WEIZMANN INSTITUTE OF SCIENCE

WEXAC workshop GUI tools and efficient WEXAC usage

Vadim Malkin
Head of HPC Section

July 25, 2023

Agenda:

- WEXAC Fast track: Wiki and useful links page.
- WEXAC real time monitor tool: RTM
- WEXAC Best Practices and our recommendations.
- WEXAC LSF Explorer, Grafana.
- WEXAC new tools: PAC, Jupytherhub, Rstudio, etc.
- WEXAC Private cloud options.
- WEXAC and Public cloud integration (AWS).

Ways to contact HPC people:

- Email: hpc@weizmann.ac.il
- Mailing list: wexac-news@weizmann.ac.il
- WhatsApp Channel: https://chat.whatsapp.com/05qVVCCcR8v9234vit28gc
- Office location: De Picciotto A building, rooms 434-436

Shlomit Afgin
Biology HPC

Lior Amir
BioChemistry
HPC

Vadim Malkin
Head of HPC

Igor Chebotar
WAIC

Alexey Konvisher
Physics HPC

Yuval Mankali
DevOps

Tal Malach
CloudOps

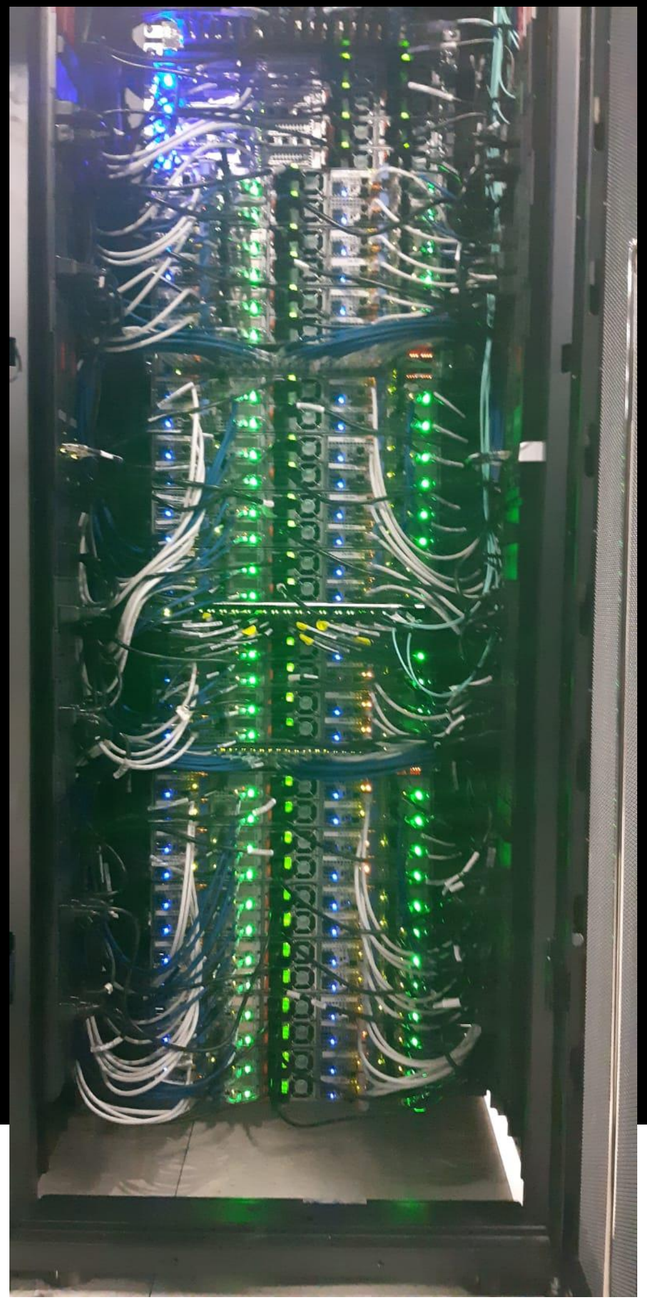
Eli Biezuher
PM

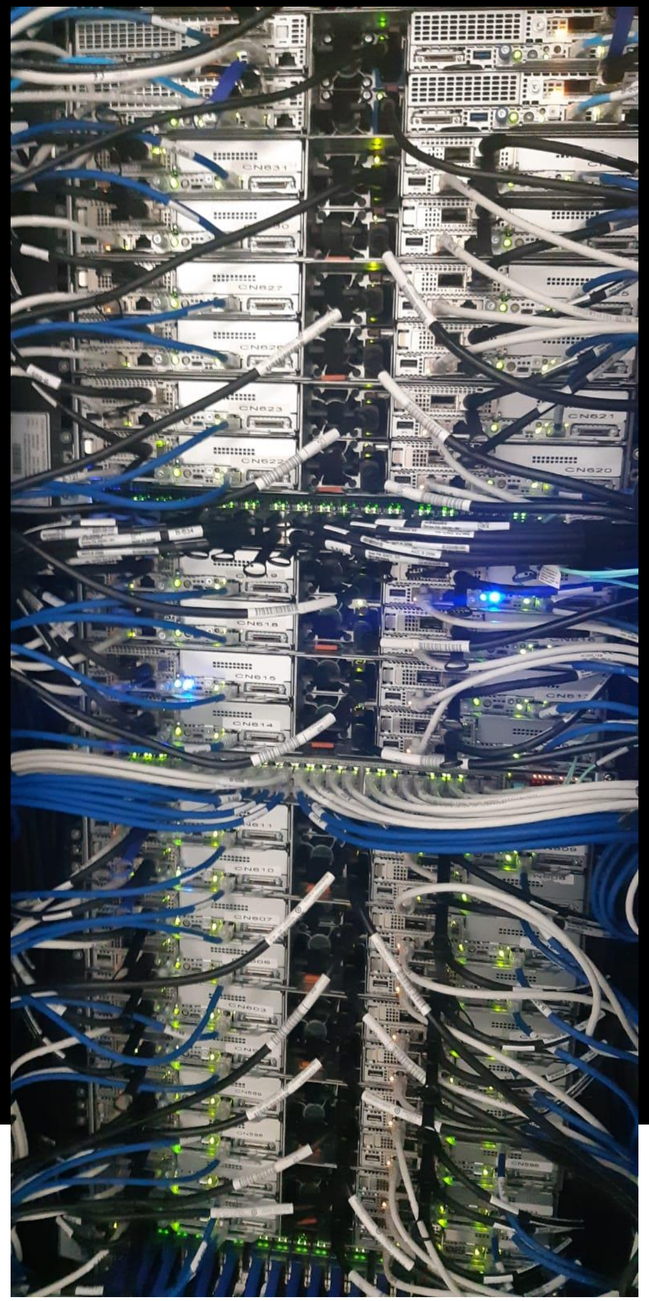
Yaron Weitz
HPC Expert

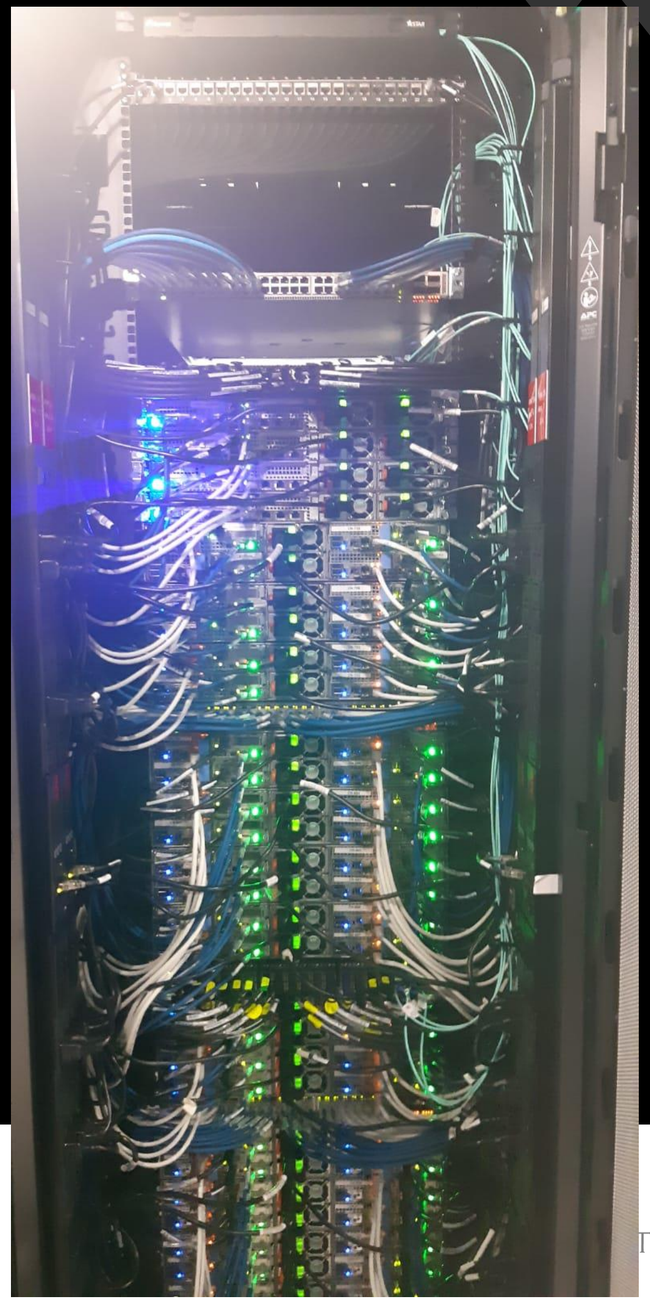
Jerry Mersel
Programmer

People behind WEXAC:

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE



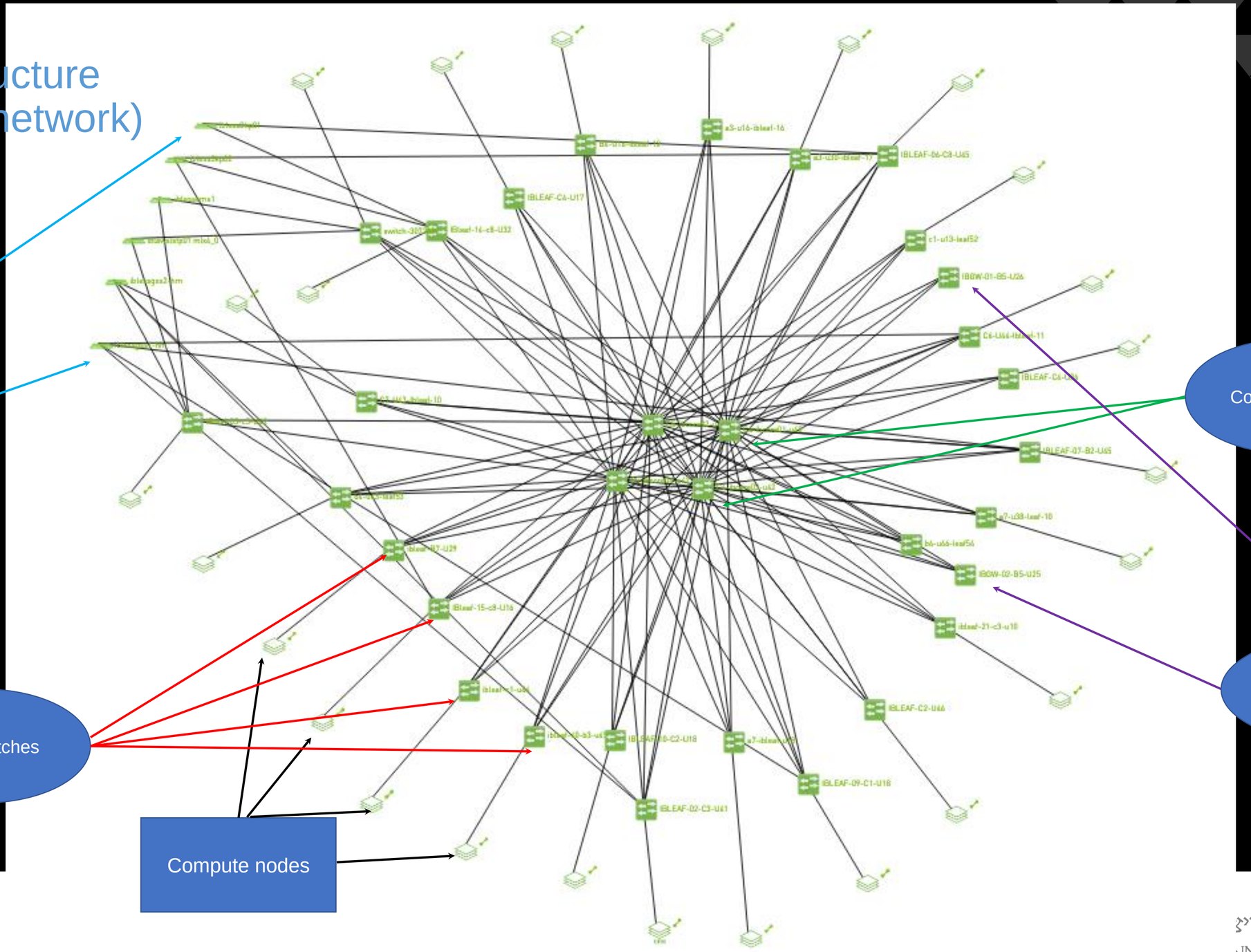




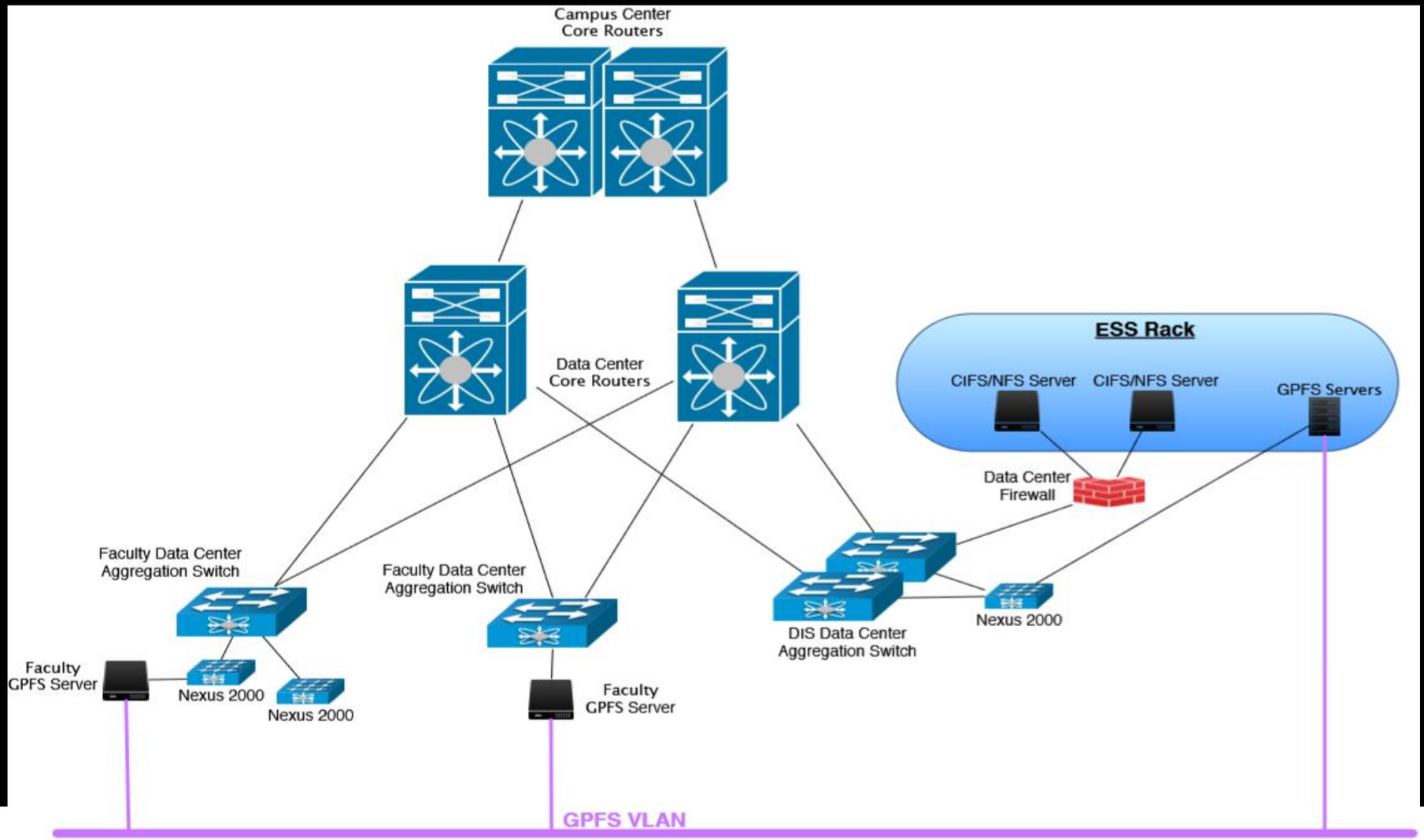
TE OF SCIENCE

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

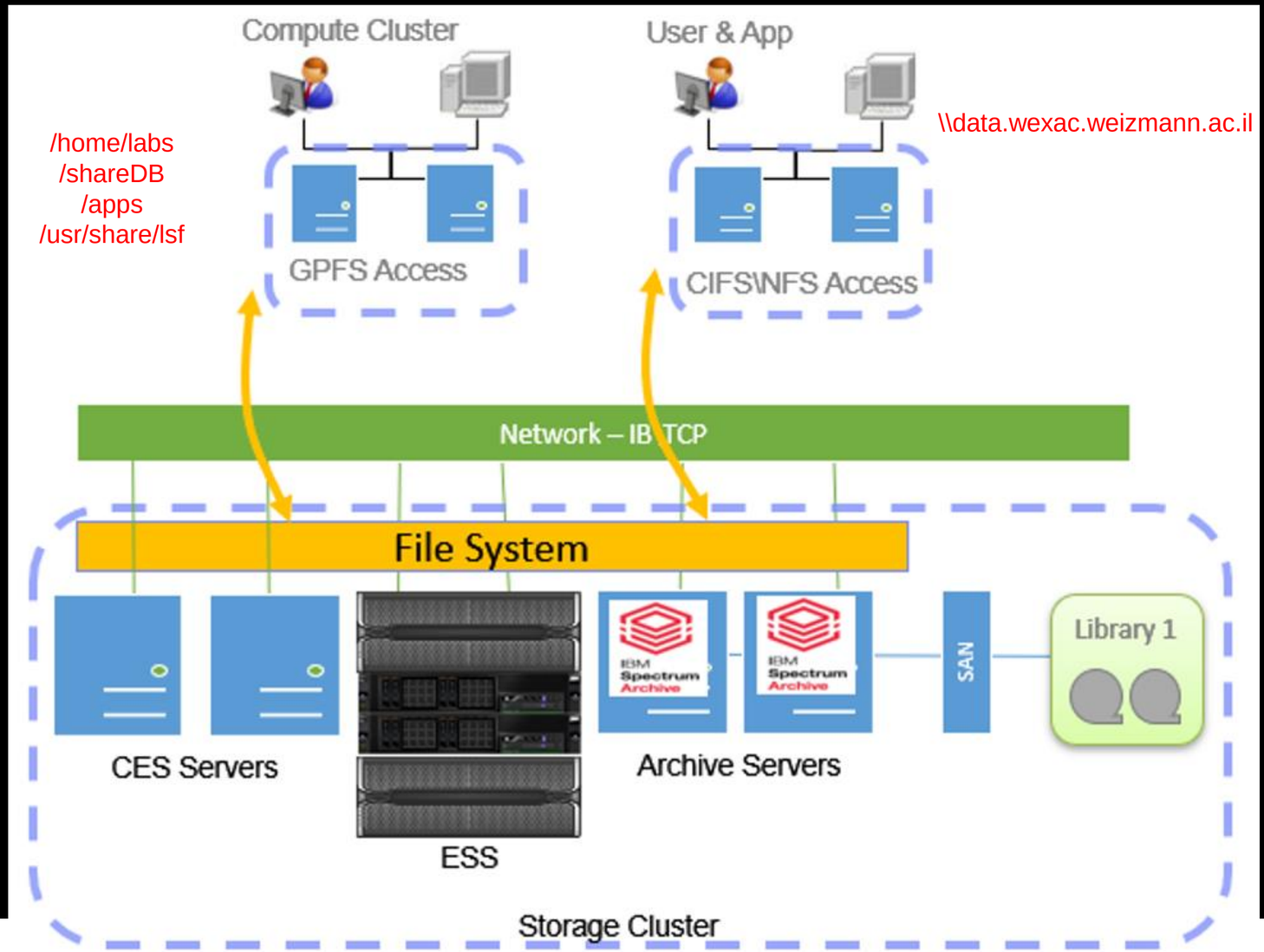
WEXAC structure (InfiniBand network)



WEXAC structure (Ethernet network)



WEXAC structure (IBM storage):



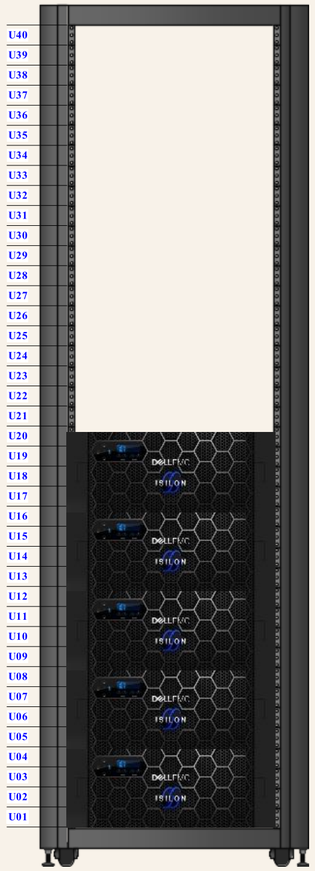
WEXAC structure (Isilon storage):

Single File System
With two storage pools:

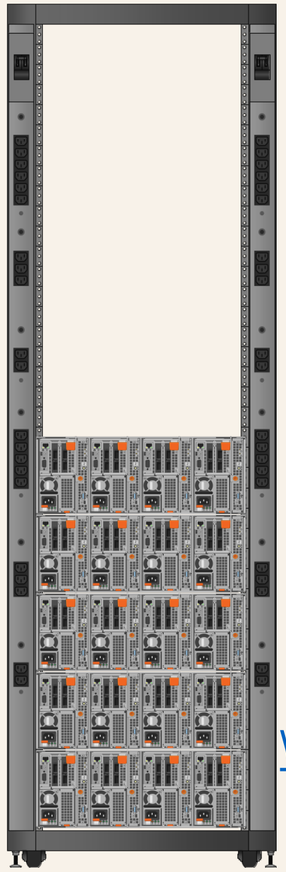
- 8 x F800
- 12 x A2000

Each pool will serve different use case :

- F800 for AI and HPC workloads
- A2000 as NL400 replacement for StorWIS service



Front View



Rear View

Standard File Sharing Services
12 X A2000 ~2,163TIBU
12 x A2000, 240TB, 64GB, 2X10GbE + 2X8G FC W/ OPTICS

