Webinars series

Live demonstrations on the e-infrastructure deployment and the risk assessment case studies

	Topic	Date & Time
Past events	Introduction sessions to the OpenRiskNet e-infrastructure	See Webinar recordings: • Session 1 (24 Sep 2018) • Session 2 (27 Sept 2018) • Session 3 (4 Oct 2018) • Session 4 (30 Oct 2018)
	Learn how to deploy the OpenRiskNet virtual research environment	See Webinar recordings (25 Feb 2019)
	Demonstration on data curation and creation of pre-reasoned datasets in the OpenRiskNet framework	Monday, **18 March 2019** 16:00 CET
	Identification and linking of data related to AOPWiki (an OpenRiskNet case study)	Tuesday, **26 March 2019** 17:00 CET
	Semantic annotation	Monday, **1 April 2019** 16:00 CET
	The Adverse Outcome Pathway Database (AOP-DB)	Monday, **8 April 2019** 16:00 CET
Current Event	Nextflow and TGX case study	Monday, **27 May 2019**

https://openrisknet.org/events/

OpenRiskNet

RISK ASSESSMENT E-INFRASTRUCTURE

Nextflow for toxicogenomics-based predictions on the OpenRiskNet Virtual Research Infrastructure

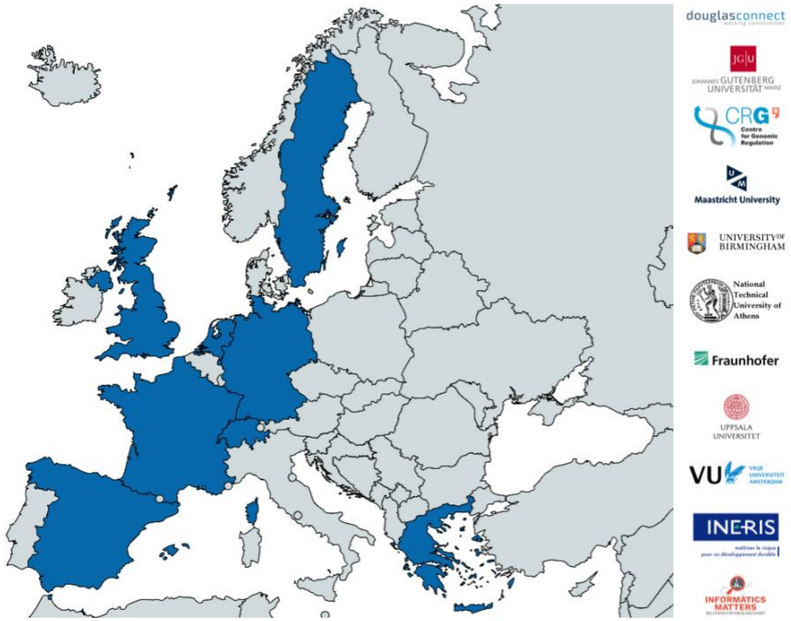
Evan Floden (Centre for Genomic Regulation)

Webinar - 27 May 2019

OpenRiskNet: Open e-Infrastructure to Support Data Sharing, Knowledge Integration and *in silico* Analysis and Modelling in Risk Assessment
Project Number 731075

About the project

OpenRiskNet is a 3-year EU Horizon 2020 project with the main objective to develop an open e-infrastructure providing resources and services to a variety of communities requiring risk assessment, including chemicals, cosmetic ingredients, therapeutic agents and nanomaterials.



Main components:

- ➔ **Case-study-driven development** - examples of tools to be integrated are selected based on the case study needs. More information: https://openrisknet.org/e-infrastructure/development/case-studies/
- ➔ Solutions for all areas by **integrating existing tools** from consortium and associated partners (via the implementation challenge)
- ➔ **Integrated approach** combining experimental data (*in vivo, in vitro, in chemico*) with analysis, modelling and simulation tools into risk assessment workflows

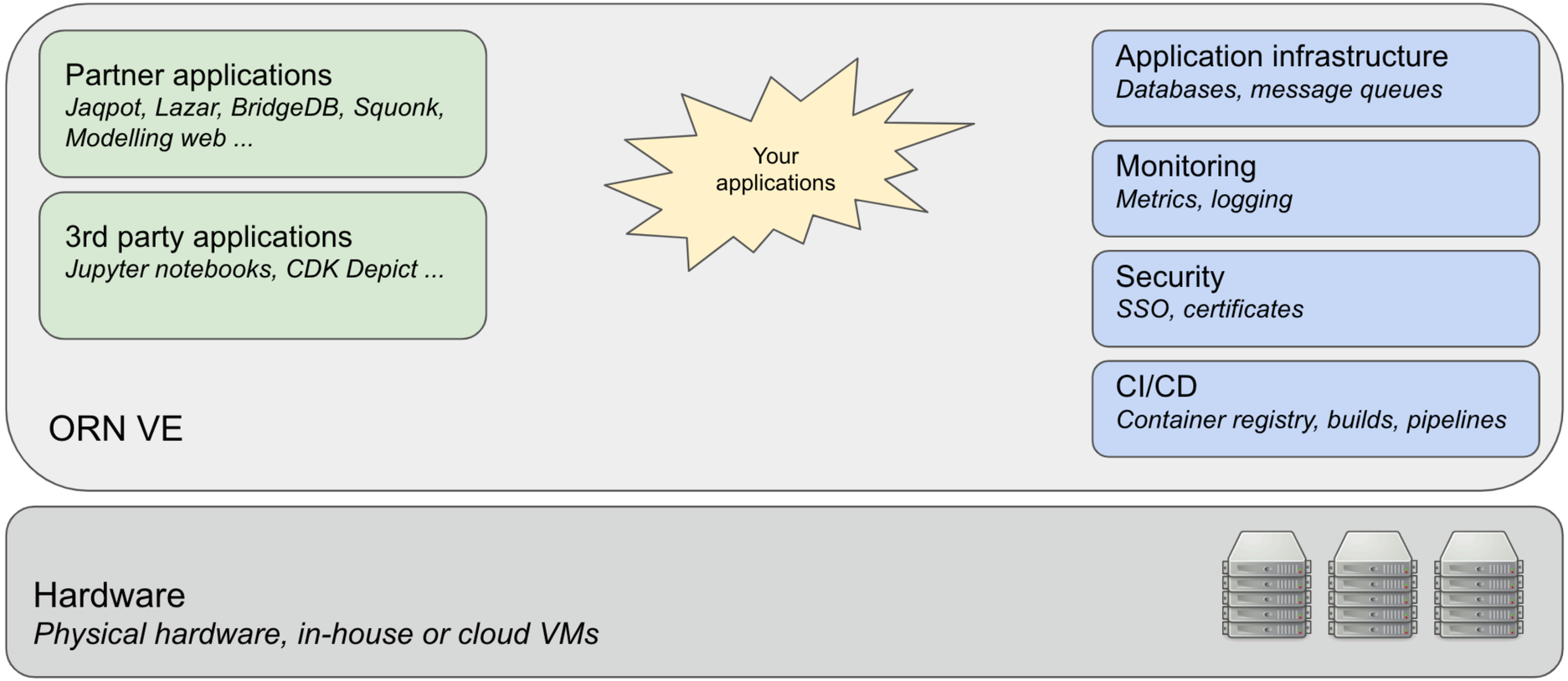
Webinars series

Live demonstrations on the e-infrastructure deployment and the risk assessment case studies

	Topic	Date & Time
Past events	Introduction sessions to the OpenRiskNet e-infrastructure	See Webinar recordings: • Session 1 (24 Sep 2018) • Session 2 (27 Sept 2018) • Session 3 (4 Oct 2018) • Session 4 (30 Oct 2018)
	Learn how to deploy the OpenRiskNet virtual research environment	See Webinar recordings (25 Feb 2019)
	Demonstration on data curation and creation of pre-reasoned datasets in the OpenRiskNet framework	Monday, **18 March 2019** 16:00 CET
	Identification and linking of data related to AOPWiki (an OpenRiskNet case study)	Tuesday, **26 March 2019** 17:00 CET
	Semantic annotation	Monday, **1 April 2019** 16:00 CET
	The Adverse Outcome Pathway Database (AOP-DB)	Monday, **8 April 2019** 16:00 CET
Current Event	Nextflow and TGX case study	Monday, **27 May 2019**

