Reproducible bioinformatics at scale:

nf-core + Nextflow

Phil Ewels · Senior Product Manager for Open Source, Seqera

phil.ewels@seqera.io

ewels.bsky.social ewels

Bioinformatics CRO Webinar · February 2026

ABOUT ME

Phil Ewels

Senior Product Manager, Seqera

Creator of MultiQC. Former head of genomics development at Swedish National Genomics Infrastructure - SciLifeLab, Sweden. Co-founder of nf-core and PM + contributor to Nextflow.

10+

Years in bioinformatics

nf-core

Co-founder, core team member

MultiQC

Creator & maintainer

THE PROBLEM

The reproducibility crisis

70%

of researchers failed to reproduce another scientist's experiments

Nature, 2016 · 10.1038/533452a

50%

failed to reproduce their own experiments

- Complex dependencies
- Diverse compute environments
- No workflow standards
- Poor data provenance

NEXTFLOW

Workflow orchestration for reproducible scientific pipelines

Flow-based syntax

Simple, readable syntax for complex data pipelines

Software built-in

Docker, Singularity, Conda for full reproducibility

Multi-platform

Laptop, HPC cluster, or any major cloud provider

Resume & cache

Automatic caching — resume from any failure point

Scalable

One sample to thousands with auto-parallelization

Active ecosystem

140+ curated pipelines via the nf-core community

HOW IT WORKS

Four simple steps

1. **Define**
   Write processes (commands) in Nextflow code

2. **Connect**
   Link processes via channels. Nextflow handles data flow automatically

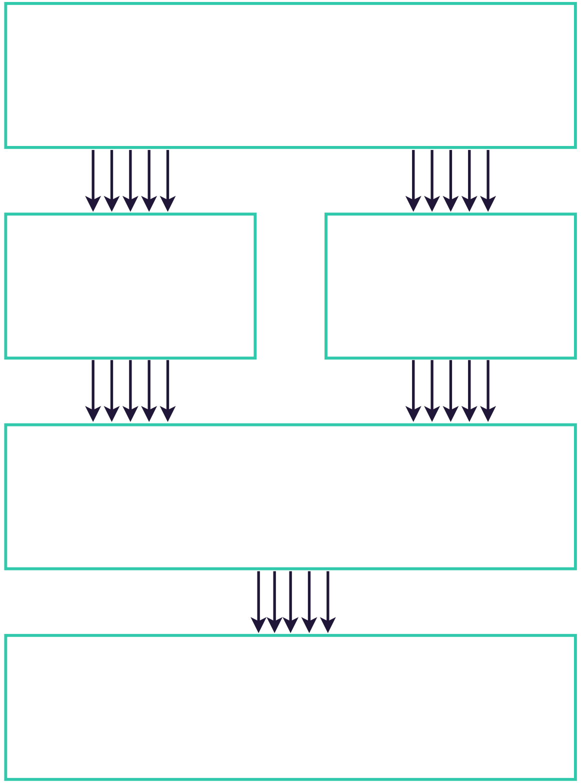
3. **Configure**
   Set compute environment, containers, and resources

4. **Execute**
   Run on laptop, HPC, or cloud

```
process {

  input:
  path 'sample.fq.gz'

  output:
  path 'report.html'

  script:
  """
  mytool sample.fq.gz \
    -o report.html
  """

}
```

HOW IT WORKS

Four simple steps



1. **Define**
Write processes (commands) in Nextflow code

2. **Connect**
Link processes via channels. Nextflow handles data flow automatically

3. **Configure**
Set compute environment, containers, and resources

4. **Execute**
Run on laptop, HPC, or cloud

HOW IT WORKS

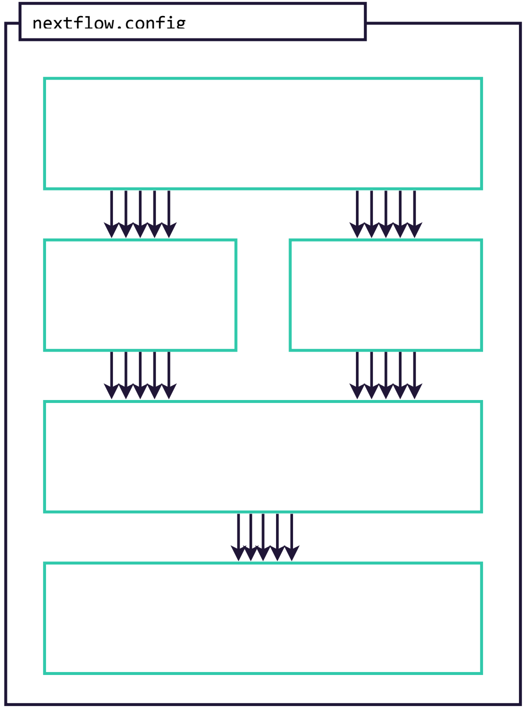
Four simple steps

1. **Define**
   Write processes (commands) in Nextflow code

2. **Connect**
   Link processes via channels. Nextflow handles data flow automatically