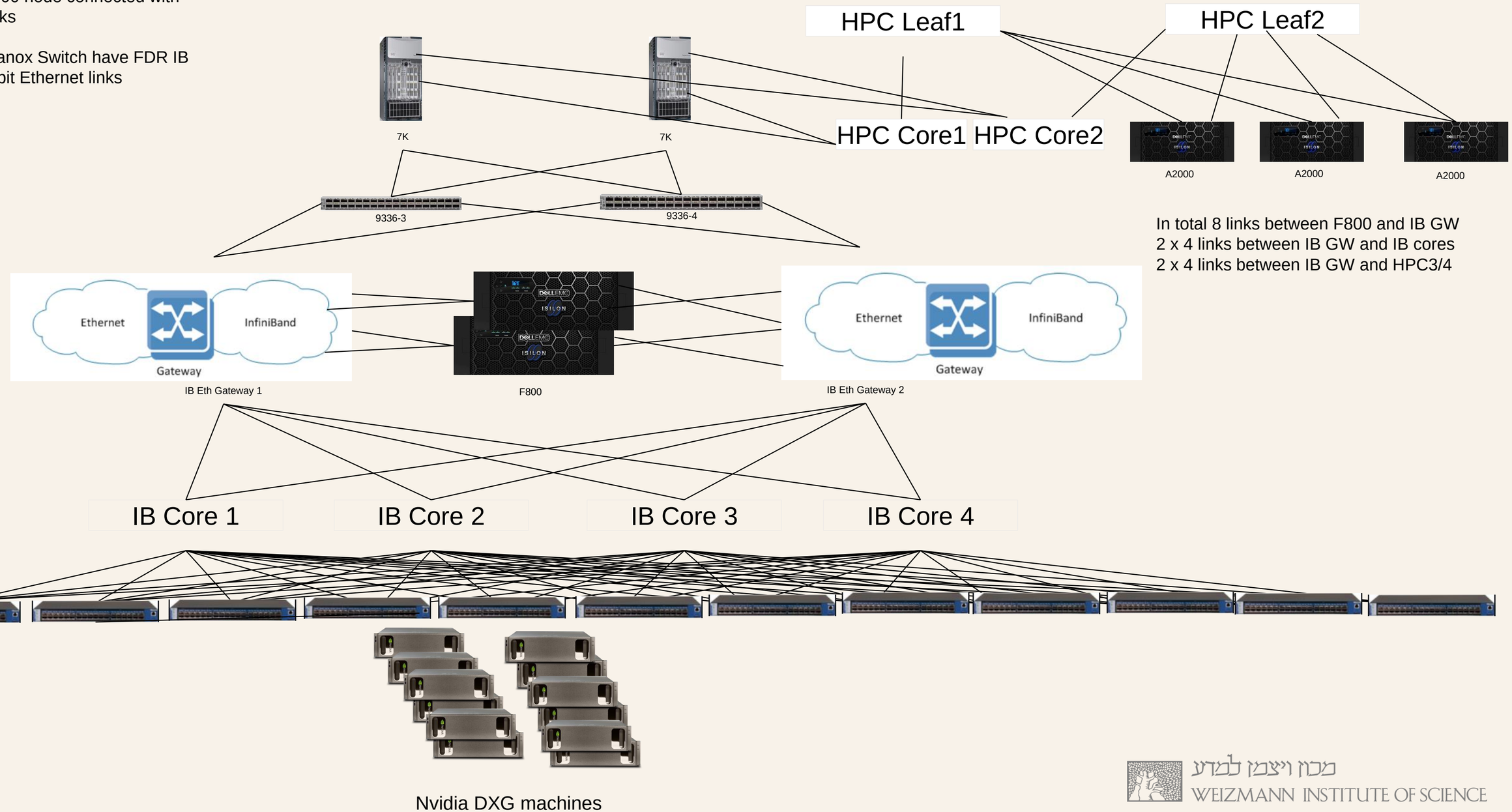
\\isi.storwis.weizmann.ac.il\labs
\\isi.storwis.weizmann.ac.il\users

HPC and DL/AI
8 X F800 ~302TIBU 5Y
8 x F800, 48TB, 256GB, 2x40GbE (QSFP+) F800
Backend Network
2 x 36 x Ports QDR Mellanox Switch

\\isi.bigdata.weizmann.ac.il\projects

Each Isilon F800 node connected with two 40GbE links

Gateway Mellanox Switch have FDR IB links and 40Gbit Ethernet links



In total 8 links between F800 and IB GW
2 x 4 links between IB GW and IB cores
2 x 4 links between IB GW and HPC3/4

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

WEXAC fast track

- Wiki - https://hpcwiki.weizmann.ac.il/
- Useful links - https://appsrv.wexac.weizmann.ac.il/

hpc@weizmann.ac.il

LAST EDITED BY
Administrator
07/06/2023

Backup Restore

Archive

See https://www.weizmann.ac.il/DIS/wexac-archive-facility

if you need to archive a big folder. It is better to create tar files not bigger than 100G.

For creating the tar:

Change DIRECTORY_NAME to the directory you need to archive and *YOUR_LAB to your lab name.*

Change the directory to the path where the directory DIRECTORY_NAME, you want to archive exists and run:

```
tar czf - DIRECTORY_NAME |  split --bytes=100G - /home/archive/labs/YOUR_LAB/DIRECTORY_NAME.tar.gz.
```

It will create tar files with the names DIRECTORY_NAME.tar.gz.aa, DIRECTORY_NAME.tar.gz.ab, ... each file will have size of 100GB.

To list the contact of the tar files, run:

```
cat /home/archive/labs/YOUR_LAB/DIRECTORY_NAME.tar.gz.* | tar tzf -
```

To extract the folder to the current directory, run:

```
cat /home/archive/labs/YOUR_LAB/DIRECTORY_NAME.tar.gz.* | tar xzvf -
```

Restore

If you lost your file/directory within the last 2 weeks, you can find your file in the snapshot.

- If your home directory in /home/labs/...
  ```
  cd .snapshots
  ```
- if your home directory in /home/projects/...
  ```
  cd .snapshot
  ```

Please note, you cannot see the snapshot directories with the command ls or ls -a.

Under snapshots, you will have directories whose names are dates.

Go to the date you need. Under that date, you will see the tree of your files.

Copy the files/directories to your home directory.

WEXAC Application Server

Software name	URLs
Rstudio Server	rstudio rstudio2
Jupytherhub	access compute access1 compute access2 access3 access4
RTM	IBM Spectrum LSF RTM
Platform Application Center	PAC
Access servers Monitoring	Monitoring
LSF Explorer	Explorer
Ganglia	Compute nodes Access servers
Wexac DOCS	Docs
Wexac Workshops	Oct 2020: mp4 timestamps pdf July 19, 2021: mp4-beginners pdf-beginners July 26, 2021: mp4-advanced pdf-advanced pdf-nvidia July 19, 2022: presentation slides

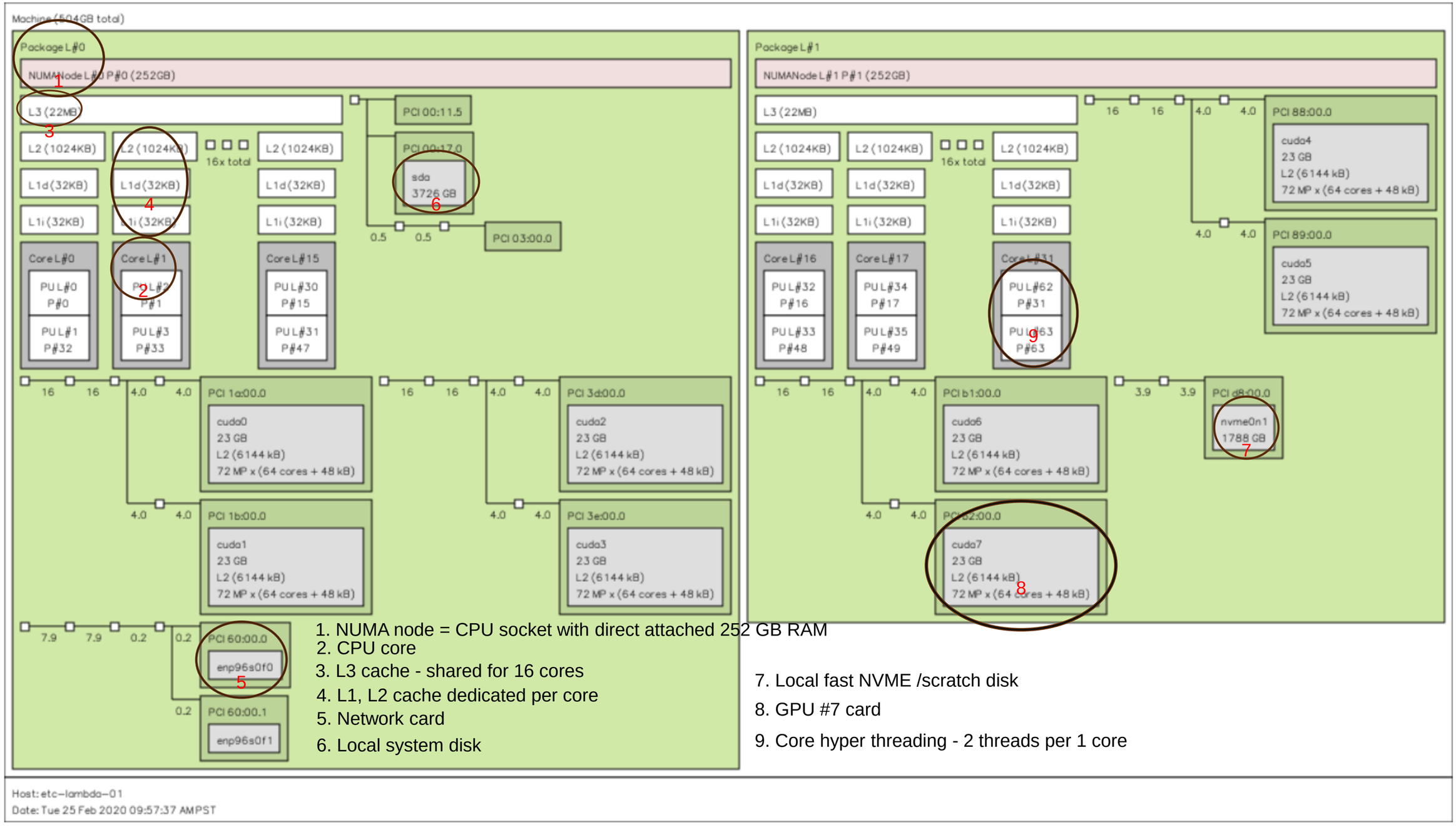
hpc@weizmann.ac.il

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

WEXAC basic concepts

- Memory: 2 types
  - CPU memory defined by -R “rusage[mem=1024]” in case of 1GB required
  - GPU memory defined as a part og GPU resources -gpu num=1:gmem=1000 means 1GB GPU RAM
- Memory allocated as # of cores multiply requested memory
- Tasks: multiply all resources. -n 4 -R “rusage[mem=1024]” -gpu num=1 means 4 cores, 4 GB CPU RAM, 4 GPU

WEXAC basic concepts



1. NUMA node = CPU socket with direct attached 252 GB RAM
2. CPU core
3. L3 cache - shared for 16 cores
4. L1, L2 cache dedicated per core
5. Network card
6. Local system disk

7. Local fast NVME /scratch disk
8. GPU #7 card
9. Core hyper threading - 2 threads per 1 core

WEXAC basic concepts

- Affinity awareness: NUMA nodes, sockets, cores, threads resources are not multiplied all other resources
  - bsub -n 2 -q gpu-short -gpu num=2:gmem=1000:aff=yes -R"span[hosts=1] rusage[mem=100]affinity[core(3,same=numa,exclusive=(numa,injob)):cpubind=numa:membind=localonly]" ***yourcode***
  - This bsub command means to start 2 independent tasks when both tasks locality will be on the same execution host and each task will use 2 GPU with strict affinity CPU - GPU. Each task will use 3 cores on same NUMA level using exclusively NUMA during job execution when CPU bind is on NUMA level as well and use only memory attached locally to specific NUMA
  - bsub -q gpu-short -gpu num=4:gmem=1000:aff=yes -R"span[hosts=1] rusage[mem=100]affinity[core(6,same=numa,exclusive=(numa,injob)):cpubind=numa:membind=localonly]"***yourcode***
  - In this case the same 4 GPU's and 6 CPU cores will act together in the same NUMA binding. Important for tightly coupled CPU/GPU workloads
- Bottom line: if you are using tasks with -n flag your code must be aware of it and spread independent tasks to different resources allocated by LSF. Otherwise, it will be just resources wasting.

WEXAC basic concepts

```
EXTERNAL MESSAGES:
MSG_ID FROM       POST_TIME      MESSAGE                                    ATTACHMENT
0      vadimm     Jul 23 12:40   hgn07:gpus=3,1,5,7;                            N

RESOURCE REQUIREMENT DETAILS:
Combined: select[(type = any ) && (type == any)] order[-slots:-ngpus_physical]
                    rusage[mem=100.00] span[hosts=1] affinity[core(3,same=num
                   a,exclusive=(numa,injob))*1:cpubind=numa:membind=localonly
                   ]
Effective: select[(((type = any ) && (type == any))) && (ngpus>0)] order[-slot
                   s:-ngpus_physical] rusage[mem=100.00,ngpus_physical=2.00:g
                   mem=1000.00] span[hosts=1] affinity[core(3,same=numa,exclu
                   sive=(numa,injob))*1:cpubind=numa:membind=localonly]

AFFINITY:
                   CPU BINDING                                MEMORY BINDING
                   -----------                                --------------
HOST               TYPE   LEVEL  EXCL   IDS                  POL   NUMA SIZE
hgn07              core   numa   numa   /0/0/3               local 0    100.0MB
                                        /0/0/4
                                        /0/0/8
hgn07              core   numa   numa   /1/1/0               local 1    100.0MB
                                        /1/1/1
                                        /1/1/2

GPU REQUIREMENT DETAILS:
Combined: num=2:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
Effective: num=2:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
```

WEXAC basic concepts

```
EXTERNAL MESSAGES:
MSG_ID FROM       POST_TIME      MESSAGE                                  ATTACHMENT
0      vadimm     Jul 23 12:55   hgn05:gpus=5,6,4,7;                          N

RESOURCE REQUIREMENT DETAILS:
Combined: select[(type = any ) && (type == any)] order[-slots:-ngpus_physical]
                     rusage[mem=100.00] span[hosts=1] affinity[core(6,same=num
                    a,exclusive=(numa,injob))*1:cpubind=numa:membind=localonly
                    ]
Effective: select[(((type = any ) && (type == any))) && (ngpus>0)] order[-slot
                    s:-ngpus_physical] rusage[mem=100.00,ngpus_physical=4.00:g
                    mem=1000.00] span[hosts=1] affinity[core(6,same=numa,exclu
                    sive=(numa,injob))*1:cpubind=numa:membind=localonly]

AFFINITY:
                    CPU BINDING                                 MEMORY BINDING
                    -----------                                 --------------
HOST                TYPE   LEVEL  EXCL   IDS                   POL   NUMA SIZE
hgn05               core   numa   numa   /1/1/0                local 1    100.0MB
                                         /1/1/1
                                         /1/1/2
                                         /1/1/3
                                         /1/1/4
                                         /1/1/8

GPU REQUIREMENT DETAILS:
Combined: num=4:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
Effective: num=4:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
```

WEXAC basic concepts

- Memory – overall job memory reservation calculated as # of cores * required memory. Example if you asked for 4 cores and 4 GB RAM – 16 GB RAM will be allocated for this job. So if only 4 GB RAM required for 4 cores jobs ask for 1GB RAM and 4 cores.
- bsub –q new-short –n 4 –R “rusage[mem=1024]” *yourscript*

```
[vadimm@access4 ~]$ bsub -q new-short -n 4 -R "rusage[mem=1024]" sleep 999
Memory reservation is (MB): 1024
Memory Limit is (MB): 1024

Amount of tasks (-n X) reserved: 4
=== Your total amount of memory reservation for this job is (MB): 4096 ===

Job <875152> is submitted to queue <new-short>.
[vadimm@access4 ~]$ bsub -q new-short -n 4 -R "rusage[mem=4096]" sleep 999
Memory reservation is (MB): 4096
Memory Limit is (MB): 4096

Amount of tasks (-n X) reserved: 4
=== Your total amount of memory reservation for this job is (MB): 16384 ===

Job <875459> is submitted to queue <new-short>.
```

WEXAC basic concepts

- GPU – allocated as # of tasks * requested GPU. For instance if you ask for 4 tasks and 2 GPU system will try to assign 8 GPU for your task causing extreme pending time and resources wasting ! Please use cores instead fo tasks.
- bsub -q gpu-short -n 4 -gpu num=2:j_exclusive=yes -o ~/output-%J.out -e ~/error-%J.err nvidia-smi

```
[vadimm@access4 ~]$ bjobs -l 879732

Job <879732>, User <vadimm>, Project <default>, User Group <testing-wx-grp-lsf>
                     , Status <DONE>, Queue <gpu-short>, Job Priority <50>, Com
                     mand <nvidia-smi>, Share group charged </vadimm>, Esub <me
                     m nonre group gpu>
Thu Jul 20 16:31:52: Submitted from host <access4>, CWD <$HOME>, Output File <o
                     utput_879732.out>, Error File <error-879732.err>, Re-runna
                     ble, 4 Task(s), Requested GPU <num=2:j_exclusive=yes>;
Thu Jul 20 16:31:53: Started 4 Task(s) on Host(s) <1*ibdgx005> <2*hgn01> <1*gac
                     cess06>, Allocated 4 Slot(s) on Host(s) <1*ibdgx005> <2*hg
                     n01> <1*gaccess06>, Execution Home </home/labs/testing/vad
                     imm>, Execution CWD </home/labs/testing/vadimm>;
Thu Jul 20 16:31:55: Done successfully. The CPU time used is 0.3 seconds.
                     HOST: ibdgx005; CPU_TIME: 0 seconds
```

Bad example – don't try at home!

WEXAC basic concepts

```
EXTERNAL MESSAGES:
MSG_ID FROM       POST_TIME      MESSAGE                                    ATTACHMENT
0      vadimm     Jul 20 16:31   gacces:06:gpus=2,1:ngn01:gpus=6,2,1            N

RESOURCE REQUIREMENT DETAILS:
Combined: select[(type = any ) && (type == any)] order[-slots:-ngpus_physical]
                     affinity[core(1)*1]
Effective: select[((type = any ) && (type == any)) && (ngpus>0)] order[-slots:
                    -ngpus_physical] rusage[ngpus_physical=2.00:gmem=6248.00]
                    affinity[core(1)*1]

GPU REQUIREMENT DETAILS:
Combined: num=2:mode=shared:mps=no:j_exclusive=yes:gmem=6248.00:aff=no:gvendor
                    =nvidia
Effective: num=2:mode=shared:mps=no:j_exclusive=yes:gmem=6248.00:aff=no:gvendo
                    r=nvidia
```

Bad example – don't try at home!

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

WEXAC basic concepts

```
Sender: LSF System <lsfadmin@weizmann.ac.il>
Subject: Job 879732: <nvidia-smi> in cluster <wexac> Done

Job <nvidia-smi> was submitted from host <access4> by user <vadimm> in cluster <wexac> at Thu Jul 20 16:31:52 2023
Job was executed on host(s) <1*ibdgx005>, in queue <gpu-short>, as user <vadimm> in cluster <wexac> at Thu Jul 20 16:3
1:53 2023
                            <2*hgn01>
                            <1*gaccess06>

</home/labs/testing/vadimm> was used as the home directory.
```

```
[vadimm@access4 ~]$ cat output-879732.out
Thu Jul 20 16:31:55 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.60.13    Driver Version: 525.60.13    CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:0A:00.0 Off |                    0 |
| N/A   35C    P0    42W / 300W |     75MiB / 32768MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:86:00.0 Off |                    0 |
| N/A   35C    P0    44W / 300W |    109MiB / 32768MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Bad example – don’t try at home!

WEXAC basic concepts

- GPU – allocated as # of tasks * requested GPU. For instance, if you ask for 4 tasks and 2 GPU system will try to assign 8 GPU for your task causing extreme pending time and resources wasting ! Please use cores instead fo tasks.
- bsub -q gpu-short -gpu num=2:j_exclusive=yes -R affinity[core*4] -o ~/output-%J.out -e ~/error-%J.err nvidia-smi

```
[vadimm@access4 ~]$ bjobs -l 880501

Job <880501>, User <vadimm>, Project <default>, User Group <testing-wx-grp-lsf>
                     , Status <DONE>, Queue <gpu-short>, Job Priority <50>, Com
                     mand <nvidia-smi>, Share group charged </vadimm>, Esub <me
                     m nonre group gpu>
Thu Jul 20 16:47:10: Submitted from host <access4>, CWD <$HOME>, Output File <o
                     utput-880501.out>, Error File <error-880501.err>, Re-runna
                     ble, Requested Resources <affinity[core*4]>, Requested GPU
                      <num=2:j_exclusive=yes>;
Thu Jul 20 16:47:11: Started 1 Task(s) on Host(s) <hgn01>, Allocated 4 Slot(s)
                     on Host(s) <4*hgn01>, Execution Home </home/labs/testing/v
                     adimm>, Execution CWD </home/labs/testing/vadimm>;
Thu Jul 20 16:47:12: Done successfully. The CPU time used is 0.2 seconds.
                     HOST: hgn01; CPU_TIME: 0 seconds
```

Good example

WEXAC basic concepts

```
EXTERNAL MESSAGES:
MSG_ID FROM       POST_TIME      MESSAGE                                    ATTACHMENT
0      vadimm     Jul 20 16:47   hgn01:gpus=0,1;                                N

RESOURCE REQUIREMENT DETAILS:
Combined: select[(type = any ) && (type == any)] order[-slots:-ngpus_physical]
                    affinity[core(1)*4]
Effective: select[(((type = any ) && (type == any))) && (ngpus>0)] order[-slot
                   s:-ngpus_physical] rusage[ngpus_physical=2.00:gmem=6248.00
                   ] affinity[core(1)*4]

GPU REQUIREMENT DETAILS:
Combined: num=2:mode=shared:mps=no:j_exclusive=yes:gmem=6248.00:aff=no:gvendor
                   =nvidia
Effective: num=2:mode=shared:mps=no:j_exclusive=yes:gmem=6248.00:aff=no:gvendo
                   r=nvidia
```

Good example

WEXAC basic concepts

- GPU – allocated as # of tasks * requested GPU. For instance, if you ask for 4 tasks and 2 GPU system will try to assign 8 GPU for your task causing extreme pending time and resources wasting ! Please use cores instead fo tasks.
- bsub -q gpu-short -gpu num=2:gmem=1000:aff=yes -R affinity[core*4] -o ~/output-%J.out -e ~/error-%J.err nvidia-smi

```
EXTERNAL MESSAGES:
MSG_ID FROM       POST_TIME      MESSAGE                                   ATTACHMENT
0      vadimm     Jul 20 17:17   hgn05:gpus=1,2;                               N

RESOURCE REQUIREMENT DETAILS:
Combined: select[(type = any ) && (type == any)] order[-slots:-ngpus_physical]
                     affinity[core(1)*4]
Effective: select[(((type = any ) && (type == any))) && (ngpus>0)] order[-slot
                    s:-ngpus_physical] rusage[ngpus_physical=2.00:gmem=1000.00
                    ] affinity[core(1)*4]

GPU REQUIREMENT DETAILS:
Combined: num=2:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
Effective: num=2:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
```

Good example #2

WEXAC basic concepts

