

Nextflow Tower (Cloud)

Procedures

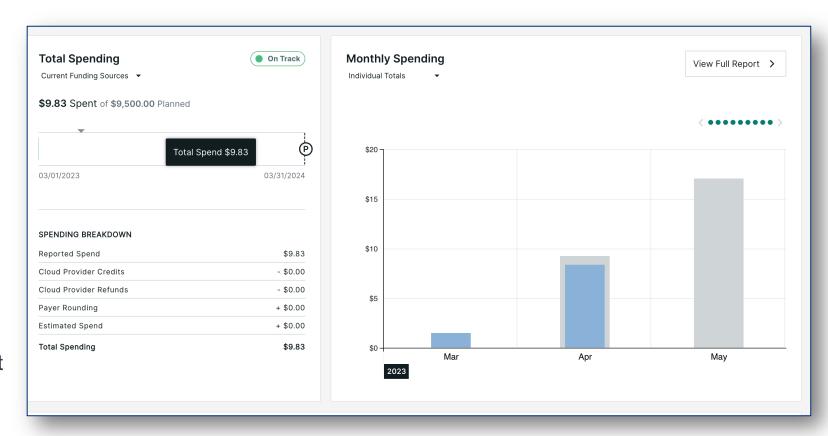
Steps for Accessing Nextflow Tower Launchpad

Nextflow Tower - Overview

- Create an account (Required)
 https://tower.nf/login
 You can use your Github or Google Account
- 2. Connects your S3 Storage and IAM user with necessary tokens (retrieve on email)
- 3. Launch your pipeline(s) on AWS Batch

2. AWS

- Receive Email (from APL) for specific IAM user access
 - Include Access and Secret Key outlined in Tower Setup
 - 2. Request one at brian.merritt@jhuapl.edu





Viewing S3 Bucket(s) and Validations

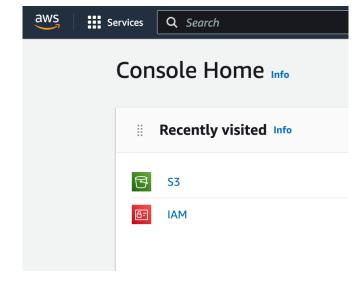
Data Uploads

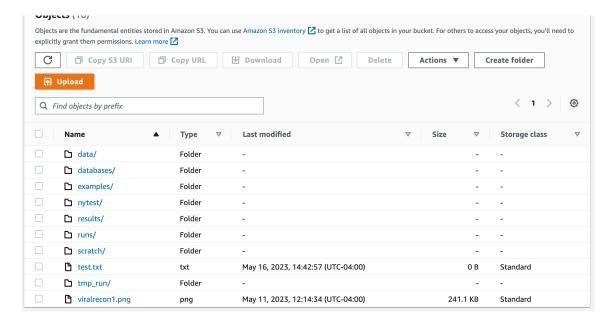
- Verifying file uploads: https://aws.amazon.com/getting-started/hands-on/amazon-s3-with-additionalchecksums/?ref=docs_gateway/amazons3/checking-object-integrity.html
 - Useful if internet is unstable / S3 upload(s) timeout
 - Be aware Windows and Unix (S3 native) handle file formats differently. Checksums will always be different between a windows and linux
- Accessing the Console from the browser:
 - https://docs.aws.amazon.com/IAM/latest/UserGuide/id users sign-in.html
 - A spreadsheet of the sign-in url, username (IAM), and password will be sent alongside the access key set on account request
 - You do NOT need an AWS account to access AWS Console from the browser
- If you plan to share an account across more than one user, please let me know all emails that are to be associated with it so I can better track and send updates to all.
 - Keep all access keys and the password guarded to a limited group of people. Misuse is always be possible and I'll be monitoring through weekly reports if cost becomes high