3. **Configure**
   Set compute environment, containers, and resources

4. **Execute**
   Run on laptop, HPC, or cloud



KEY CONCEPTS

Nextflow key points

Language

Code for describing workflow steps and data flow

Configuration

Separate configuration avoids modifying pipeline source code

Orchestration

Nextflow runs the pipeline, submitting and monitoring jobs

Reproducible

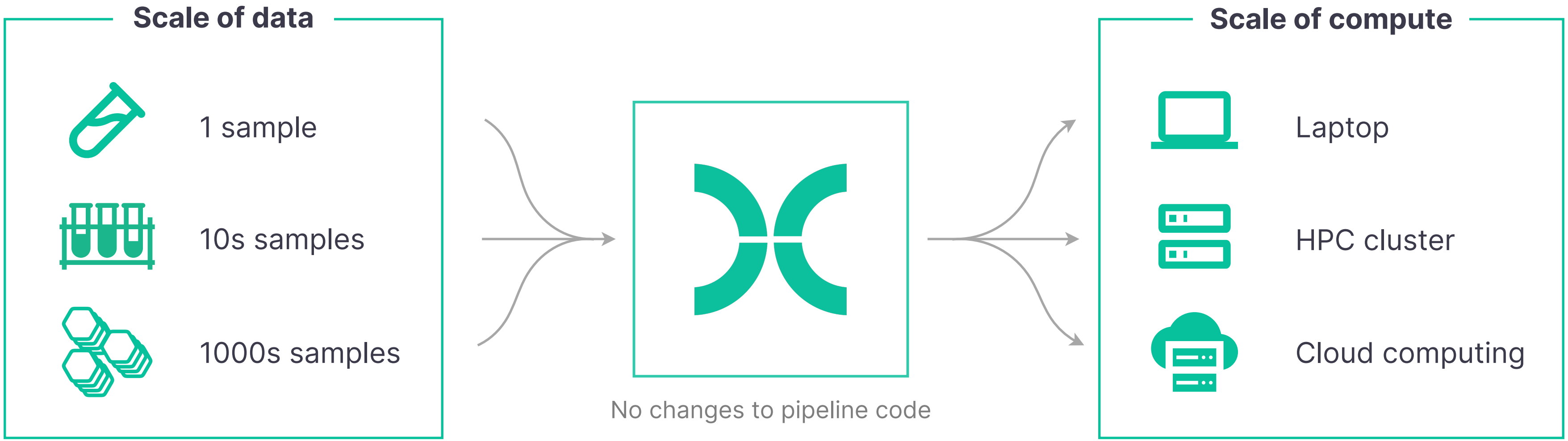
Same results, every time

Portable

Write once, run anywhere

AT SCALE

Write once, scale everywhere



Scale of data

- 1 sample
- 10s samples
- 1000s samples

No changes to pipeline code

Automatic parallelization

Scale of compute

- Laptop
- HPC cluster
- Cloud computing

GROWTH

Nextflow usage continues to climb

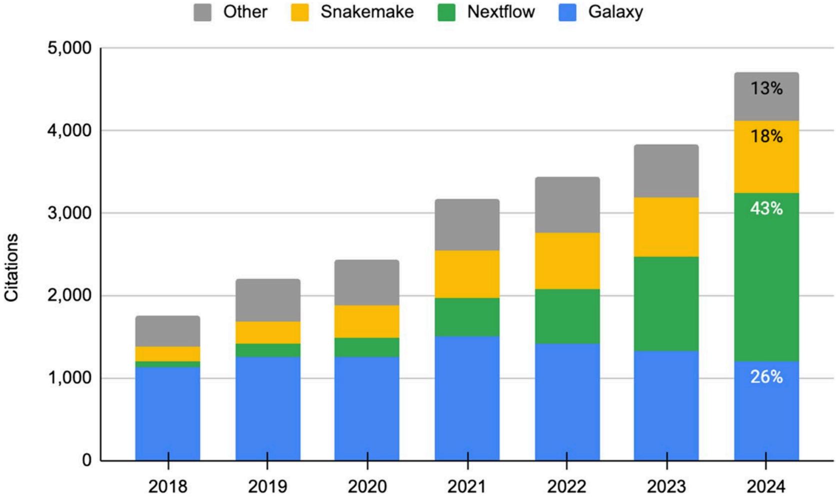
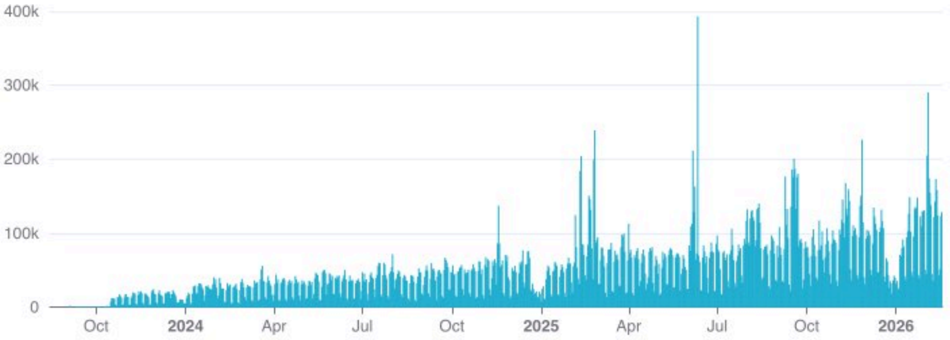


Fig. 1 Google Scholar citation counts for bioinformatics workflow management systems. Sum of citations of the major publications of Galaxy, Nextflow, and Snakemake between 2018 and 2024 (Data in Additional File 1: Supp. Table 1)

DOI: 10.1186/s13059-025-03673-9

Nextflow daily runs



nf-core

A global community collaborating to build open-source Nextflow components and pipelines