```
[vadimm@access4 ~]$ cat output-881819.out
Thu Jul 20 17:17:48 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.60.13    Driver Version: 525.60.13    CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Quadro RTX 6000     On   | 00000000:15:00.0 Off |                  Off |
| 33%   26C    P8    13W / 260W |   3949MiB / 24576MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Quadro RTX 6000     On   | 00000000:39:00.0 Off |                  Off |
| 33%   25C    P8    15W / 260W |   3917MiB / 24576MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      2515      C   python                            164MiB |
|    0   N/A  N/A     28899      C   python                            164MiB |
|    1   N/A  N/A     33083      C   python                            164MiB |
|    1   N/A  N/A     76287      C   python                            164MiB |
+-----------------------------------------------------------------------------+
```

Good example #2

WEXAC basic concepts

- Other affinity job CPU only NUMA binding, memory relaxed to any available place:
  - bsub -n 2 -R"span[hosts=1] rusage[mem=100]affinity[core(3,same=numa,exclusive=(numa,injob)):cpubind=numa]" ***yourcode***

```
RESOURCE REQUIREMENT DETAILS:
Combined: select[(type = any ) && (type == any)] order[-slots:-maxslots:-mem]
                    rusage[mem=100.00] span[hosts=1] affinity[core(3,same=numa
                    ,exclusive=(numa,injob))*1:cpubind=numa]
Effective: select[((type = any ) && (type == any))] order[-slots:-maxslots:-me
                    m] rusage[mem=100.00] span[hosts=1] affinity[core(3,same=n
                    uma,exclusive=(numa,injob))*1:cpubind=numa]

AFFINITY:
                    CPU BINDING                          MEMORY BINDING
                    -----------------------              ------------------
HOST                TYPE   LEVEL  EXCL   IDS            POL   NUMA SIZE
cn110               core   numa   numa   /1/2/4         -     -    -
                                         /1/2/5
                                         /1/2/6
cn110               core   numa   numa   /1/3/16        -     -    -
                                         /1/3/17
                                         /1/3/18
```

WEXAC basic concepts

- Other affinity job involving both CPU and GPU with only NUMA binding, memory aligned to task NUMA level:
  - bsub -n 2 -q gpu-short -gpu num=2:gmem=1000:aff=yes -R"span[hosts=1] rusage[mem=100]affinity[core(3,same=numa):cpubind=numa:membind=localonly]"

```
EXTERNAL MESSAGES:
MSG_ID FROM       POST_TIME      MESSAGE                                    ATTACHMENT
0      vadimm     Jul 23 12:39   hgn13:gpus=2,3,5,6;                           N

RESOURCE REQUIREMENT DETAILS:
Combined: select[(type = any ) && (type == any)] order[-slots:-ngpus_physical]
                     rusage[mem=100.00] span[hosts=1] affinity[core(3,same=num
                    a)*1:cpubind=numa:membind=localonly]
Effective: select[(((type = any ) && (type == any))) && (ngpus>0)] order[-slot
                    s:-ngpus_physical] rusage[mem=100.00,ngpus_physical=2.00:g
                    mem=1000.00] span[hosts=1] affinity[core(3,same=numa)*1:cp
                    ubind=numa:membind=localonly]

AFFINITY:
                    CPU BINDING                                 MEMORY BINDING
                    -----------------------                     --------------------
HOST                TYPE    LEVEL   EXCL    IDS                 POL    NUMA SIZE
hgn13               core    numa    -       /0/0/17             local 0    100.0MB
                                            /0/0/18
                                            /0/0/19
hgn13               core    numa    -       /1/1/9              local 1    100.0MB
                                            /1/1/10
                                            /1/1/11

GPU REQUIREMENT DETAILS:
Combined: num=2:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
Effective: num=2:mode=shared:mps=no:j_exclusive=no:gmem=1000.00:gvendor=nvidia
```

WEXAC basic concepts - summary

- LSF tasks multiply all other resources
- Affinity threads doesn’t multiple all other resources
- Pickup of right job size in terms of CPU cores, CPU memory, GPU memory, GPU setup is crucial for effective jobs, pending time reduction and achieve high fair share score.

SHARE_INFO_FOR: new-short/

USER/GROUP	SHARES	PRIORITY	STARTED	RESERVED	CPU_TIME	RUN_TIME	ADJUST	GPU_RUN_TIME
aihubadm	100	33.333	0	0	0.0	0	0.000	0
nirc	100	33.333	0	0	0.0	0	0.000	0
leep	100	33.333	0	0	0.0	0	0.000	0
tomerant	100	33.333	0	0	0.0	0	0.000	0
alexkon	100	33.331	0	0	1.2	0	0.000	0
gabril	100	33.330	0	0	1.3	0	0.000	0
vitalyg	100	33.324	0	0	4.6	0	0.000	0
golann	100	33.220	0	0	52.8	0	0.000	0
amy	100	32.390	0	0	449.4	0	0.000	0
eliranso	100	27.668	0	0	3159.0	0	0.000	0
vadimm	100	6.665	4	0	0.0	1	0.000	0
nathanl	100	2.281	10	0	23.8	3902	0.000	0
fabrizio	100	1.495	0	0	328540.8	0	0.000	0
nachumn	100	1.425	1	0	20135.1	21695	0.000	0
saarsh	100	1.414	10	0	56.6	13578	0.000	0
efratw	100	1.351	4	0	20688.3	19802	0.000	0
angolo	100	1.141	10	0	2597.9	19496	0.000	0
arielo	100	1.106	4	0	1914.7	27025	0.000	0
tamark	100	0.885	10	0	32.8	28780	0.000	0
adiwi	100	0.399	10	0	1597.2	78341	0.000	0
yufei	100	0.362	1	0	90156.7	91113	0.000	0

SHARE_INFO_FOR: gpu-short/

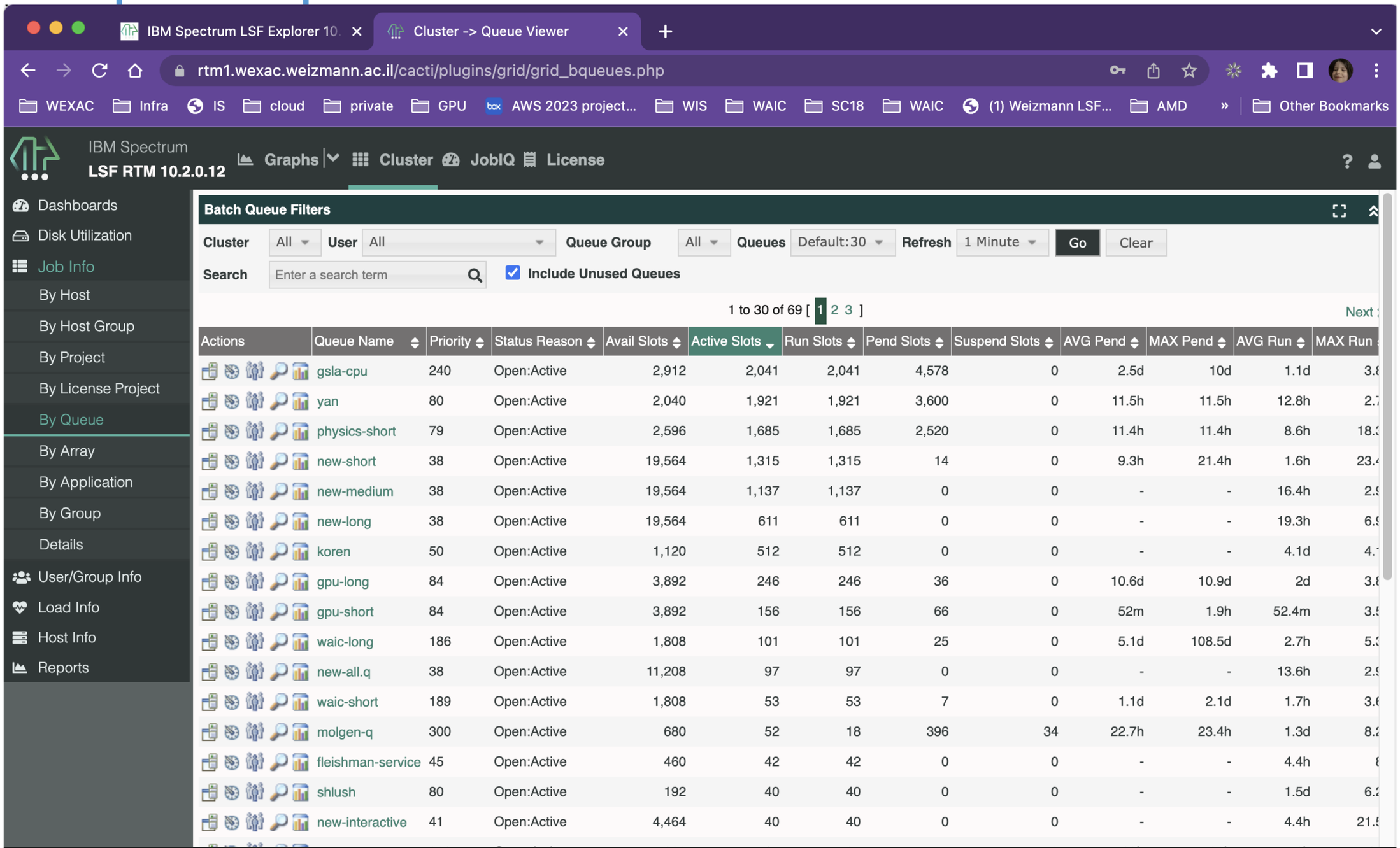
USER/GROUP	SHARES	PRIORITY	STARTED	RESERVED	CPU_TIME	RUN_TIME	ADJUST	GPU_RUN_TIME
vadimm	1	0.953	0	0	0.9	0	0.000	17
alexkon	1	0.952	0	0	1.7	0	0.000	17
vitalyg	1	0.569	0	0	4198.2	0	0.000	0
nancyy	1	0.218	0	0	1525.6	0	0.000	1190
nathanl	1	0.055	6	0	12.9	3637	0.000	3637
angolo	1	0.036	0	0	3600.7	0	0.000	9384
sagima	1	0.016	8	0	6744.6	17459	0.000	17459
karbati	1	0.008	36	0	443903.3	43030	0.000	0
dijin18	1	0.006	40	0	594210.6	30814	0.000	0
gavriel	1	0.002	6	0	150993.3	1358807	0.000	0
saurabhm	1	0.002	1	0	19129.8	4639	0.000	176274
erezy	1	0.002	77	0	1330228.1	1196698	0.000	0
yuril	1	0.001	4	0	220667.3	219886	0.000	219905
sagyk	1	0.001	2	0	378272.8	262115	0.000	262363
joel	1	0.001	21	0	3757288.8	637295	0.000	0
ronsar	1	0.001	8	0	245799.0	466469	0.000	466469
dekely	1	0.000	12	0	244582.0	815497	0.000	815498
jonasm	1	0.000	9	0	923955.8	190838	0.000	1717542
guylu	1	0.000	468	0	1840099.9	78374	0.000	1622840
yaelami	1	0.000	157	0	112651.7	1148122	0.000	2005626
jonatham	1	0.000	187	0	8395377.0	8530559	0.000	8534280

Platform RTM

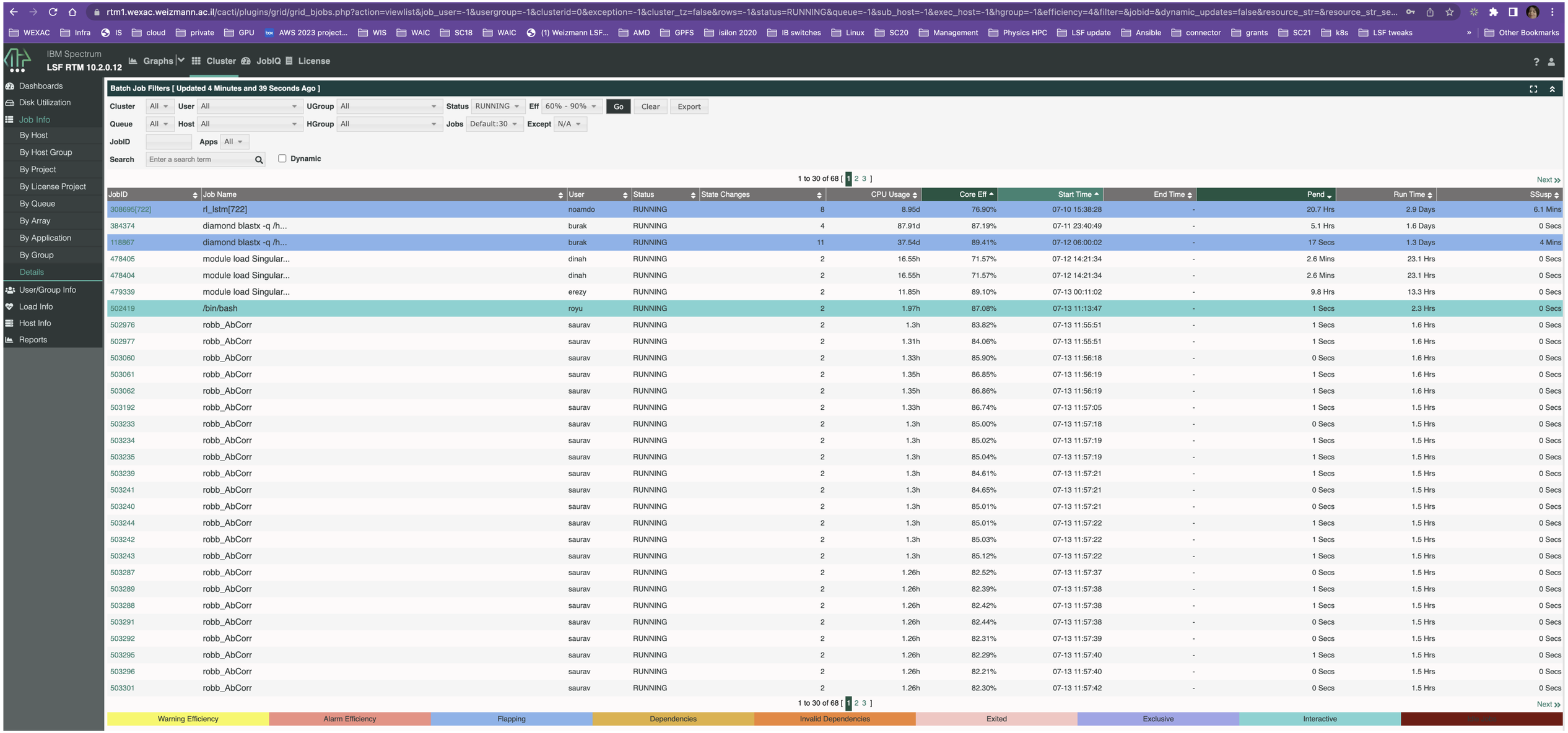
- Can be used to monitor and manage LSF jobs in real-time

- https://rtm1.wexac.weizmann.ac.il/ – WEXAC's RTM

RTM Graphs https://rtm1.wexac.weizmann.ac.il

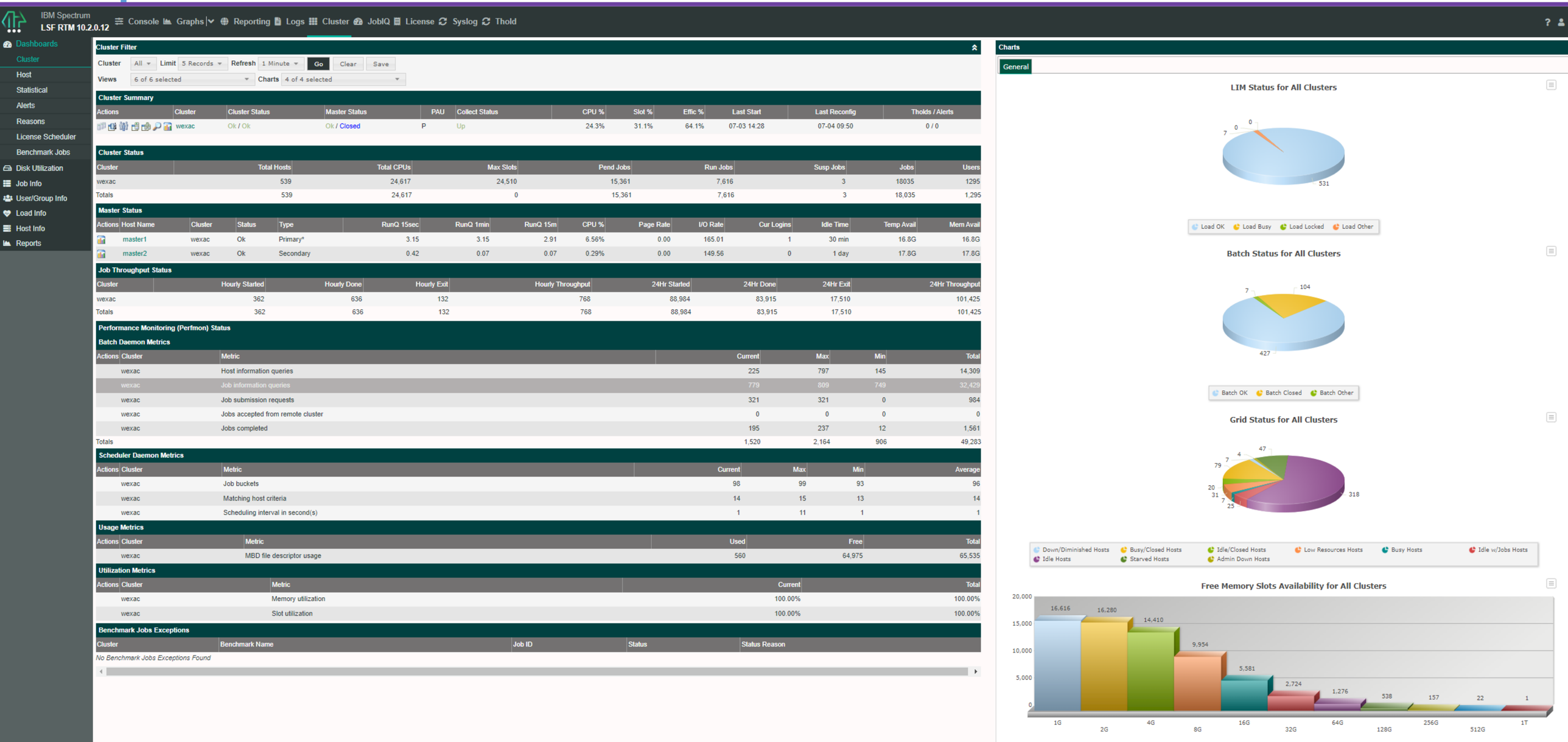


RTM Graphs https://rtm1.wexac.weizmann.ac.il

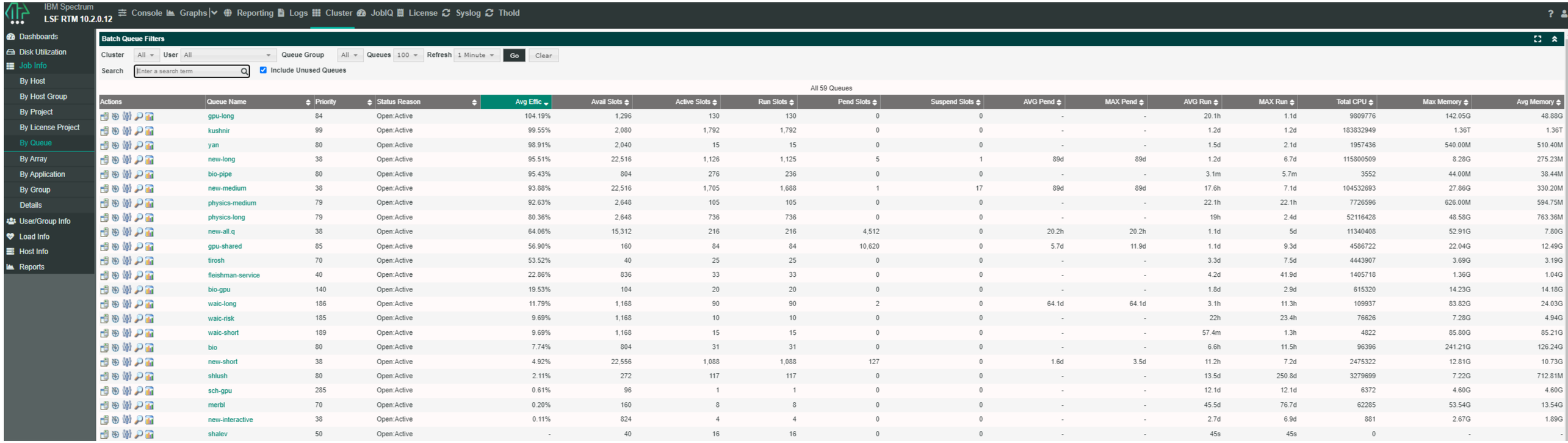


RTM – Real Time Monitoring

https://rtm1.wexac.weizmann.ac.il/



RTM Queues status



Actions	Queue Name	Priority	Status Reason	Avg Effic	Avail Slots	Active Slots	Run Slots	Pend Slots	Suspend Slots	AVG Pend	MAX Pend	AVG Run	MAX Run	Total CPU	Max Memory	Avg Memory
	gpu-long	84	Open:Active	104.19%	1,296	130	130	0	0	-	-	20.1h	1.1d	9809776	142.05G	48.88G
	kushnir	99	Open:Active	99.55%	2,080	1,792	1,792	0	0	-	-	1.2d	1.2d	183832949	1.36T	1.36T
	yan	80	Open:Active	98.91%	2,040	15	15	0	0	-	-	1.5d	2.1d	1957436	540.00M	510.40M
	new-long	38	Open:Active	95.51%	22,516	1,126	1,125	5	1	89d	89d	1.2d	6.7d	115800509	8.28G	275.23M