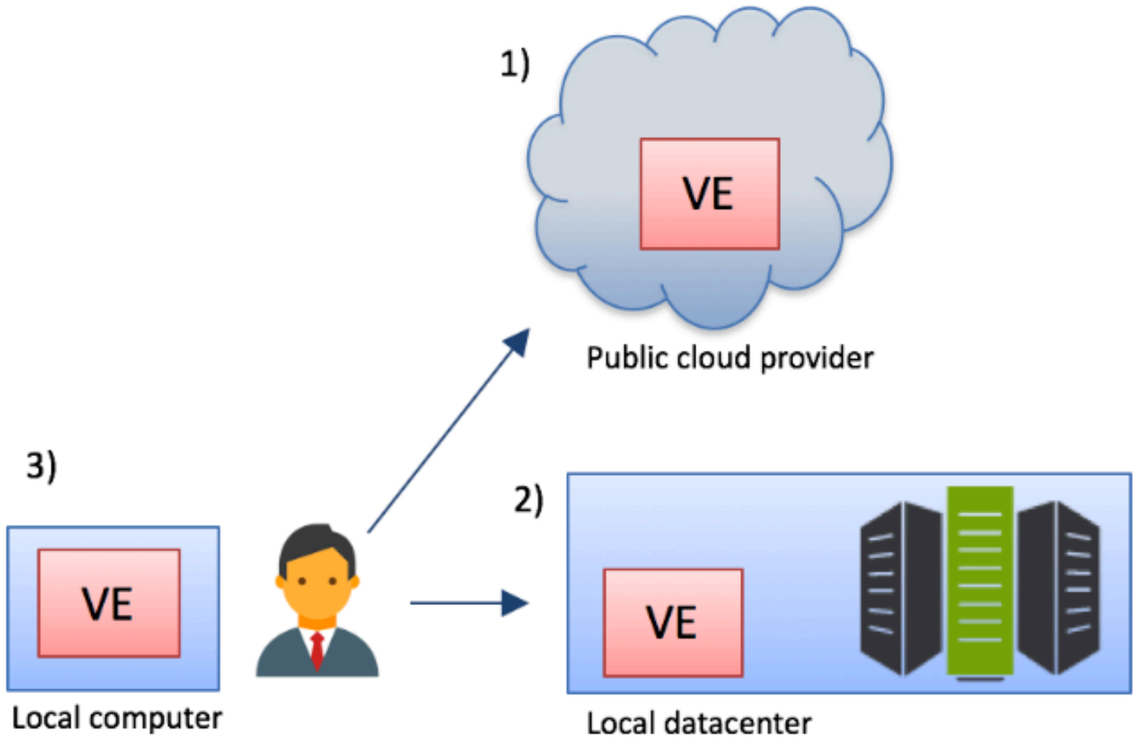
https://openrisknet.org/events/

The OpenRiskNet VE



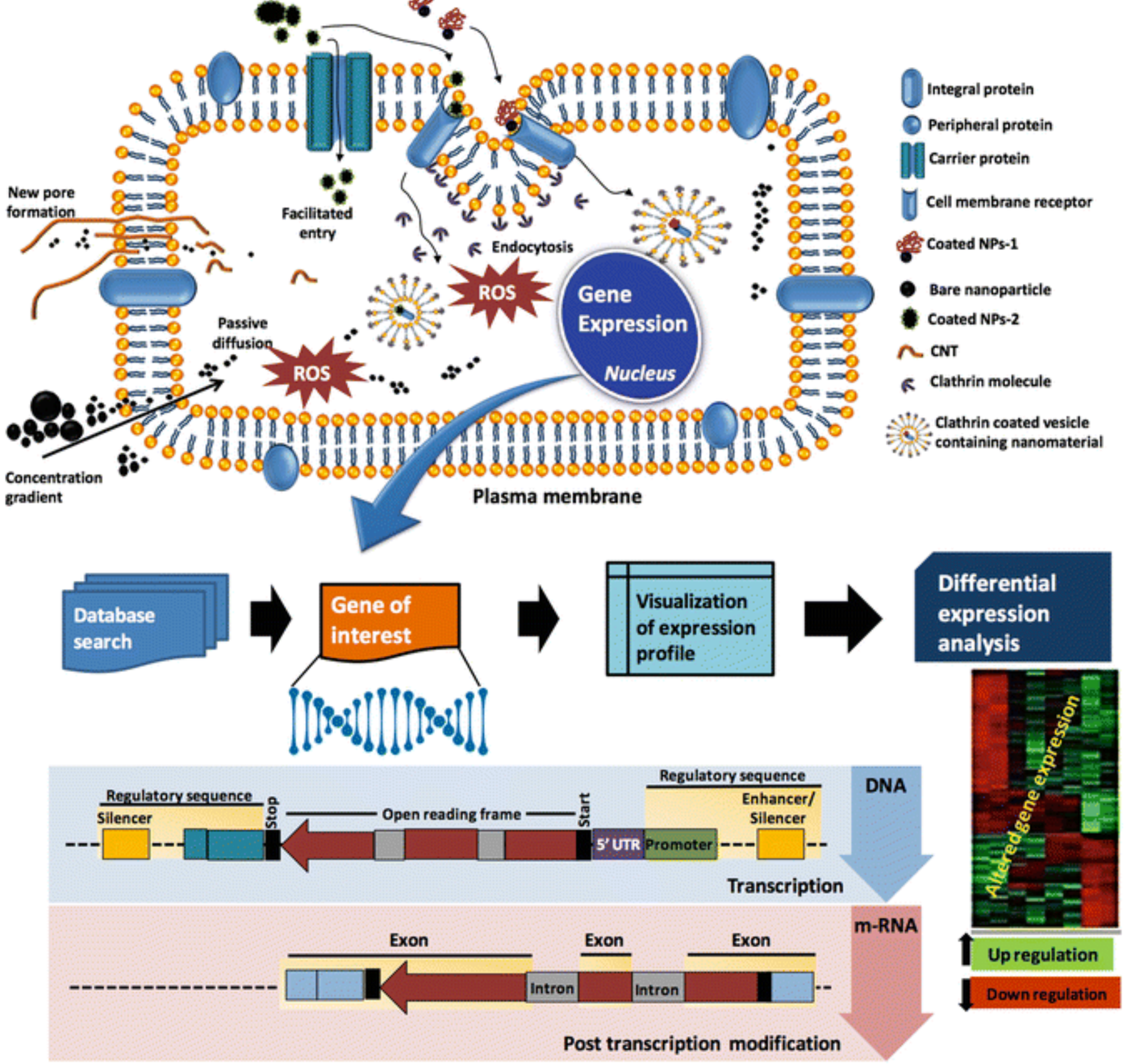
https://prod.openrisknet.org/

The OpenRiskNet VE

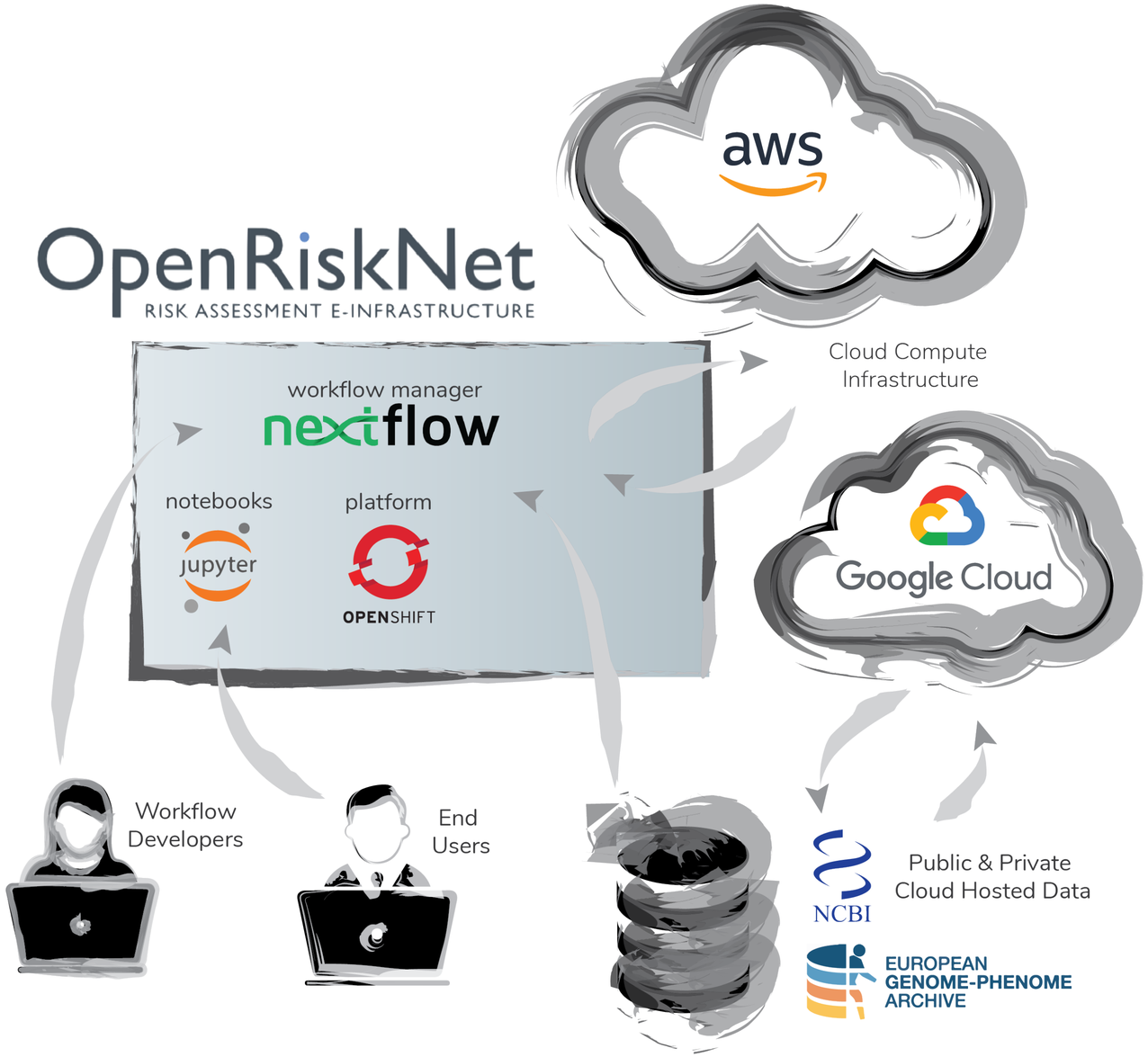


https://github.com/OpenRiskNet/home/wiki

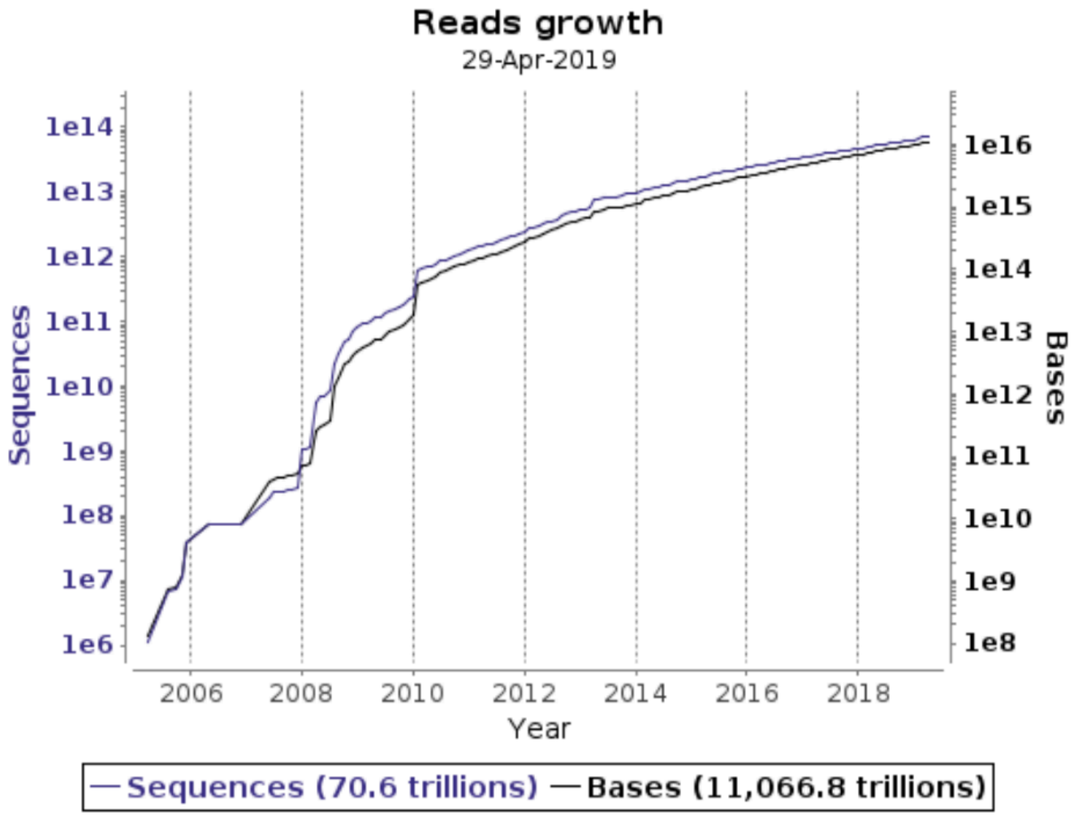
What is Toxicogenomics?

