MemVerge

Workflows in the Cloud

MDI Biological Laboratory

2nd May, 2025

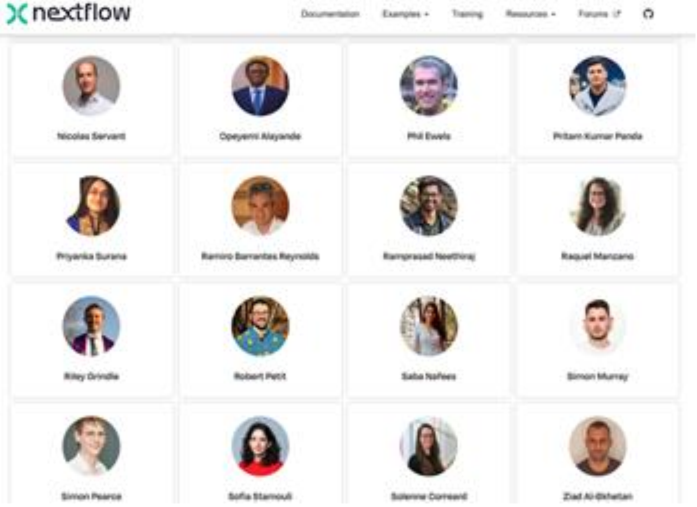
MDI Biological Laboratory
Computational Biology Core

Saba Nafees, Ph.D.
MemVerge, Nextflow Ambassador

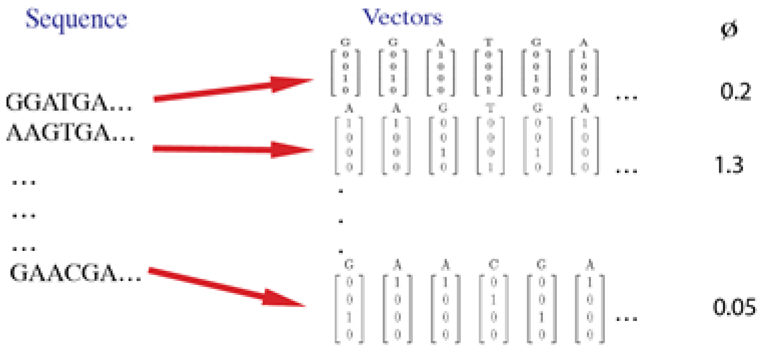
Background

- Research: Understanding sequence–phenotype relationships using novel analytical methods.
- Bioinformatics (analysis, pipeline development & optimization, etc.); computational biology problems (gene expression, immune/antibody analysis, etc.); methods development.
- community building (nf-core), training/teaching, and mentorship!

HORIZONS
— OMICS COLLECTIVE —



https://www.nextflow.io/our_ambassadors.html



Basis of ortho_seqs work

SUPERFLUID

CHAN ZUCKERBERG BIOHUB

MemVerge

nf-core

Little bit about MemVerge…

Raised ~$100M of investment

Headquarters in San Jose, CA

SK hynix intel capital NetApp JVP

Lightspeed cisco investments

Founders created $B product lines

Powered By vmware vSAN XtremIO

Software is used by genomics customers in four continents

Launched Memory Machine for AWS at re:Invent 2022

Columbia University oNKo-innate HZPC MDI Biological Laboratory Traitology

tgen Nuanced Health Dana-Farber Cancer Institute Analytical Biosciences Stanford University

BD Regenerative Bio 寻因生物 SEEKGENE PennState A|M

Award winning solution helps automate & lower the cost of running genomics workloads on AWS

BEST of SHOW WINNER Bio-IT World Europe

BEST of SHOW WINNER Bio-IT World 22 BEST of SHOW PEOPLE'S CHOICE WINNER Bio-IT World 22 USA

Running pipelines at scale while optimizing resources

Running multiple pipelines at scale, given lots of sequence samples and limited resources can be a bottleneck for R&D.

Have you used EC2 Spot instances?

First the benefits…

Customers have saved over $8B

EC2 Spot prices can be 50-80% less vs. On-Demand

Spot instances introduce unpredictability

Costs can end up being higher than just running with on-demand instances

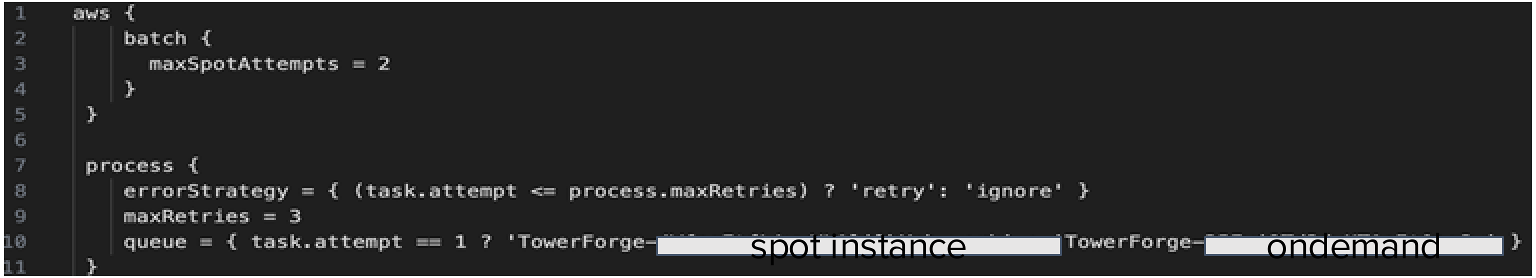
Runtimes can be much longer when jobs are interrupted, 2-minute warning

Pain Point: Spot reclamations & long runtimes

Use case example:

Molecular diagnostics startup working on cell free mRNA assay for Alzheimer's

- running nf-core rnaseq pipeline on samples
- need fast turnaround times for bioinformatics analysis + results
- analysis informs further iterations of assay
- without results, subsequent experiments are delayed
- pipeline runs monitored on Seqera, spot reclamations led to increased runtimes
  - hybrid strategy of switching to OnDemand (Nextflow config) reduced failures.
- cost & time optimization was key



```
aws {
    batch {
      maxSpotAttempts = 2
    }
}

process {
    errorStrategy = { (task.attempt <= process.maxRetries) ? 'retry': 'ignore' }
    maxRetries = 3
    queue = { task.attempt == 1 ? 'TowerForge-[redacted]spot instance TowerForge-[redacted]ondemand }
}
```

Pain Point: Cloud Costs

“Stuck between a rock and a hard place”