	bio-pipe	80	Open:Active	95.43%	804	276	236	0	0	-	-	3.1m	5.7m	3552	44.00M	38.44M
	new-medium	38	Open:Active	93.88%	22,516	1,705	1,688	1	17	89d	89d	17.6h	7.1d	104532693	27.86G	330.20M
	physics-medium	79	Open:Active	92.63%	2,648	105	105	0	0	-	-	22.1h	22.1h	7726596	626.00M	594.75M
	physics-long	79	Open:Active	80.36%	2,648	736	736	0	0	-	-	19h	2.4d	52116428	48.58G	763.36M
	new-all.q	38	Open:Active	64.06%	15,312	216	216	4,512	0	20.2h	20.2h	1.1d	5d	11340408	52.91G	7.80G
	gpu-shared	85	Open:Active	56.90%	160	84	84	10,620	0	5.7d	11.9d	1.1d	9.3d	4586722	22.04G	12.49G
	tirosh	70	Open:Active	53.52%	40	25	25	0	0	-	-	3.3d	7.5d	4443907	3.69G	3.19G
	fleishman-service	40	Open:Active	22.86%	836	33	33	0	0	-	-	4.2d	41.9d	1405718	1.36G	1.04G
	bio-gpu	140	Open:Active	19.53%	104	20	20	0	0	-	-	1.8d	2.9d	615320	14.23G	14.18G
	waic-long	186	Open:Active	11.79%	1,168	90	90	2	0	64.1d	64.1d	3.1h	11.3h	109937	83.82G	24.03G
	waic-risk	185	Open:Active	9.69%	1,168	10	10	0	0	-	-	22h	23.4h	76626	7.28G	4.94G
	waic-short	189	Open:Active	9.69%	1,168	15	15	0	0	-	-	57.4m	1.3h	4822	85.80G	85.21G
	bio	80	Open:Active	7.74%	804	31	31	0	0	-	-	6.6h	11.5h	96396	241.21G	126.24G
	new-short	38	Open:Active	4.92%	22,556	1,088	1,088	127	0	1.6d	3.5d	11.2h	7.2d	2475322	12.81G	10.73G
	shlush	80	Open:Active	2.11%	272	117	117	0	0	-	-	13.5d	250.8d	3279699	7.22G	712.81M
	sch-gpu	285	Open:Active	0.61%	96	1	1	0	0	-	-	12.1d	12.1d	6372	4.60G	4.60G
	merbl	70	Open:Active	0.20%	160	8	8	0	0	-	-	45.5d	76.7d	62285	53.54G	13.54G
	new-interactive	38	Open:Active	0.11%	824	4	4	0	0	-	-	2.7d	6.9d	881	2.67G	1.89G
	shalev	50	Open:Active	-	40	16	16	0	0	-	-	45s	45s	0	-	-

RTM - Bad job resource allocation example #1

Job Detail | Job Graphs | Host Graphs | License Usage

Show User Jobs

General Information

JobId:	47604	**Status:**	DONE
Jobname:	HLA-A01; 01_3.txt		
Project:	default	**License Project:**	-
Queue:	gpu-medium	**Cluster Name:**	wexac
User:	[redacted]	**User Group:**	[redacted]
Charged SAAP:			

Submission Details

Submit Time:	2021-05-18 11:34:13	**Number of CPUs:**	12
Submit Host:	access3		
Asked Hosts/Groups:	-	**Runtime Estimate:**	-
Combined ResReq:	select[((type = any) && (ngpus>0)) && (type == any)] order[gpu_maxfactor] rusage[mem=2000.00:ngpus_physical=1.00] affinity[thread(1)*1]		
Submit Command:	[redacted]pan-4.0/netMHCpan -a HLA-A01: 01 -l 8,9,10,11,12,13,14 -f 'HLA-A01; 01_3.txt' > ../netMHCpan_output/'HLA-A01; 01_3.txt'		
Submit Dir:	$HOME/TCGA-Data driven approach/TCGA HLA-I/_4_netMHCpan/fasta2		
Resource Requirements:	rusage[mem=2000]		
Output File:	../parallel_output/HLA-A01;01_3.txt-%J.o		
Error File:	../parallel_output/HLA-A01;01_3.txt-%J.e		

Execution Environment

Ex Host:	12*hgn02 (6*hgn02, 6*hgn11)		
Start Time:	2021-05-18 11:34:23		
Username:	[redacted]	**UID String:**	42234
User Home:	[redacted]	**Working Dir:**	[redacted]h approach/TCGA HLA-I/_4_netMHCpan/fasta2
Max Processors:	12	**Max Allocated Slots:**	12
CPU Limit:	-	**Run Time Limit:**	12 Hours
Memory Limit:	1.953G	**Swap Limit:**	0.000M
Effective ResReq:	select[(((type = any) && (type == any))) && (ngpus>0)] order[gpu_maxfactor] rusage[mem=2000.00,ngpus_physical=1.00] affinity[thread(1)*1]		

Current/Last Status

PGIDS:	64330				
PIDS:	46739, 46740, 46741, 64330, 64371, 64373, 64380				
Threads:	8				
Pend Time:	10 Seconds	**PROV Time:**	-	**Run Time:**	4.47 Hours
Effective Pending Time Limit:	-				
PSUSP Time:	-	**USUSP Time:**	-	**SSUSP Time**	-
Unknown Time	-				
Cumulative CPU:	4.46 Hours	**System Time:**	-	**User Time:**	4.46 Hours
Cur Mem Used:	38.000M	**Max Mem Used:**	38.000M	**Job Efficiency:**	8.31%
Cur V.Memory Size:	0.000M	**Max V.Memory Size:**	0.000M		
Exit Code:	0	**End Time:**	2021-05-18 16:02:23		

RTM - Bad job resource allocation example #1

- In the above example, we see job's detailed information. This can help us identify jobs that was submitted with wrong resource allocation.
- We can see that job was executed on a gpu queue while this program using only cpu power
- We can see that this job was submitted with 12 cores allocation while the job was really using only 1 core (look job efficency parameter)
- We can see that 24GB of memory was requested for job, while only 38MB(!) was actually used
- Understanding the most optimal resource requirements for programs is very important for jobs and overall system usage efficency.
- Our recommendation prior massive array or MPI job submission just run 1-2 jobs to understand the actual resources demands.

RTM - Bad job resource allocation example #2

JobID	Job Name	Queue	User	UGroup	JGroup	Status	State Changes	Mem Request	Mem Wasted	Max Memory	CPU Usage	CPU Effic	Num Nodes	Num CPUs	Execution Host
950371	ciwd	[redacted]	[redacted]	[redacted]	-	DONE	2	-	4611.72G	638.28G	4442.81d	99.65%	32	896	896*cn462

- In the above job details table, we can see a parallel mpi job that requested 896 CPU's per job and more than 5TB of RAM memory.
- While we can see that CPU efficency of the job is very good, we can see that there are more than 4.6TB(!) of memory was completely wasted.
- Always check your jobs real resource usage in order to prevent such big resource waste.

RTM - Bad job resource allocation example #3

JobID	Job Name	Queue	User	UGroup	JGroup	Status	State Changes	Mem Request	Mem Wasted	Max Memory	CPU Usage	CPU Effic	Num Nodes	Num CPUs	Execution Host
144643	module load matlab/R...	gpu-short	[redacted]	[redacted]wx-grp-lsf	-	DONE	2	-	4.79G	3.02G	3.97d	2481.48%	1	1	ibdgx001

- In the above job details table, we can see the opposite situation comparing to other examples.
- Here we can see a job that has over efficency rate of 2481%
- What does it mean? It means that the requested resources for that job was significantly smaller than was really needed for that job.
- In this case, we see that CPU request for that job is only one single CPU, while the job was actually using almost 25 CPU's.
- This situation is bad too because user is taking more resources than was requested and it affect badly on the LSF scheduler performance and resource calculations.

WEXAC Best practicies and recommendations

- Minimal Linux knowledge required
- Load only essential modules
- Don’t mix different toolchains (for instance gcc/10.3.0 with gcc/8.3.0)
- If you must have multiple modules loaded – make sure they belong to same toolchain (example: foss/2021b)
- You should plan which 1 environment type you want to use out of below 3 options (**don’t mix them!**) :
  - Conda
  - Modules
  - Containers

WEXAC Best practicies and recommendations

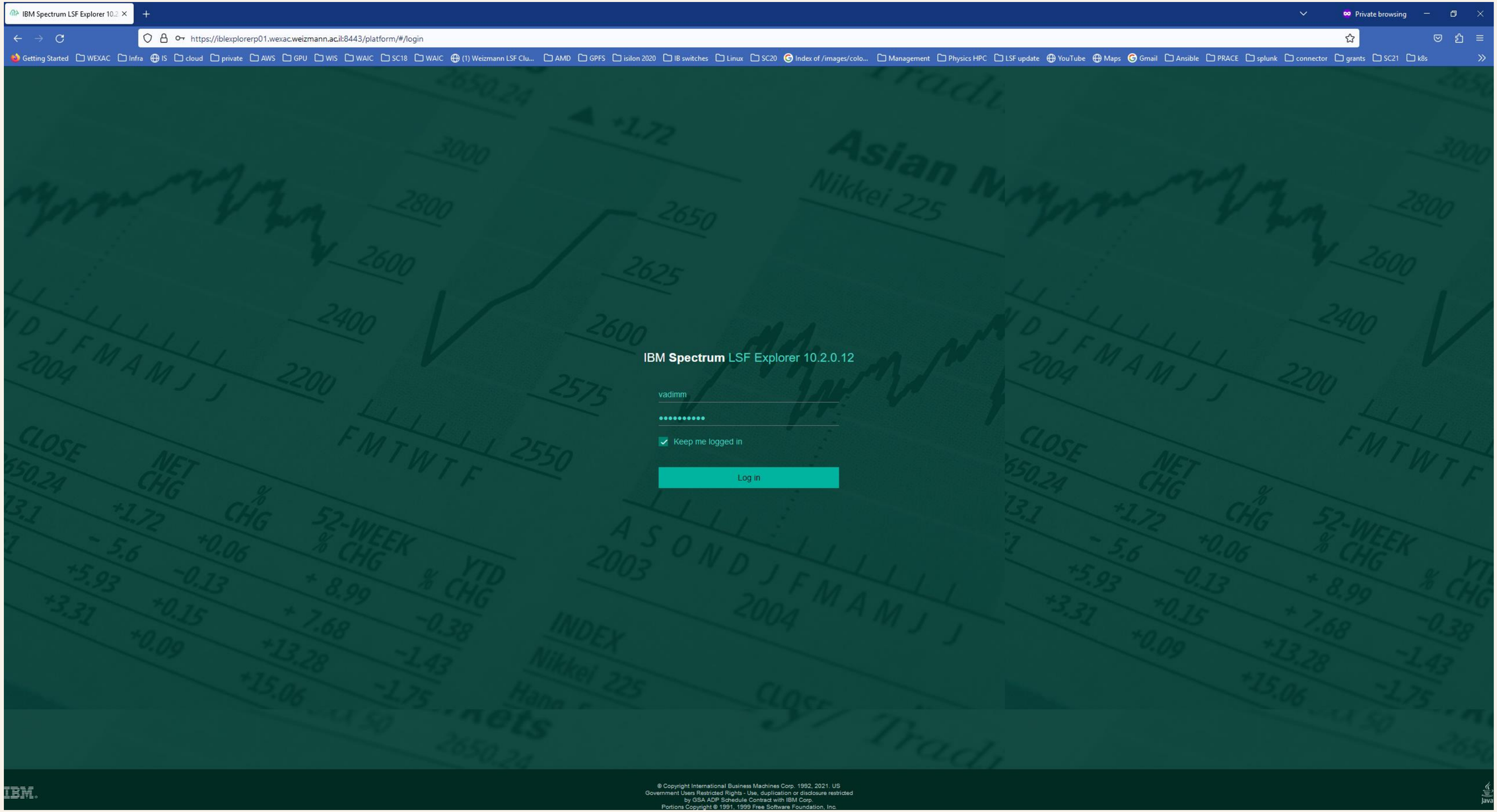
- If you are using container – all your environment for current job should be inside the container !
- In case of modules usage – **don’t** use conda !
- Conda users – **don’t use** containers neither modules !
- Make sure you have clean enviroment and clean ~/.bashrc to avoid compatibility issues.
- Pipelines involves multiple environment types should be separated to different job types or including sterilization between different environment types switching like module purge or conda deactivate commands between different pipelines steps.

WEXAC Best practicies and recommendations

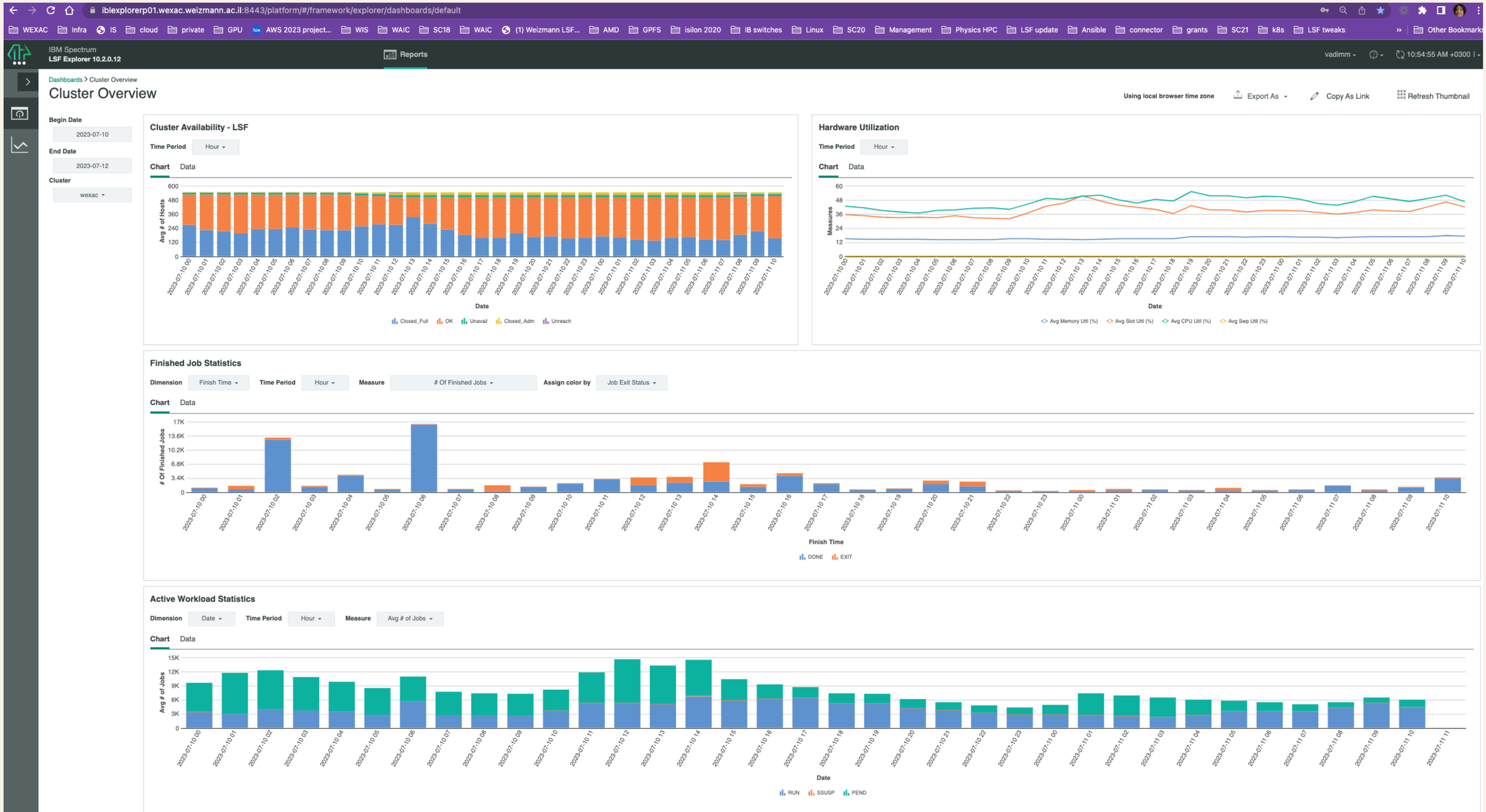
```
[vadimm@access4 ~]$ ml
No modules loaded
[vadimm@access4 ~]$ ml foss/2021b
[vadimm@access4 ~]$ ml