nf-core

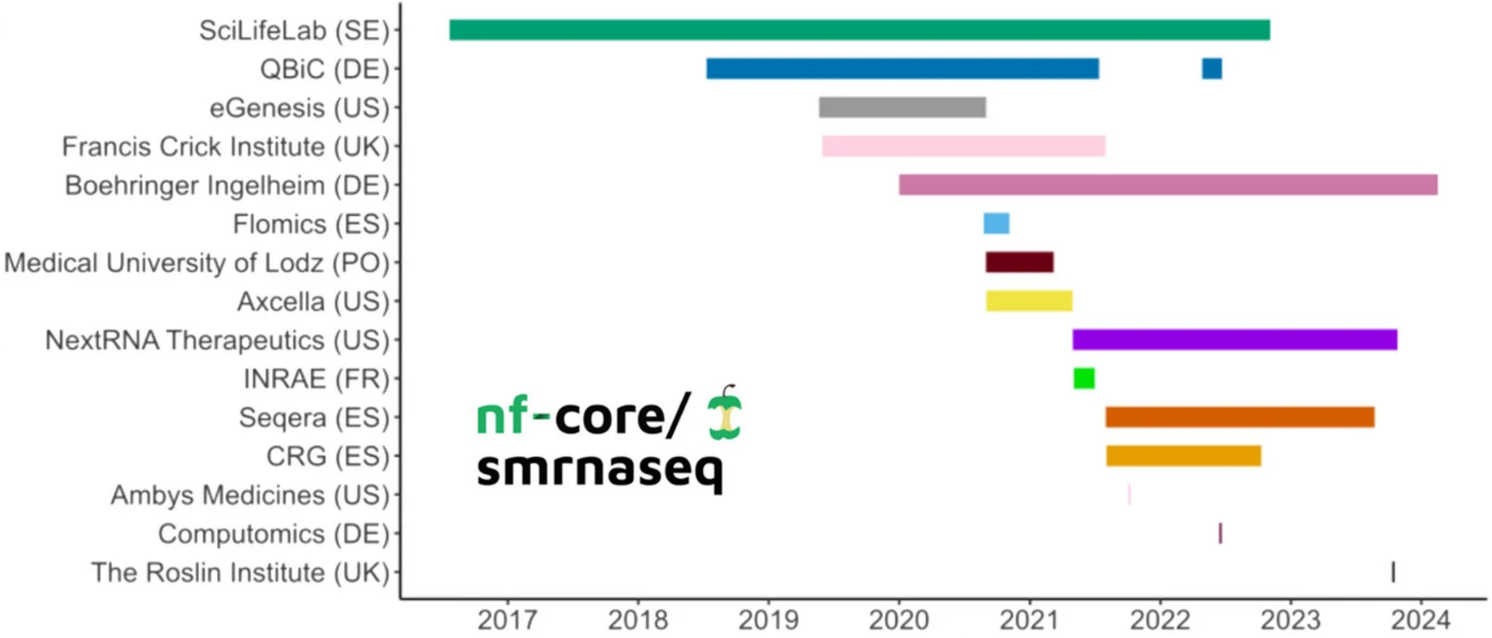
A global community collaborating to build open-source Nextflow components and pipelines

140+	1,700+	13,600+
Pipelines	Modules	Community members

- → Best-practice pipeline design patterns
- → Standardized tooling, configuration & testing
- → Extensive documentation for every pipeline

NF-CORE ECOSYSTEM

Community owned



BUILD WITH NF-CORE

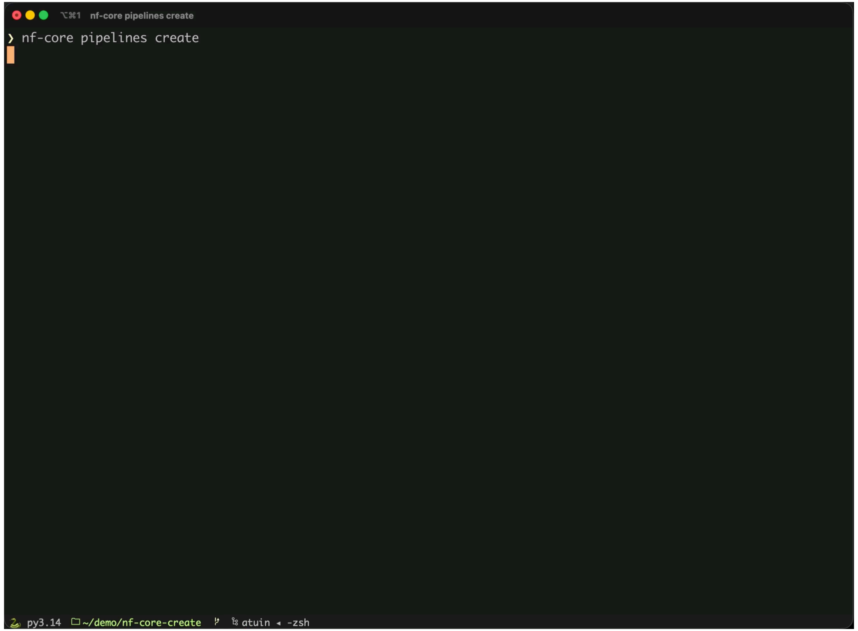
Not just pipelines

Modules

Reusable process definitions for 1,700+ bioinformatics tools. Mix and match for custom workflows.

Tools

nf-core/tools CLI for creating, linting, testing, and launching pipelines with best practices.

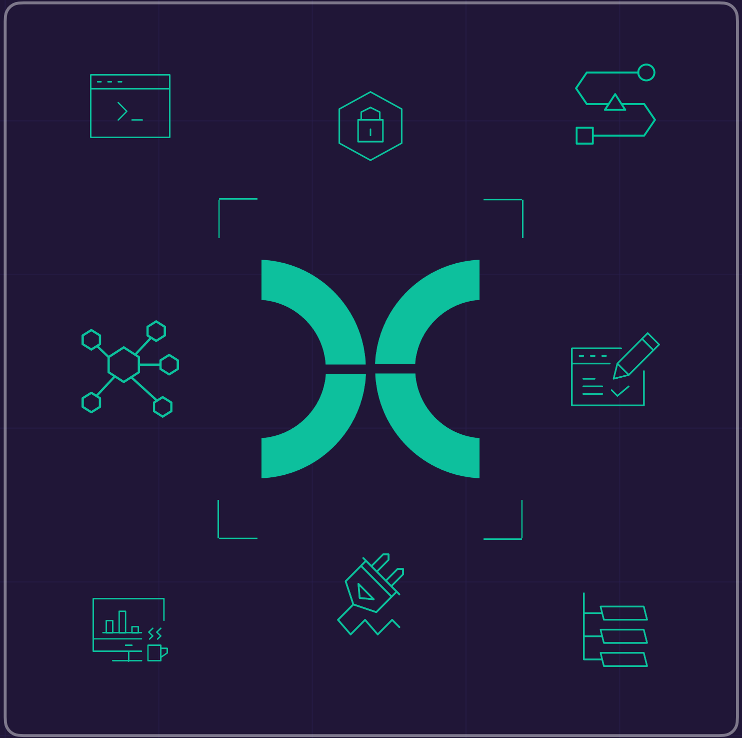


```
❯ nf-core pipelines create
```

```
py3.14 ~/demo/nf-core-create atuin ◂ -zsh
```