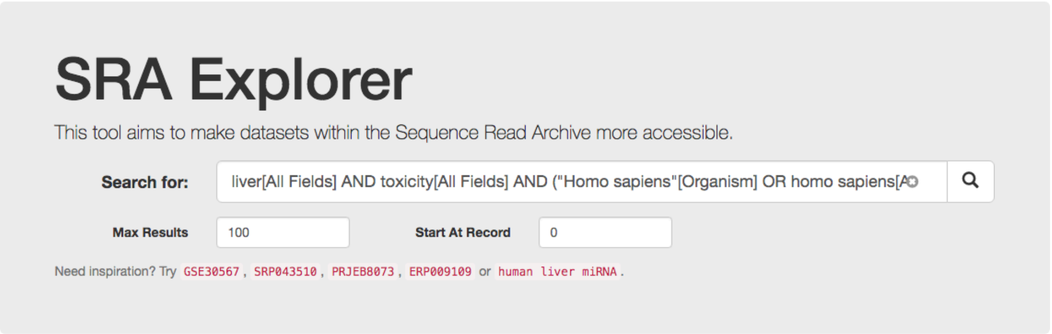


External Compute Resources



External Data Resources





https://ewels.github.io/sra-explorer/

AWS iGenomes

Common reference genomes hosted on AWS S3

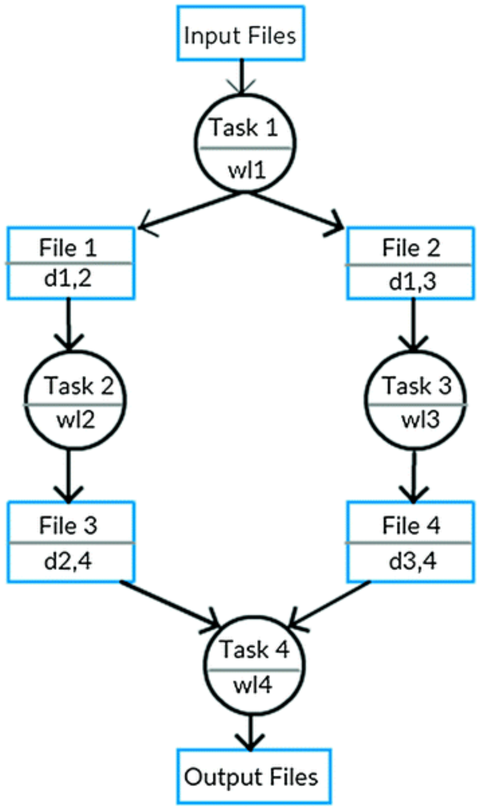
https://ewels.github.io/AWS-iGenomes/

Portable Computation

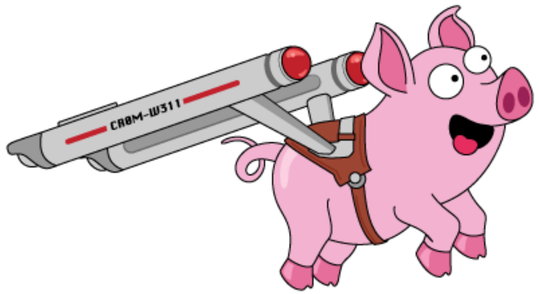
Virtual Infrastructure

Application

Scientific Workflow Managers



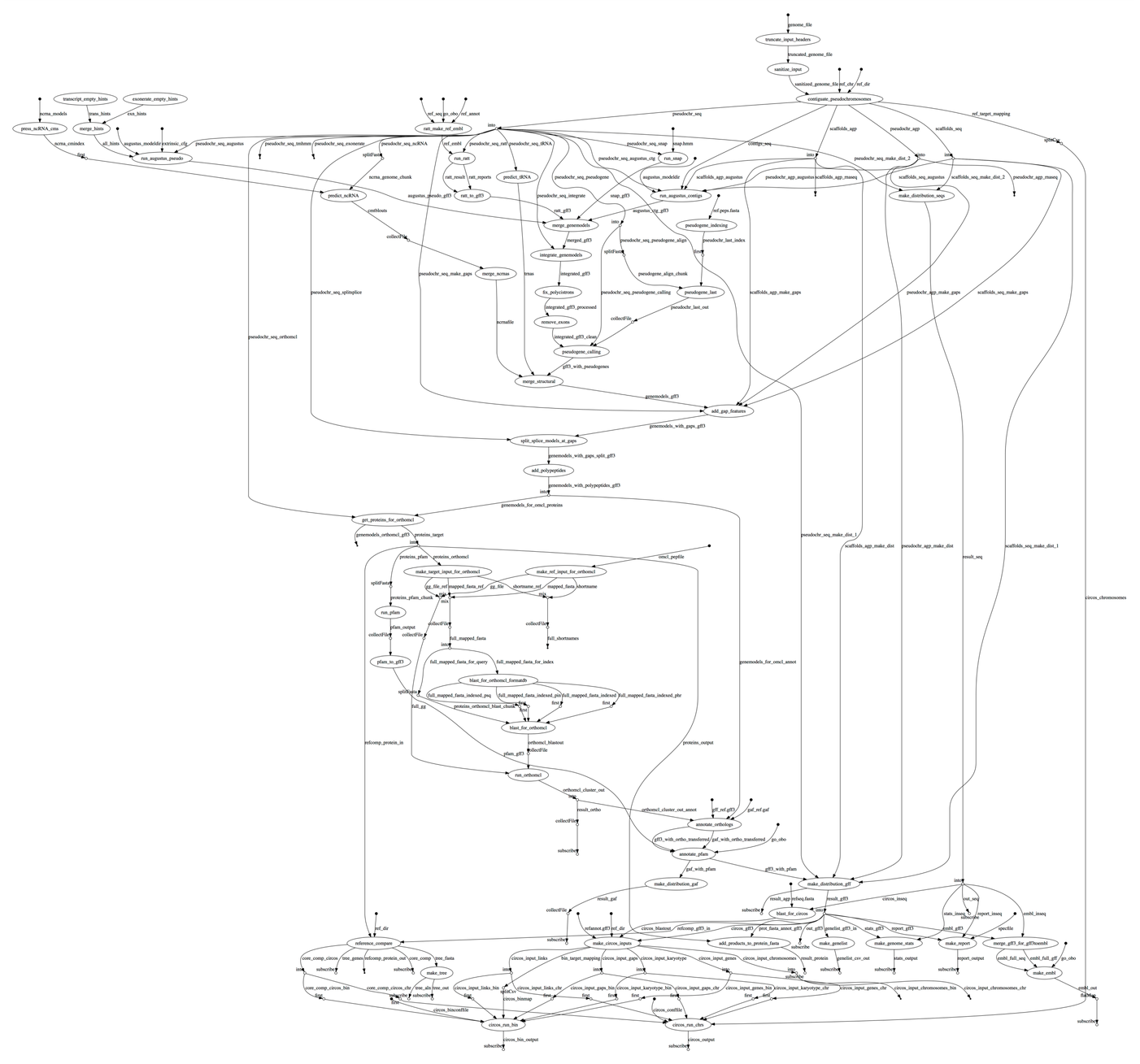
Galaxy



nextflow

Toxicogenomic workflows

- Data analysis applications performs computation to generate information from large genomic datasets (resource requirements)
- Embarrassingly parallelisation, can spawn 100s-100k jobs over distributed cluster
- Mash-up of many different tools and scripts (dependancies!)
- Complex dependency trees and configuration → very fragile ecosystem



Steinbiss et al., *Companion parassite genome annotation pipeline*, DOI: 10.1093/nar/gkw292

a lot of moving parts

70 tasks

55 external scripts

39 software tools & libraries

OPEN ACCESS Freely available online

PLOS | ONE

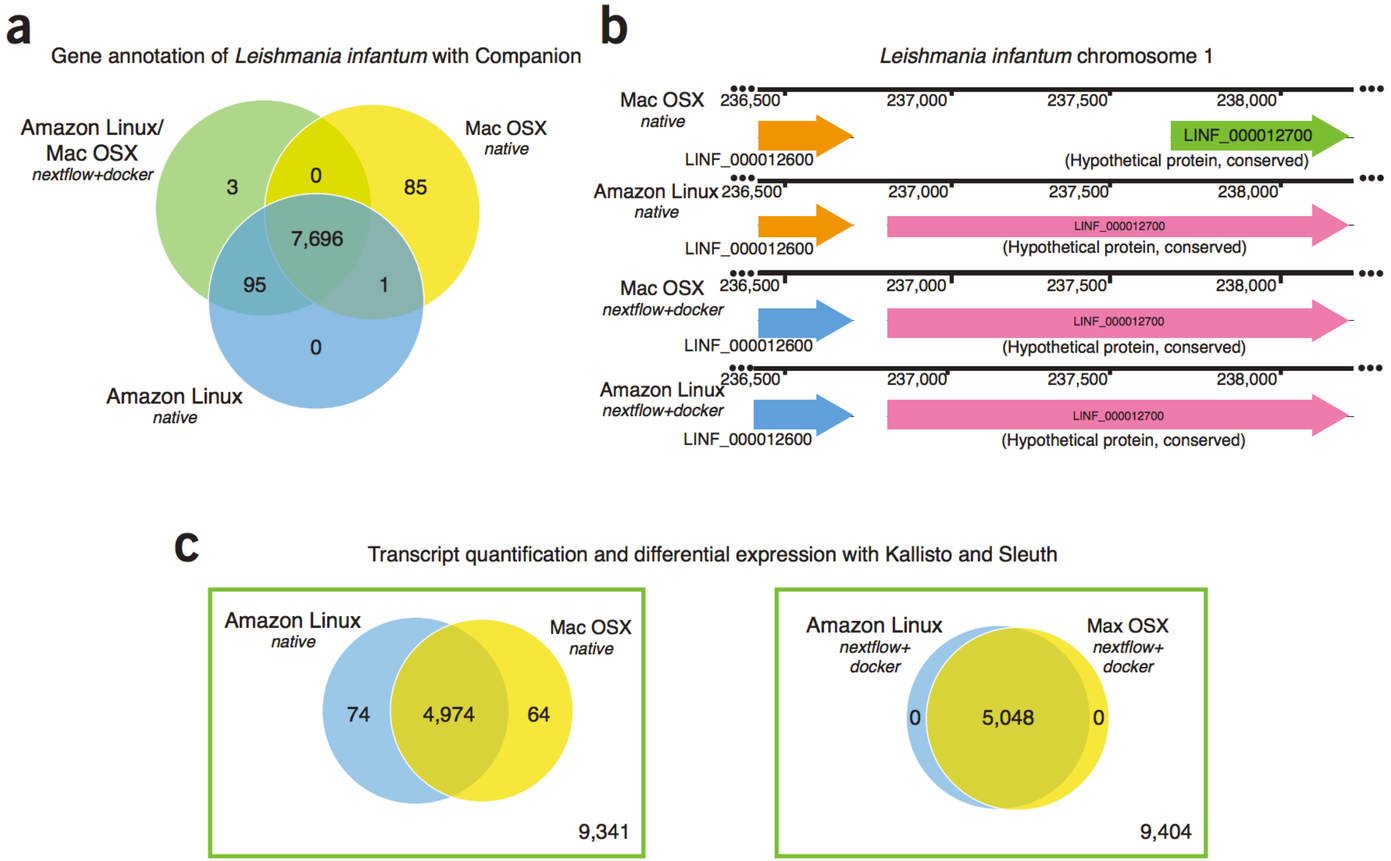
Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome

Daniel Garijo[1], Sarah Kinnings[2], Li Xie[3], Lei Xie[4], Yinliang Zhang[5], Philip E. Bourne[3]*, Yolanda Gil[6]*

1 Ontology Engineering Group, Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain, **2** Department of Chemistry and Biochemistry, University of California San Diego, La Jolla, California, United States of America, **3** Skaggs School of Pharmacy and Pharmaceutical Sciences, University of California San Diego, La Jolla, California, United States of America, **4** Department of Computer Science, Hunter College, The City University of New York, New York, New York, United States of America, **5** School of Life Sciences, University of Science and Technology of China, Hefei, Anhui, China, **6** Information Sciences Institute and Department of Computer Science, University of Southern California, LosAngeles, California, United States of America

To reproduce the result of a typical computational biology paper requires 280 hours.

≈ 1.7 months!



VOLUME 35 NUMBER 4 APRIL 2017 **NATURE BIOTECHNOLOGY**

* Di Tommaso P, et al., *Nextflow enables computational reproducibility*, Nature Biotech, 2017

Comparison of the Companion pipeline annotation of *Leishmania infantum* genome executed across different platforms *

Platform	Amazon Linux	Debian Linux	Mac OSX
Number of chromosomes	36	36	36
Overall length (bp)	32,032,223	32,032,223	32,032,223
Number of genes	7,781	7,783	7,771
Gene density	236.64	236.64	236.32
Number of coding genes	7,580	7,580	7570
Average coding length (bp)	1,764	1,764	1,762
Number of genes with multiple CDS	113	113	111
Number of genes with known function	4,147	4,147	4,142
Number of t-RNAs	88	90	88

* Di Tommaso P, et al., *Nextflow enables computational reproducibility*, Nature Biotech, 2017

challenges for risk assessment entering into the omics era

Reproducibility

Portability

Scalability

Usability

Traceability

PUSH-THE-BUTTON PIPELINES

The nextflow fundamentals for scaleable genomic workflows

code

orchestration

dependencies

deployment

sharing & reproducibility

how to achieve this?

- Fast prototyping ⇒ custom DSL that enables tasks composition, simplifies most use cases + general purpose programming lang. for corner cases
- Easy parallelisation ⇒ declarative reactive programming model based on dataflow paradigm, implicit portable parallelism
- Self-contained ⇒ functional approach, a task execution is idempotent ie. cannot modify the state of other tasks + isolate dependencies with containers
- Portable deployments ⇒ executor abstraction layer + deployment configuration from implementation logic

task example

```
bwa mem reference.fa sample.fq \
      | samtools sort -o sample.bam
```

task example

```
process align_sample {

  input:
  file 'reference.fa' from genome_ch
  file 'sample.fq' from reads_ch

  output:
  file 'sample.bam' into bam_ch

  script:
  """
  bwa mem reference.fa sample.fq \
        | samtools sort -o sample.bam
  """

}
```

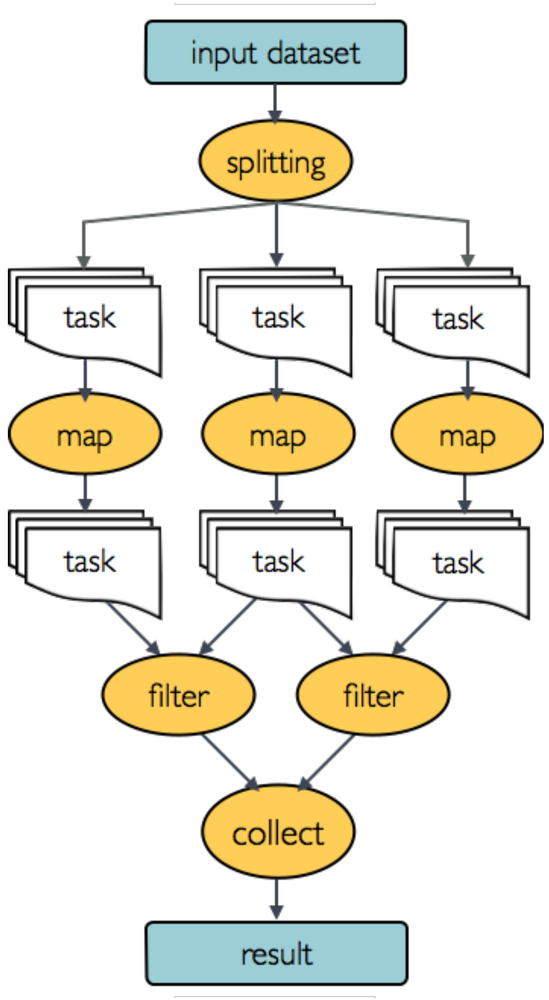
tasks composition

```
process align_sample {

    input:
    file 'reference.fa' from genome_ch
    file 'sample.fq' from reads_ch

    output:
    file 'sample.bam' into bam_ch


    script:
    """
    bwa mem reference.fa sample.fq \
          | samtools sort -o sample.bam
    """

}
```