Accessing the console

- Open a browser and go to the sign-in url in the CSV sent after account creation by JHUAPL
- Copy + paste the url and enter the IAM Username and Password (same spreadsheet)
- Select "Services" in the top-left or recently visited (if you've done this before)
- Select S3
- Select nftass
- Upload data to the "data" folder. Mark all subfolders appropriately
 - You can add any kraken2 database into the databases folder
 - Make sure to add your samplesheet.csv for your runs as well, to be referenced in the TaxTriage Pipeline in Nextflow Tower
 - Remember: all paths from Nextflow tower will be relative to the root level of nftass i.e. "S3://nftass"
- Run pipeline(s) on data as necessary. Delete when no longer needed.

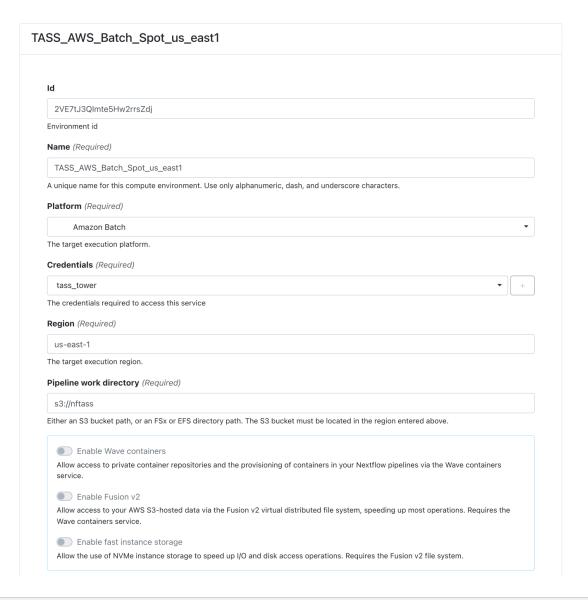




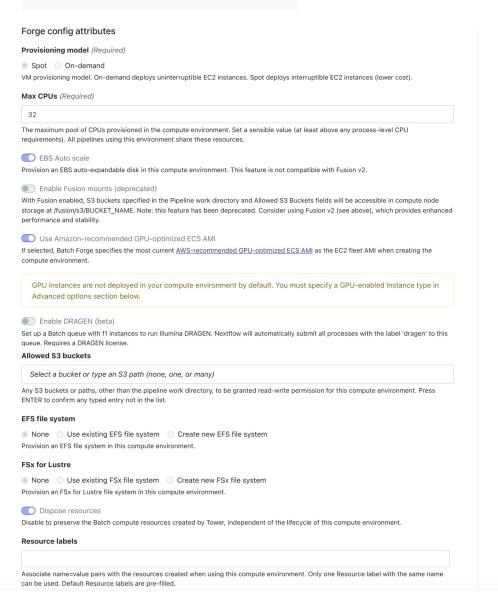
- 1. In a workspace, select Compute Environments and then New Environment.
- 2. Enter a descriptive name for this environment: TASS_AWS_Batch_Spot_us-east-1
- 3. Select **Amazon Batch** as the target platform.
- 4. Select your AWS credentials or add new credentials by selecting the + button.
- 5. Enter a name, e.g. "AWS Credentials".
- 6. Add the Access key and Secret key. These keys are to be emailed to each user wanting access to the workflow on NF Tower
- 7. Select a **Region**, put: "us-gov-east-1"
- 8. Enter the **Pipeline work directory**: "s3://tass"
- 9. Set the Config mode to Batch Forge.
- 10. Select a Provisioning model. In most cases this will be Spot.
- 11. Select Enable GPUs to allow the deployment of GPU-enabled EC2 instances
- 12. Select **Dispose resources**
- 13. Create/Save the environment