Currently Loaded Modules:
  1) GCCcore/11.2.0                    5) numactl/2.0.14-GCCcore-11.2.0
     9) hwloc/2.5.0-GCCcore-11.2.0        13) PMIx/4.1.0-GCCcore-11.2.0
 17) gompi/2021b
  2) zlib/1.2.11-GCCcore-11.2.0        6) XZ/5.2.5-GCCcore-11.2.0
    10) OpenSSL/1.1                       14) OpenMPI/4.1.1-GCC-11.2.0
 18) FFTW/3.3.10-gompi-2021b
  3) binutils/2.37-GCCcore-11.2.0      7) libxml2/2.9.10-GCCcore-11.2.0
    11) libevent/2.1.12-GCCcore-11.2.0  15) OpenBLAS/0.3.18-GCC-11.2.0
 19) ScaLAPACK/2.1.0-gompi-2021b-fb
  4) GCC/11.2.0                        8) libpciaccess/0.16-GCCcore-11.2.
0   12) UCX/1.11.2-GCCcore-11.2.0          16) FlexiBLAS/3.0.4-GCC-11.2.0
 20) foss/2021b
```

WEXAC LSF explorer https://iblexplorerp01.wexac.weizmann.ac.il:8443

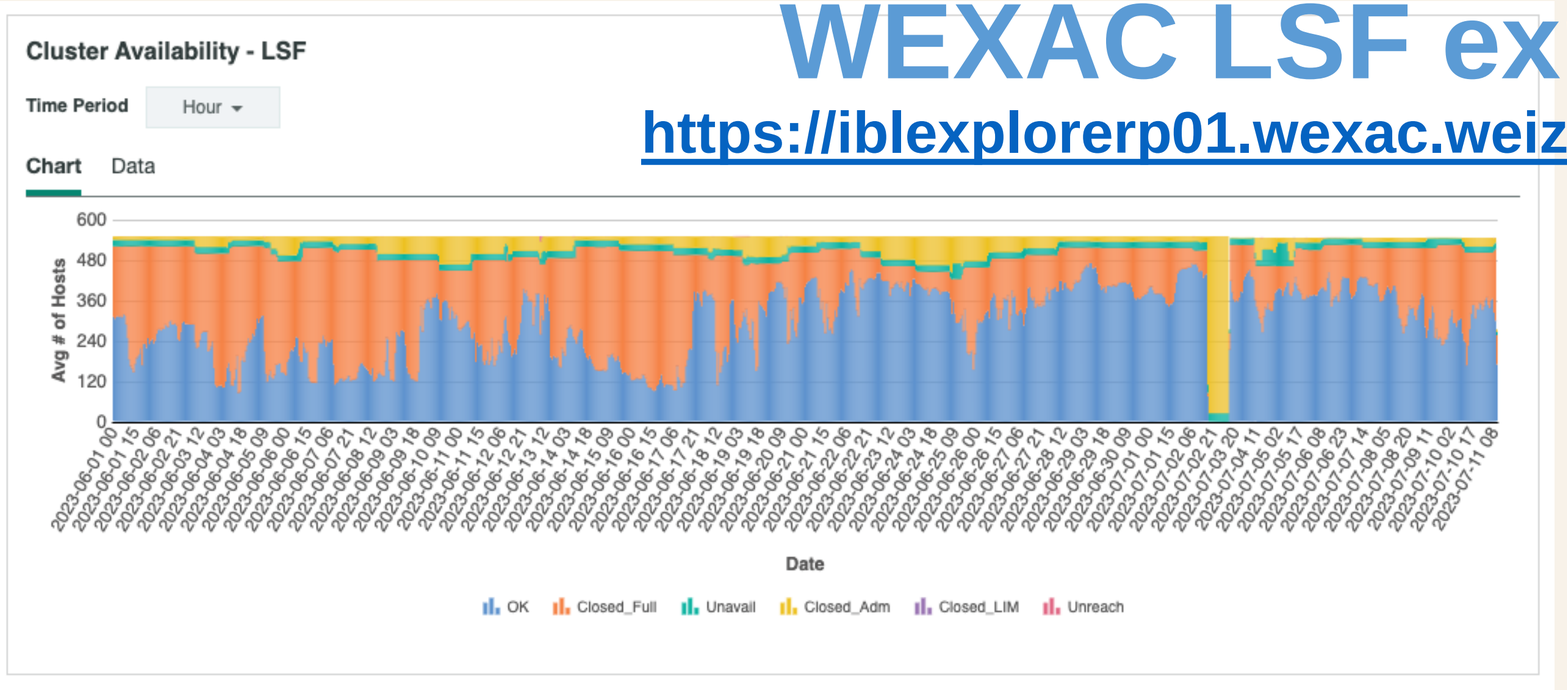


WEXAC LSF explorer https://iblexplorerp01.wexac.weizmann.ac.il:8443

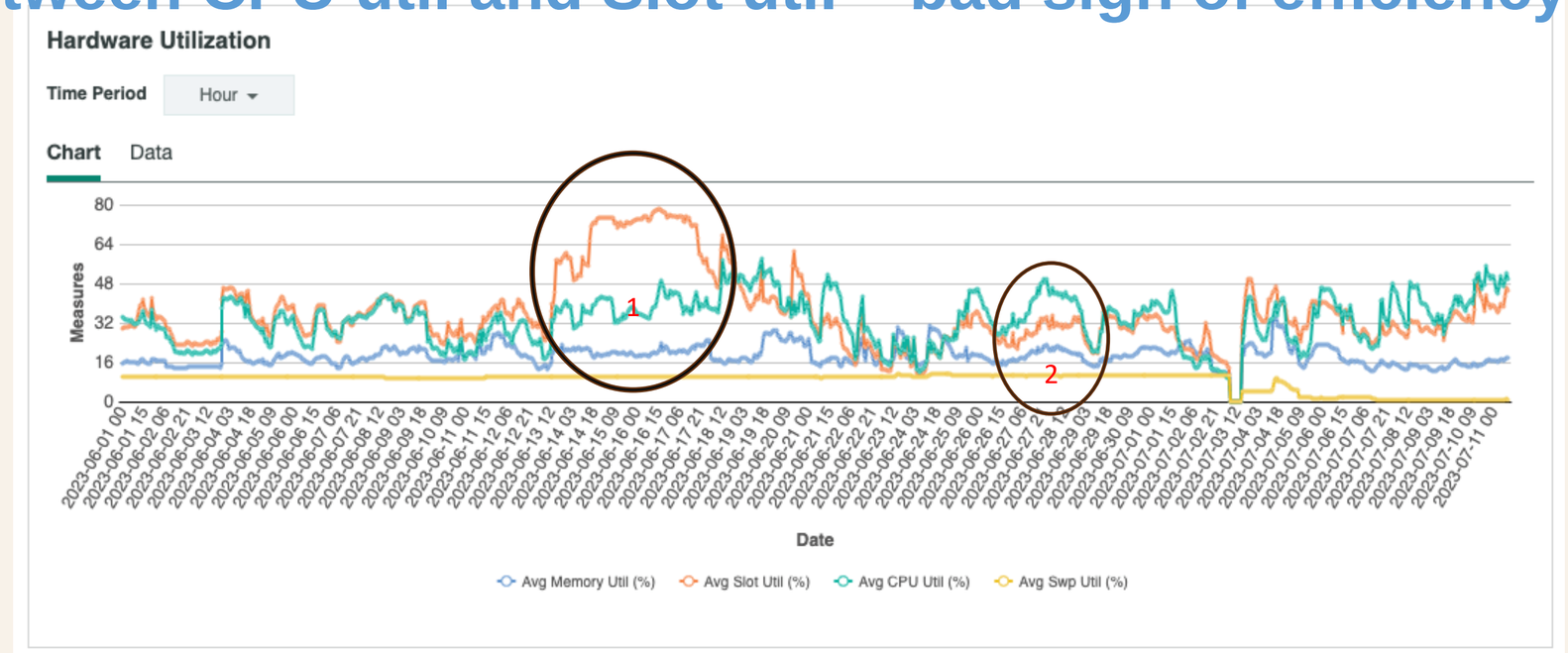


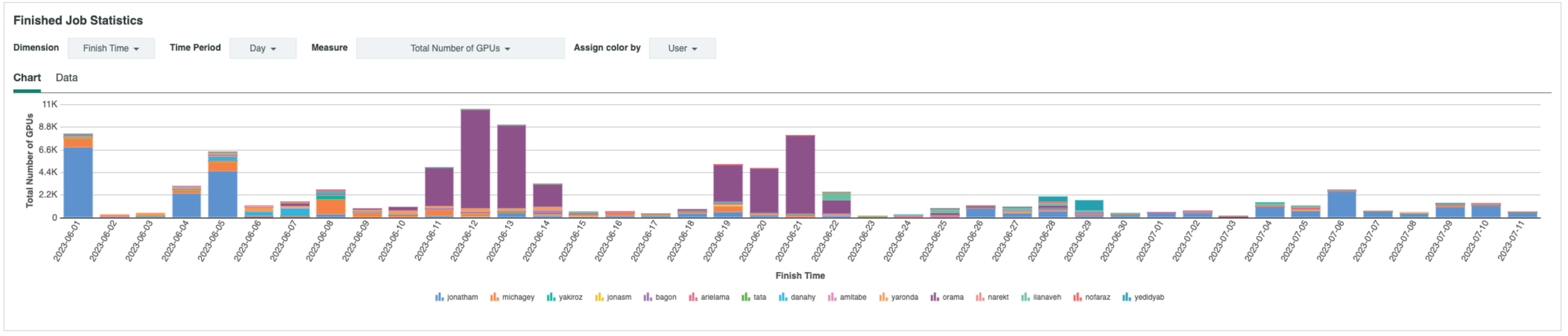
WEXAC LSF explorer

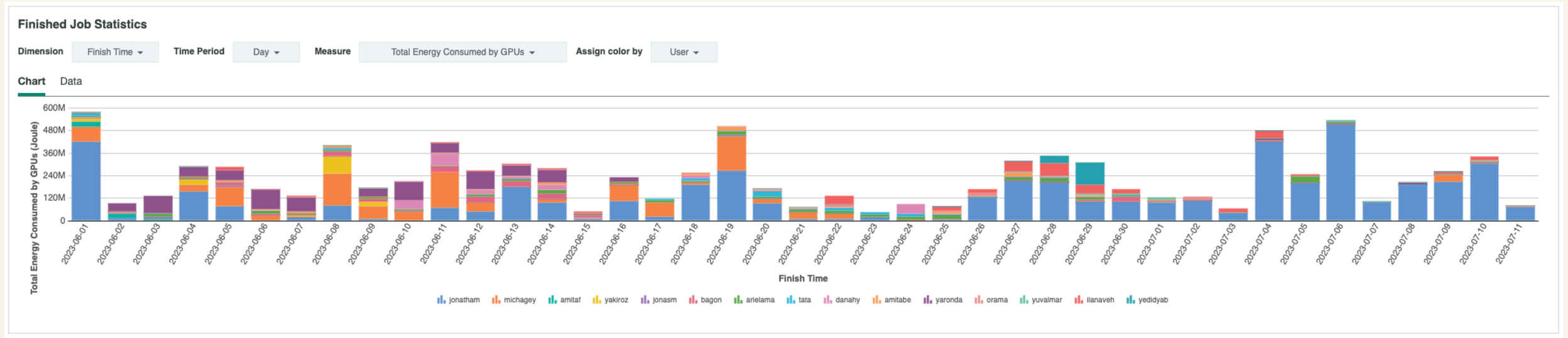
https://iblexplorerp01.wexac.weizmann.ac.il:8443



High delta between CPU util and Slot util – bad sign of efficiency problems





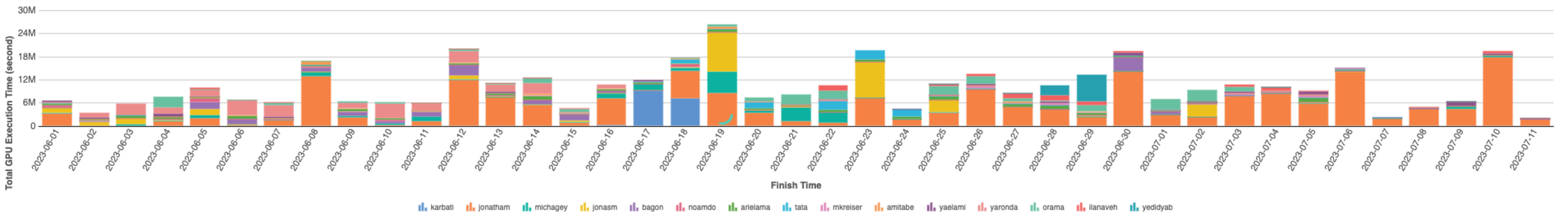


Looking for correlation between GPU allocation and energy consumption

Finished Job Statistics

Dimension Finish Time | **Time Period** Day | **Measure** Total GPU Execution Time | **Assign color by** User

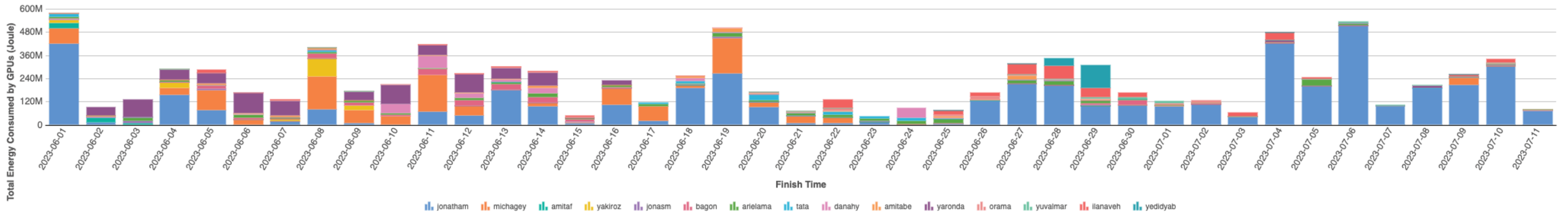
Chart Data

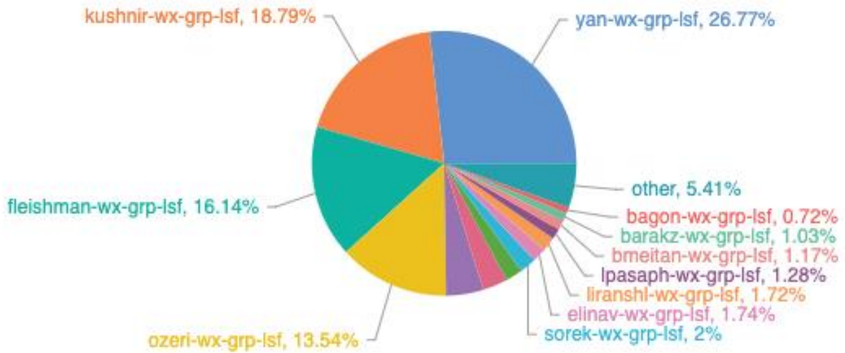


Finished Job Statistics

Dimension Finish Time | **Time Period** Day | **Measure** Total Energy Consumed by GPUs | **Assign color by** User

Chart Data

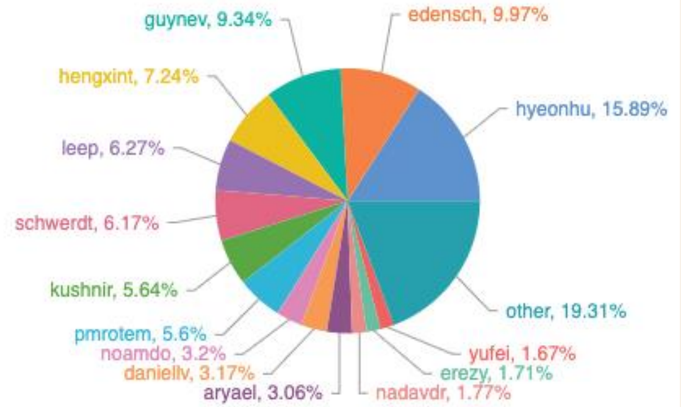




Charts > Chargeback > Workload Slots Charge

Workload Slots Charge

Dimension User | **Begin Date** 2023-06-01 | **End Date** 2023-07-12



Charts > Chargeback > Workload Slots Charge Summary

Workload Slots Charge Summary

Begin Date 2023-07-05 | **End Date** 2023-07-12 | **Cluster** wexac | **Project** All selected (9) | **User** All selected (200) | **User Group** All selected (78) | **Dimension** User Group

User Group	Slot Duration (Hour)	Slot Charge
yan-wx-grp-lsf	484105.93	1771.83
kushnir-wx-grp-lsf	181055.05	662.66
schneidmann-wx-grp-lsf	174125.81	637.3
sorek-wx-grp-lsf	92222.65	337.53
elinav-wx-grp-lsf	61852.9	226.38

Showing 1 to 10 of 78 entries

Charts > Chargeback > Workload Slot Charge Distribution

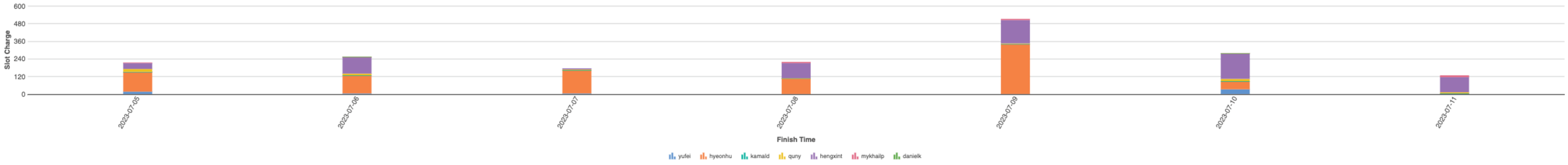
Workload Slot Charge Distribution

Export As | Copy As Link | Refresh Thumbnail

Time Period Day | **Assign color by** User | **Begin Date** 2023-07-05 | **End Date** 2023-07-12 | **Cluster** wexac | **Project** default | **User** All selected (7) | **User Group** yan-wx-grp-lsf

danielk, hengxint, hyeonhu, kamald, mykhailp, quny, yufei

Chart | Data



LSF explorer dashboards

Dashboards > Workload Statistics

Workload Statistics

Using local browser time zone | Export As | Copy As Link | Refresh Thumbnail

Begin Date
2023-07-12

End Date
2023-07-14

Cluster
wexac

User
All selected (122)

User Group
All selected (66)

Project
All selected (6)

Queue
All selected (38)

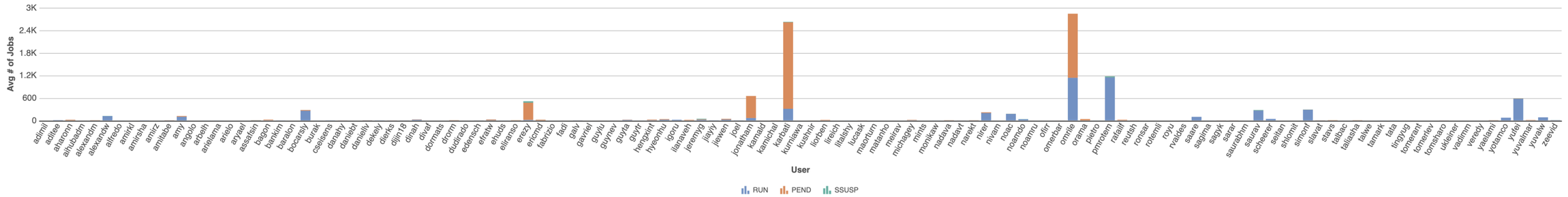
Application
All selected (4)

Job Status
All selected (3)

Active Workload Statistics

Dimension User | **Time Period** Hour | **Measure** Avg # of Jobs

Chart | Data



Workload Statistics Summary

Dimension 2 selected | **Measure** 3 selected

User	Queue	Avg Memory Usage (KB)	Total CPU Time (s)	Avg # of Jobs
omrile	physics-medium	0	0	1707
karbati	gpu-short	0	0	1287
omrile	physics-medium	296173289.88	788404.86	1136
karbati	gpu-medium	0	0	702
yufei	new-long	8298690208.63	6876.83	686

Showing 1 to 10 of 278 entries

Previous 1 2 3 4 5 … 28 Next

Workload Accounting

Using local browser time zone | Export As | Copy As Link | Refresh Thumbnail

Begin Date
2023-07-07

End Date
2023-07-14

Cluster
wexac

Project
All selected (7)

Queue
All selected (45)

User
All selected (185)

User Group
All selected (76)

Job Exit Status
All selected (2)

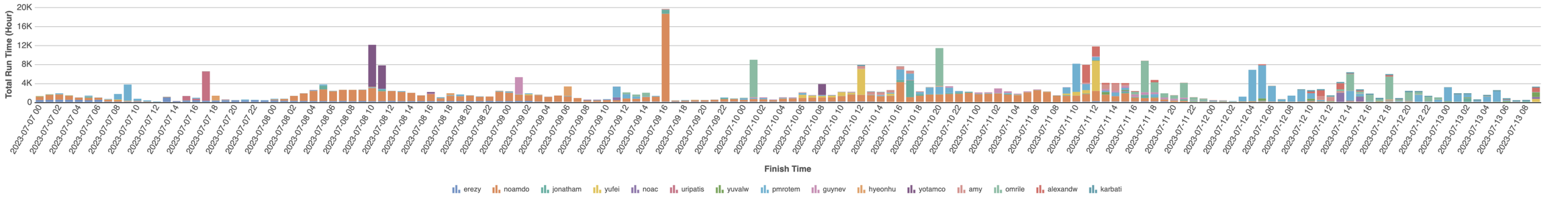
Job Run Time Rank
All selected (10)

Application
None selected

Finished Job Statistics

Dimension: Finish Time | Time Period: Hour | Measure: Total Run Time | Assign color by: User

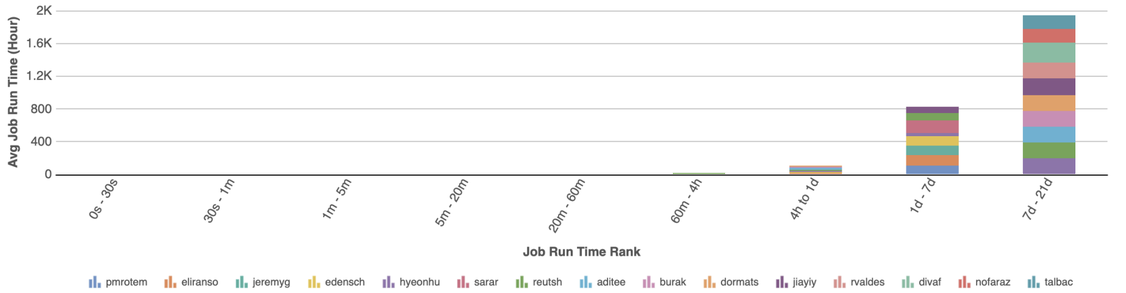
Chart | Data



Job Run Time Rank

Measure: Avg Job Run Time | Assign color by: User

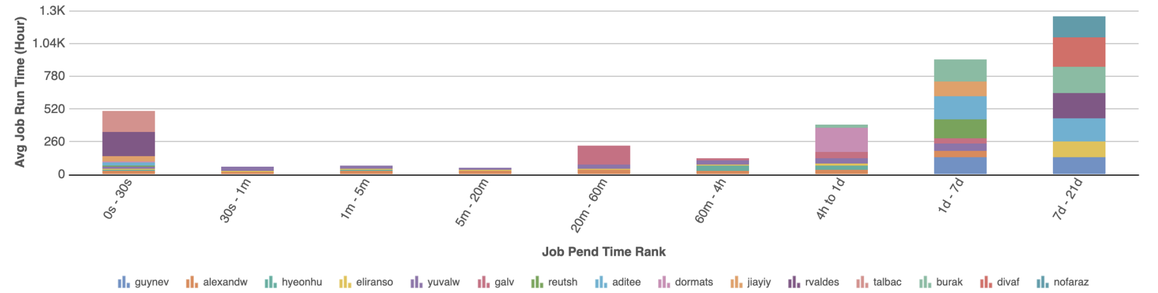
Chart | Data



Job Pend Time Rank

Measure: Avg Job Run Time | Assign color by: User

Chart | Data



Workload Summary

Dimension: 2 selected | Measure: All selected (5)

Queue	User	# Of Finished Jobs	Total Run Time (Hour)	Total Pend Time (Hour)	Total CPU Time (Hour)	Total Slot Duration (Hour)
sorek-gpu	erezy	21267	13907.42	1096248.4	13365.43	13907.42
new-short	avnim	3918	11.62	169463.36	6.12	11.57
new-medium	noamdo	5292	55482.69	82309.41	185072.8	55482.69
sch-gpu	jonatham	827	550.76	44431.36	549.31	550.76

Workload Accounting

Using local browser time zone | Export As | Copy As Link | Refresh Thumbr

Begin Date
2023-07-01

End Date
2023-07-14

Cluster
wexac

Project
All selected (11)

Queue
All selected (47)

User
All selected (241)

User Group
All selected (84)

Job Exit Status
All selected (2)

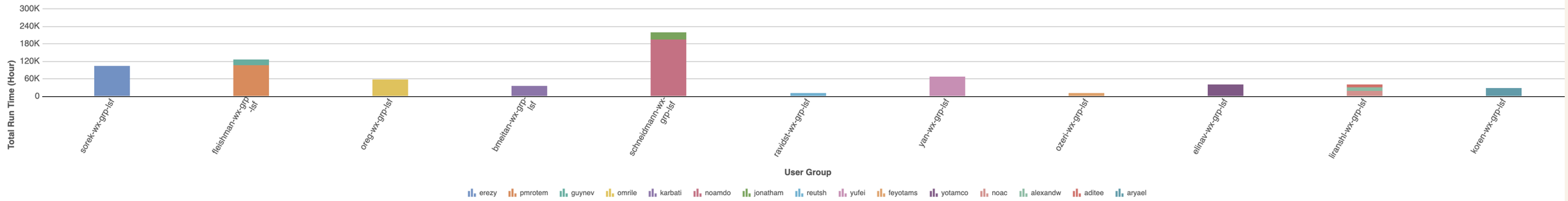
Job Run Time Rank
All selected (10)

Application
None selected

Finished Job Statistics

Dimension: User Group | Time Period: Day | Measure: Total Run Time | Assign color by: User

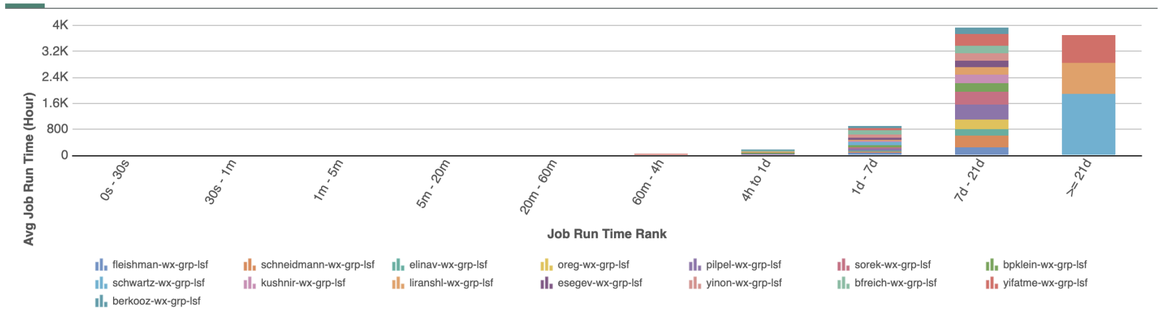
Chart | Data



Job Run Time Rank

Measure: Avg Job Run Time | Assign color by: User Group

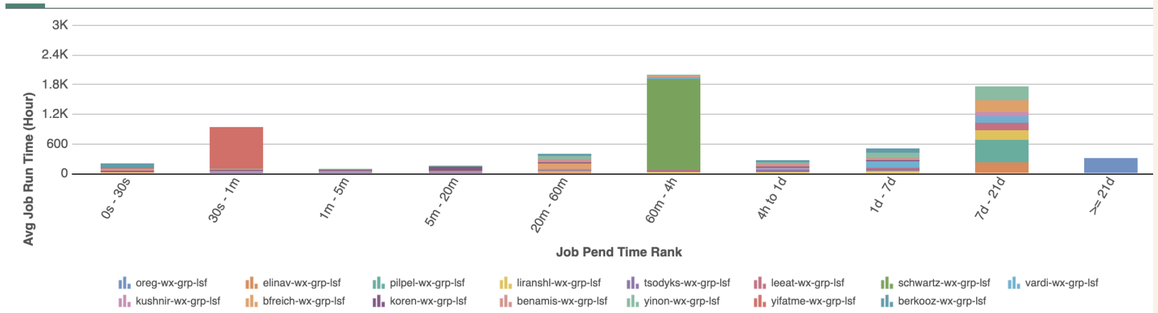
Chart | Data



Job Pend Time Rank

Measure: Avg Job Run Time | Assign color by: User Group

Chart | Data



Workload Summary

Dimension: 2 selected | Measure: All selected (5)

Queue	User	# Of Finished Jobs	Total Run Time (Hour)	Total Pend Time (Hour)	Total CPU Time (Hour)	Total Slot Duration (Hour)
new-short	pmrotem	98322	40421.72	1543.46	39526.89	40421.72
new-short	saurav	27775	2956.14	4567.08	2932.59	2956.14
new-short	amy	26018	8988.89	19102.7	8681.15	8988.89

Begin Date
2023-07-01

End Date
2023-07-14

Cluster
wexac

Project
default

Queue
All selected (4)

User
All selected (6)

User Group
liranshl-wx-grp-lsf…

Job Exit Status
All selected (2)

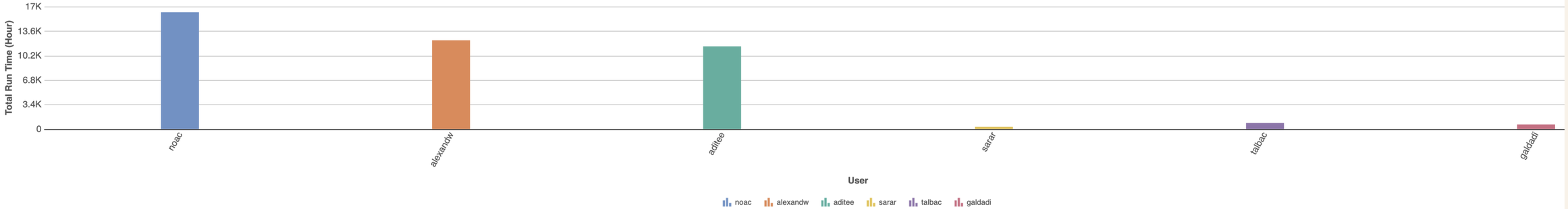
Job Run Time Rank
All selected (10)

Application
None selected

Finished Job Statistics

Dimension User | **Time Period** Day | **Measure** Total Run Time | **Assign color by** User

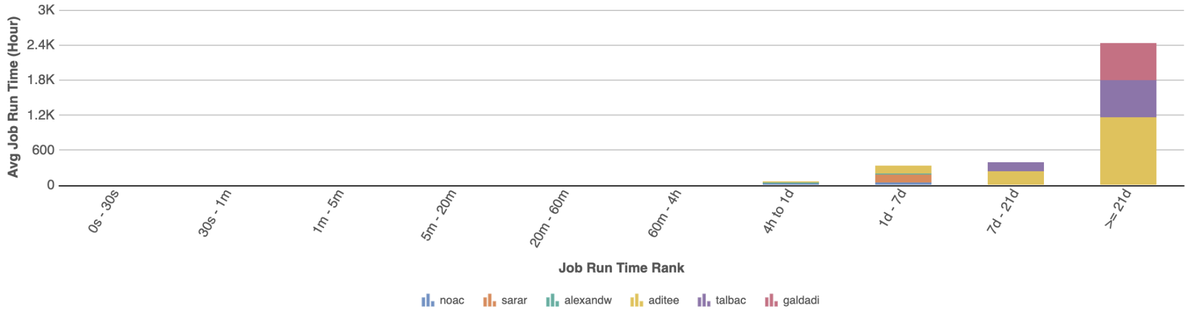
Chart | Data



Job Run Time Rank

Measure Avg Job Run Time | **Assign color by** User

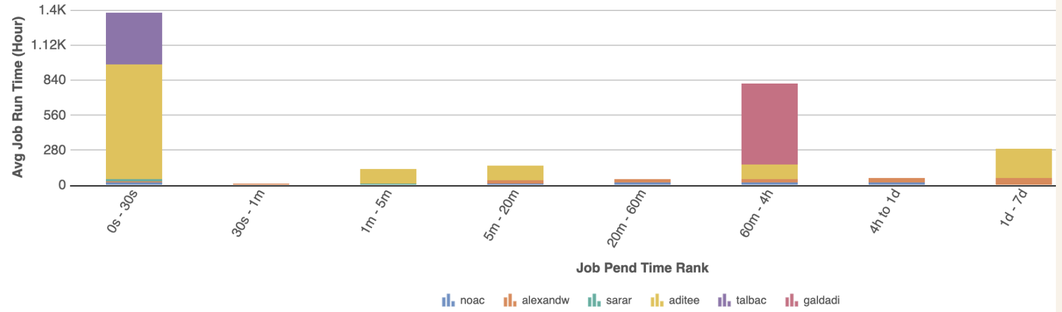
Chart | Data



Job Pend Time Rank

Measure Avg Job Run Time | **Assign color by** User

Chart | Data



Workload Summary

Dimension 2 selected | **Measure** All selected (5)

Queue	User	# Of Finished Jobs	Total Run Time (Hour)	Total Pend Time (Hour)	Total CPU Time (Hour)	Total Slot Duration (Hour)
shlush	aditee	5	4614.33	0	109	4614.33
shlush	talbac	2	816.69	0	0.71	816.69
new-long	noac	7	234.08	0.03	4.87	4915.74
shlush	sarar	22	288.46	0.07	2.38	288.46

Pending Job Analysis

Using local browser time zone | Export As | Copy As Link | Refresh Thumbnail

Begin Date
2023-07-08

End Date
2023-07-14

Cluster
wexac

User
All selected (14…

User Group
All selected (58)…

Project
All selected (5)

Queue
All selected (42)…

Application
All selected (3)

Is Eligible Pending
None selected

Pending Reason Type
All selected (5)

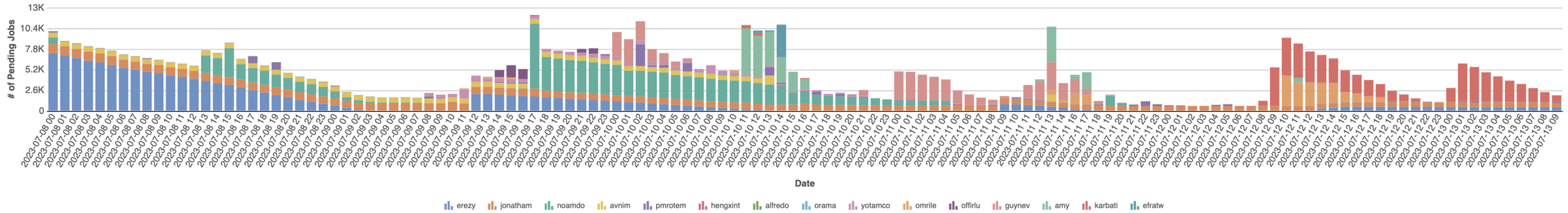
Pending Reason
All selected (26)…

Detailed Pending Info
All selected (18)…

Pending Jobs Over Time

Time Period: Hour | Assign color by: User | Measure: # of Pending Jobs

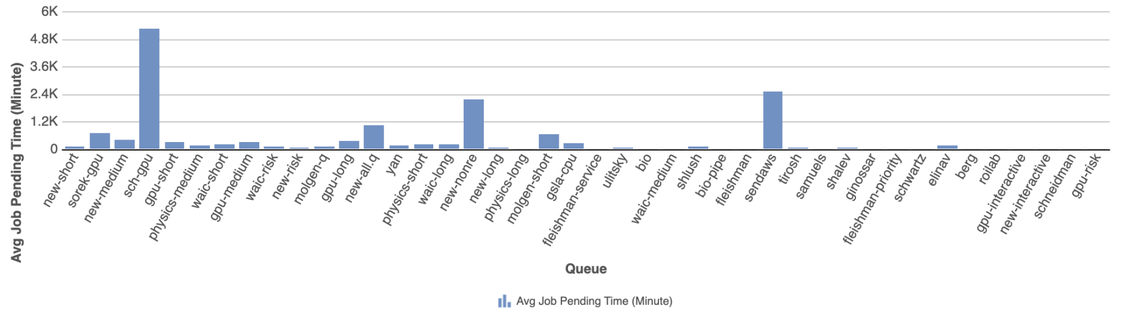
Chart | Data



Pending Time Analysis

Dimension: Queue | Measure: Avg Job Pending Time

Chart | Data



Top Pending Reasons

Dimension: 4 selected | Measure: All selected (2)

Pending Reason	Queue	User	User Group	# of Pending Jobs	Total Job Pending Time (Minute)
Job's requirements for resource reservation not satisfied	sorek-gpu	erezy	sorek-wx-grp-lsf	10283	7288328.95
Resource limit defined on queue has been reached	new-medium	noamdo	schneidmann-wx-grp-lsf	9256	4412344.02
Job's requirements for resource reservation not satisfied	sch-gpu	jonatham	schneidmann-wx-grp-lsf	1072	4177721
Job has reached its requeue limit	new-short	avnim	efrats-wx-grp-lsf	595	2218800

Showing 1 to 10 of 554 entries

Previous 1 2 3 4 5 … 56 Next

Pending jobs troubleshooting

First, use bjobs -l *yourjobid* command

Then add bjobs -l -p1 *yourjobid* or bjobs -l -p2 *yourjobid* or bjobs -l -p3 *yourjobid*