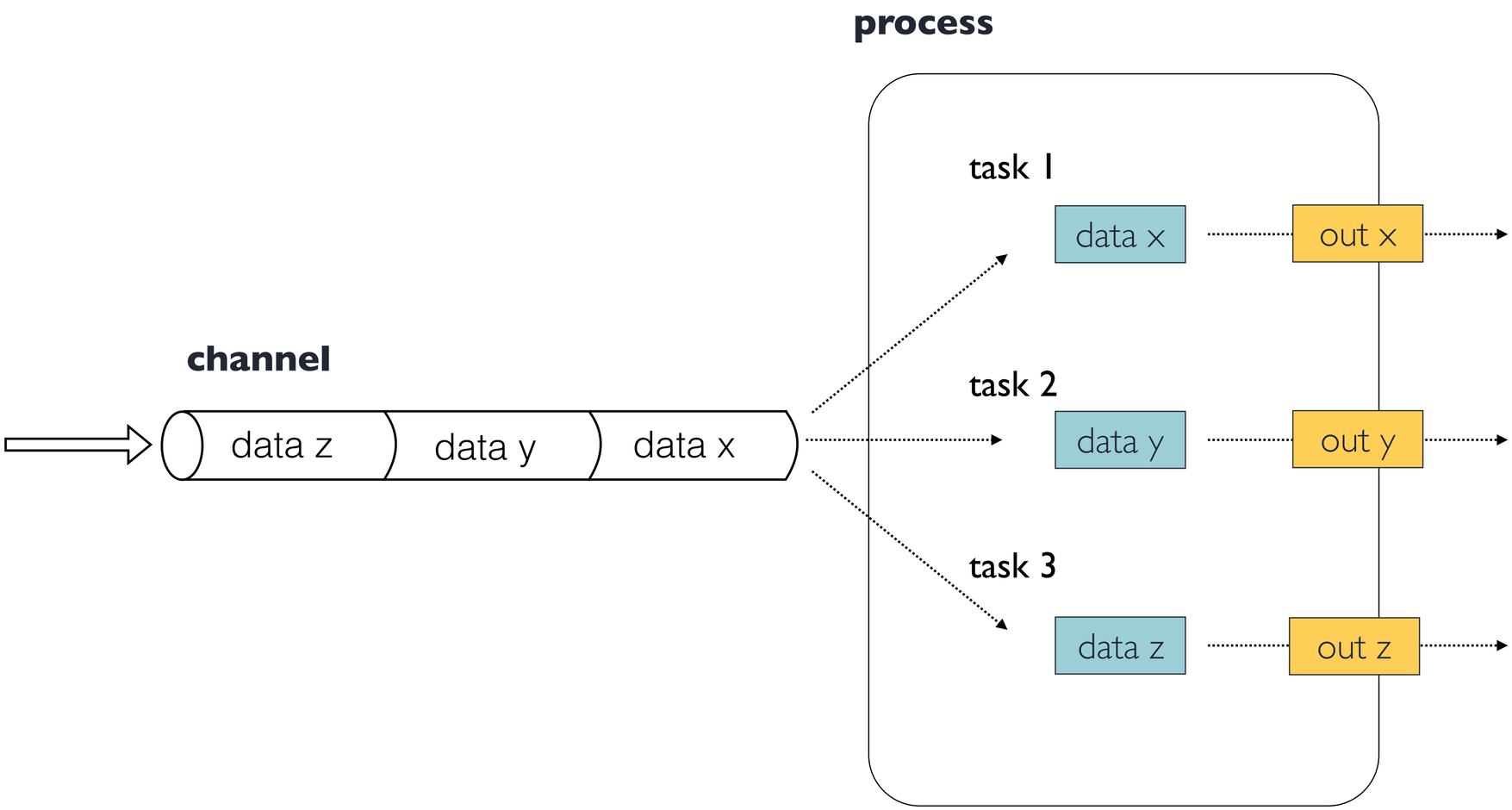
```
process index_sample {

    input:
    file 'sample.bam' from bam_ch

    output:
    file 'sample.bai' into bai_ch


    script:
    """
    samtools index sample.bam
    """
}
```

dataflow programming model

- Declarative computational model for parallel process executions
- Processes wait for data, when an input set is ready the process is executed
- They communicate by using dataflow variables i.e. async FIFO queues called channels
- Parallelisation and tasks dependencies are implicitly defined by process in/out declarations



How parallelisation works



how parallelisation works

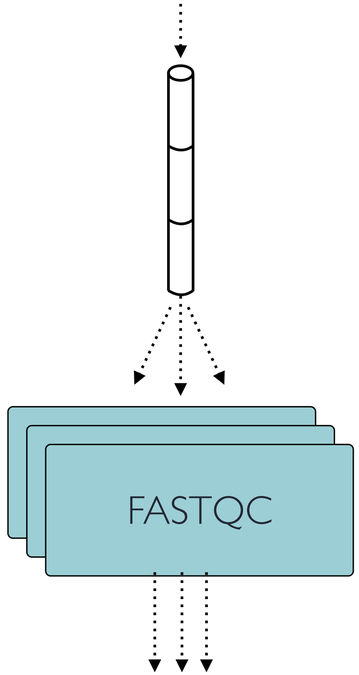
```
samples_ch = Channel.fromPath('data/sample.fastq')


process FASTQC {

   input:
     file reads from samples_ch

   output:
     file 'fastqc_logs' into fastqc_ch

   script:
   """
   mkdir fastqc_logs
   fastqc -o fastqc_logs -f fastq -q ${reads}
   """
}
```

how parallelisation works

```
samples_ch = Channel.fromPath('data/*.fastq')

process FASTQC {

   input:
     file reads from samples_ch

   output:
     file 'fastqc_logs' into fastqc_ch

   script:
   """
   mkdir fastqc_logs
   fastqc -o fastqc_logs -f fastq -q ${reads}
   """
}
```

implicit parallelism

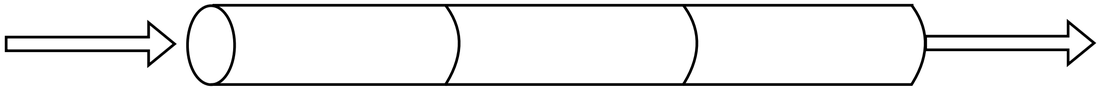
```
Channel.fromPath("data/*.fastq")
```



handling file pairs

```
Channel.fromFilePairs("*_{1,2}.fq")
```

```
gut_1.fq
gut_2.fq

liver_1.fq
liver_2.fq

lung_1.fq
lung_2.fq
```



```
( gut, [gut_1.fq, gut_2.fq] )

( lung, [lung_1.fq, lung_2.fq] )

( liver, [liver_1.fq, liver_2.fq] )
```

basic example

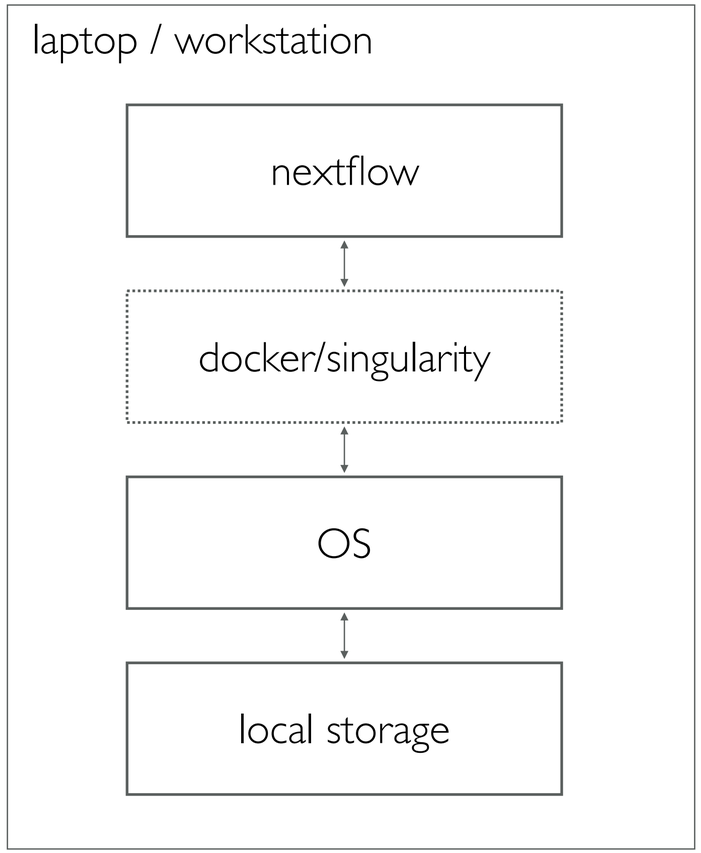
```
( gut, [gut_1.fq, gut_2.fq] )

( lung, [lung_1.fq, lung_2.fq] )

( liver, [liver_1.fq, liver_2.fq] )
```

```
process FASTQC {

   input:
     set pair_id, file(reads) from samples_ch

   output:
     file 'fastqc_logs' into fastqc_ch

   """
    mkdir fastqc_logs
    fastqc -o fastqc_logs -f fastq -q ${reads}
   """
}
```

deployment scenarios

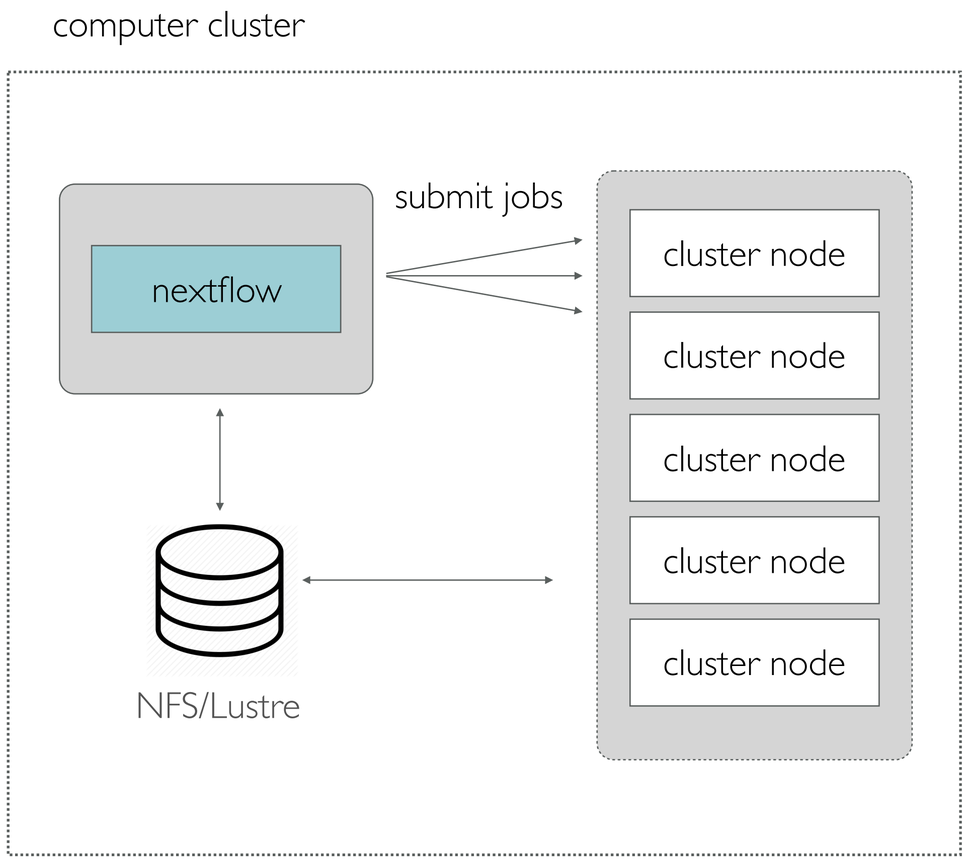
local execution

- Common development scenario
- Dependencies can be managed using a container runtime
- Parallelisations is managed spawning posix processes
- Can scale vertically using fat server / shared mem. machine



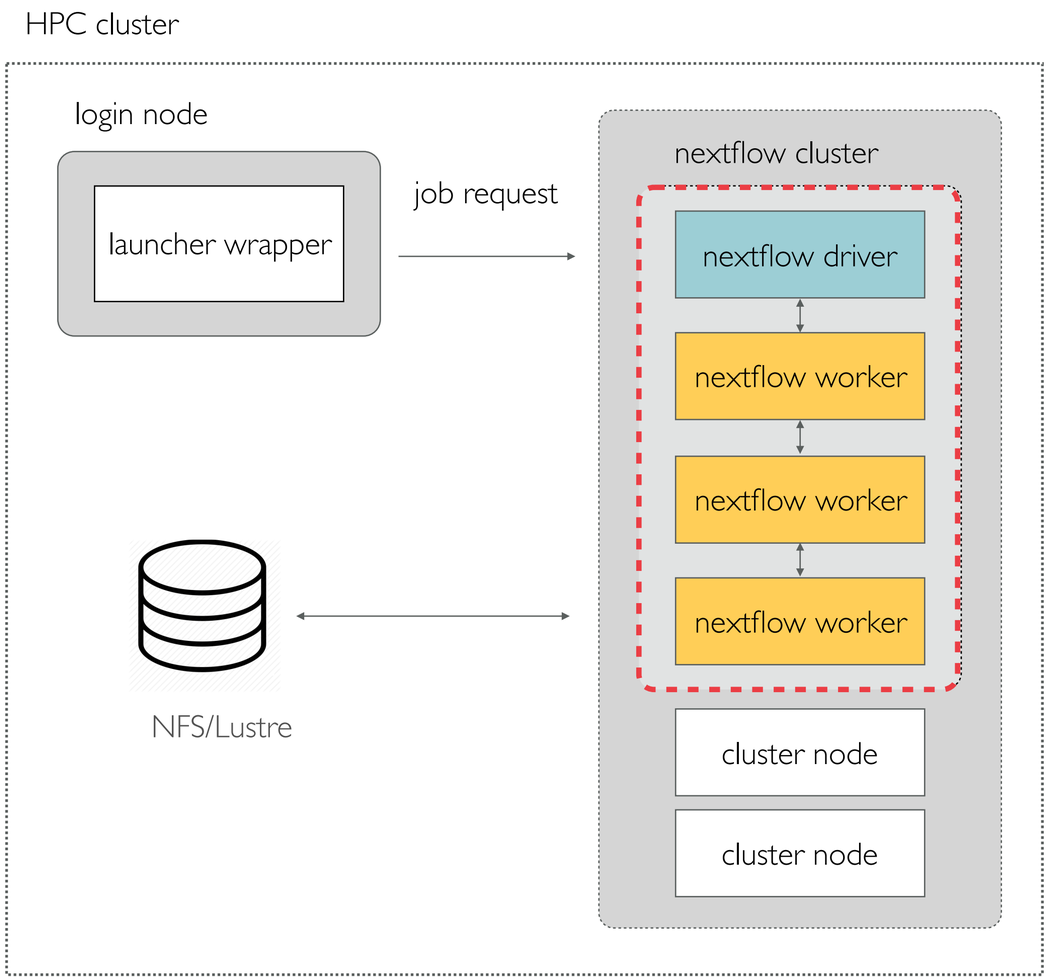
centralised orchestration

- Nextflow orchestrates workflow execution submitting jobs to a compute cluster eg. SLURM
- It can run in the head node or a compute node
- Requires a shared storage to exchange data between tasks
- Ideal for corse-grained parallelisms