What's new in Nextflow

Releases 25.04, 25.10, and beyond

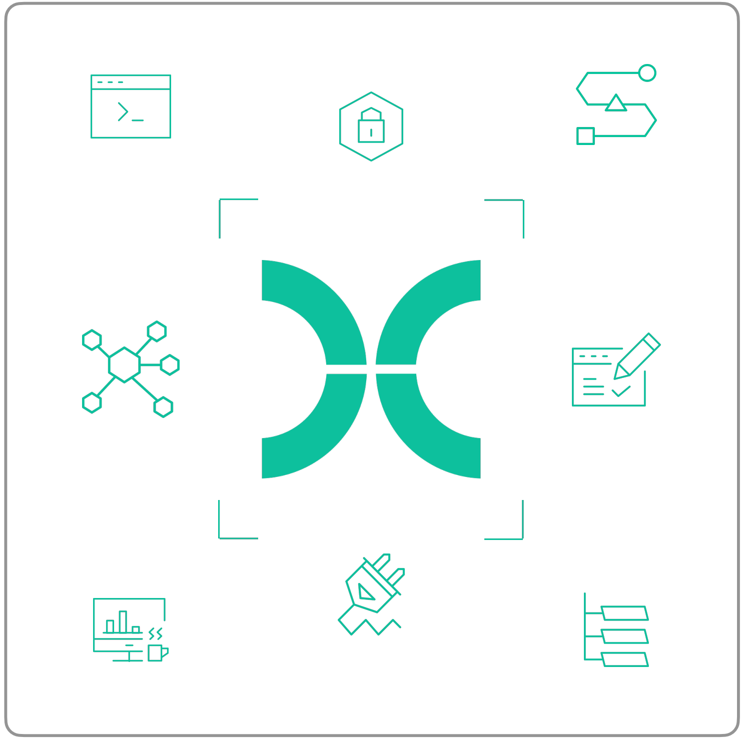


Phil Ewels · Seqera

We've rewritten the heart of Nextflow

Nextflow is no longer a "Groovy DSL"