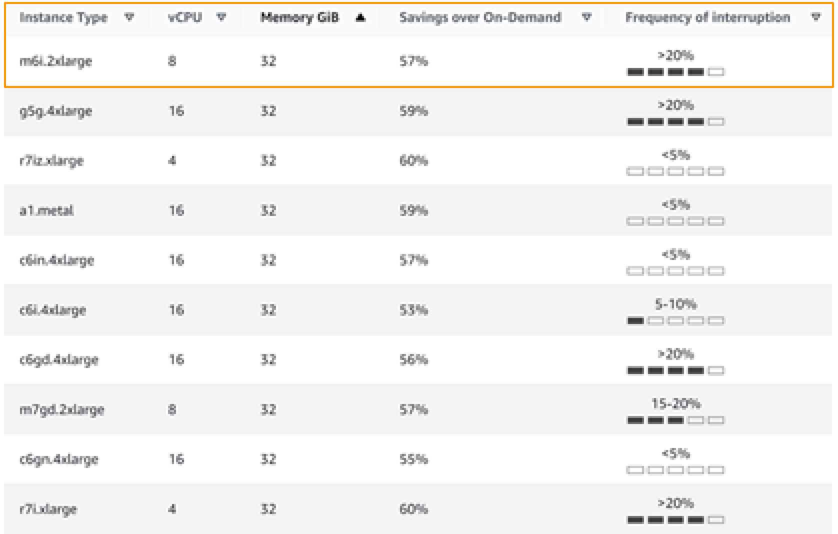
On-Demand $$$
no commitment

~~Reserved Instances~~ $$ long term
commit

~~Cloud Savings Plans~~ $$ long term
commit

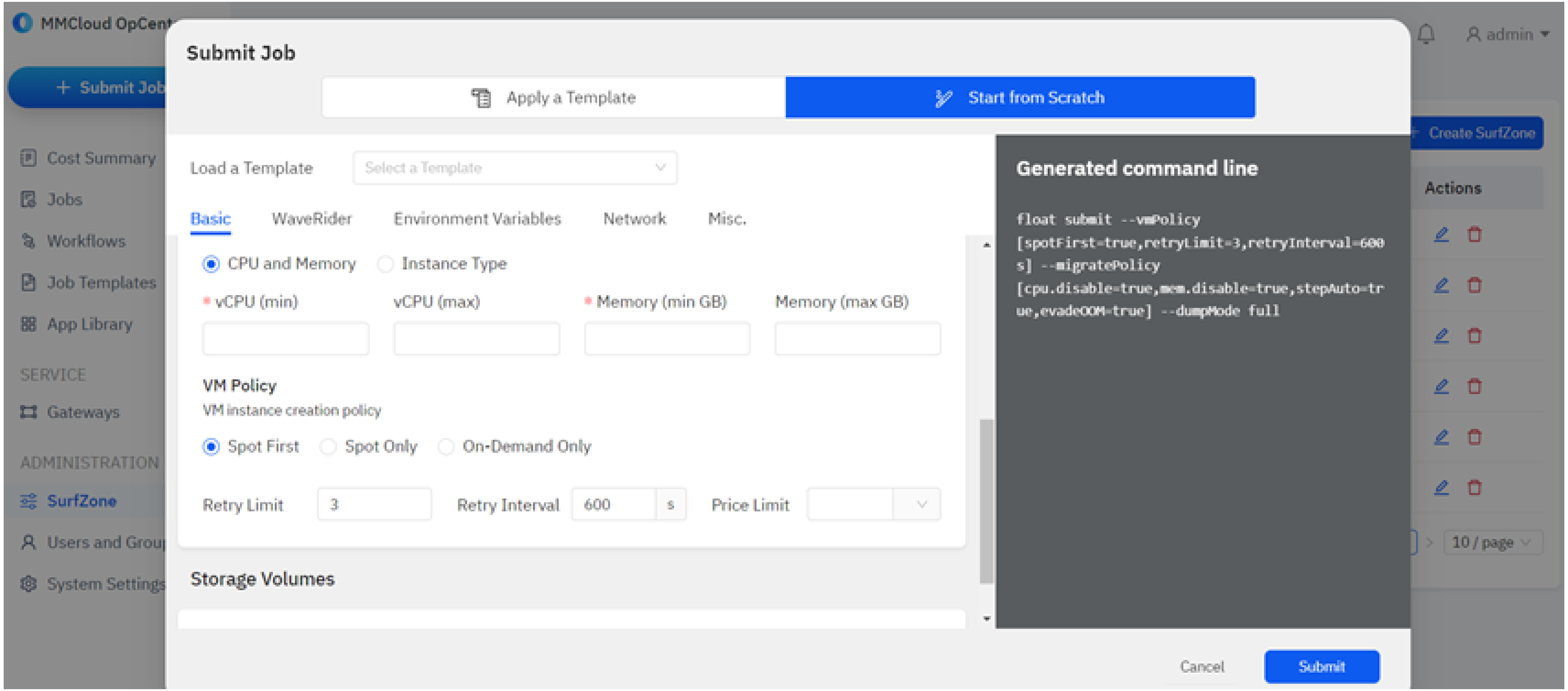
Spot Instances $?
no commitment

“Savings vs. On-Demand vs. Interruption Frequency”



Instance Type	vCPU	Memory GiB	Savings over On-Demand	Frequency of interruption
m6i.2xlarge	8	32	57%	>20% ■■■■□
g5g.4xlarge	16	32	59%	>20% ■■■■□
r7iz.xlarge	4	32	60%	<5% □□□□□
a1.metal	16	32	59%	<5% □□□□□
c6in.4xlarge	16	32	57%	<5% □□□□□
c6i.4xlarge	16	32	53%	5-10% ■□□□□
c6gd.4xlarge	16	32	56%	>20% ■■■■□
m7gd.2xlarge	8	32	57%	15-20% ■■■□□
c6gn.4xlarge	16	32	55%	<5% □□□□□
r7i.xlarge	4	32	60%	>20% ■■■■□

We make running on Spot simple…and…



Submit Job

Apply a Template | Start from Scratch

Load a Template: Select a Template

Basic | WaveRider | Environment Variables | Network | Misc.

● CPU and Memory ○ Instance Type

* vCPU (min) | vCPU (max) | * Memory (min GB) | Memory (max GB)

VM Policy

VM instance creation policy

● Spot First ○ Spot Only ○ On-Demand Only

Retry Limit: 3 Retry Interval: 600 s Price Limit:

Storage Volumes

Generated command line

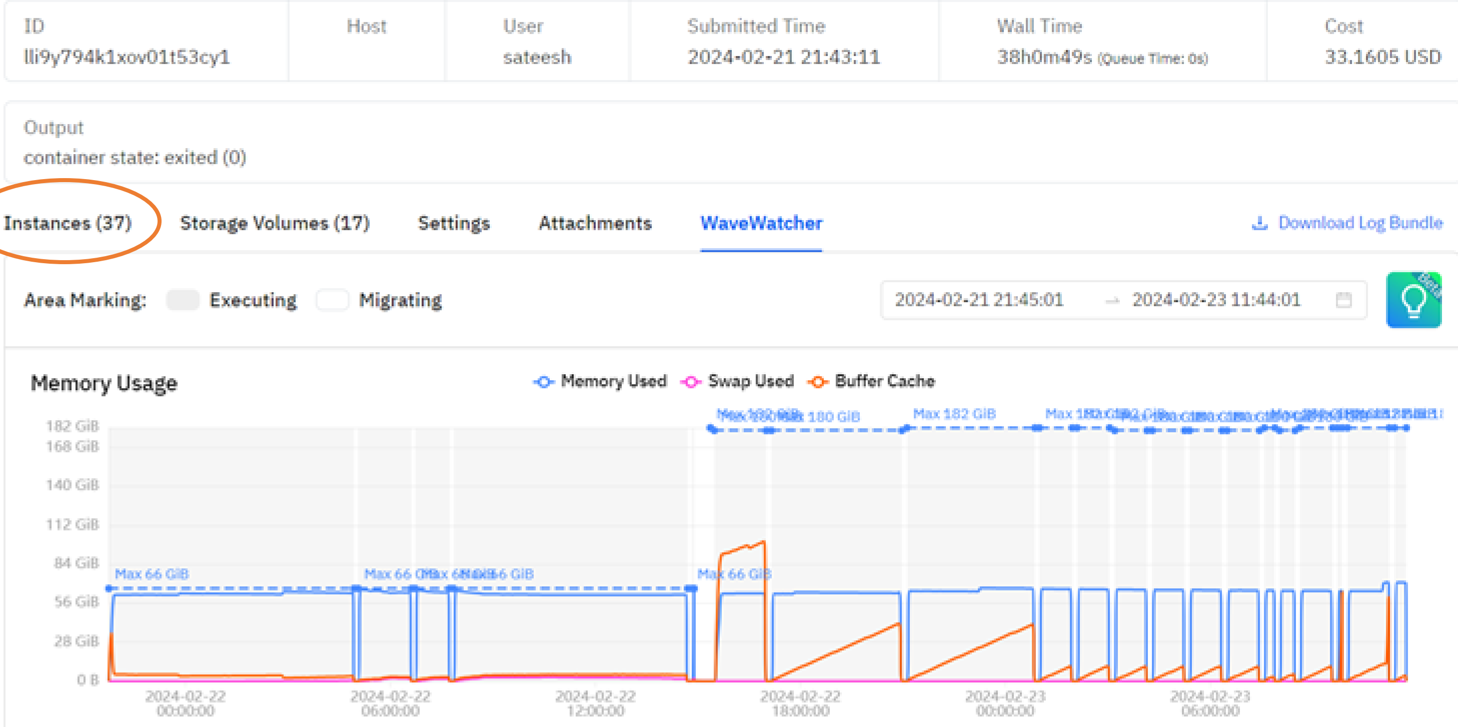
```
float submit --vmPolicy
[spotFirst=true,retryLimit=3,retryInterval=600
s] --migratePolicy
[cpu.disable=true,mem.disable=true,stepAuto=tr
ue,evadeOOM=true] --dumpMode full
```