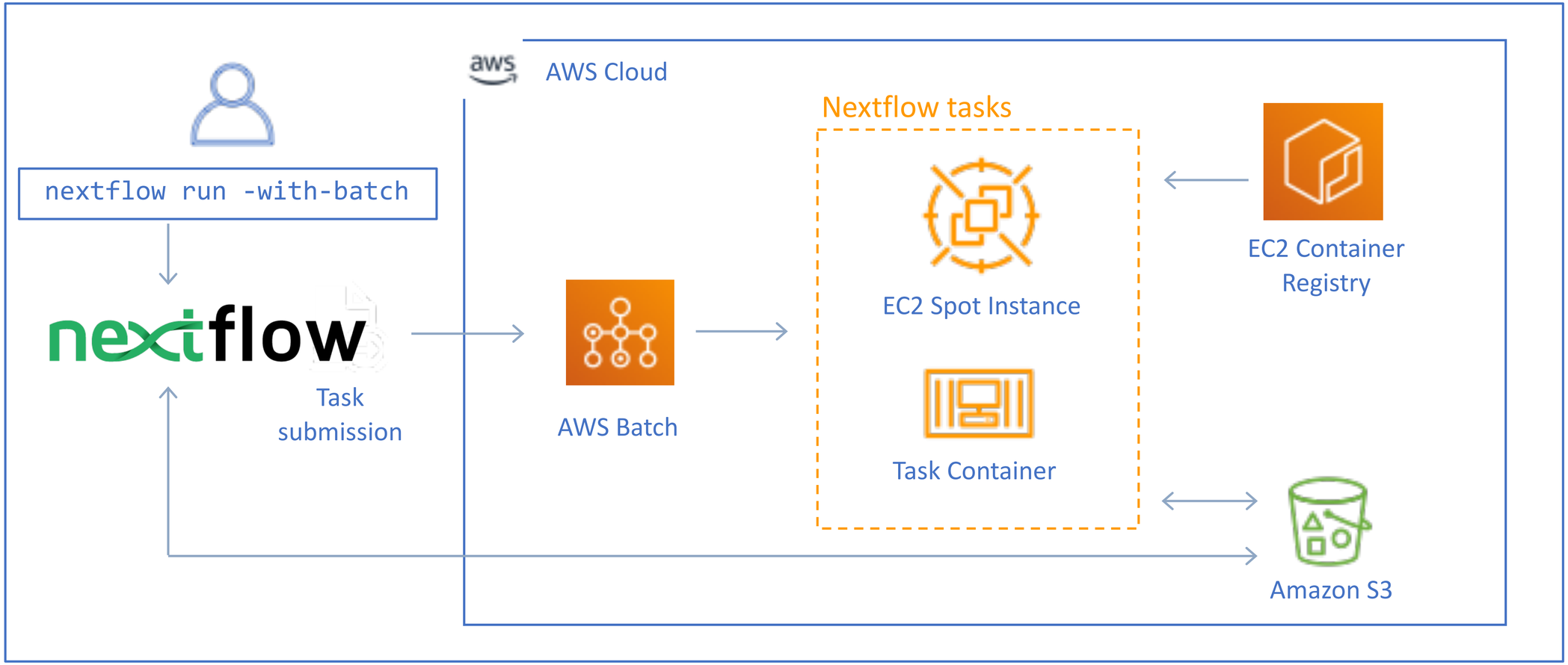


distributed orchestration

- A single job request allocates the desired computes nodes
- Nextflow deploys its own embedded compute cluster
- The main instance orchestrate the workflow execution
- The worker instances execute workflow jobs (work stealing approach)

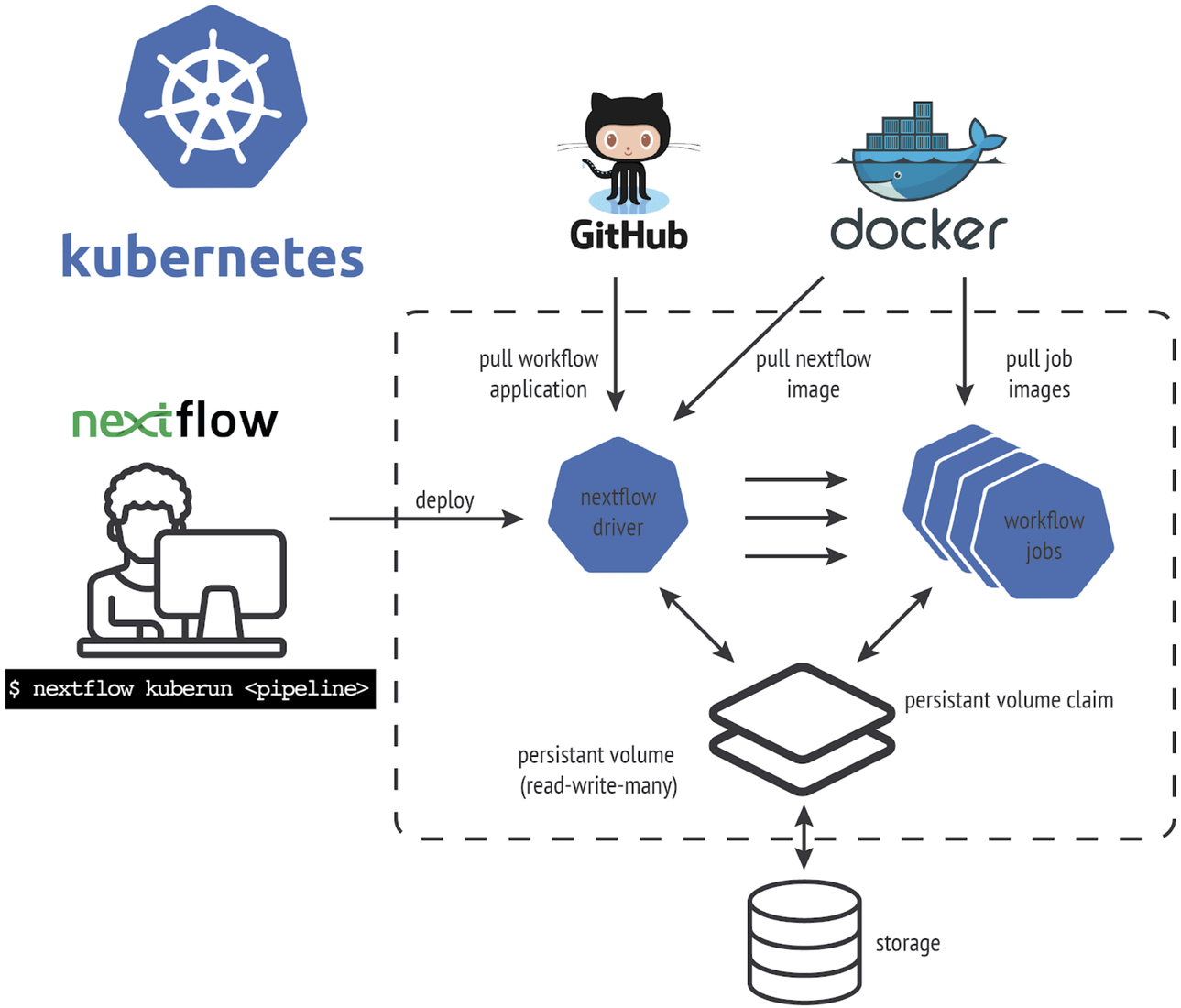


AWS batch deployment



kubernetes / OpenShift

- Next generation native cloud clustering for containerised workloads
- There's the need of workflow orchestration
- K8S executor works well with OpenShift



RED HAT OPENSHIFT

portability

portability


```
process {
 executor = 'slurm'
 queue = 'my-queue'
 memory = '8 GB'
 cpus = 4
 container = 'user/image'
}
```

portability

nextflow

```
process {
 executor = 'awsbatch'
 queue = 'my-queue'
 memory = '8 GB'
 cpus = 4
 container = 'user/image'
}
```

slurm
workload manager

configuration

decoupling

is the key to

portable deployments

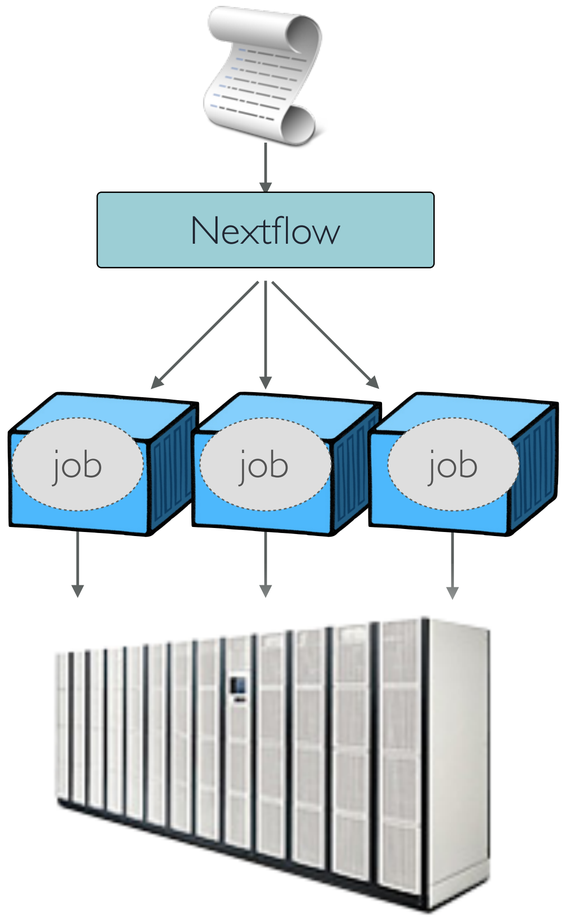
CONTAINERISATION

container vs. VM

- Lighter: MB vs GB
- Faster startup: ms/secs vs minutes
- Virtualise a process/application instead of a OS/ Hardware
- Immutable: don't change over time, thus guarantee replicability over executions.
- Composable: the output of one container is directly consumable as input by another container.
- Transparent: they are created with a well defined automated procedure.

containerisation

- Nextflow envisioned the use of software containers to fix computational reproducibility
- Mar 2014 (ver 0.7), support for Docker
- Dec 2016 (ver 0.23), support for Singularity



nf-core

- Community effort to collect production ready analysis pipelines built with Nextflow
- Initially supported by SciLifeLab, QBiC and A*Star Genome Institute Singapore
- https://nf-co.re

Phil Ewels

Alexander Peltzer

Andreas Wilm

execution reports

Nextflow Report Summary Resources Tasks [trusting_cuvier]

Nextflow workflow report

[trusting_cuvier] *(resumed run)*

Workflow execution completed successfully!