A new language parser means we can develop the language for users



DEVELOPER EXPERIENCE

Nextflow Language Server & VS Code Extension

A modern IDE experience for Nextflow development

Inline error reporting

Warnings and errors highlighted as you type

DAG preview

Visualize your workflow graph as you work

Quick navigation

Rename objects, click to jump straight to includes

Code completion

Context-aware suggestions at your cursor

Auto formatting

Format on save, with pre-commit or in CI

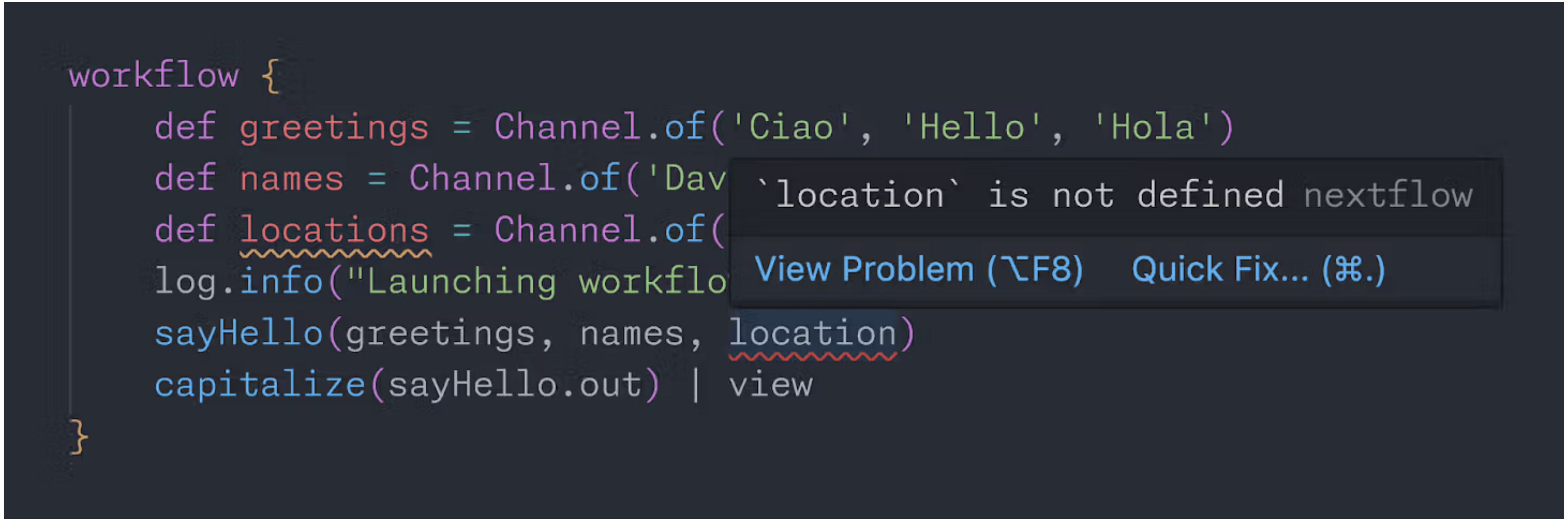
Compile time errors

Errors are detected at launch, rather than mid-run

DEVELOPER EXPERIENCE

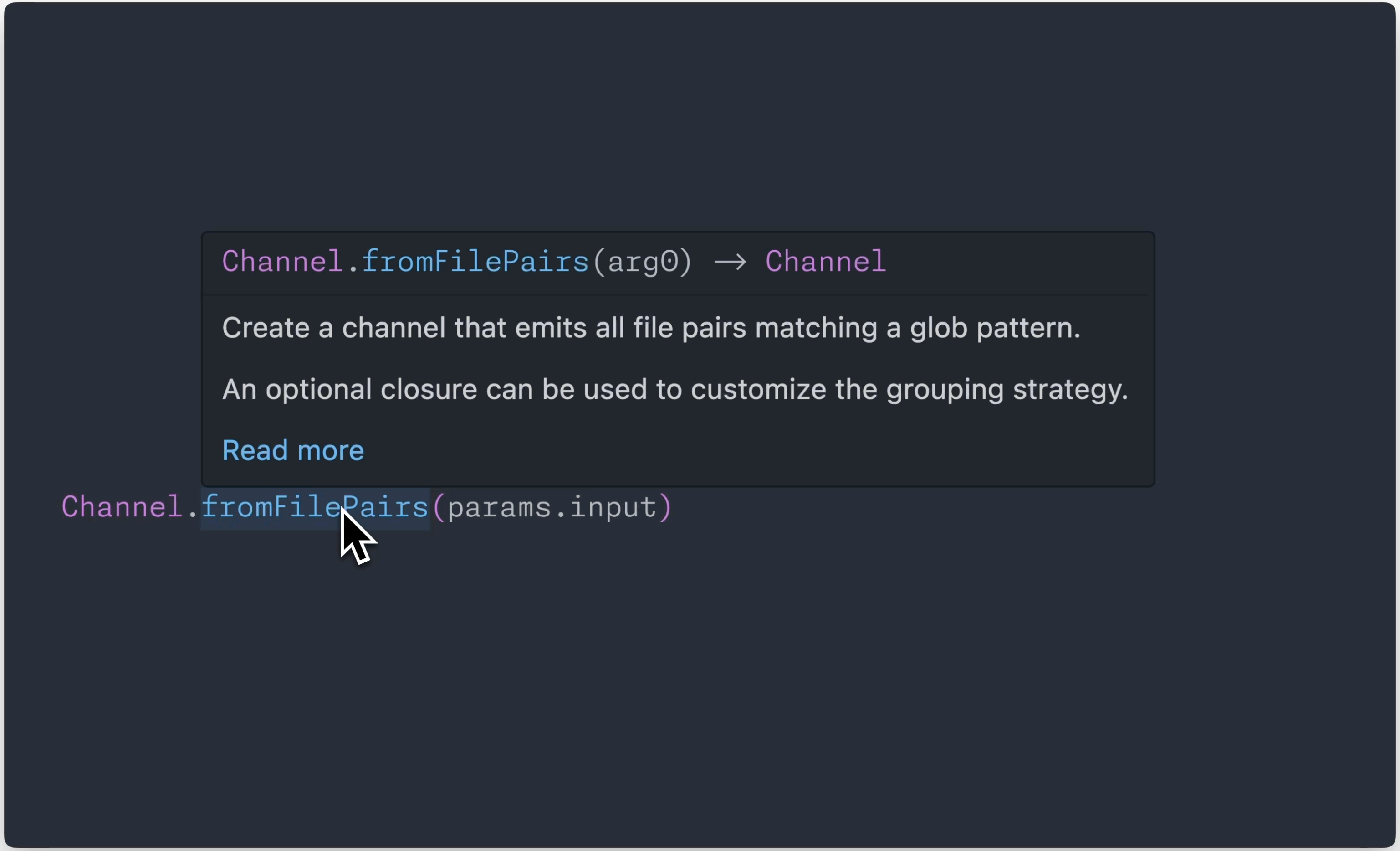
Nextflow Language Server & VS Code Extension

A modern IDE experience for Nextflow development



```
workflow {
    def greetings = Channel.of('Ciao', 'Hello', 'Hola')
    def names = Channel.of('Dav
    def locations = Channel.of(
    log.info("Launching workflo
    sayHello(greetings, names, location)
    capitalize(sayHello.out) | view
}
```