Cancel Submit

Memory Machine = Spot Reclaim Immunity

Sample: Axolotl **Runtime:** 38hrs, 49 secs **EC2 Instances:** 37 Spot Instances

Workflow: Nextflow **Process:** NFCORE-ATACSEQ-ATACSEQ-FASTQ-ALIGN-BWA-BWA-MEM

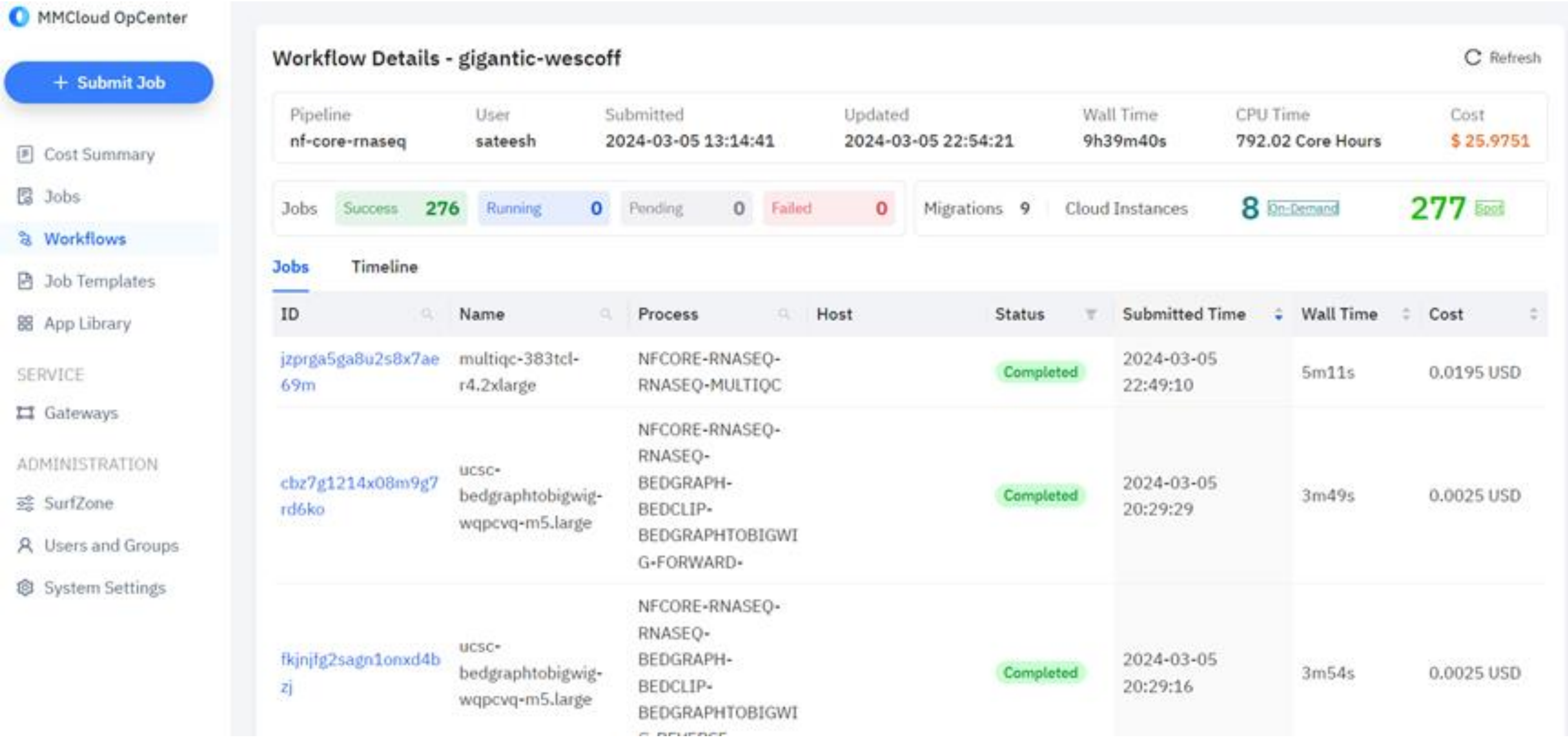


ID	Host	User	Submitted Time	Wall Time	Cost
lli9y794k1xov01t53cy1		sateesh	2024-02-21 21:43:11	38h0m49s (Queue Time: 0s)	33.1605 USD

Output
container state: exited (0)

2024-02-21 21:45:01 → 2024-02-23 11:44:01

How is my Nextflow pipeline run going?



Workflow Details - gigantic-wescoff

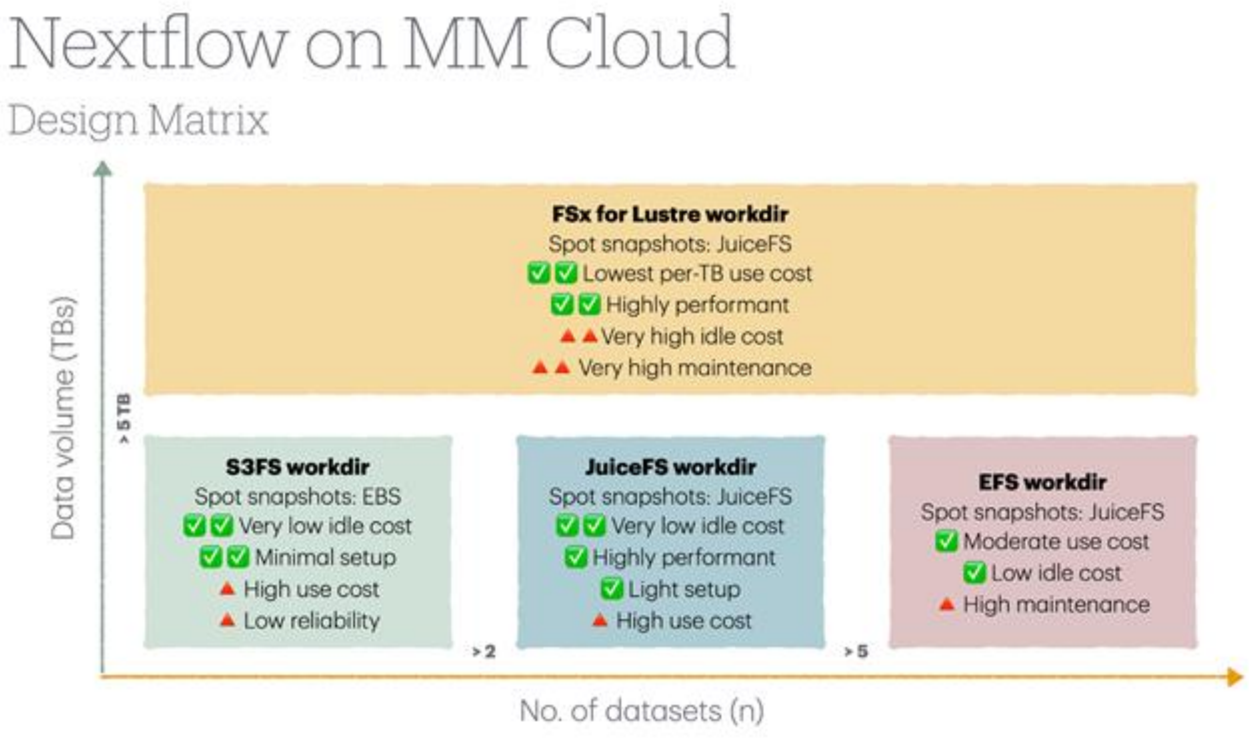
Pipeline	User	Submitted	Updated	Wall Time	CPU Time	Cost
nf-core-rnaseq	sateesh	2024-03-05 13:14:41	2024-03-05 22:54:21	9h39m40s	792.02 Core Hours	$ 25.9751

Jobs Success 276 Running 0 Pending 0 Failed 0

Migrations 9 Cloud Instances 8 On-Demand 277 Spot

ID	Name	Process	Host	Status	Submitted Time	Wall Time	Cost
jzprga5ga8u2s8x7ae69m	multiqc-383tcl-r4.2xlarge	NFCORE-RNASEQ-RNASEQ-MULTIQC		Completed	2024-03-05 22:49:10	5m11s	0.0195 USD
cbz7g1214x08m9g7rd6ko	ucsc-bedgraphtobigwig-wqpcvq-m5.large	NFCORE-RNASEQ-RNASEQ-BEDGRAPH-BEDCLIP-BEDGRAPHTOBIGWIG-FORWARD-		Completed	2024-03-05 20:29:29	3m49s	0.0025 USD
fkjnjfg2sagn1onxd4bzj	ucsc-bedgraphtobigwig-wqpcvq-m5.large	NFCORE-RNASEQ-RNASEQ-BEDGRAPH-BEDCLIP-BEDGRAPHTOBIGWIG-REVERSE		Completed	2024-03-05 20:29:16	3m54s	0.0025 USD

Use the right file system for various workloads without getting vendor locked



Nextflow on MM Cloud

Design Matrix

FSx for Lustre workdir
Spot snapshots: JuiceFS
✓✓ Lowest per-TB use cost
✓✓ Highly performant
▲▲ Very high idle cost
▲▲ Very high maintenance