Run times

Fri Apr 27 23:19:53 CEST 2018 - Sat Apr 28 03:18:15 CEST 2018 (completed a day ago, duration: **3h 58m 21s**)

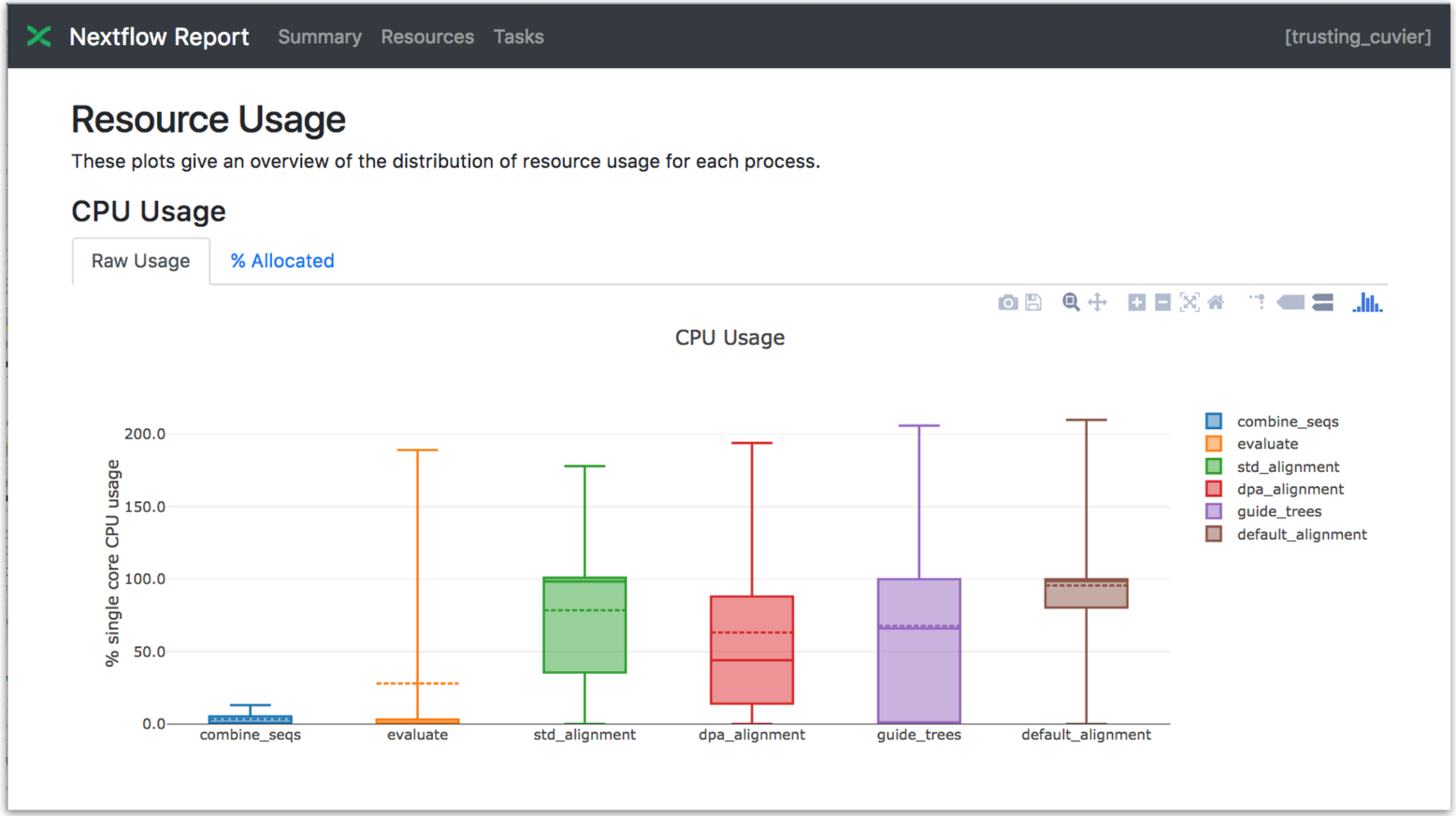
5329 succeeded | 2849 cached | ...

Nextflow command

```
nextflow run main.nf -profile crg --std_align=true --default_align=true --align_method=CLUSTALO,MAFFT --
tree_method=CLUSTALO,MAFFT_PARTTREE --seqs=/users/cn/egarriga/datasets/homfamClustalo/seqs/*.fa --
refs=/users/cn/egarriga/datasets/homfamClustalo/refs/*.ref -with-report -with-trace -resume -bg
```

CPU-Hours	156.6 (31.5% cached, 4.6% failed)
Launch directory	/nfs/users2/cn/egarriga/projects/dpa_cp
Work directory	/nfs/users2/cn/egarriga/projects/dpa_cp/work
Project directory	/nfs/users2/cn/egarriga/projects/dpa_cp
Script name	main.nf
Script ID	6ff267a42e50448d41927a6e5a9787fc
Workflow session	087c9bc8-e488-4311-88aa-961138c42fd6
Workflow profile	crg
Workflow container	cbcrg/regressive-msa:v0.2.4
Container engine	singularity
Nextflow version	version 0.28.2, build 4782 (06-04-2018 12:25 UTC)

execution reports



execution reports

Nextflow Report Summary Resources Tasks [angry_babbage]

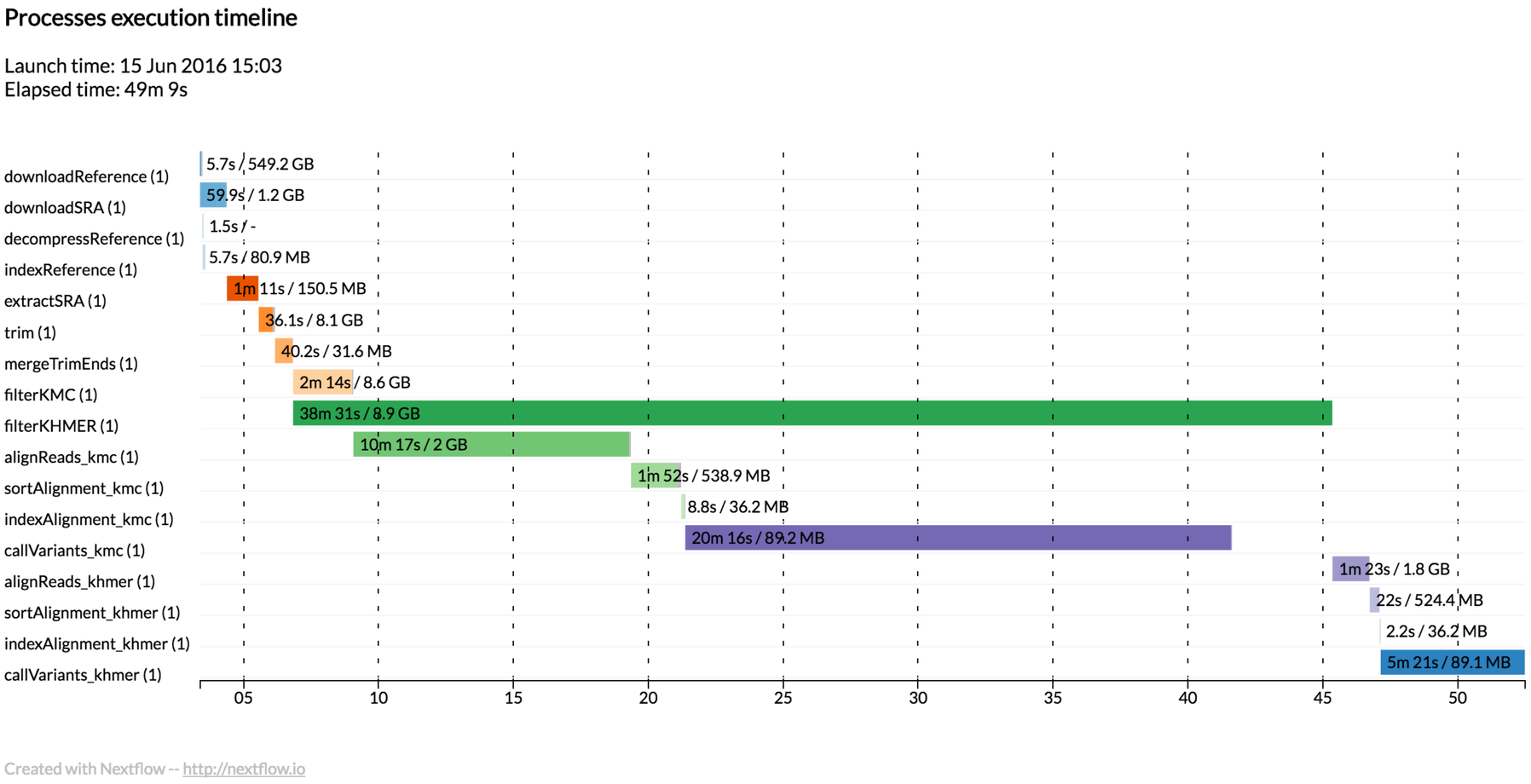
Tasks

This table shows information about each task in the workflow. Use the search box on the right to filter rows for specific values. Clicking headers will sort the table by that value and scrolling side to side will reveal more columns.

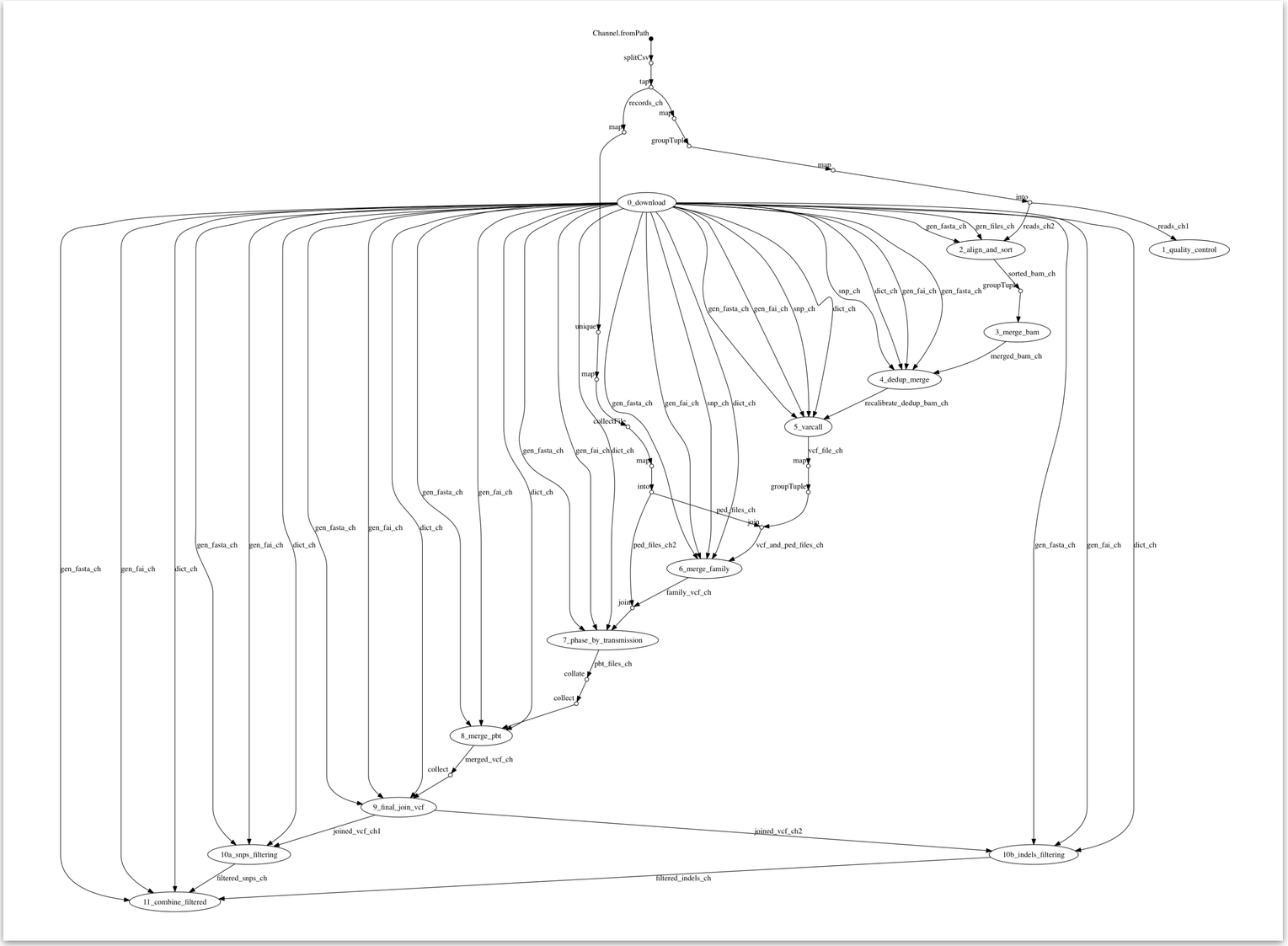
Show 25 entries Search:

task_id	process	tag	status	hash	allocated cpus	%cpu	allocated memory (bytes)	%mem	vmem	rss
1	index	Homo_sapiens.GRCh38.cdna.all.fa.g	COMPLETED	f4/a72585	2	195.0	8589934592	31.9	5272805376	51318
2	parseEncode	/home/pditommaso/projects/rnaseq encode-nf/data/metadata.tsv	COMPLETED	12/bdfd13	1	0.0	-	0.0	17960960	5324
3	fastqc	FASTQC on SRR5210435	COMPLETED	ba/5068a0	2	46.4	6442450944	0.0	4088819712	3685
4	fastqc	FASTQC on SRR3192620	COMPLETED	fa/3e8db3	2	76.7	6442450944	0.0	4089171968	5049
5	fastqc	FASTQC on SRR3192621	FAILED	6b/f753e2	2	-	6442450944	-	-	-
6	fastqc	FASTQC on SRR3192434	COMPLETED	1e/d7f3c2	2	68.8	6442450944	0.0	4088832000	4153
7	fastqc	FASTQC on SRR3192433	COMPLETED	5e/4886ef	2	70.2	6442450944	0.0	4031012864	3843