Viewing your compute environment



Compute Environments

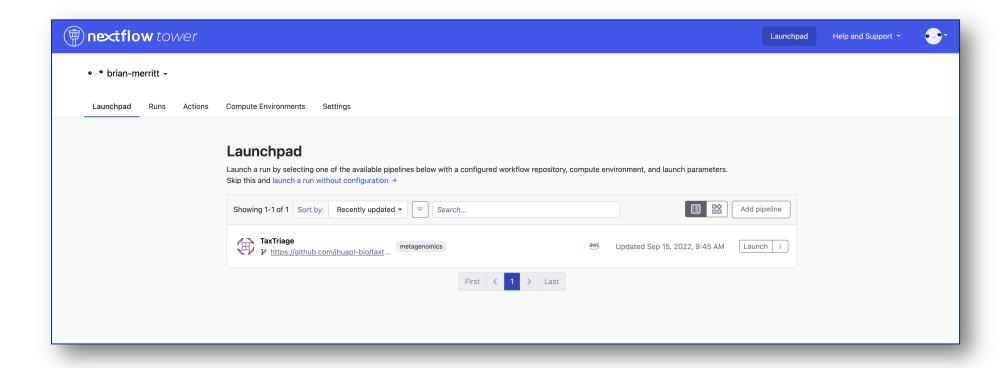




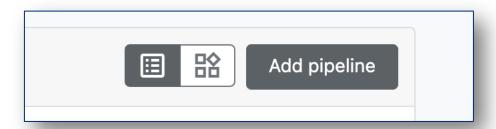
Nextflow Tower

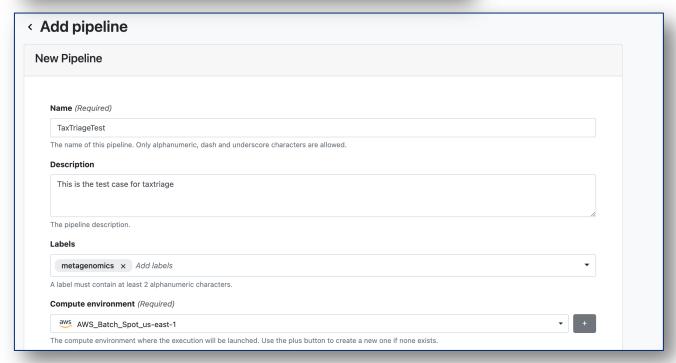
Viewing your launchpad environment

- View current and previous runs
- 2. Attach or add any other nextflow-based pipeline directly from the web-ui
- 3. Relaunch or import existing pipelines (like TaxTriage)



- 1. Add Pipeline
- 2. Add Name
- 3. Set basic description
- Select Compute environment connected to your NF Tower account we made earlier
 (TASS_AWS_Batch_Spot_useast-1)

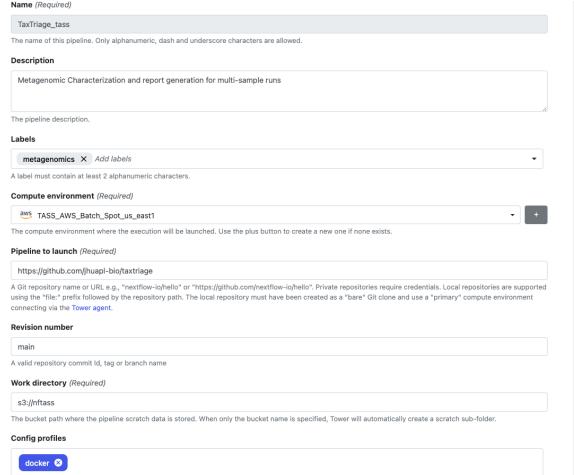




https://help.tower.nf/21.10/compute-envs/aws-batch/#forge-compute-environment



- 5. Set pipeline launch to:
 https://github.com/jhuapl-bio/taxtriage
- 6. Revision: main
 - This is the main branch for TaxTriage, you can specify any branch in the "Revision number" field
- 7. Workdir: s3://nftass
- 8. Config: docker