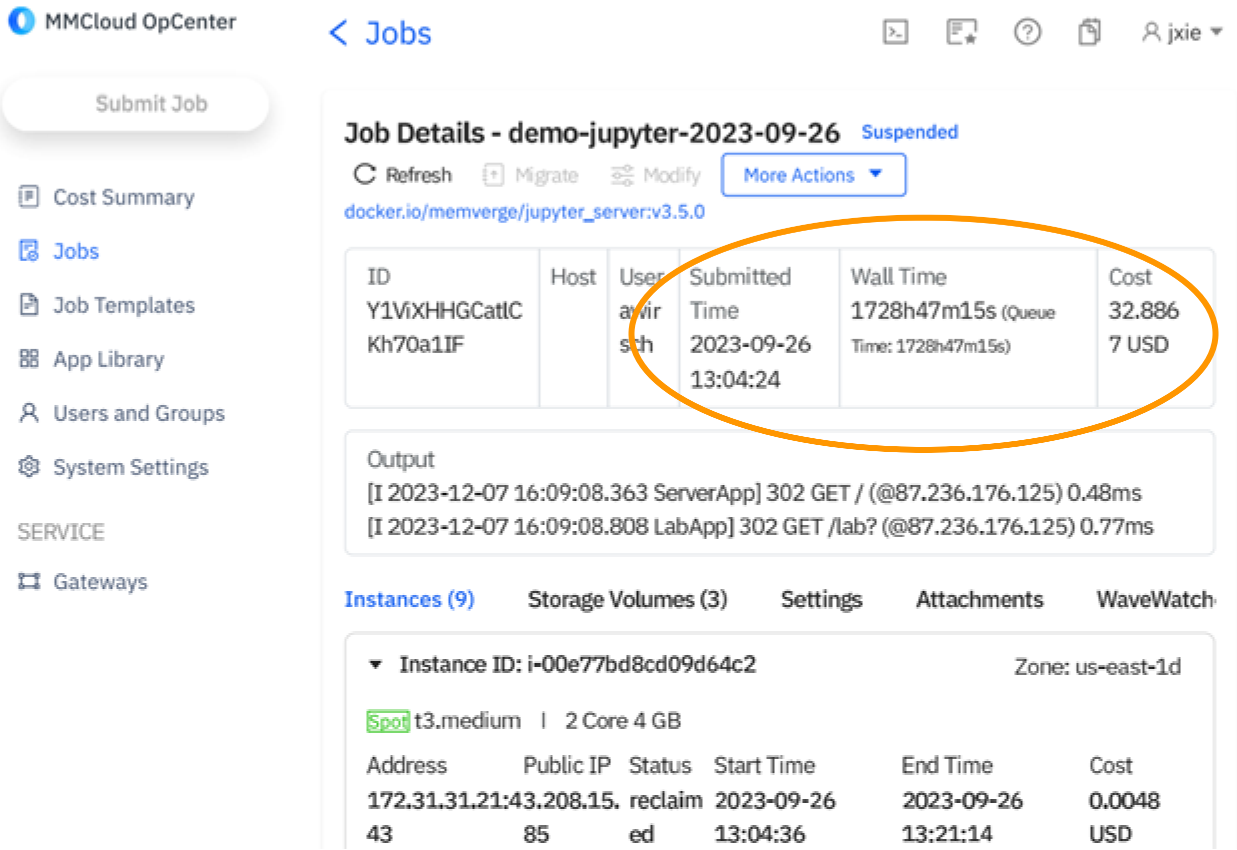
S3FS workdir
Spot snapshots: EBS
✓✓ Very low idle cost
✓✓ Minimal setup
▲ High use cost
▲ Low reliability

JuiceFS workdir
Spot snapshots: JuiceFS
✓✓ Very low idle cost
✓ Highly performant
✓ Light setup
▲ High use cost

EFS workdir
Spot snapshots: JuiceFS
✓ Moderate use cost
✓ Low idle cost
▲ High maintenance

MemVerge

Jupyter notebooks that can EC2 Spot-Surf



Job Details - demo-jupyter-2023-09-26 Suspended

docker.io/memverge/jupyter_server:v3.5.0

ID	Host	User	Submitted Time	Wall Time	Cost
Y1ViXHHGCatlCKh70a1IF		awirsth	2023-09-26 13:04:24	1728h47m15s (Queue Time: 1728h47m15s)	32.8867 USD

Output

```
[I 2023-12-07 16:09:08.363 ServerApp] 302 GET / (@87.236.176.125) 0.48ms
[I 2023-12-07 16:09:08.808 LabApp] 302 GET /lab? (@87.236.176.125) 0.77ms
```

Instance ID: i-00e77bd8cd09d64c2 Zone: us-east-1d

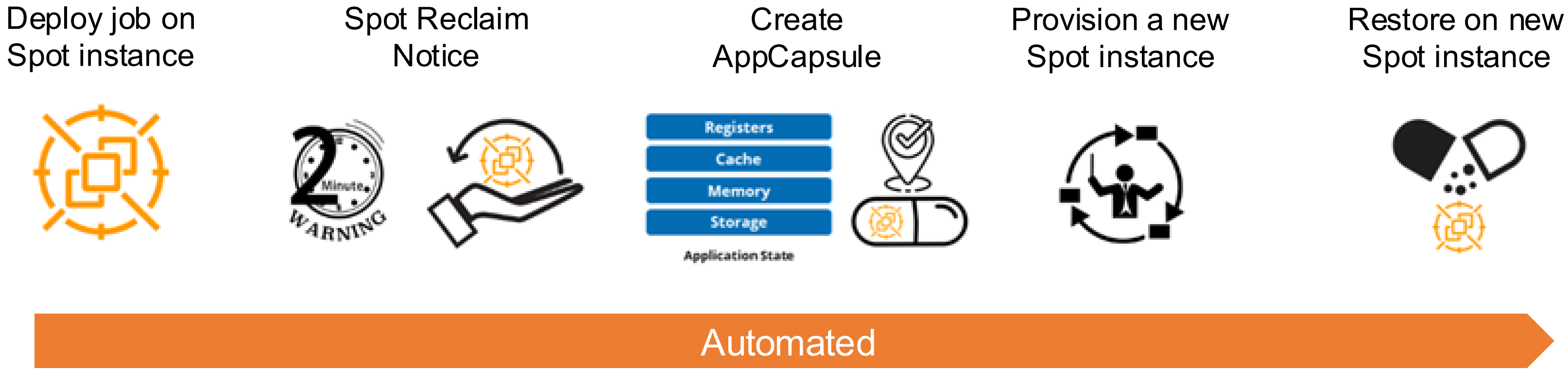
Spot t3.medium | 2 Core 4 GB

Address	Public IP	Status	Start Time	End Time	Cost
172.31.31.21:43	43.208.15.85	reclaimed	2023-09-26 13:04:36	2023-09-26 13:21:14	0.0048 USD

72 days running continuously on AWS EC2 for less than your daily coffee

How does it work?

Memory Machine provides an automated checkpoint recovery service called **SpotSurfer** that enables long-running batch and even stateful apps to run on low-cost Spot instances



Deploy job on Spot instance

Spot Reclaim Notice

Create AppCapsule

Provision a new Spot instance

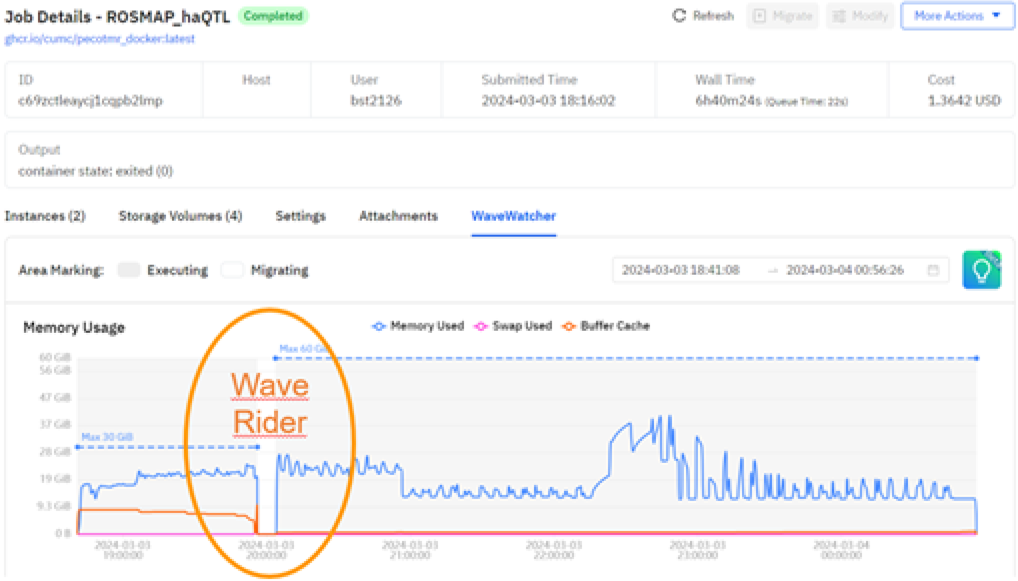
Restore on new Spot instance

Automated

If no Spot instances are available SpotSurfer can recover using on-demand instances

Have you ever run out of memory?