```
[vadimm@ibdgxa01 ~]$ bjobs  53690
JOBID      USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
53690      amy     PEND  new-medium access                  *_fcgrH_16 Jul 23 11:09
[vadimm@ibdgxa01 ~]$ bjobs -l 53690

PENDING REASONS:
Job requirements for reserving resource (mem) not satisfied: 55 hosts;
```

```
bjobs -l -p1 53690
```

```
PENDING REASONS:
Job's requirements for resource reservation not satisfied (Resource: mem): cn6
                     53, cn652, cn651, cn650, cn789, cn788, cn787, cn784, cn783
                     , cn782, cn781, cn780, cn779, cn778, cn773, cn770, cn110,
                     cn769, cn767, cn109, cn766, cn106, cn763, cn105, cn762, cn
                     102, cn100, cn759, cn758, cn757, cn754, cn753, cn751, cn74
                     9, cn748, cn747, cn746, cn745, cn743, cn742, cn741, cn098,
                      cn097, cn096, cn095, cn094, cn093, cn092, cn091, cn088, c
                     n087, cn710, cn086, cn084;
```

```
bjobs -l -p2 53690
```

```
PENDING REASONS:
Candidate host pending reasons (85 of 545 hosts):
  Job's requirements for resource reservation not satisfied (Resource: mem): c
                     n653, cn652, cn651, cn650, cn789, cn788, cn787, cn784, cn7
                     83, cn782, cn781, cn780, cn779, cn778, cn773, cn770, cn110
                     , cn769, cn767, cn109, cn766, cn106, cn763, cn105, cn762,
                     cn102, cn100, cn759, cn758, cn757, cn754, cn753, cn751, cn
                     749, cn748, cn747, cn746, cn745, cn743, cn742, cn741, cn09
                     8, cn097, cn096, cn095, cn094, cn093, cn092, cn091, cn088,
                      cn087, cn086, cn084;
  Job slot limit reached: cn752, cn108, cn750, cn765, cn709, cn107, cn262, cn7
                     64, cn774, cn055, cn744, cn772, cn771, cn104, cn739, cn099
                     , cn103, cn760, cn184, cn101, cn183, cn786, cn768, cn785,
                     cn090, cn089, cn756, cn755, cn710, cn180, cn085, cn775;
```

```
bjobs -l -p3 53690
```

```
PENDING REASONS:
Candidate host pending reasons (85 of 545 hosts):
  Job's requirements for resource reservation not satisfied (Resource: mem): c
                     n653, cn652, cn651, cn650, cn789, cn788, cn787, cn784, cn7
                     83, cn782, cn781, cn780, cn779, cn778, cn773, cn770, cn110
                     , cn769, cn767, cn109, cn766, cn106, cn763, cn105, cn762,
                     cn102, cn100, cn759, cn758, cn757, cn754, cn753, cn751, cn
                     749, cn748, cn747, cn746, cn745, cn743, cn742, cn741, cn09
                     7, cn096, cn095, cn094, cn093, cn092, cn091, cn088, cn087,
                      cn086, cn084;
  Job slot limit reached: cn775, cn752, cn108, cn750, cn765, cn709, cn107, cn2
                     62, cn764, cn774, cn055, cn744, cn772, cn771, cn104, cn739
                     , cn099, cn098, cn103, cn760, cn184, cn101, cn183, cn786,
                     cn768, cn090, cn089, cn785, cn756, cn710, cn755, cn085, cn
                     180;

Non-candidate host pending reasons (460 of 545 hosts):
  Not specified in job submission: cn561, cn232, cn718, cn562, cn233, hgn30, c
                     n719, cn563, cn234, hgn31, cn564, cn235, hgn32, cn565, cn2
                     36, cn080, hgn33, cn566, cn237, cn081, hgn34, cn567, cn238
```