9. Set Pipeline Params

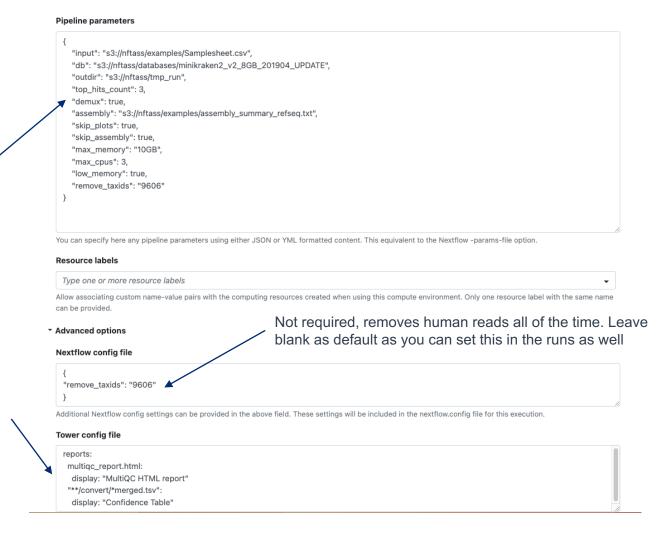
Pipeline Params are in JSON format, and should reference whatever Samplesheet, outdir, and db (kraken2) you want.

See TaxTriage documentation for updating arguments in this JSON

Adjustable Fields also available in the Launch page for your pipeline

Useful for redoing the pipeline, but with different parameters like sample sheet or database

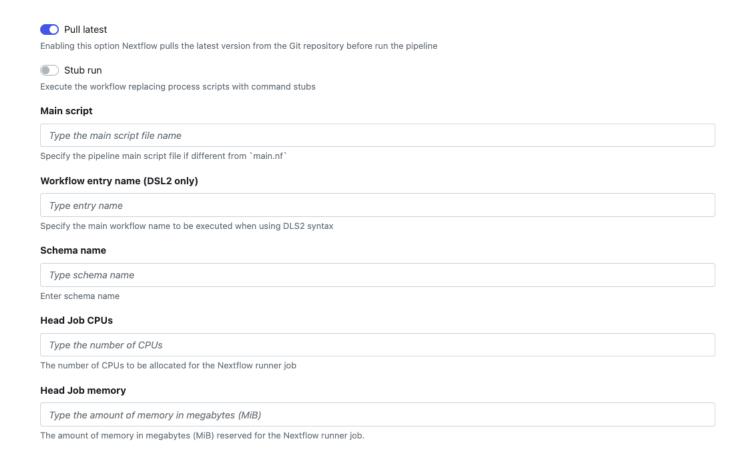
Set Nextflow Tower to Render the HTML reports directly on the dashboard on completion



Finish it up

Pull Latest if you always want the most up to date version

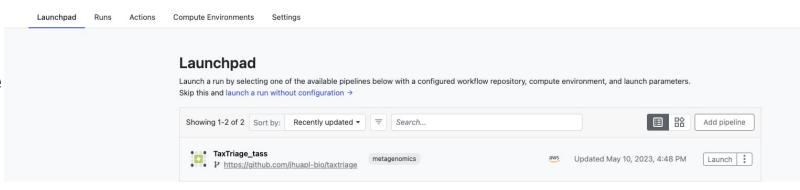
Leave the rest of the fields blank

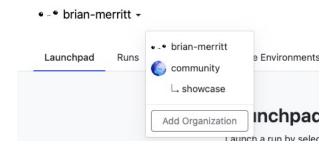




Running the pipeline

- Select Launchpad Tab
- 2. Select Launch for your pipeline
- 3. Review Parameters
 - These are where you need to either point to paths on AWS or set numbers/string values for the pipeline
 - 2. See https://github.com/jhuapl-bio/taxtriage/blob/main/docs/usage.md