Memory Machine provides an automated service called **WaveRider** that migrates workloads to larger or smaller instances based on real-time CPU & memory utilization



Job Details - ROSMAP_haQTL Completed

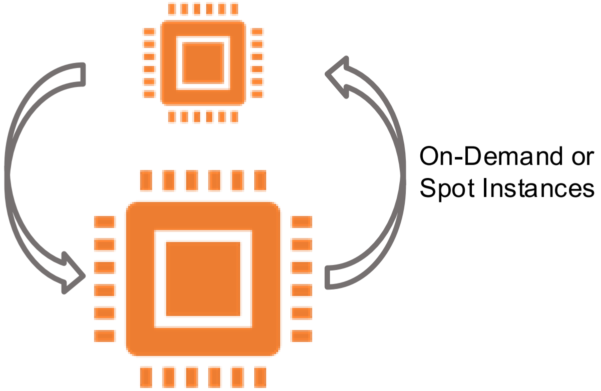
ghcr.io/cumc/pecotmr_docker:latest

ID	Host	User	Submitted Time	Wall Time	Cost
c69zctleaycj1cqpb2lmp		bst2126	2024-03-03 18:16:02	6h40m24s (Queue Time: 22s)	1.3642 USD

Output
container state: exited (0)

Area Marking: ▢ Executing ▢ Migrating

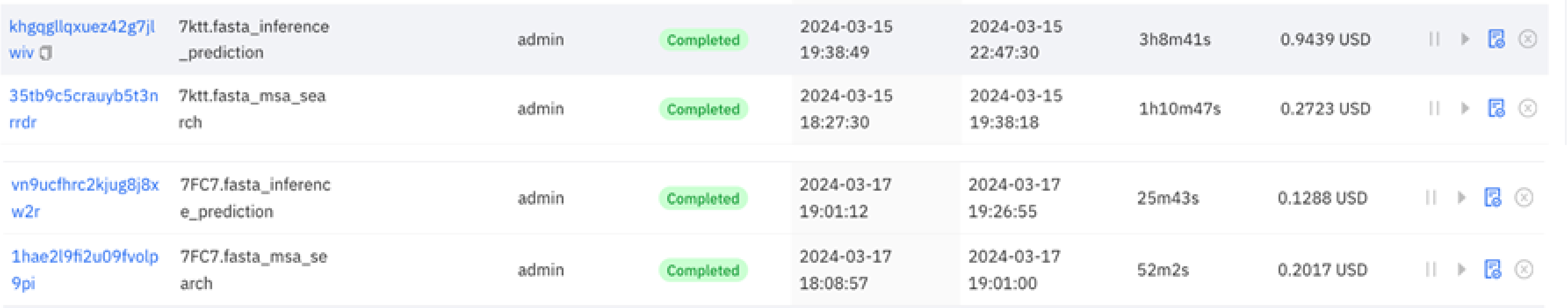
2024-03-03 18:41:08 → 2024-03-04 00:56:26



On-Demand or Spot Instances

Running Alphafold on AWS for less than $2

Alphafold is an open-source workflow developed by Google Deepmind. Its purpose is to predict protein structure base on the provided protein sequence. Alphafold can be considered as two parts: MSA search(rely on CPU) and inference prediction(heavily rely on GPU)



khgqgllqxuez42g7jlwiv	7ktt.fasta_inference_prediction	admin	Completed	2024-03-15 19:38:49	2024-03-15 22:47:30	3h8m41s	0.9439 USD
35tb9c5crauyb5t3nrrdr	7ktt.fasta_msa_search	admin	Completed	2024-03-15 18:27:30	2024-03-15 19:38:18	1h10m47s	0.2723 USD
vn9ucfhrc2kjug8j8xw2r	7FC7.fasta_inference_prediction	admin	Completed	2024-03-17 19:01:12	2024-03-17 19:26:55	25m43s	0.1288 USD
1hae2l9fi2u09fvolp9pi	7FC7.fasta_msa_search	admin	Completed	2024-03-17 18:08:57	2024-03-17 19:01:00	52m2s	0.2017 USD

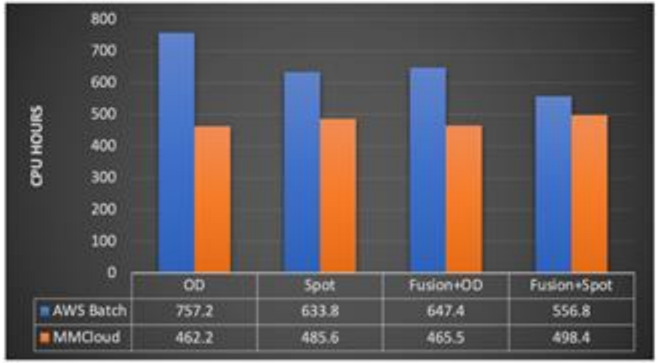
- 7ktt -- 1142 residues
- 7FC7 -- 96 residues

Cost per run = $1.22 using MMCloud

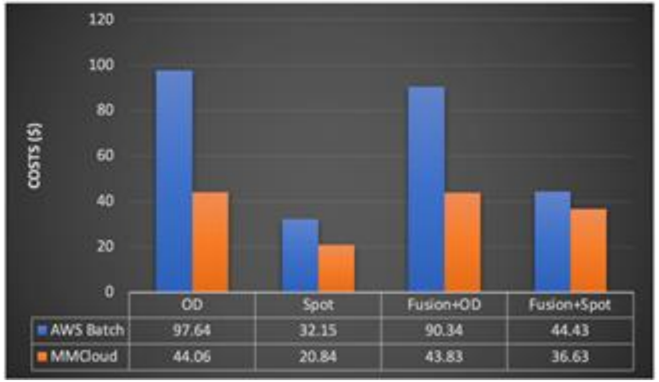
85-90% less than alternative options…

MemVerge

Benchmarking nf-core/rnaseq



	OD	Spot	Fusion+OD	Fusion+Spot
■ AWS Batch	757.2	633.8	647.4	556.8
■ MMCloud	462.2	485.6	465.5	498.4



	OD	Spot	Fusion+OD	Fusion+Spot
■ AWS Batch	97.64	32.15	90.34	44.43
■ MMCloud	44.06	20.84	43.83	36.63

☑ MemVerge delivered equal or better rnaseq performance and cost without the use of a commercial file system called Fusion from Seqera

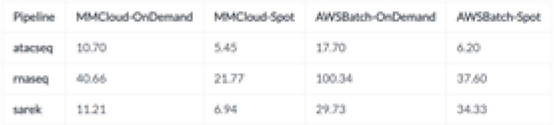
☑ MemVerge's superior EC2 automation and selection logic outperformed AWS Batch when comparing both "OD" On-Demand runs and Spot runs

☑ The cost of using MemVerge managed Spot instances for rnaseq was 78.7% lower than using AWS Batch managed On-Demand instances

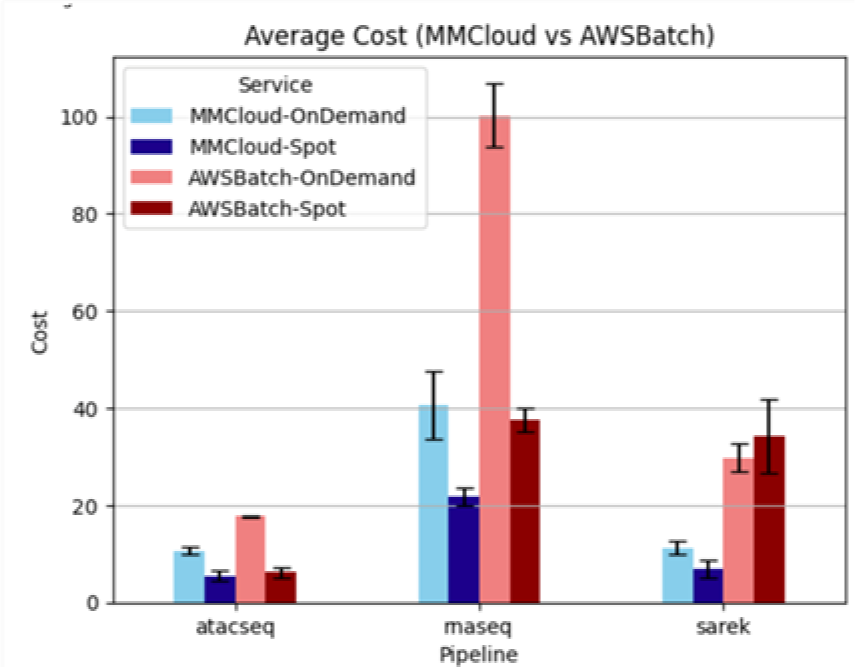
☑ MemVerge SpotSurfer can guarantee consistent CPU hours and cost per run on Spot instances at scale while AWS Batch cannot (see next slide)

