a nf-core pipeline.

1. Install Java

In order to successfully install Nextflow your system must have Java preinstalled.

```
sudo apt update

sudo apt-get install default-jdk -y

java –version
```

2. Specify Plug-ins

If you are using a cloud platform to run your Nextflow analysis you can specify the associated Nextflow plug-in, in the example below we are installing the google plug-in.

```
export NXF_MODE=google
```

3. Install Nextflow

Install Nextflow, make it executable, and update it.

```
curl https://get.nextflow.io | bash

chmod +x nextflow

./nextflow self-update
```

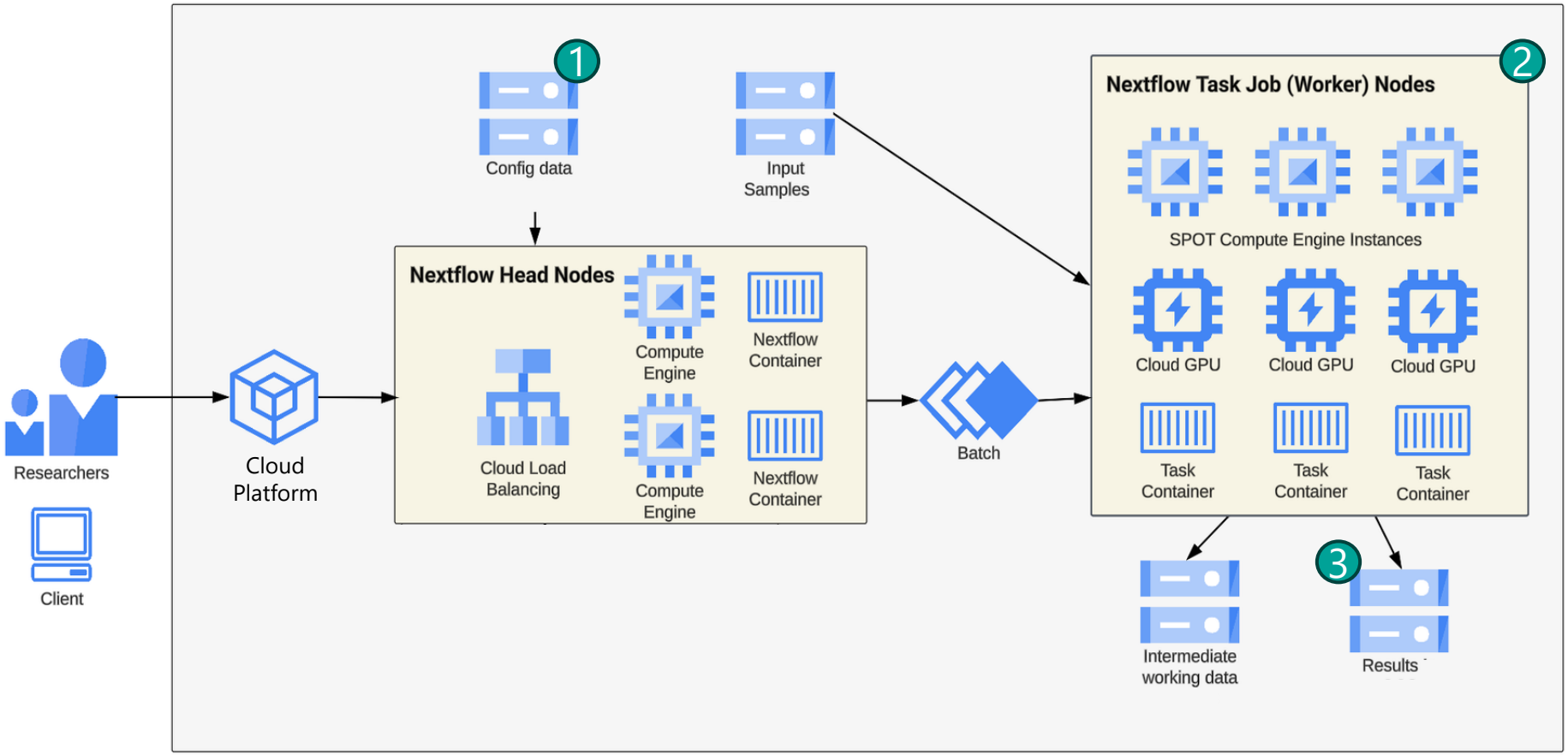
4. Run a nf-core pipeline

```
./nextflow run nf-core/methylseq \
      --input 'SRR067701.fastq.gz' \
      --genome GRCh37 \
      --single_end \
      --max_cpus 32 \
      --max_memory '110.GB'
```

Utilizing Schedulers

Cloud platforms like Azure, Google Cloud, and AWS all have batch systems that provides users a serverless, scalable, and cost-effective way to run Nextflow pipelines.

1. Configure specified Batch system, eliminating manual cluster management
2. Auto provision and deallocates resources, handling parallelism and scaling
3. Store results in cloud storage for long term access



NIH STRIDES
Accelerating biomedical research

Demo

NIH STRIDES
Accelerating biomedical research

Poll and Questions, Call to Action Banner

The Cloud Lab team can help users navigate the complexities of cloud computing by offering informative sessions tailored to their needs providing clear explanations of how various cloud services function, focusing on practical application and addressing specific user questions.

Support We Provide

1. Technical support related to tutorial challenges
2. General Troubleshooting
3. Ensuring work done within Cloud Lab can be migrated to a production ready environment

Collaboration Opportunities

Cloud Services is creating tutorials monthly and would love to collaborate with your IC to develop scientifically focused content, if interested please email us at CloudLab@nih.gov.

User Poll: Tutorial Content

Vote on what tutorial topic you find most beneficial to you.

Join at:
Slido.com

NIH STRIDES
Accelerating biomedical research

Q&A

NIH STRIDES
Accelerating biomedical research

NIH STRIDES

Accelerating biomedical research

For more information, please visit cloud.nih.gov
or send an email to STRIDES@nih.gov.