```
                     hgn04, cn381, cn208, cn053, hgn05, cn382, cn209, hgn06, c
                     n383, hgn08, hgn07, cn384, cn056;
 Load information unavailable: cn082, ibdgx001, cn273, cn279, cn432, cn761, c
                     n601, cn602, cn457, ops, cn463, cn162, iblpacp02, cn668, c
                     n044, cn054;
 Closed by LSF administrator: cn078, dgxws01, dgxws02, hgn44, hgn45, hgn47, c
                     n433, cn451, cn456, iblpacp01, cn182, cn186;
```

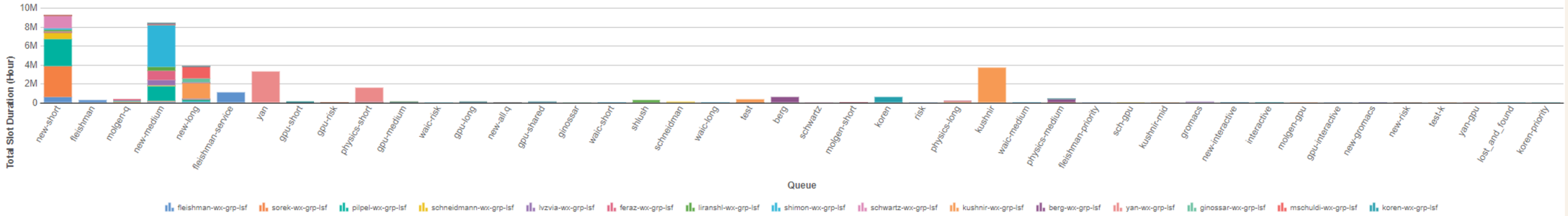
LSF Explorer - Workload insights and statistics

https://iblexplorerp01.wexac.weizmann.ac.il:8443

Finished Job Statistics

Dimension: Queue | Time Period: Hour | Measure: Total Slot Duration | Assign color by: User Group

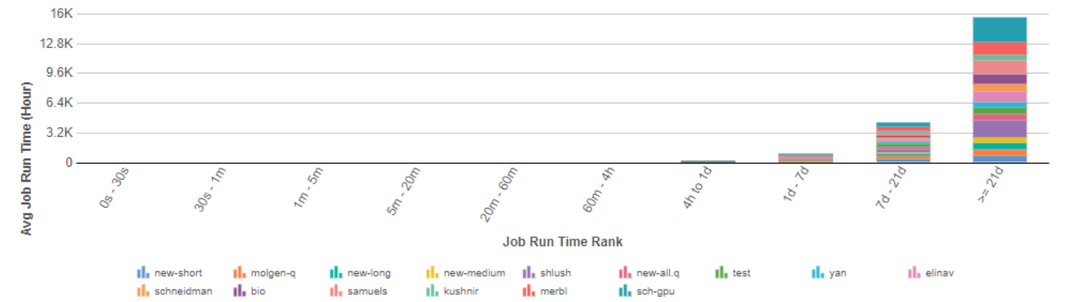
Chart | Data



Job Run Time Rank

Measure: Avg Job Run Time | Assign color by: Queue

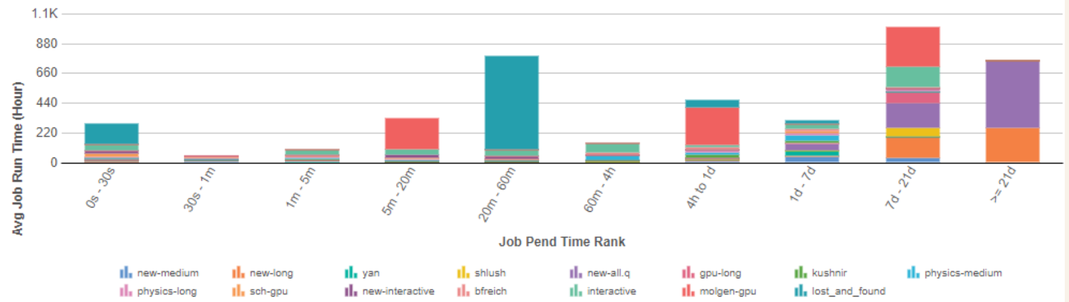
Chart | Data



Job Pend Time Rank

Measure: Avg Job Run Time | Assign color by: Queue

Chart | Data



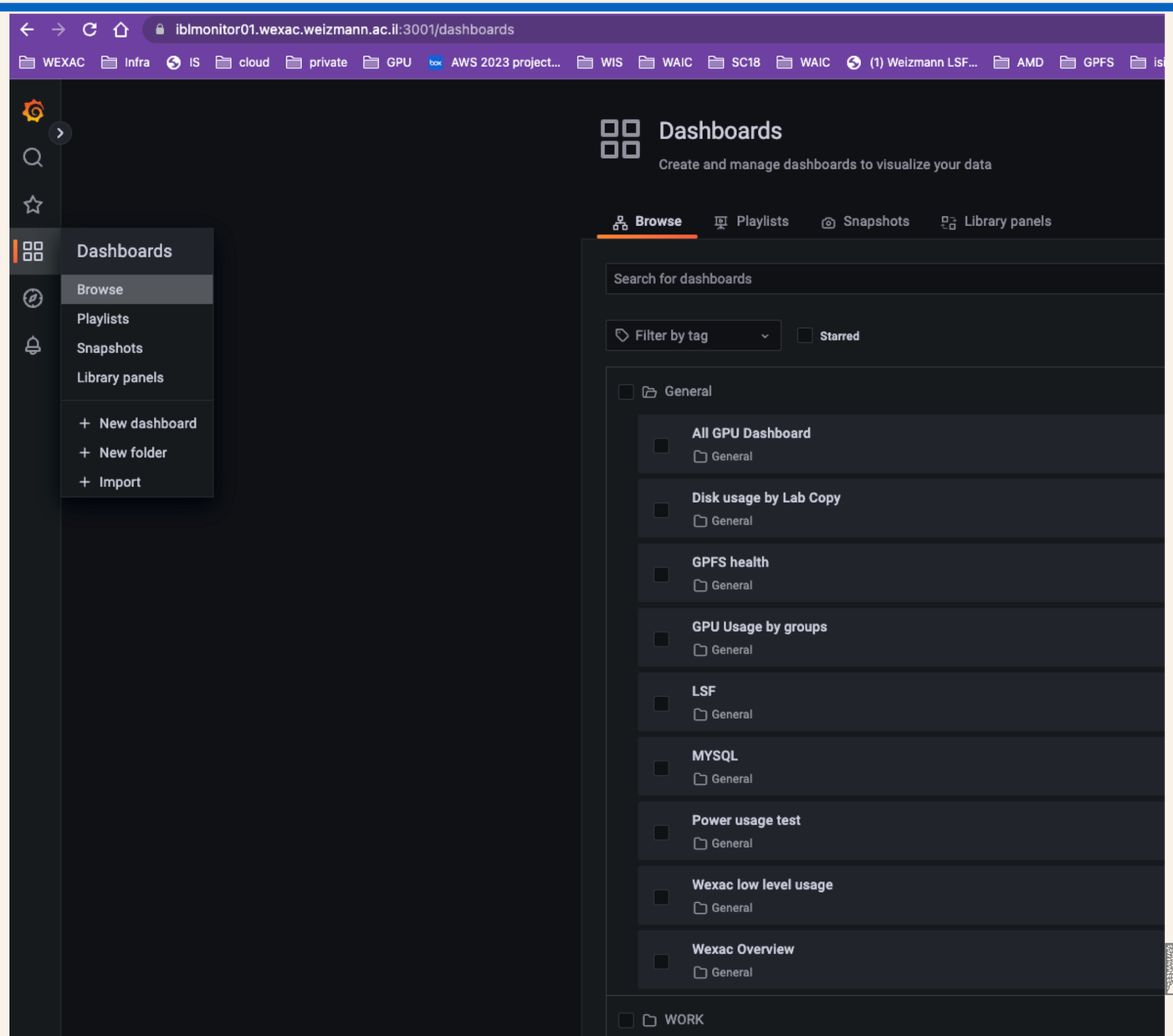
Workload Summary

Dimension: 3 selected | Measure: All selected (5)

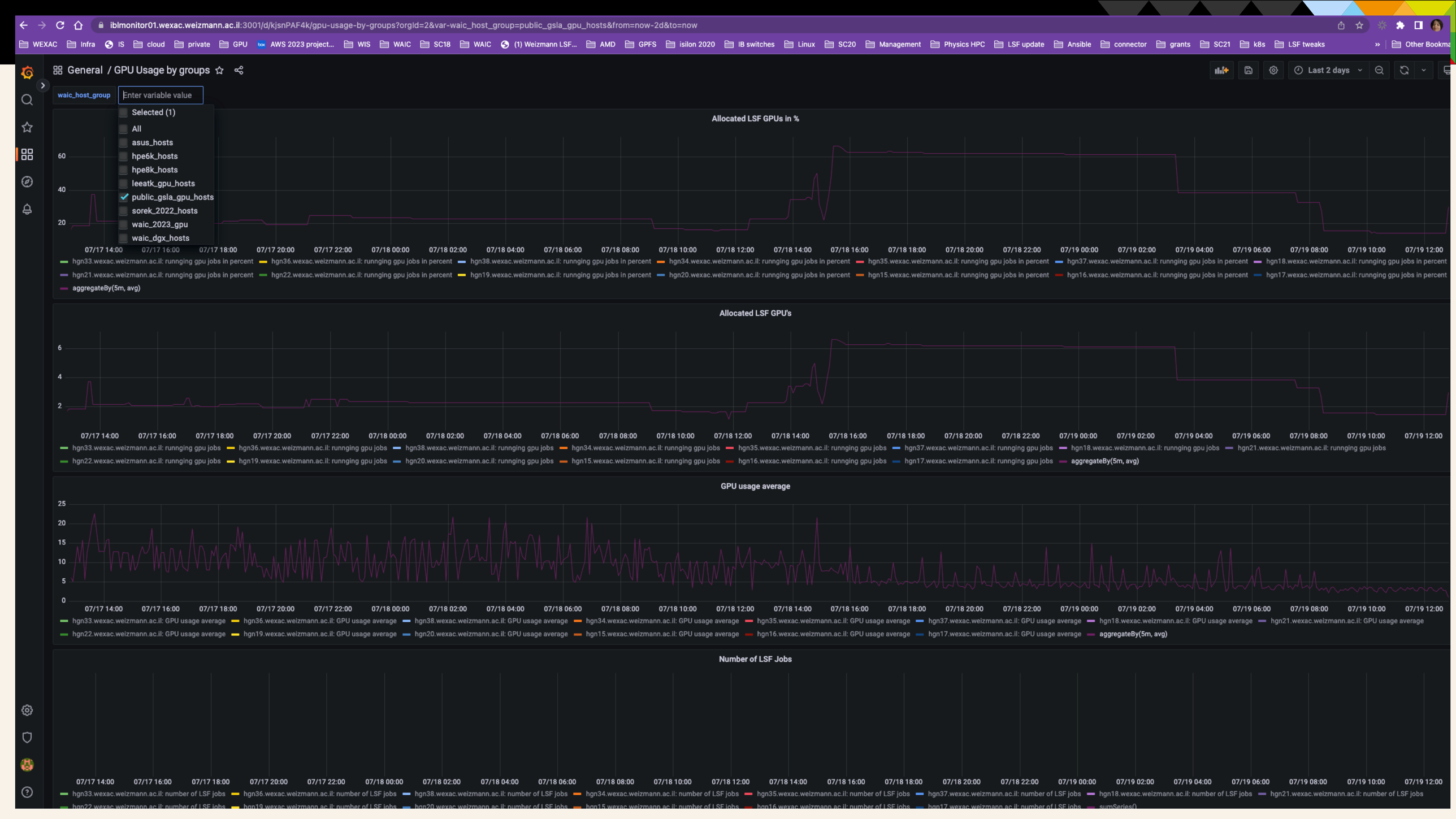
Cluster	Job Exit Status	Queue	# Of Finished Jobs	Total Run Time (Hour)	Total Pend Time (Hour)	Total CPU Time (Hour)	Total Slot Duration (Hour)
wexac	DONE	new-short	7179959	6028167.79	9950204.71	6672012.58	6805774.79
wexac	EXIT	new-medium	369879	4419716.36	956286.6	4076385.39	5207028.15
wexac	DONE	new-medium	600220	3591698.74	2564619.29	3598863.01	3762813.01

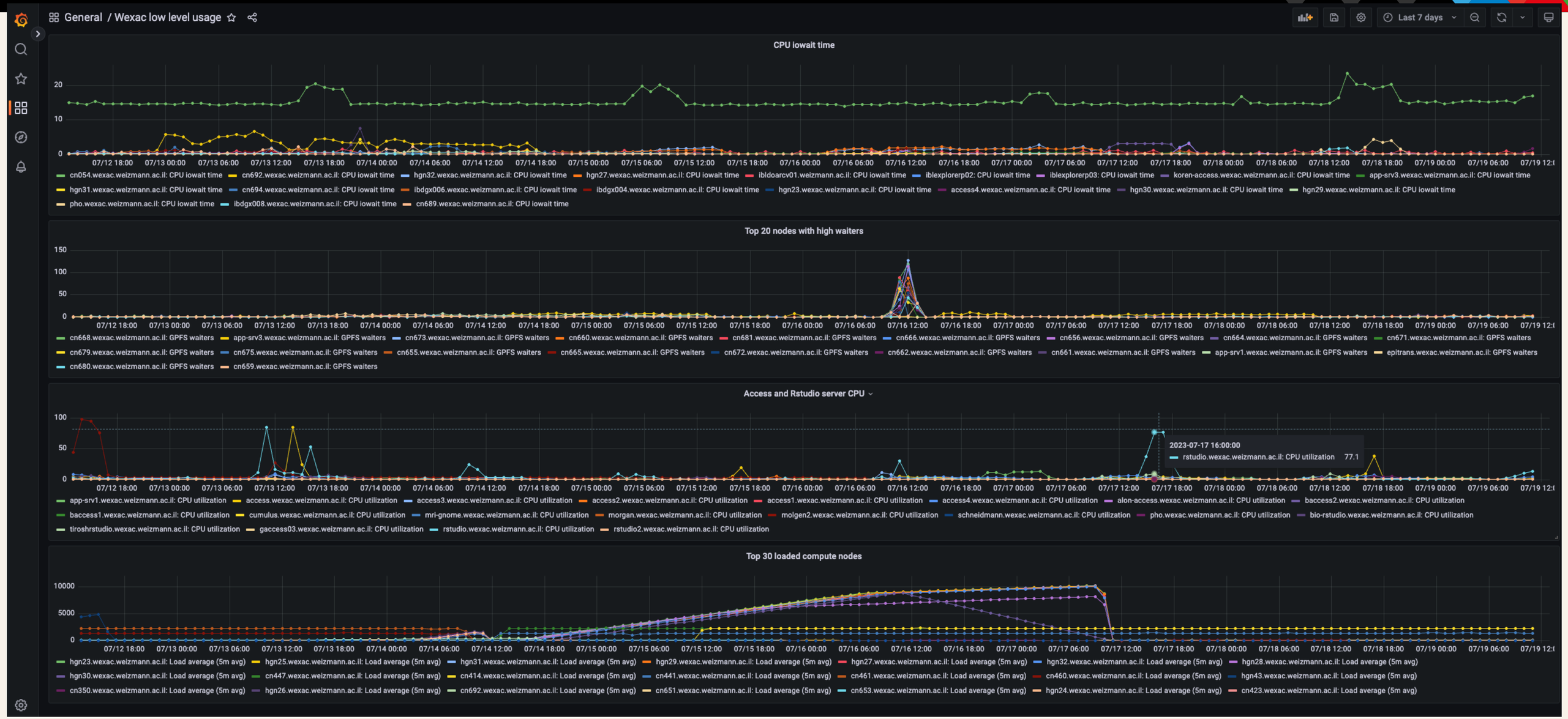
WEXAC Grafana

https://iblmonitor01.wexac.weizmann.ac.il:3001

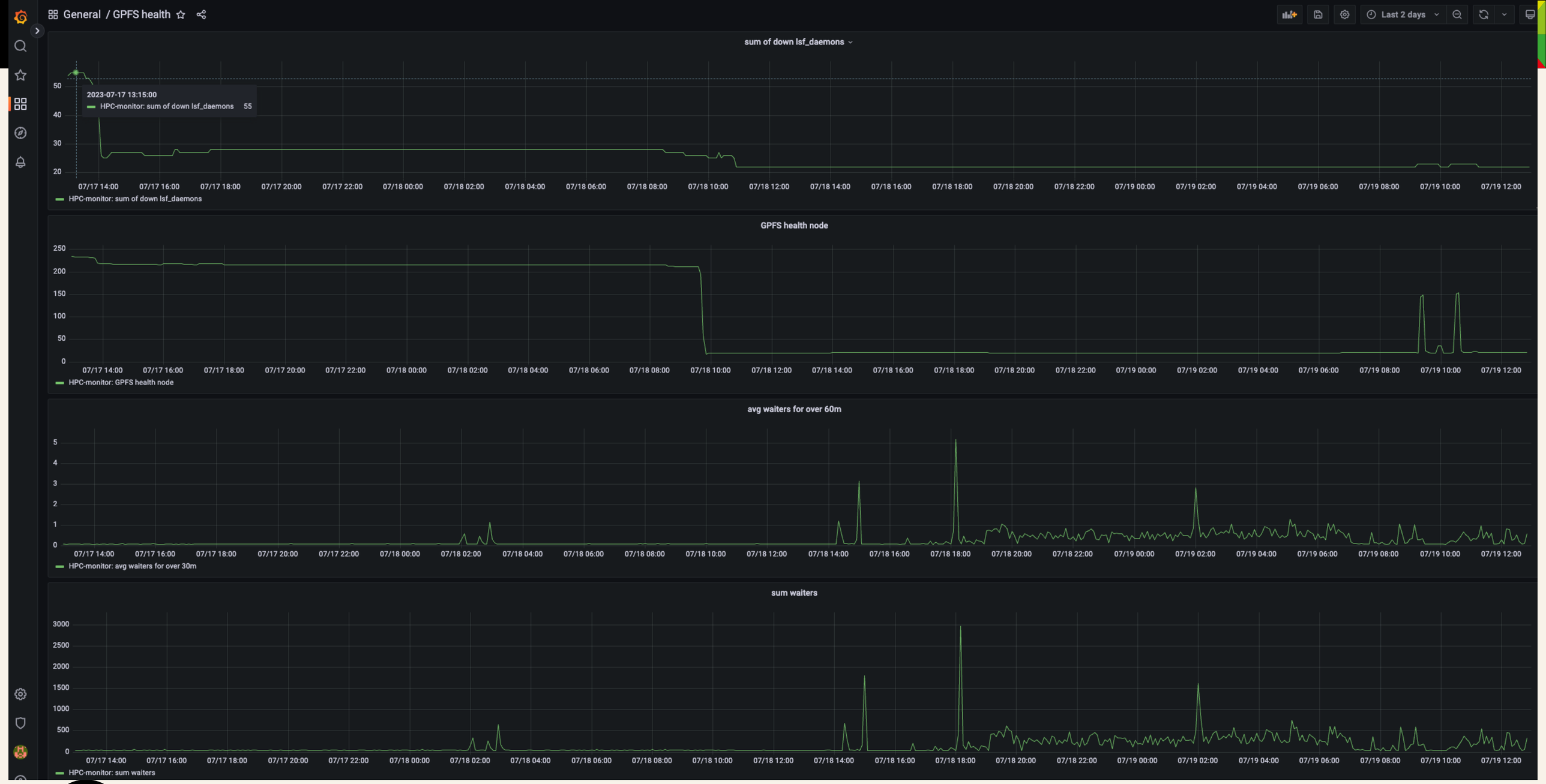


מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE


iblmonitor01.wexac.weizmann.ac.il:3001/d/kjsnPAF4k/gpu-usage-by-groups?orgId=2&var-waic_host_group=public_gsla_gpu_hosts&from=now-2d&to=now
General / GPU Usage by groups
Last 2 days
waic_host_group
Enter variable value
Selected (1)
All
asus_hosts
hpe6k_hosts
hpe8k_hosts
leeatk_gpu_hosts
public_gsla_gpu_hosts
sorek_2022_hosts
waic_2023_gpu
waic_dgx_hosts
Allocated LSF GPUs in %
Allocated LSF GPU's
GPU usage average
Number of LSF Jobs

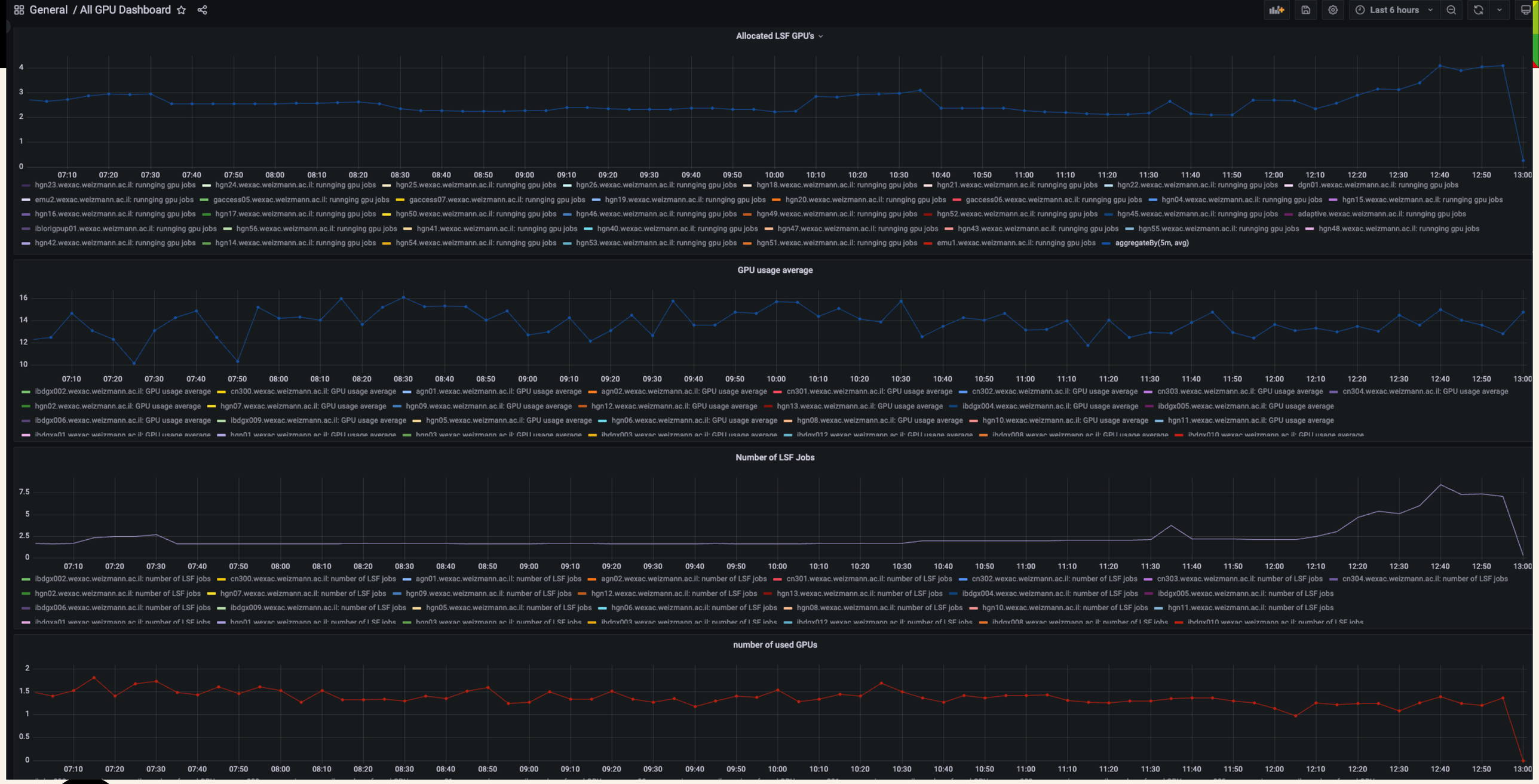


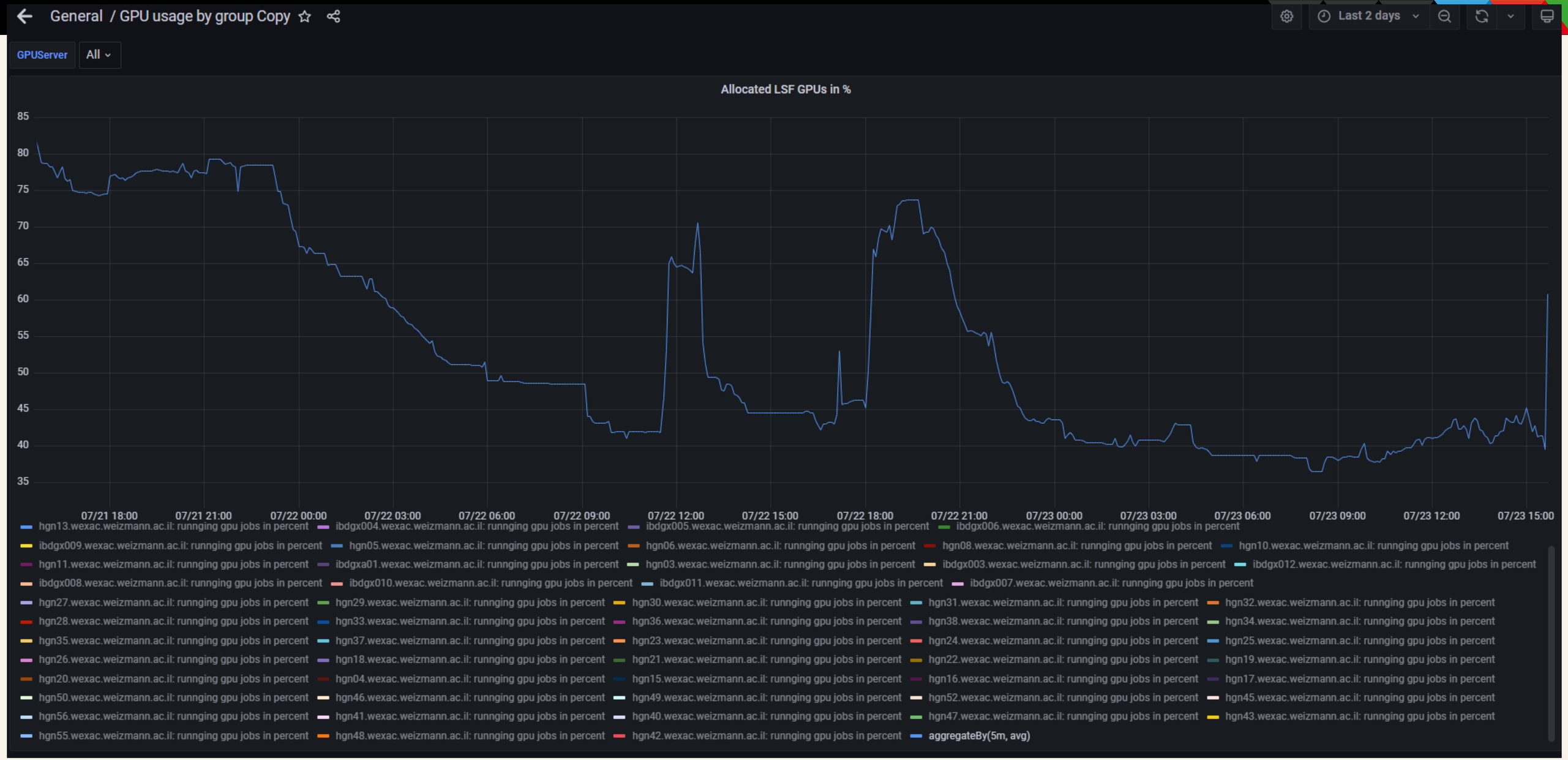
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE



מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

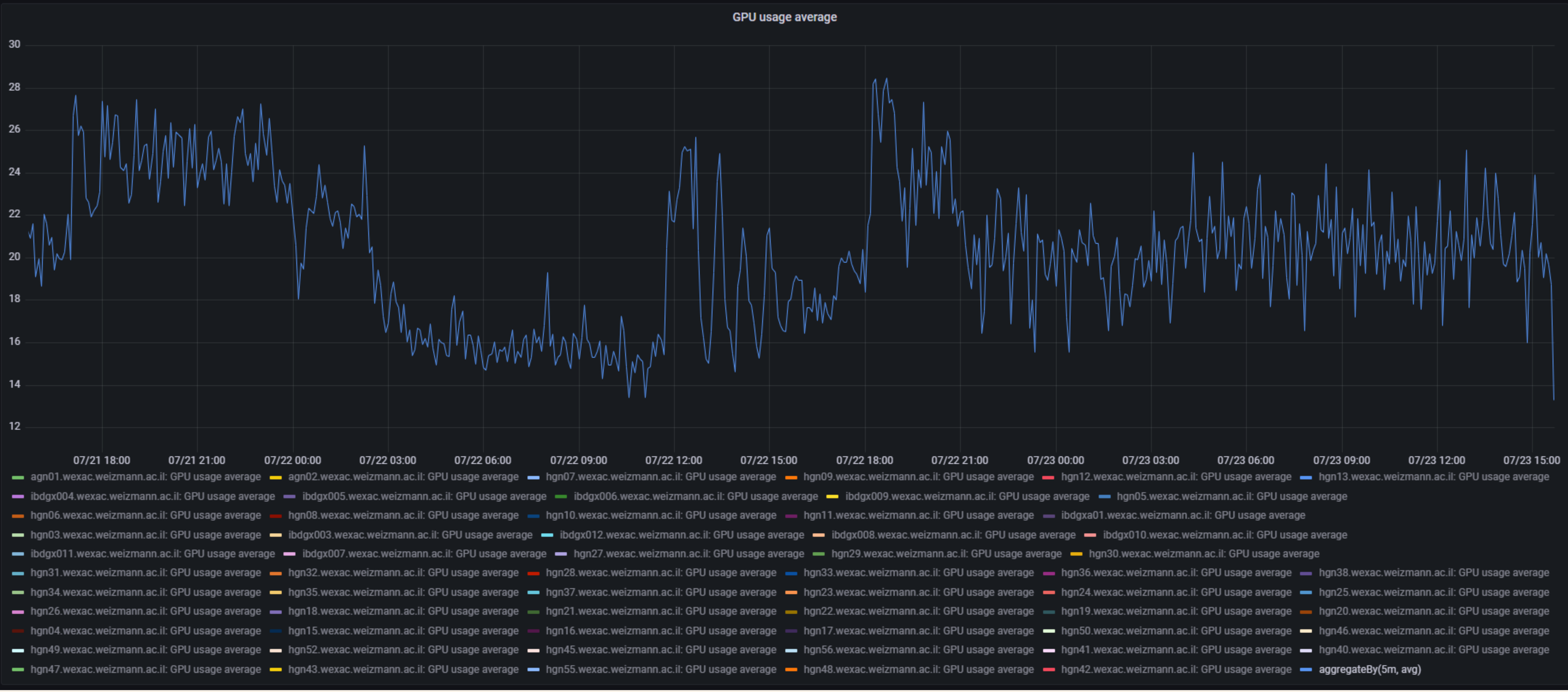

General / All GPU Dashboard
Last 6 hours
Allocated LSF GPU's
GPU usage average
Number of LSF Jobs
number of used GPUs


General / GPU usage by group Copy
Last 2 days
GPUServer
All
Allocated LSF GPUs in %
85
80
75
70
65
60
55
50
45
40
35
07/21 18:00
07/21 21:00
07/22 00:00
07/22 03:00
07/22 06:00
07/22 09:00
07/22 12:00
07/22 15:00
07/22 18:00
07/22 21:00
07/23 00:00
07/23 03:00
07/23 06:00
07/23 09:00
07/23 12:00
07/23 15:00
hgn13.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx004.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx005.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx006.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx009.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn05.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn06.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn08.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn10.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn11.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgxa01.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn03.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx003.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx012.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx008.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx010.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx011.wexac.weizmann.ac.il: runnging gpu jobs in percent
ibdgx007.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn27.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn29.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn30.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn31.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn32.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn28.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn33.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn36.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn38.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn34.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn35.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn37.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn23.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn24.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn25.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn26.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn18.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn21.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn22.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn19.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn20.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn04.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn15.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn16.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn17.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn50.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn46.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn49.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn52.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn45.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn56.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn41.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn40.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn47.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn43.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn55.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn48.wexac.weizmann.ac.il: runnging gpu jobs in percent
hgn42.wexac.weizmann.ac.il: runnging gpu jobs in percent
aggregateBy(5m, avg)

General / GPU usage by group Copy

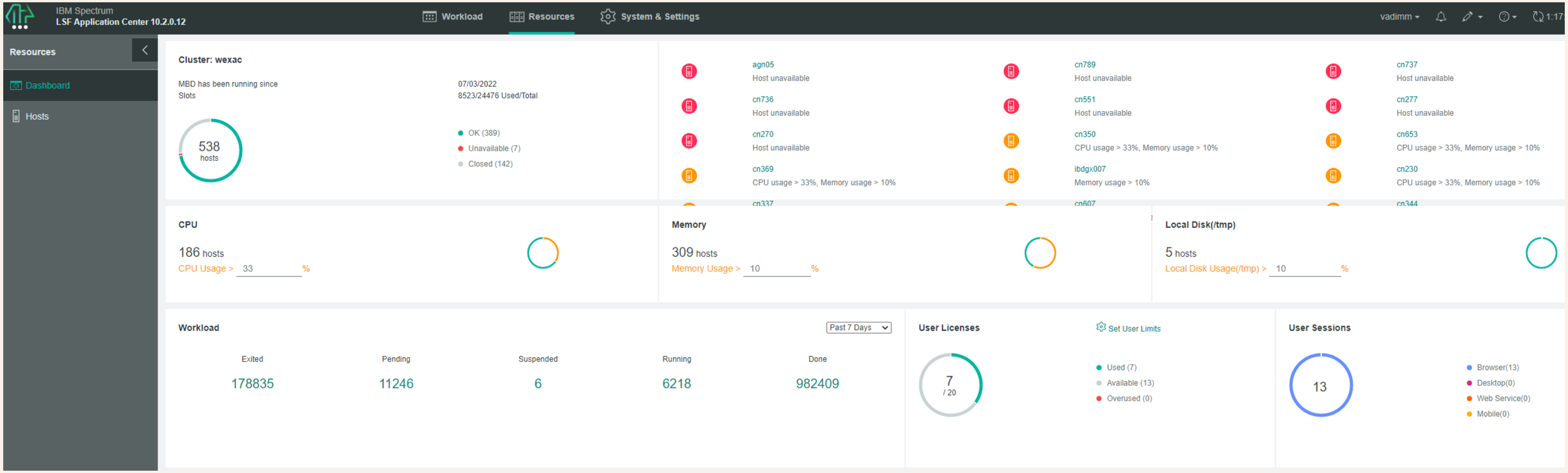
Last 2 days

GPUServer All



LSF PAC - Platform Application Center

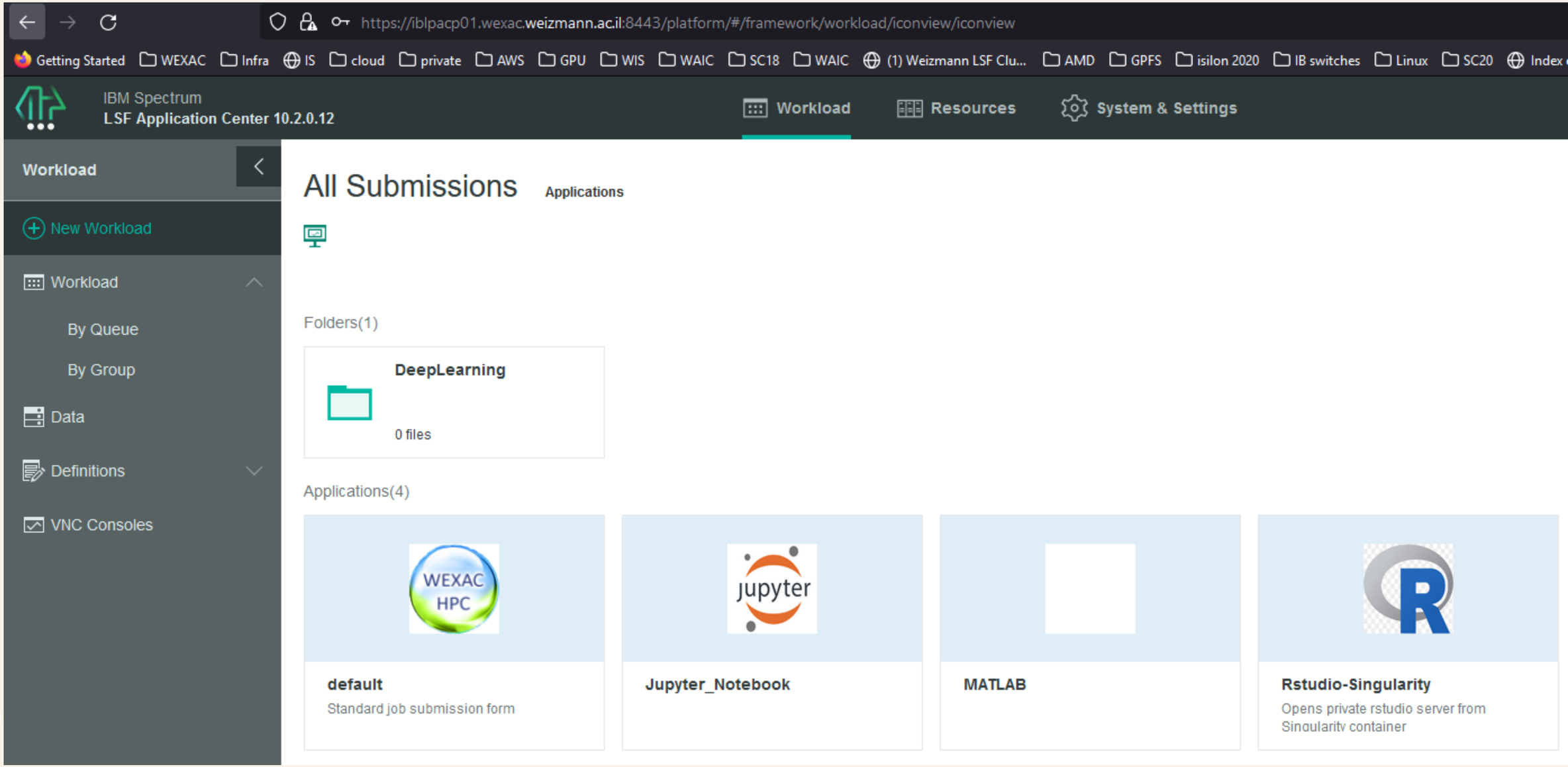
https://iblpacp01.wexac.weizmann.ac.il:8443/



מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

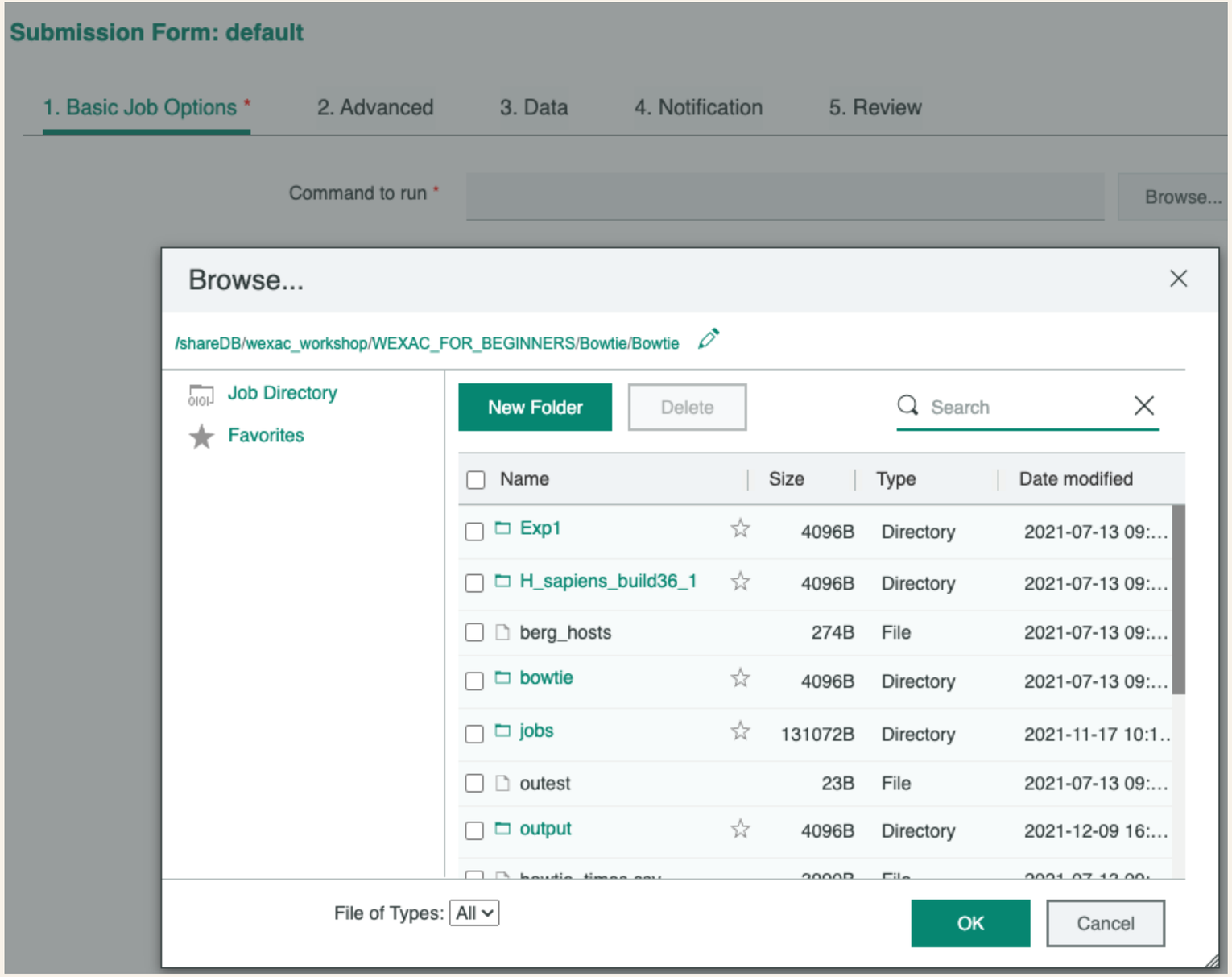
LSF PAC - Platform Application Center

https://iblpacp01.wexac.weizmann.ac.il:8443/



LSF PAC - Platform Application Center

https://iblpacp01.wexac.weizmann.ac.il:8443/



מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Submission Form: default

1. Basic Job Options * | 2. Advanced | 3. Data | 4. Notification | 5. Review

Number of processors: 2

Memory: 4096 MB

Allocate processors on single host: ☑

Resource Requirement string:

Submit to this Queue: new-short

Additional Parameters:

Requirements

--Walltime for your job

Hour: 1

Minutes: 00

Additional Job Options

This job is rerunnable: No

Application profile name: <Use Default>

Project: July25

Reservation ID: <Use Default>

Login Shell: <Use Default>

Submission Form: default

1. Basic Job Options * | 2. Advanced | 3. Data | 4. Notification | 5. Review

Input file

Drag here or browse to upload local file

Add Server File

Delete | Move to... | Open with..

File Name	Type	Size	Actions
/shareDB/wexac_workshop/WEXAC_FOR_BEGINNERS/Bowtie/Bowtie/Exp1	Server	4096B	

Output file: Enter the file path — Browse...

Error file: Enter the file path — Browse...

Submission Form: default

1. Basic Job Options * | 2. Advanced | 3. Data | 4. Notification | 5. Review

Specify when you want to be notified about your workload.

- [x] **Notify me:**
  - [x] When workload starts
  - [x] When workload ends
  - [x] If workload exits
  - [x] If workload is suspended

Notify via:

Browser | Email | Desktop Client | Mobile

Submission Form: default

Memory reservation is (MB): 4096Memory Limit is (MB): 4096 Amount of tasks (-n X) reserved: 2 === Your total amount of memory reservation for this job is (MB): 8192 === Job <561894> is submitted to queue <new-short>.

Basic Job Options

Command to run * :	/shareDB/wexac_workshop/WEXAC_FOR_BEGINNERS/Bowtie/Bowtie//benchm_bowtie.sh	Job Name :	bowtie July25

Advanced

Number of processors :	2	Memory :	4096 MB
Allocate processors on single host :	Yes	Submit to this Queue:	new-short
		Hour :	1
Minutes :	00	This job is rerunnable:	No
Project :	July25		

Data

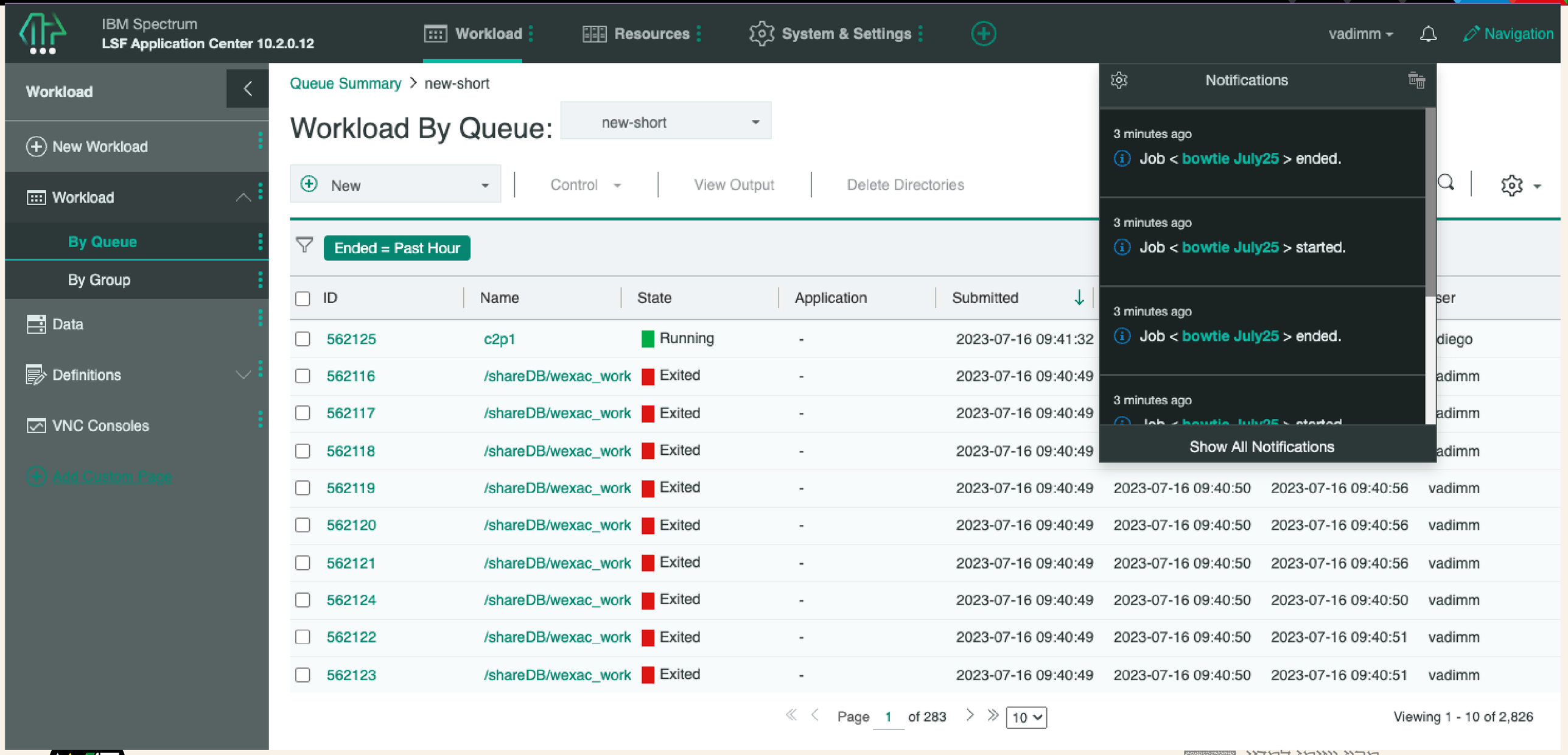
Input file:	/shareDB/wexac_workshop/WEXAC_FOR_BEGINNERS/Bowtie/Bowtie/Exp1

Notification

- [x] Notify me:
  - [x] When workload starts
  - [x] When workload ends
  - [x] If workload exits