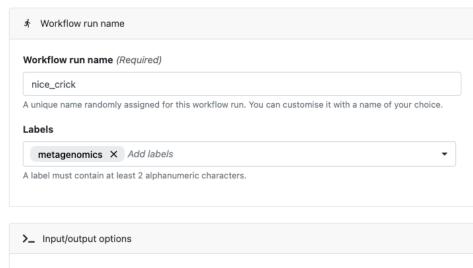


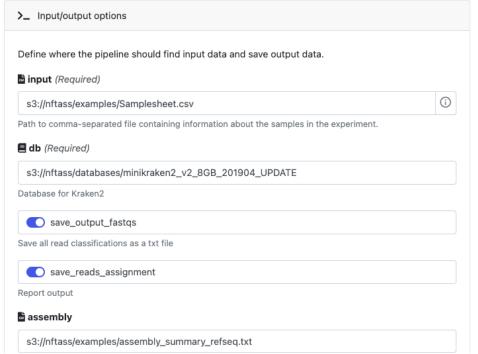
Upper left: Ensure you are viewing your launchpad and not the showcase

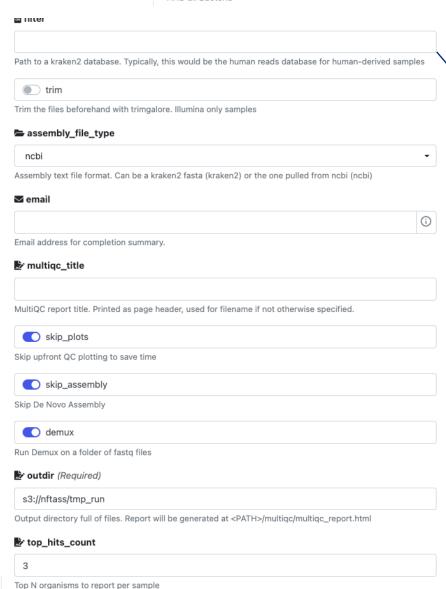
Example Input Parameters in Action



Taxid numbers in double quotes (e.g. "9606" for human). For multiple, example: "9606 2" removes human AND all Bacteria



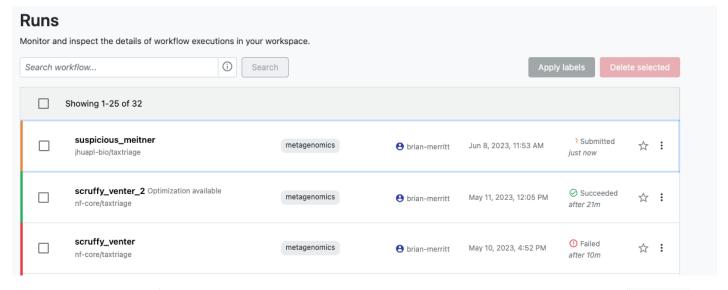


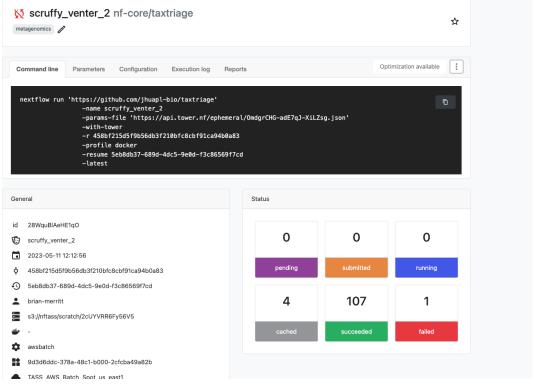


Filter: not necessary if you know taxid of organism. See remove taxids to pull all taxid(s) out of kraken2 output i.e. 9606 for human reads using kraken2