78.7% Savings

Benchmarking nf-core cont’d…



Pipeline	MMCloud-OnDemand	MMCloud-Spot	AWSBatch-OnDemand	AWSBatch-Spot
atacseq	10.70	5.45	17.70	6.20
rnaseq	40.66	21.77	100.34	37.60
sarek	11.21	6.94	29.73	34.33

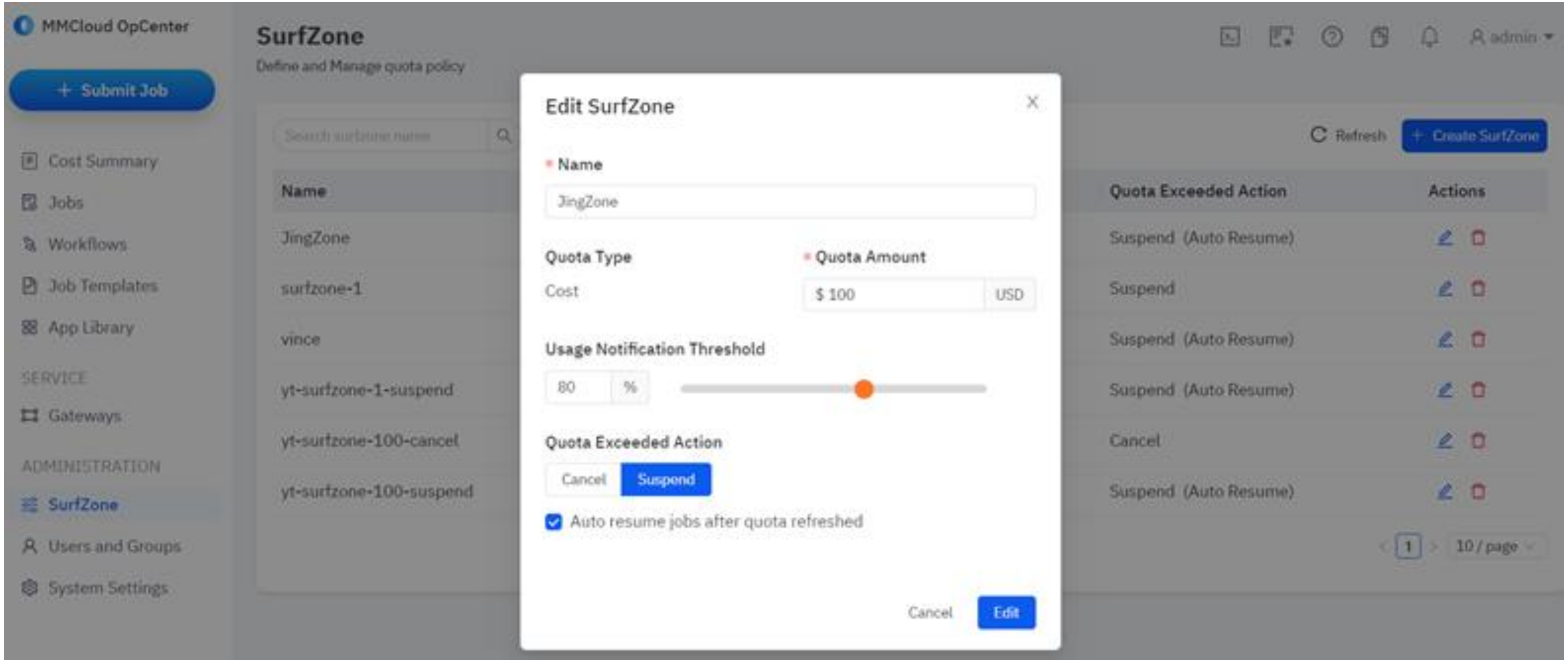


- ☑ MemVerge’s superior EC2 automation and selection logic outperformed AWS Batch when comparing both “OD” On-Demand runs and Spot runs
- ☑ The cost of using MemVerge managed Spot instances for ranged from 69% to 78% lower vs AWS Batch managed On-Demand instances
- ☑ MemVerge SpotSurfer can guarantee consistent CPU hours and cost per run on Spot instances at scale while AWS Batch cannot (see Sarek data)

69-78% Savings

Have you ever had AWS billing surprises?

Memory Machine **SurfZone** is a service that minimizes the risk of runaway compute spend by cancelling or suspending jobs when a user's "quota" is exceeded



SurfZone

Define and Manage quota policy

Name	Quota Exceeded Action	Actions
JingZone	Suspend (Auto Resume)	
surfzone-1	Suspend	
vince	Suspend (Auto Resume)	
yt-surfzone-1-suspend	Suspend (Auto Resume)	
yt-surfzone-100-cancel	Cancel	
yt-surfzone-100-suspend	Suspend (Auto Resume)	

Edit SurfZone

* Name: JingZone

Quota Type: Cost

* Quota Amount: $ 100 USD

Usage Notification Threshold: 80 %

Quota Exceeded Action: Cancel [✓ Suspend]

☑ Auto resume jobs after quota refreshed

Cancel Edit

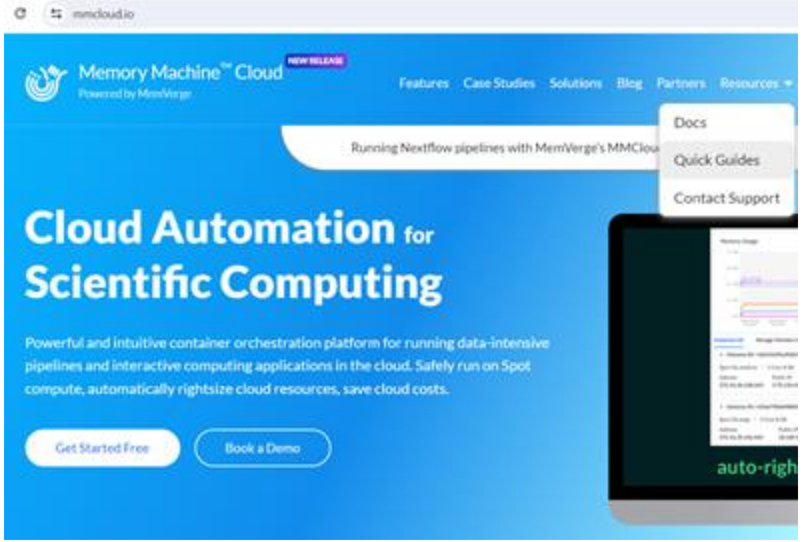
Thank you!

If you have questions or want to collaborate, problem solve, etc., please reach out!

saba.nafees@memverge.com

Interested? Here’s how to get started!

Email Jing for more info:
jing.xie@memverge.com
Quick start guides: www.mmcloud.io



Cloud Automation for Scientific Computing

Powerful and intuitive container orchestration platform for running data-intensive pipelines and interactive computing applications in the cloud. Safely run on Spot compute, automatically rightsize cloud resources, save cloud costs.

Sign up for a free trial

Slack support is available to all users
https://www.mmcloud.io/customer/register

Supplementary

Pain Point: Cloud Skills

Cloud Computing comes with a learning curve

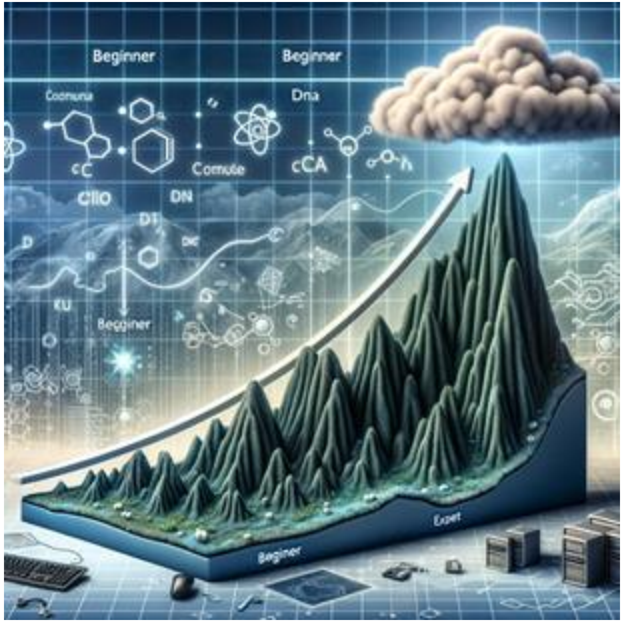
-How do I “submit a job” on AWS, GCP?