`location` is not defined nextflow



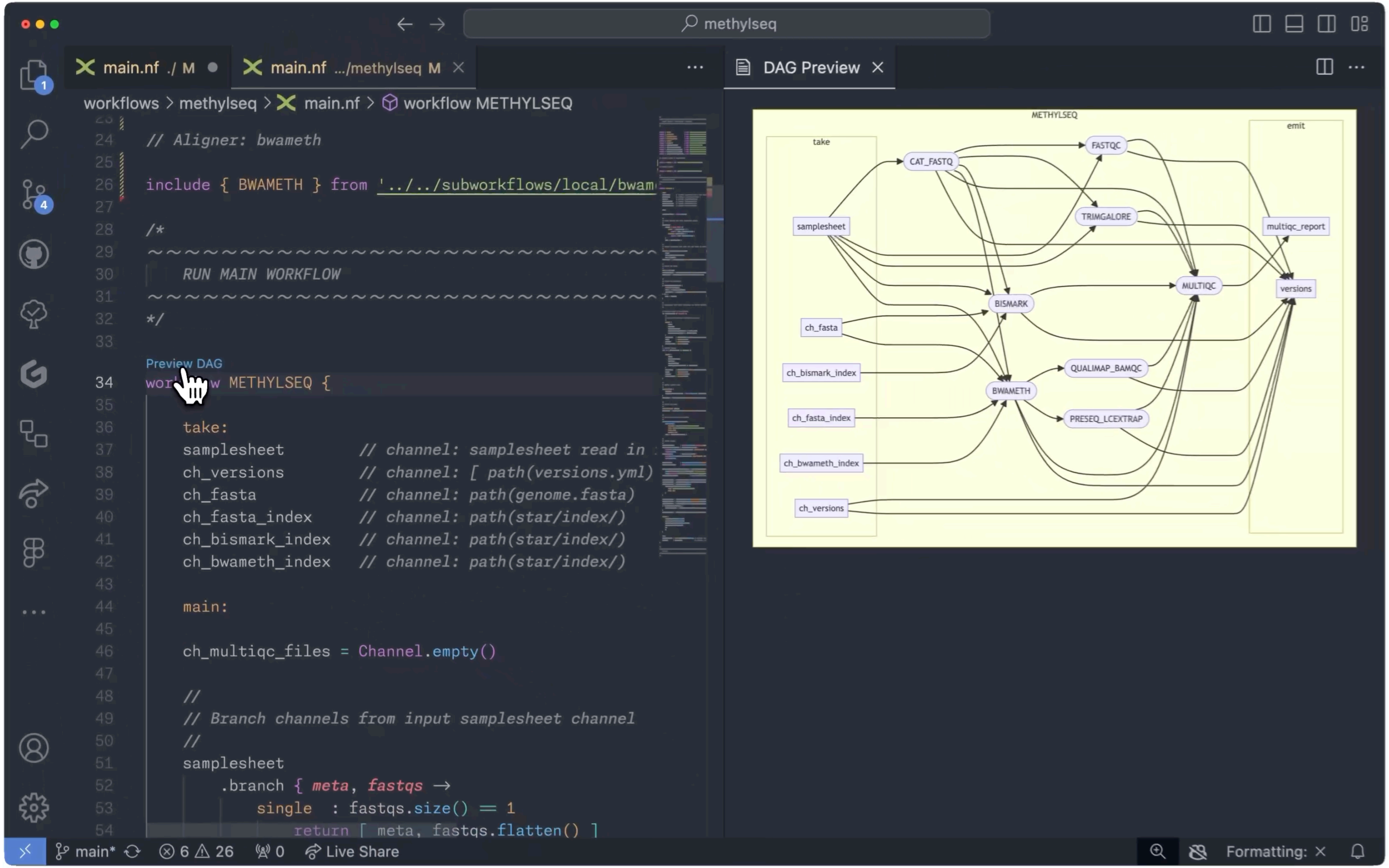
```
Channel.fromFilePairs(arg0) → Channel
```

Create a channel that emits all file pairs matching a glob pattern.

An optional closure can be used to customize the grouping strategy.

Read more

```
Channel.fromFilePairs(params.input)
```



```
// Aligner: bwameth

include { BWAMETH } from '../../subworkflows/local/bwam

/*
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    RUN MAIN WORKFLOW
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
*/

workflow METHYLSEQ {

    take:
    samplesheet        // channel: samplesheet read in
    ch_versions        // channel: [ path(versions.yml)
    ch_fasta           // channel: path(genome.fasta)
    ch_fasta_index     // channel: path(star/index/)
    ch_bismark_index   // channel: path(star/index/)
    ch_bwameth_index   // channel: path(star/index/)

    main:

    ch_multiqc_files = Channel.empty()

    //
    // Branch channels from input samplesheet channel
    //
    samplesheet
        .branch { meta, fastqs ->
            single  : fastqs.size() == 1
                return [ meta, fastqs.flatten() ]
```