Running the Pipeline

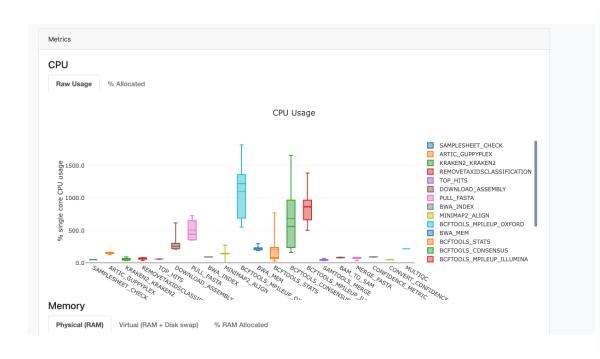
- Hit Launch at the bottom
- Select "Runs" in the top tab
- View each job submissions
 - Lines indicate status
 - Green: Complete
 - Orange: submitted, requisitioning compute
 - Red: Failed at some point
 - Blue: Running

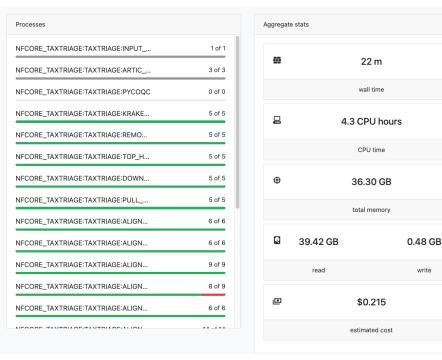




View Logs and Report

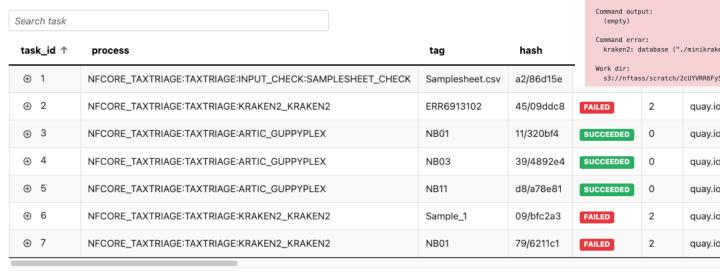
- Inside a specific job, select Reports to see the MultiQC report (if completed successfully)
- Select Execution log to see what logs are, as it would seem if you ran from the CLI
- Scroll down to see cost analysis and compute used at each step





Checking Errors

 IF your run fails, check that all parameters are appropriately set. You can also select specific subtasks in the table halfway down to see what parts did not complete. Hit the "+" sign to view more



Workflow execution completed unsuccessfully The exit status of the task that caused the workflow execution to fail was: 2 Error executing process > 'NFCORE_TAXTRIAGE:TAXTRIAGE:KRAKEN2_KRAKEN2 (ERR6913102)' Caused by: Essential container in task exited Command executed: kraken2 \ --db minikraken2_v2_8GB_201904_UPDATE \ --threads 12 \ --report ERR6913102.kraken2.report.txt \ --gzip-compressed \ --unclassified-out ERR6913102.unclassified#.fastq \ --classified-out ERR6913102.classified#.fastq \ --output ERR6913102.kraken2.classifiedreads.txt \ --paired \ ERR6913102_1.fastq.gz ERR6913102_2.fastq.gz pigz -p 12 *.fastq cat <<-END VERSIONS > versions.vml "NFCORE_TAXTRIAGE:TAXTRIAGE:KRAKEN2_KRAKEN2": kraken2: \$(echo \$(kraken2 --version 2>&1) | sed 's/^.*Kraken version //; s/ .*\$//') pigz: \$(pigz --version 2>&1 | sed 's/pigz //g') END VERSIONS Command exit status: kraken2: database ("./minikraken2_v2_8GB_201904_UPDATE") does not contain necessary file hash.k2d s3://nftass/scratch/2cUYVRR6Fy56V5/45/09ddc88e919d299994c5b46f58f2b7 quay.io quay.io In this example, the kraken2 quay.io database was incompletely quay.io downloaded quay.io quay.io

Configuration

Command line

Error report (1)

Execution log