-How do I know I’m using the right compute?

-How can I optimize and run this job for less?

-How do I monitor my lab’s job(s) and user(s)?

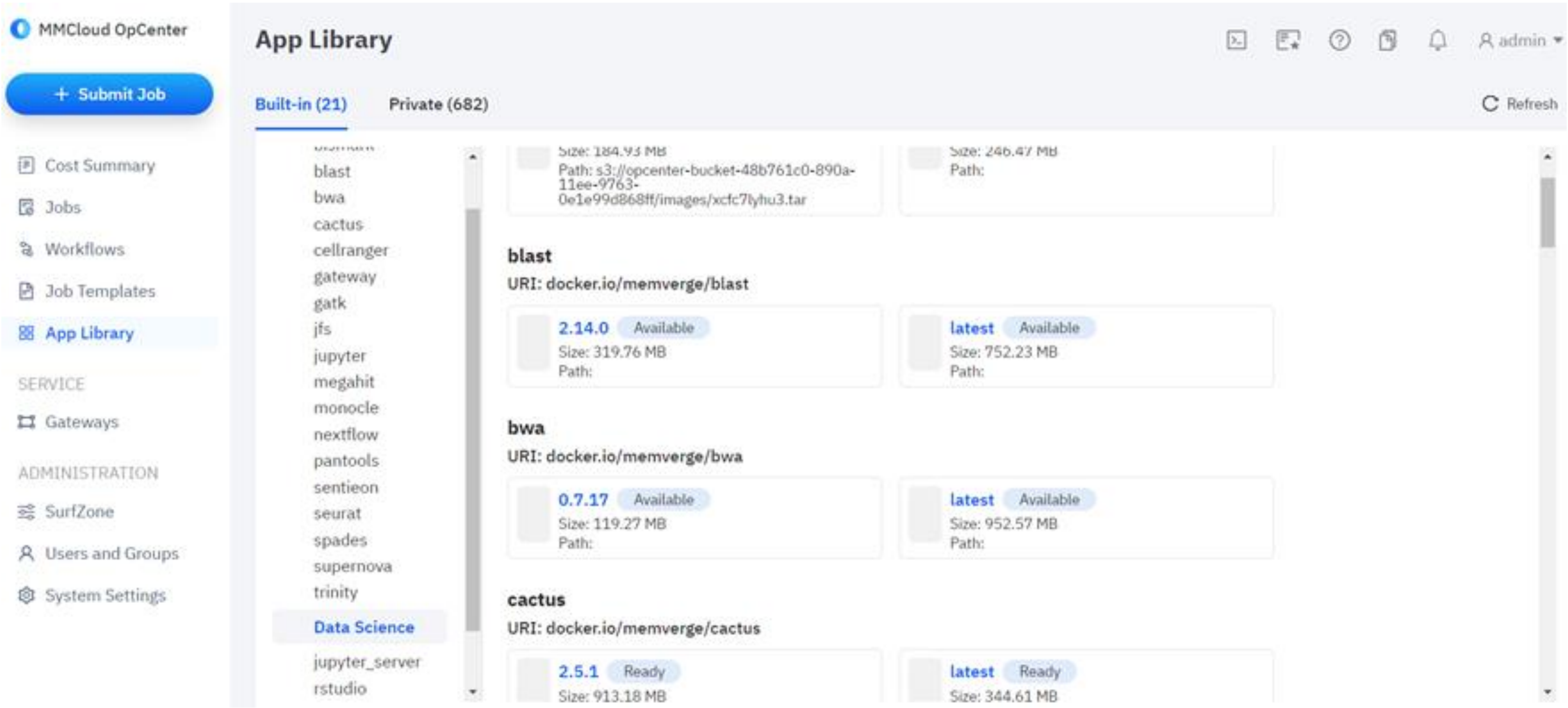
-How can I do the above in a more automated way?



How MemVerge MMCloud Cuts Cloud Costs

“WaveRider” VM auto-right-sizing	5-10%
EC2 automation, instance selection	5-10%
High perf file system automation	5-10%
Suspend / Resume / Gateway	5-10%
“SpotSurfer” vs. On Demand	50%
	50-90%

Easily run containerized apps, tools on AWS



Size: 184.93 MB
Path: s3://opcenter-bucket-48b761c0-890a-11ee-9763-0e1e99d868ff/images/xcfc7lyhu3.tar

Size: 246.47 MB
Path:

blast

URI: docker.io/memverge/blast

2.14.0 Available
Size: 319.76 MB
Path:

latest Available
Size: 752.23 MB
Path:

bwa

URI: docker.io/memverge/bwa

0.7.17 Available
Size: 119.27 MB
Path:

latest Available
Size: 952.57 MB
Path:

cactus

URI: docker.io/memverge/cactus

2.5.1 Ready
Size: 913.18 MB

latest Ready
Size: 344.61 MB

Customer testimonials

"MMCloud has helped our team in the Computational Biology Core at MDI Biological Laboratory better utilize AWS Spot EC2 instances to save cloud computing costs. In addition, the WaveWatcher tool has let us observe resource usage in real time, providing new insights into optimizing resource allocations for our work with Nextflow pipelines"

-Joel Graber, Director of the Computational Biology and Bioinformatics Core

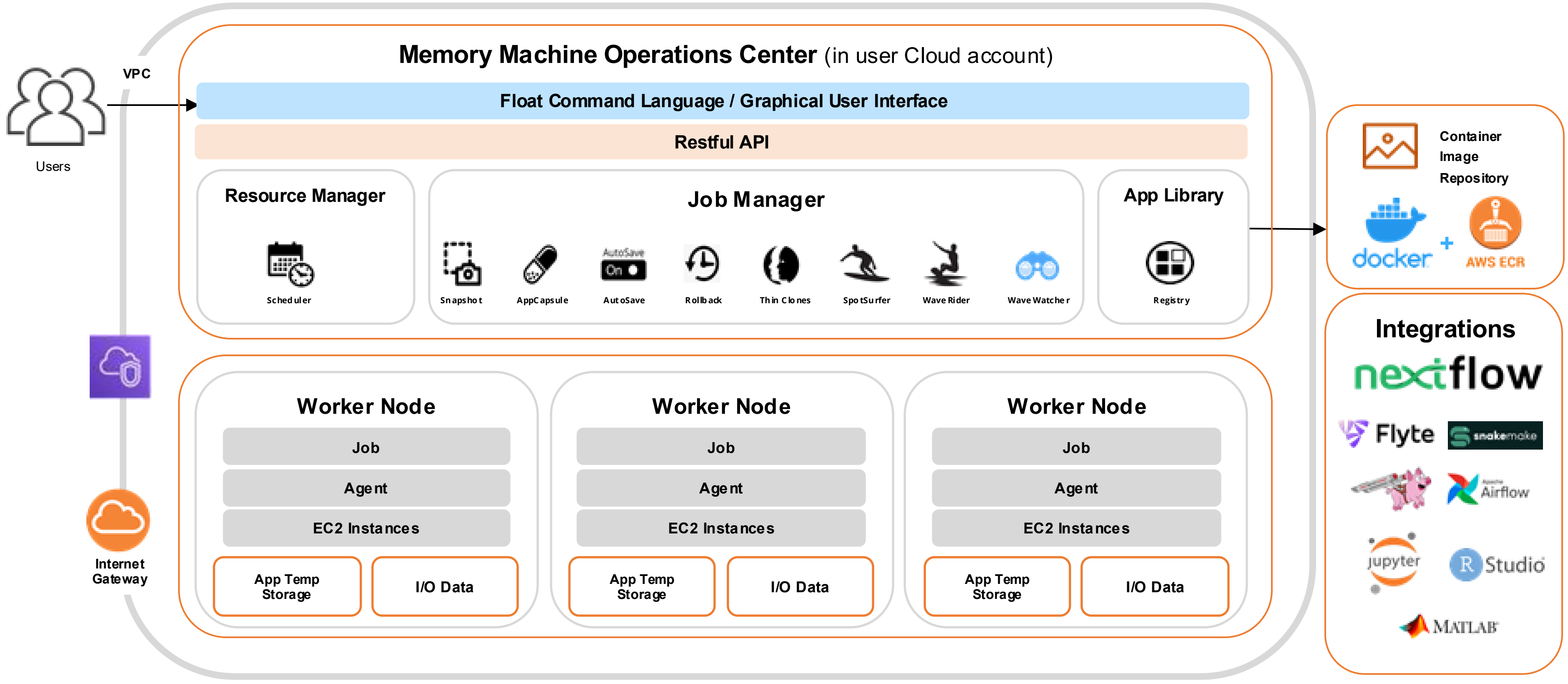
"We have been impressed by the performance of the WaveRider for Sentieon solution. Early testing shows 4-5x speed-ups out-of-the-box and we expect to improve the results as we fine tune the solution. Faster processing coupled with leveraging Spot instances is dramatically lowering our cloud costs too. This solution enables us to complete our NGS more efficiently than ever and will help elevate our research to a new level."