NEXTFLOW 25.10

Static types & strict syntax

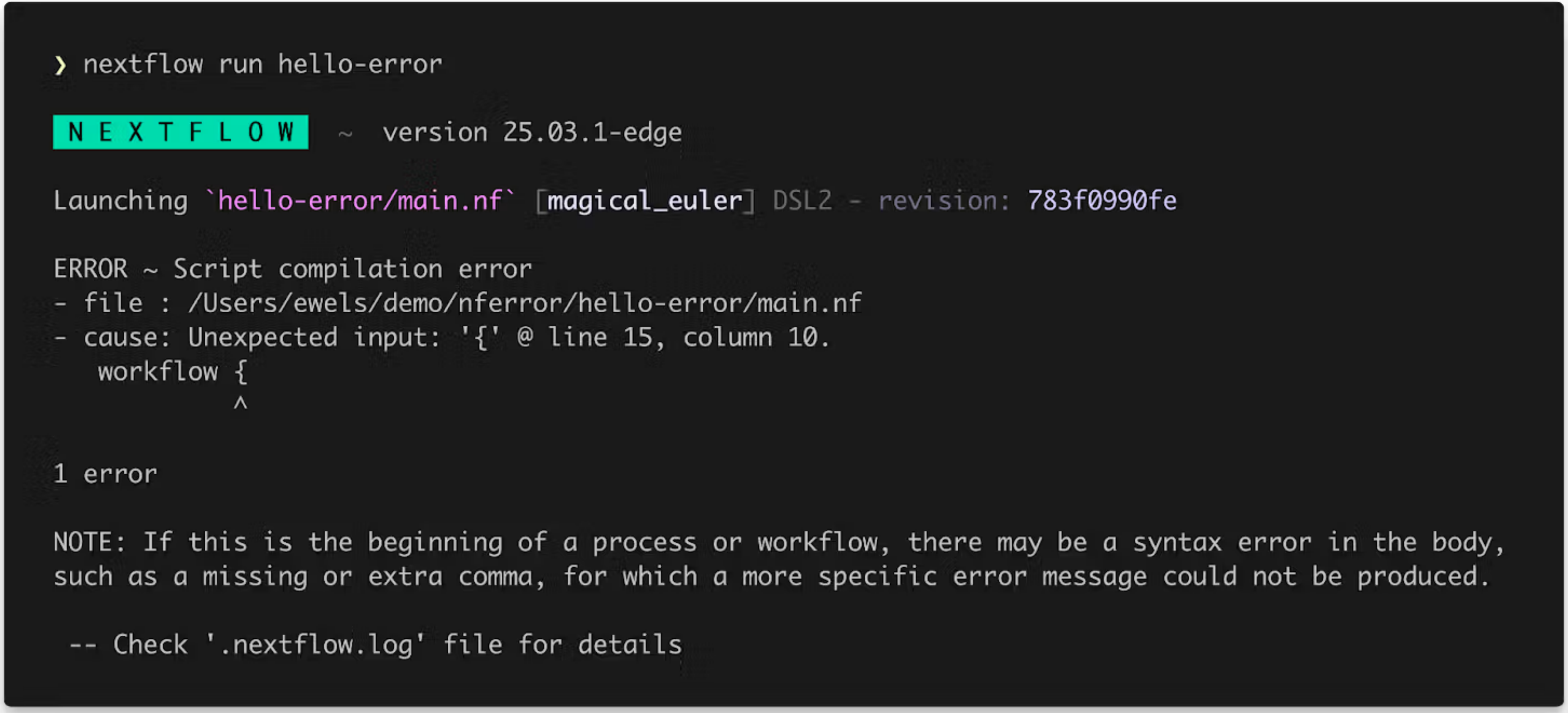
Nextflow's new native type system

Before

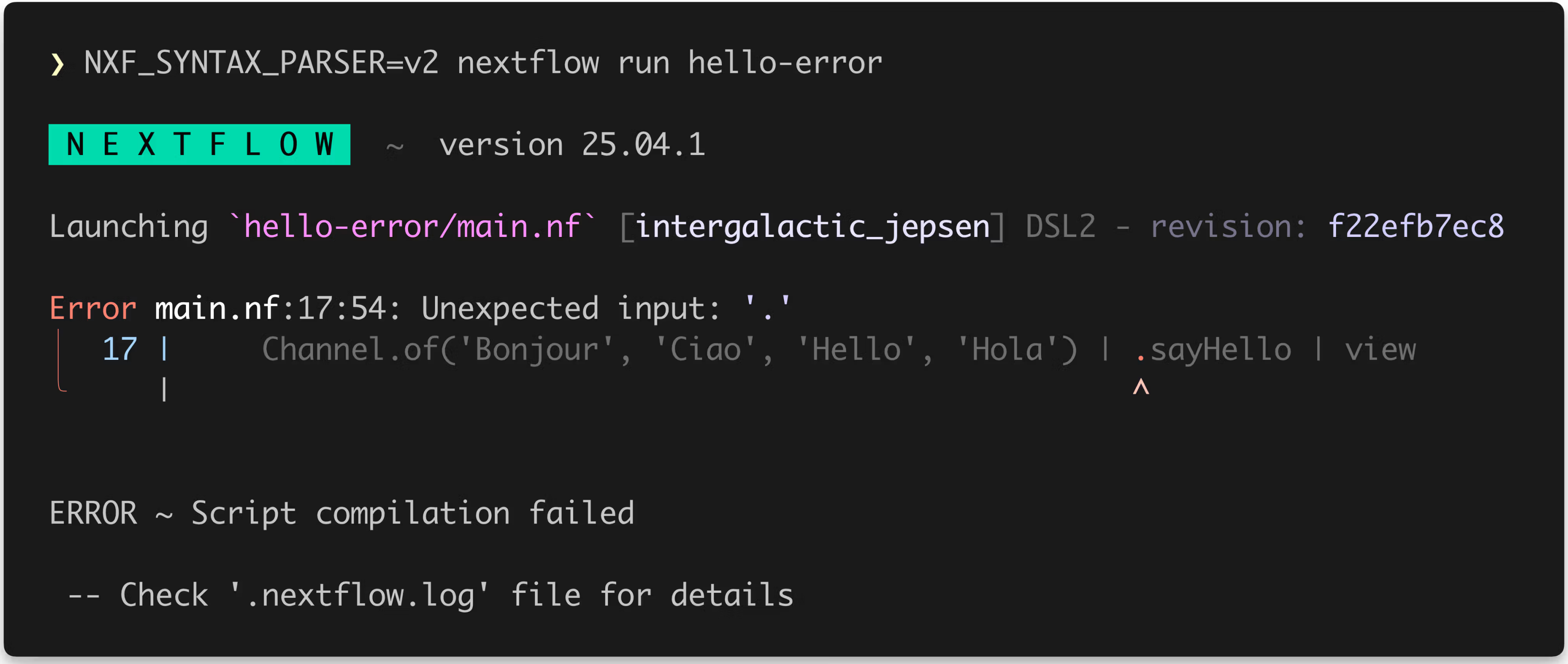
```
params.input = null

params.save_intermeds = false
params.threshold = 200
params.labels = 'one,two'
```

After

```
params {
  // No default value - required parameter
  input: Path

  save_intermeds: Boolean = false
  threshold: Integer = 200
  labels: String = 'one,two'
}
```



```
❯ nextflow run hello-error

 N E X T F L O W   ~  version 25.03.1-edge

Launching `hello-error/main.nf` [magical_euler] DSL2 - revision: 783f0990fe

ERROR ~ Script compilation error
- file : /Users/ewels/demo/nferror/hello-error/main.nf
- cause: Unexpected input: '{' @ line 15, column 10.
   workflow {
            ^

1 error

NOTE: If this is the beginning of a process or workflow, there may be a syntax error in the body,
such as a missing or extra comma, for which a more specific error message could not be produced.

 -- Check '.nextflow.log' file for details
```



```
❯ NXF_SYNTAX_PARSER=v2 nextflow run hello-error

 N E X T F L O W   ~  version 25.04.1

Launching `hello-error/main.nf` [intergalactic_jepsen] DSL2 - revision: f22efb7ec8

Error main.nf:17:54: Unexpected input: '.'
│  17 |     Channel.of('Bonjour', 'Ciao', 'Hello', 'Hola') | .sayHello | view
╰     |                                                      ^


ERROR ~ Script compilation failed

 -- Check '.nextflow.log' file for details
```