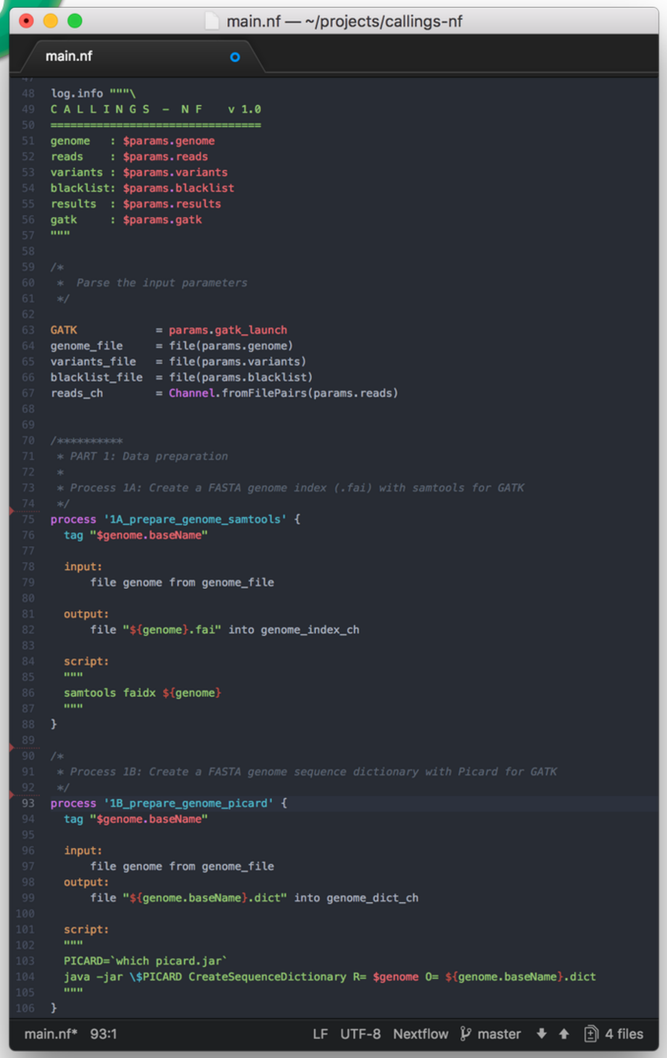
execution timelines

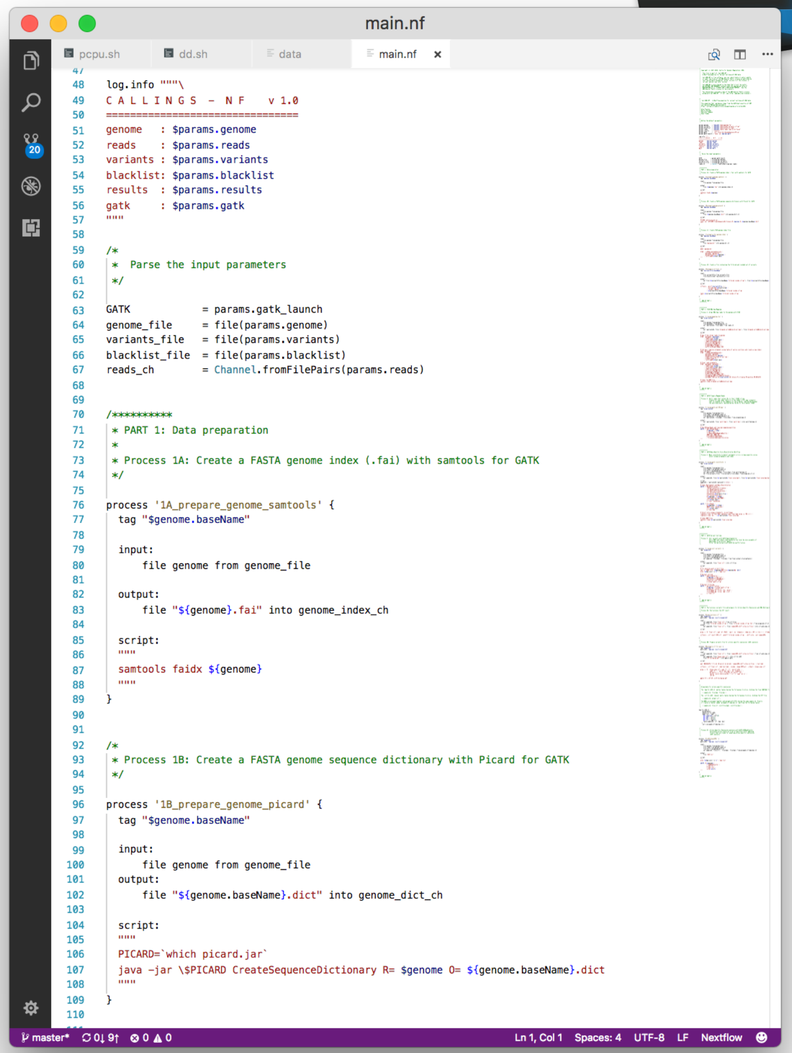


dag visualisation



code editors + syntax highlighting

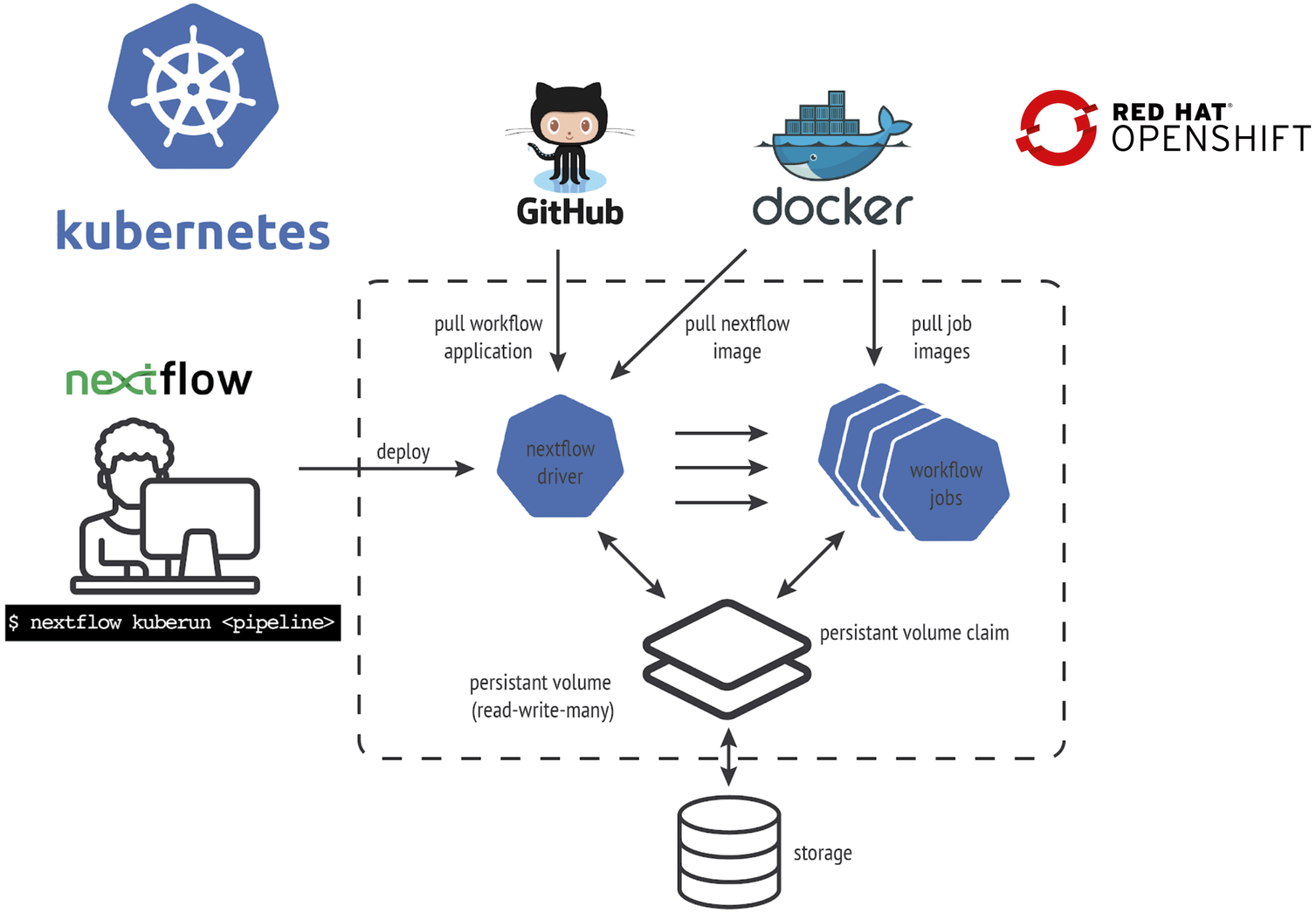




In production since 2014

EMBL-EBI
Roche
THE FRANCIS CRICK INSTITUTE
READCOOR
BERKELEY LAB
SciLifeLab
VIR
University of Colorado Boulder
wellcome sanger institute
International Agency Research on Cancer
World Health Organization
Institut Pasteur
Curtin University
Boehringer Ingelheim
PAWSEY supercomputing centre
UTSouthwestern Medical Center
NATIONAL MARROW DONOR PROGRAM®
Universität Bielefeld
JGI JOINT GENOME INSTITUTE DEPARTMENT OF ENERGY
UCSF
gritstone ONCOLOGY
HARVARD UNIVERSITY
Colorado State University
bina
BENAROYA RESEARCH INSTITUTE Virginia Mason
gmi genomics medicine ireland
Weill Cornell Medical College
QBiC Quantitative Biology Center
enterome bioscience
ACHILLES THERAPEUTICS
Karolinska Institutet
CRG Centre for Genomic Regulation
FABRIC GENOMICS™
Agency for Science, Technology and Research SINGAPORE
SYNTHETIC GENOMICS

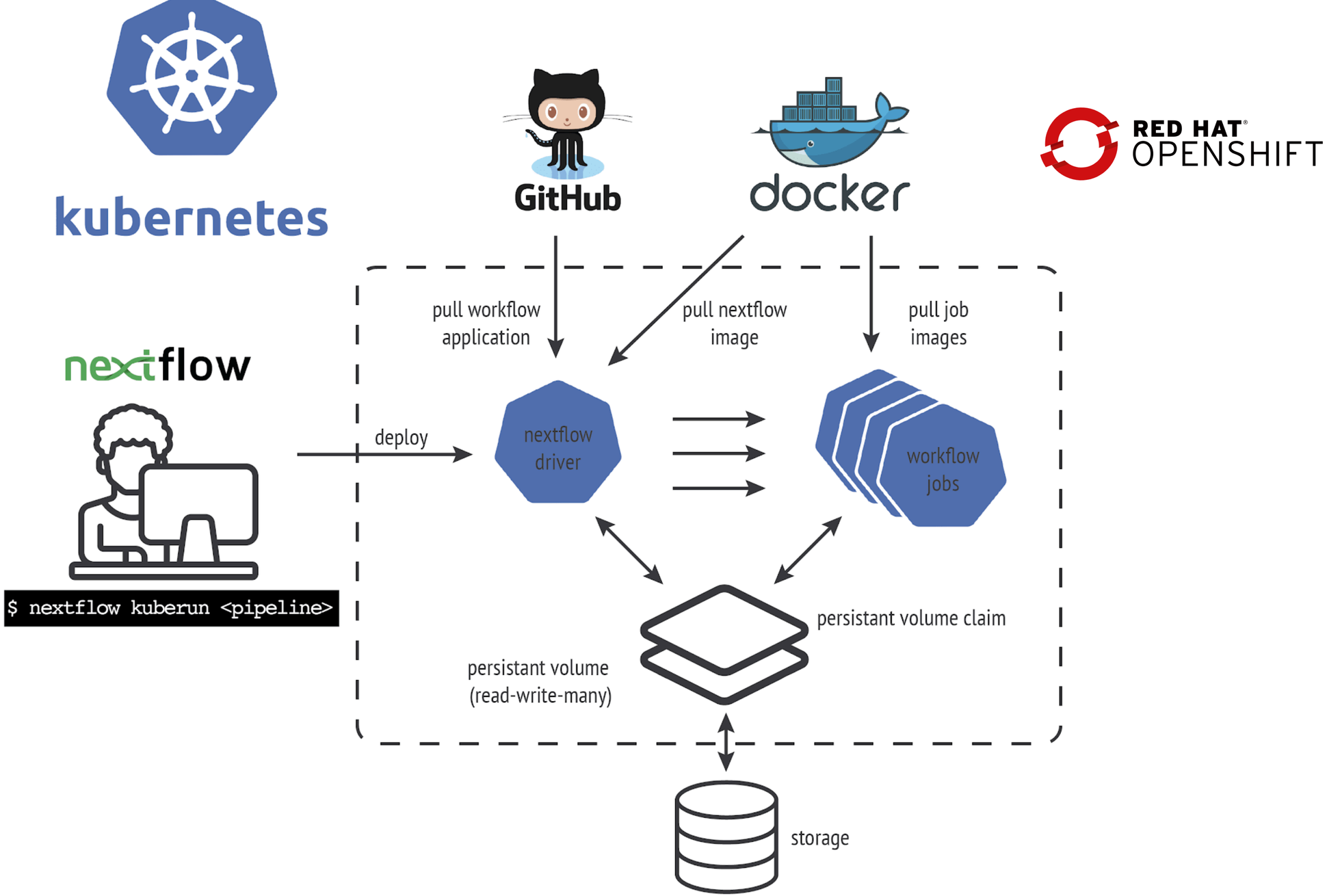
Nextflow in the OpenRiskNet VE



Hello World, Hello OpenRiskNet

1. SSH into VE (ssh -i ~/.ssh/openrisknet evan@130.238.28.49)
2. oc login https://prod.openrisknet.org -u developer
3. oc project nextflow
4. nextflow kuberun nextflow-io/hello -v nf-0001

https://github.com/nextflow-io/hello


kubernetes
GitHub
docker
RED HAT
OPENSHIFT
nextflow
pull workflow application
pull nextflow image
pull job images
deploy
nextflow driver
workflow jobs
$ nextflow kuberun <pipeline>
persistant volume claim
persistant volume (read-write-many)
storage