-Remco Ursem, Program Leader of Bioinformatics

"I was getting up to 80% batch failure rates with Spot EC2, now with SpotSurfer we have already brought failure rates due to Spot reclaims to below 1%, and we are just getting started. WaveRider picks the most appropriate VM for each of the thousands of jobs launched by Nextflow. That is giving me more efficiency than ever. Very cool."

-Vince Pagano, Senior Scientific Programmer

Full Stack Solution or Embeddable



MemVerge

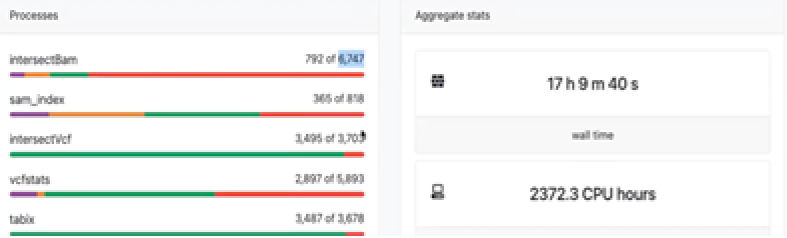
Case Study: Leading Genomics Research Institute (TGen)

Background:

Large 501(c)(3) research institution focused on accelerating genetic research was struggling to cost-effectively run Nextflow on AWS

Pain Points:

- Huge # of worker nodes, > 20K+ VMs per workflow
- ~75% failure rate using when using Spot EC2



Processes

intersectBam 792 of 6,747

sam_index 365 of 818

intersectVcf 3,495 of 3,703

vcfstats 2,897 of 5,893

tabix 3,487 of 3,678

Aggregate stats

17 h 9 m 40 s

wall time

2372.3 CPU hours

Solution:

Memory Machine Cloud on AWS

Results:

70% EC2 savings

Reduced Failure Rate to Zero

Real-time automated EC2 optimization

nextflow **Integration**

SpotSurfer
Spot Protection

WaveRider
Resource Rightsizing

Memory Machine Differentiators

Current implementation supports scaling to 1,000 concurrent worker nodes per Memory Machine Op Center, larger solutions can be supported via multiple Memory Machine Op Centers.

Case Study: Leading Genomic Medicines Company

Background:

Large biotech pioneering the future of genomic medicine with multiple clinical and pre-clinical product pipelines

Pain Points:

- Limited IT budget, multiple days to get compute spun up for researchers
- No easy way to refactor pipelines and cost-optimize for running in the cloud

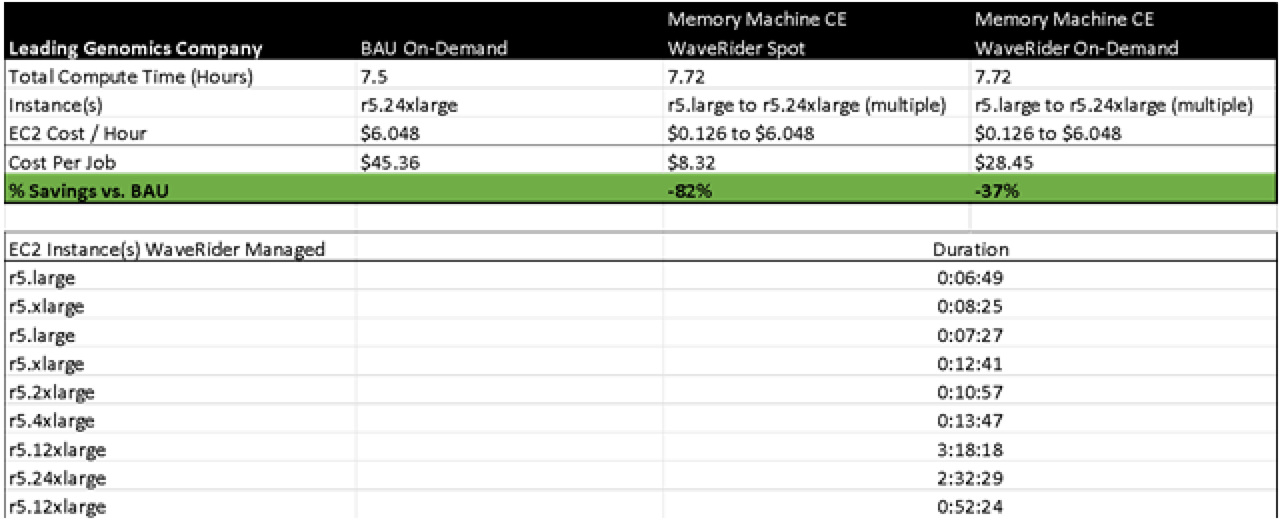
Solution:

Memory Machine Cloud on AWS

Results:

>35% cost savings w/ On-Demand EC2
>80% cost savings w/ Spot EC2

AWS resource provisioning in seconds vs. days

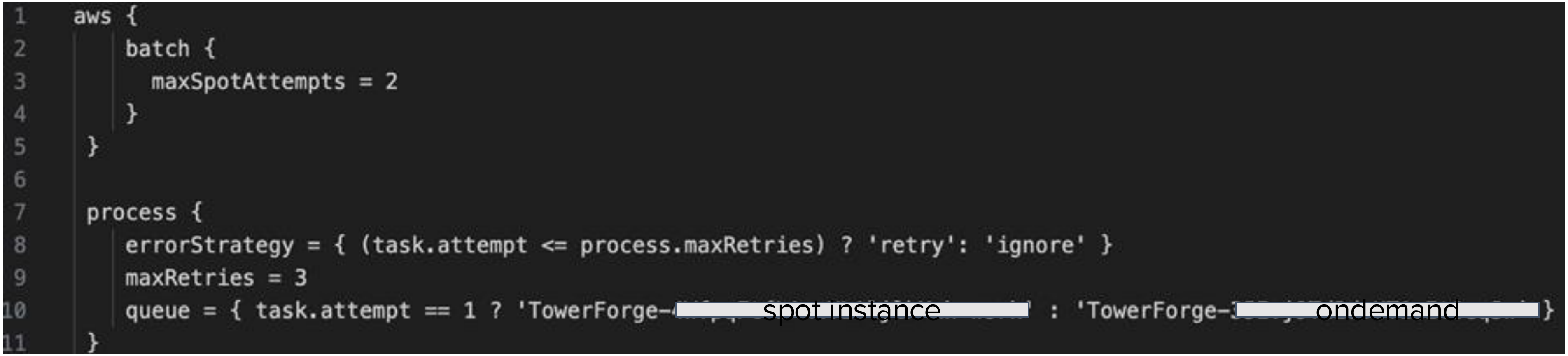


Leading Genomics Company	BAU On-Demand	Memory Machine CE WaveRider Spot	Memory Machine CE WaveRider On-Demand
Total Compute Time (Hours)	7.5	7.72	7.72
Instance(s)	r5.24xlarge	r5.large to r5.24xlarge (multiple)	r5.large to r5.24xlarge (multiple)
EC2 Cost / Hour	$6.048	$0.126 to $6.048	$0.126 to $6.048
Cost Per Job	$45.36	$8.32	$28.45
% Savings vs. BAU		**-82%**	**-37%**

EC2 Instance(s) WaveRider Managed	Duration
r5.large	0:06:49
r5.xlarge	0:08:25
r5.large	0:07:27
r5.xlarge	0:12:41
r5.2xlarge	0:10:57
r5.4xlarge	0:13:47
r5.12xlarge	3:18:18
r5.24xlarge	2:32:29
r5.12xlarge	0:52:24

Optimizing AWS resources

- Reduce the # of spot attempts under the hood
- Error strategy
  - Retry
  - Ignore
- Tell nextflow to go from spot to ondemand after 1 attempt so that resource consuming tasks can be done via ondemand



```
aws {
    batch {
      maxSpotAttempts = 2
    }
 }

 process {
    errorStrategy = { (task.attempt <= process.maxRetries) ? 'retry': 'ignore' }
    maxRetries = 3
    queue = { task.attempt == 1 ? 'TowerForge-[redacted]spot instance : 'TowerForge-[redacted]ondemand}
 }
```