NEXTFLOW LINT

Check syntax ahead of time

```
❯ nextflow lint .
Error main.nf:21:1: Invalid include source: '/Users/ewels/GitHub/nextflow-io/rnaseq-nf/modules/dnaseq.nf'
│  21 | include { DNASEQ } from './modules/dnaseq'
╰     | ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Warn  main.nf:37:3: Variable was declared but not used
│  37 |   debug_variable = "Hello world"
╰     |   ^^^^^^^^^^^^^^

Nextflow linting complete!
 ❌ 1 file had 1 error
 ✅ 6 files had no errors
```

NEXTFLOW 25.04+

Workflow outputs & data lineage

Declarative output publishing and built-in provenance tracking

Workflow Outputs

Now out of preview in 25.10

→ Declare outputs in a publish: section

→ Assignment syntax for clear mapping

→ Only in entry workflow — cleaner design

→ Typed outputs with Channel annotations

Data Lineage

Built-in provenance (25.04+)

→ Enable with: `lineage.enabled = true`

→ Records every run, task, and output

→ Explore lineage from the CLI

→ Full audit trail for compliance

seqera

The complete platform for scientific data analysis

Built by the creators of Nextflow

SEQERA PLATFORM

Launch, manage & monitor at scale

Launchpad

One-click pipeline launches across any infrastructure. Pre-configured parameters and compute environments.

Real-time monitoring

Live task tracking, resource metrics, cost estimation, and automatic failure notifications.

Governance & compliance

Complete audit trails, versioning, and role-based access control for regulated environments.

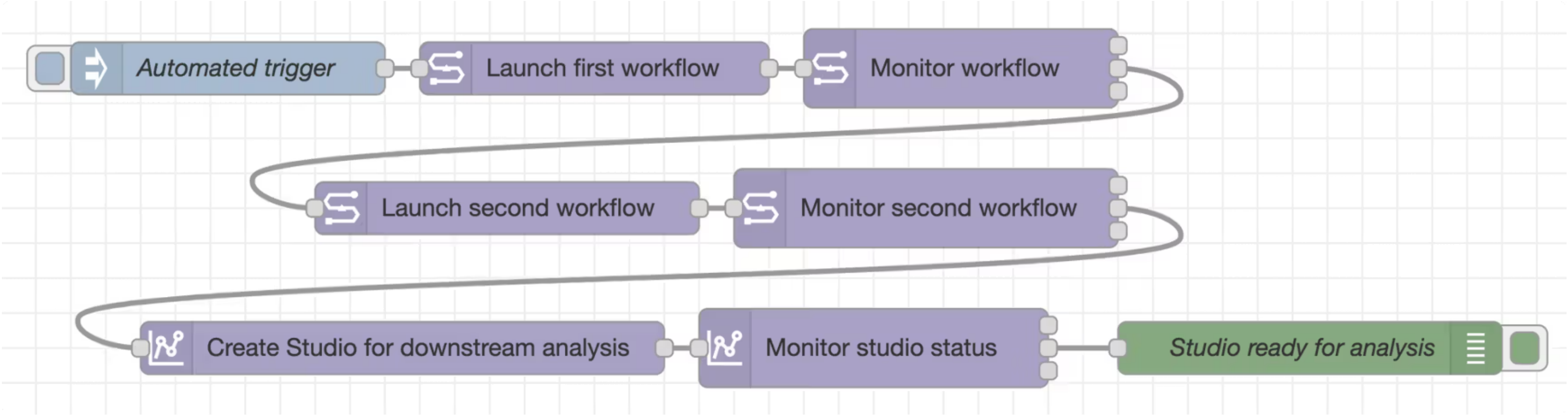
Multi-cloud

AWS, Google Cloud, Azure, and on-prem HPC. All from a single interface.

SEQERA PLATFORM

Node-RED + Seqera

Seqera Platform is your control plane for Nextflow



`https://seqeralabs.github.io/node-red-seqera/`

FUSION FILE SYSTEM

Cloud-native storage for Nextflow

fusion

Zero-copy I/O

POSIX interface to cloud storage. No "stage-in" copy.

Fusion Snapshots

Freeze task state on interruption, resume on new instance in < 2m

Multi-cloud

Works with all major cloud providers

SEQERA PLATFORM

And so much more..

The best place to run Nextflow

Seqera Studios

Interactive analysis environments on your compute

Wave: On-demand container provisioning

Builds optimized containers on the fly

Seqera Compute

Managed, optimized infrastructure

Seqera AI

AI-powered pipeline development

Seqera MCP, CLI, API, Terraform

Control your infrastructure however you want

Data Explorer

Browse your cloud data within Seqera

Join the community

`nextflow.io`

Documentation, examples, resources

`seqera.io/podcasts/`

The Nextflow Podcast

`training.nextflow.io`

Free, self-paced courses

`seqera.io/blog/`

The Open Science Blog

`community.seqera.io`

Community forum

`community.seqera.io`

Community forum

HEADER

Upcoming events

nf-core/hackathon

Online / distributed hackathon.
March 11-13

Nextflow Summit BOS

Boston Summit with talks,
training and hackathon. April 28

Nextflow Summit BCN

Online summit talks. Training +
hackathon in Barcelona. Oct '26

Plus many more: https://seqera.io/events/upcoming/

Thank you

Questions & Discussion

Phil Ewels

Senior Product Manager, Seqera

phil.ewels@seqera.io

nextflow.io · nf-co.re · seqera.io

nextflow nf-core seqera