RNA-Seq Analysis

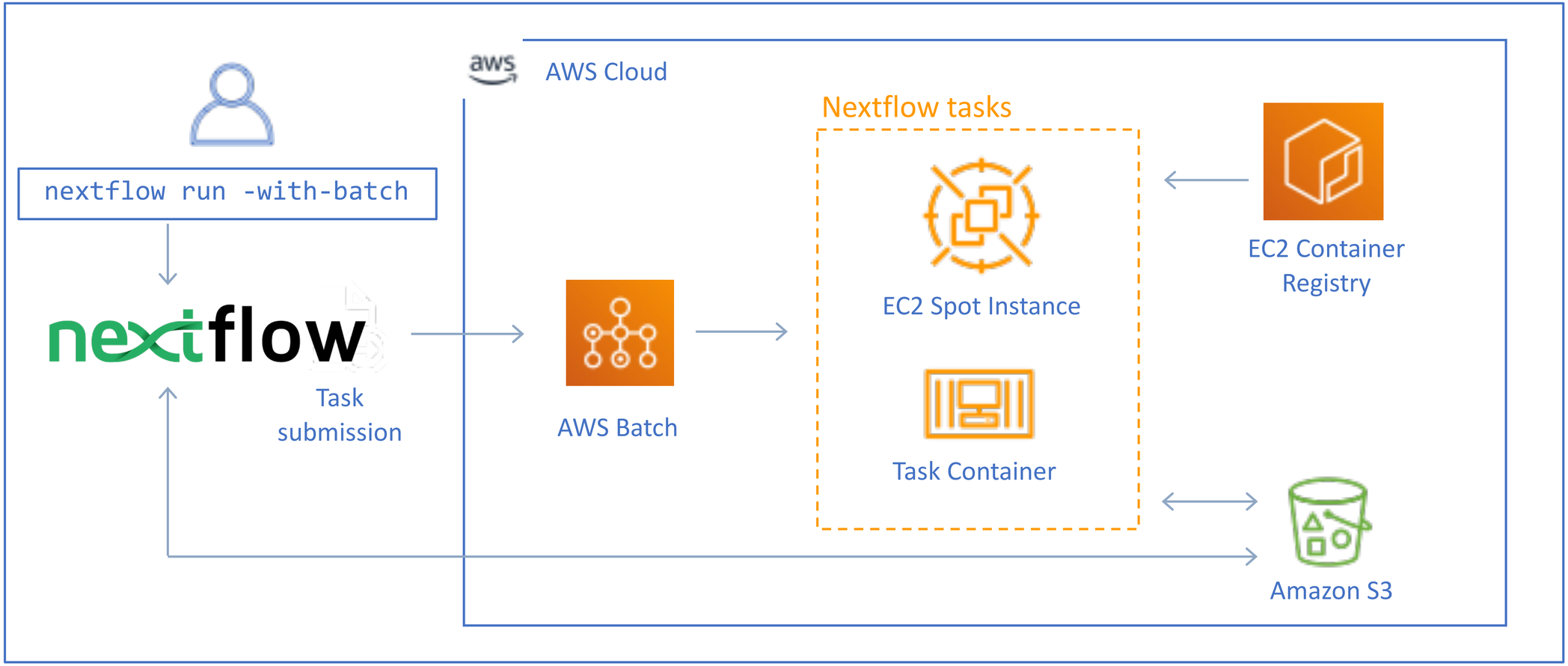
1. `nextflow kuberun nextflow-io/rnaseq-nf -v nf-0002`
2. See pod with `oc get pod`

https://github.com/nextflow-io/rnaseq-nf

Hybrid & Bursting Into the Public Cloud

1. Was configure / NF Config / Env Variables
2. nextflow kuberun nextflow-io/rnaseq-nf -r hybrid -v nf-0002
3. https://github.com/nextflow-io/rnaseq-nf/tree/hybrid
4. https://cbcrg.signin.aws.amazon.com/console

Hybrid & Bursting Into the Public Cloud



External datasources & data localisation

```
// Create a channel from an experiment ID
Channel
    .fromSRA('SRX3859232')
    .println()

// returns the sample ID & FTP address
[SRR6911292, /vol1/fastq/SRR691/002/SRR6911292/SRR6911292.fastq.gz]
[SRR6911293, /vol1/fastq/SRR691/003/SRR6911293/SRR6911293.fastq.gz]
[SRR6911294, /vol1/fastq/SRR691/004/SRR6911294/SRR6911294.fastq.gz]
...

// Create a channel from a text search
Channel
   .fromSRA('liver[All Fields] AND toxicity[All Fields] AND "Homo sapiens"[Organism]')
   .println()
```

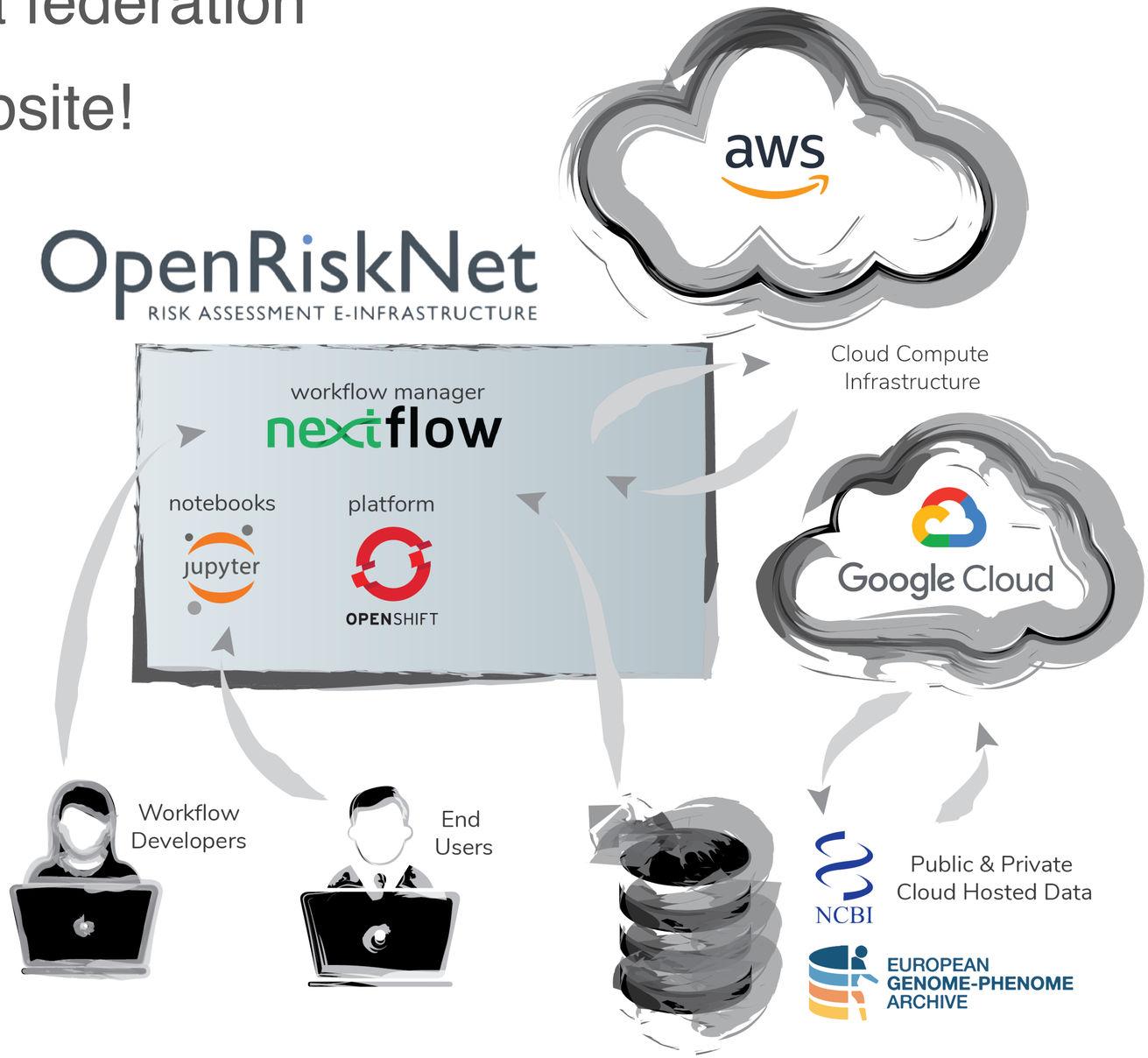
https://github.com/ewels/AWS-iGenomes

Continuing Work

- Integration with the JypyterHub launcher https://jupyterhub-jupyter.prod.openrisknet.org
- Consolidate the Toxicogenomics case study at https://github.com/OpenRiskNet/nf-toxomix
- Expand the case study to include other datasets

Report on computational and data federation

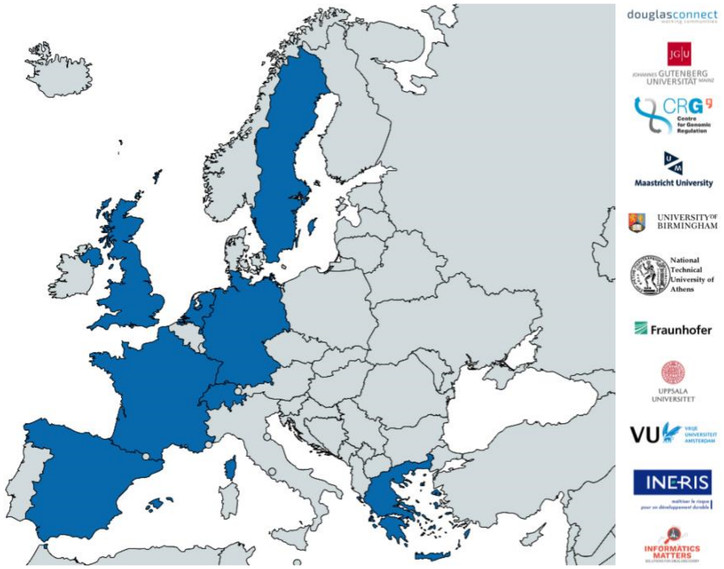
out soon on the OpenRiskNet website!



Acknowledgements

OpenRiskNet (Grant Agreement 731075) is a project funded by the European Commission within Horizon 2020 Programme

Project partners:



P1 Douglas Connect GmbH, Switzerland (DC)
P2 Johannes Gutenberg-Universität Mainz, Germany (JGU)
P3 Fundacio Centre De Regulacio Genomica, Spain (CRG)
P4 Universiteit Maastricht, Netherlands (UM)
P5 The University Of Birmingham, United Kingdom (UoB)
P6 National Technical University Of Athens, Greece (NTUA)
P7 Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V., Germany (Fraunhofer)
P8 Uppsala Universitet, Sweden (UU)
P9 Medizinische Universität Innsbruck, Austria (MUI)
P10 Informatics Matters Limited, United Kingdom (IM)
P11 Institut National De L'environnement Et Des Risques INERIS, France (INERIS)
P12 Vrije Universiteit Amsterdam, Netherlands (VU)

Webinars series

Live demonstrations on the e-infrastructure deployment and the risk assessment case studies

	Topic	Date & Time
Past events	Introduction sessions to the OpenRiskNet e-infrastructure	See Webinar recordings: • Session 1 (24 Sep 2018) • Session 2 (27 Sept 2018) • Session 3 (4 Oct 2018) • Session 4 (30 Oct 2018)
	Learn how to deploy the OpenRiskNet virtual research environment	See Webinar recordings (25 Feb 2019)
	Demonstration on data curation and creation of pre-reasoned datasets in the OpenRiskNet framework	Monday, **18 March 2019** 16:00 CET
	Identification and linking of data related to AOPWiki (an OpenRiskNet case study)	Tuesday, **26 March 2019** 17:00 CET
	Semantic annotation	Monday, **1 April 2019** 16:00 CET
	The Adverse Outcome Pathway Database (AOP-DB)	Monday, **8 April 2019** 16:00 CET
Current Event	Nextflow and TGX case study	Monday, **27 May 2019**

https://openrisknet.org/events/