  - [x] If workload is suspended

Notify via:

Browser Email


IBM Spectrum
LSF Application Center 10.2.0.12
Workload
Resources
System & Settings
vadimm
Navigation
Workload
New Workload
Workload
By Queue
By Group
Data
Definitions
VNC Consoles
Add Custom Page
Queue Summary > new-short
Workload By Queue: new-short
New
Control
View Output
Delete Directories
Ended = Past Hour
ID
Name
State
Application
Submitted
562125 c2p1 Running - 2023-07-16 09:41:32 diego
562116 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 adimm
562117 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 adimm
562118 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 adimm
562119 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 2023-07-16 09:40:50 2023-07-16 09:40:56 vadimm
562120 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 2023-07-16 09:40:50 2023-07-16 09:40:56 vadimm
562121 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 2023-07-16 09:40:50 2023-07-16 09:40:56 vadimm
562124 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 2023-07-16 09:40:50 2023-07-16 09:40:50 vadimm
562122 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 2023-07-16 09:40:50 2023-07-16 09:40:51 vadimm
562123 /shareDB/wexac_work Exited - 2023-07-16 09:40:49 2023-07-16 09:40:50 2023-07-16 09:40:51 vadimm
Notifications
3 minutes ago
Job < bowtie July25 > ended.
3 minutes ago
Job < bowtie July25 > started.
3 minutes ago
Job < bowtie July25 > ended.
3 minutes ago
Show All Notifications
Page 1 of 283
10
Viewing 1 - 10 of 2,826

IBM Spectrum
LSF RTM 10.2.0.12

Console | Graphs | Reporting | Logs | Cluster | JobIQ | License | Syslog | Thold

Batch Job Filters for User 'vadimm' [Updated 4 Minutes and 24 Seconds Ago] [JobIQ Dashboard]

User: vadimm | Cluster: wexac | Queue: All | Project: All | Status: DONE | Jobs: 30 | Go | Clear | Export

History: 1 Hour | Effic: All | ShowOnly: N/A | MemSize: N/A | RunTime: N/A | Dynamic | Cluster TZ

JobID | Search: Enter a search term | ResReq: Enter an LSF select statement | Search ResReq by: Host

1 to 30 of 58 [1 2]

JobID	Job Name	Queue	User	UGroup	App	Status	Exec Host	SubHost	Lic	Idle Job	Long RJob	Pend Dpnd	Mem Use	State Changes	Mem Res	Mem Wasted	Max Mem	Max GPU Mem	CPU Usage	Core Eff	Num Nodes	Num CPUs
561593	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn603	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.52m	97.25%	1	1
561592	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn603	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.44m	97.29%	1	1
561590	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn603	cn395		●	●	●	●	2	4.00G	878.00M	3.14G	-	6.38m	97.14%	1	1
561594	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn603	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.37m	97.21%	1	1
561591	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn603	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.33m	97.20%	1	1
561584	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	878.00M	3.14G	-	6.2m	97.60%	1	1
561583	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.2m	97.56%	1	1
561578	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	877.00M	3.14G	-	6.15m	97.38%	1	1
561579	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.13m	97.55%	1	1
561582	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.05m	97.60%	1	1
561585	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.04m	97.76%	1	1
561581	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	877.00M	3.14G	-	6.03m	97.57%	1	1
561580	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn591	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	6.01m	97.75%	1	1
561576	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn588	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	5.43m	96.97%	1	1
561575	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn588	cn395		●	●	●	●	2	4.00G	877.00M	3.14G	-	5.4m	96.75%	1	1
561577	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn588	cn395		●	●	●	●	2	4.00G	878.00M	3.14G	-	5.41m	96.84%	1	1
561595	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn709	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	5.41m	96.90%	1	1
561574	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn588	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	5.25m	96.95%	1	1
561573	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn588	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	5.23m	96.85%	1	1
561550	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn362	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	5.07m	96.28%	1	1
561596	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn709	cn395		●	●	●	●	2	4.00G	878.00M	3.14G	-	5.06m	96.66%	1	1
561599	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn709	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	5.04m	96.91%	1	1
561598	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn709	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	4.99m	96.96%	1	1
561600	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn709	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	4.99m	96.92%	1	1
561597	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn709	cn395		●	●	●	●	2	4.00G	879.00M	3.14G	-	4.99m	96.89%	1	1
561562	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn186	cn395		●	●	●	●	2	4.00G	886.00M	3.13G	-	4.86m	96.47%	1	1
561561	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn186	cn395		●	●	●	●	2	4.00G	886.00M	3.13G	-	4.82m	96.43%	1	1
561560	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn186	cn395		●	●	●	●	2	4.00G	886.00M	3.13G	-	4.8m	95.97%	1	1
561564	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn186	cn395		●	●	●	●	2	4.00G	887.00M	3.13G	-	4.8m	96.63%	1	1
561563	/shareDB/wexac_works...	new-short	vadimm	testing-wx-grp-lsf	-	DONE	cn186	cn395		●	●	●	●	2	4.00G	886.00M	3.13G	-	4.79m	96.51%	1	1

1 to 30 of 58 [1 2]

Warning Efficiency | Alarm Efficiency | Flapping | Dependencies | Invalid Dependencies | Exited

Batch Job 561592 for User All Users [JobIQ Dashboard] [JobIQ Jobs]

Job Detail | Job Graphs | Host Graphs | License Usage

General Information			
Job ID	561592	Status	DONE
Job Name	/shareDB/wexac_workshop/Bowtie/bowtie/bowtie2-2.0.5_x86/bowtie2 -x /shareDB/wexac_workshop/Bowtie/H_sapiens_build36_1/b		
Project	default	License Project	-
Queue	new-short	Cluster Name	wexac
User	vadimm	User Group	testing-wx-grp-lsf
Charged SAAP	/vadimm		
Mail User		Job Priority	50

Submission Details			
Submit Time	2023-07-16 09:17:06	Number of CPUs	1
Submit Host	cn395		
Asked Hosts/Groups	-	Runtime Estimate	-
Combined ResReq	select[(type = any) && (type == any)] order[-slots:-maxslots:-mem] rusage[mem=4096.00] affinity[thread(1)*1]		
Submit Command	/shareDB/wexac_workshop/Bowtie/bowtie/bowtie2-2.0.5_x86/bowtie2 -x /shareDB/wexac_workshop/Bowtie/H_sapiens_build36_1/build36.1 --end-to-end --sensitive -U /shareDB/wexac_workshop/Bowtie/Exp1/Inputbr -S /sha 1-outbr.sam		
Submit Directory	/shareDB/wexac_workshop/Bowtie		
Resource Requirements	rusage[mem=4096]		
Output File	/shareDB/wexac_workshop/Bowtie/jobs/out.%J		
Error File	/shareDB/wexac_workshop/Bowtie/jobs/err.%J		

Execution Environment			
Exec Host	cn603		
Start Time	2023-07-16 09:17:07		
Username	vadimm	UID String	25003
User Home	/home/labs/testing/vadimm	Working Dir	/shareDB/wexac_workshop/Bowtie
Max Processors	1	Max Allocated Slots	1
CPU Limit	-	Run Time Limit	1 Days
Memory Limit	4.000G	Swap Limit	-
Effective ResReq	select[((type = any) && (type == any))] order[-slots:-maxslots:-mem] rusage[mem=4096.00] affinity[thread(1)*1]		

Current/Last Status					
PGIDS	10755				
PIDS	10755, 10783, 10796, 10803				
Threads	5				
Pend Time	1 Seconds	PROV Time	-	Run Time	6.62 Minutes
Effective Pending Time Limit	-				
PSUSP Time	-	USUSP Time	-	SSUSP Time	-
Unknown Time	-				
Cumulative CPU	6.44 Minutes	System Time	-	User Time	6.43 Minutes
Cur Memory Used	3.142G	Max Memory Used	3.142G	Job Efficiency	97.29%
Cur V.Memory Size	-	Max V.Memory Size	-		
Exit Code	0	End Time	2023-07-16 09:23:44		

Submission Form: test

Memory reservation is (MB): 3080Memory Limit is (MB): 3080 Amount of tasks (-n X) reserved: 2 === Your total amount of memory reservation for this job is (MB): 6160 === Job <562127> is submitted to queue <new-short>.

Form Name: test

1. Basic Job Options * | 2. Advanced | 3. Data | 4. Notification | 5. Review

Basic Job Options

Command to run * :	/shareDB/wexac_workshop/WEXAC_FOR_BEGINNERS/Bowtie/Bowtie/benchm_bowtie.sh	Job Name :	Bowtie July 25 - test2

Advanced

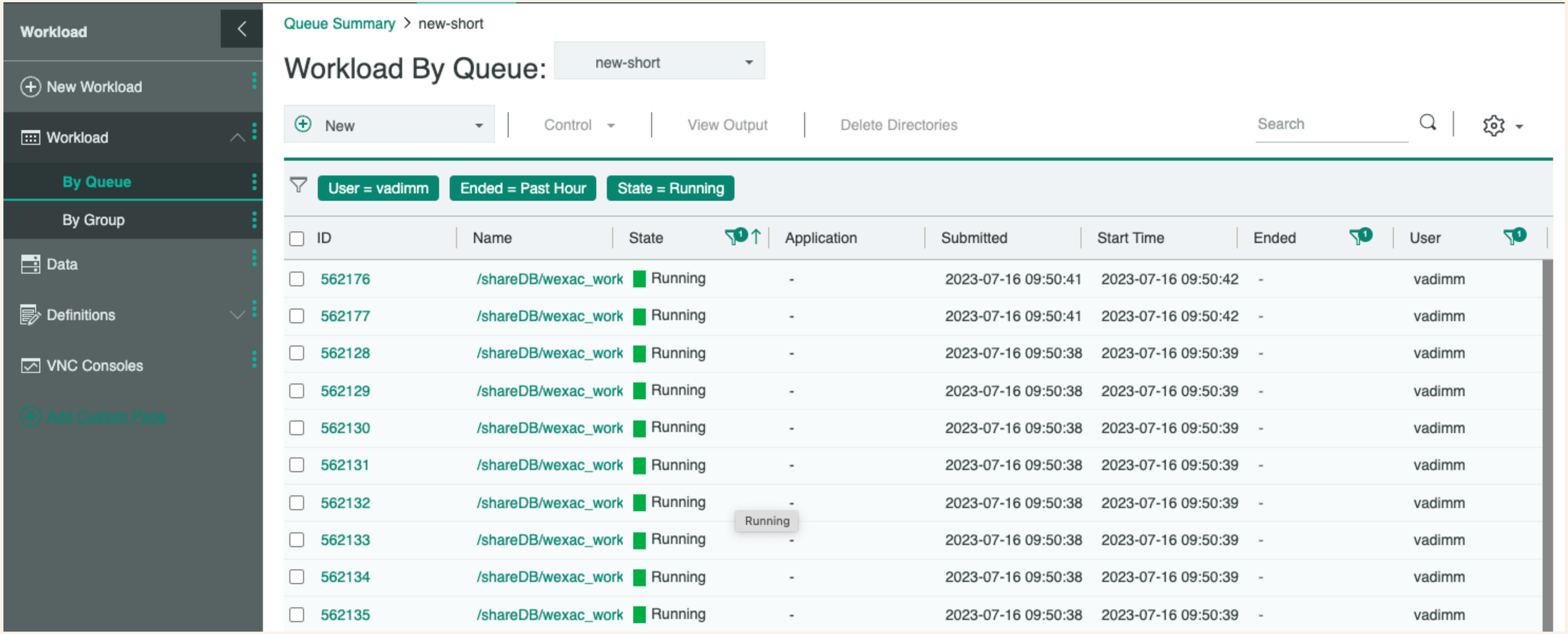
Number of processors :	2	Memory :	3080 MB
Allocate processors on single host :	Yes	Submit to this Queue:	new-short
		Hour :	0
Minutes :	30	This job is rerunnable:	No
Project :	bowtie5		

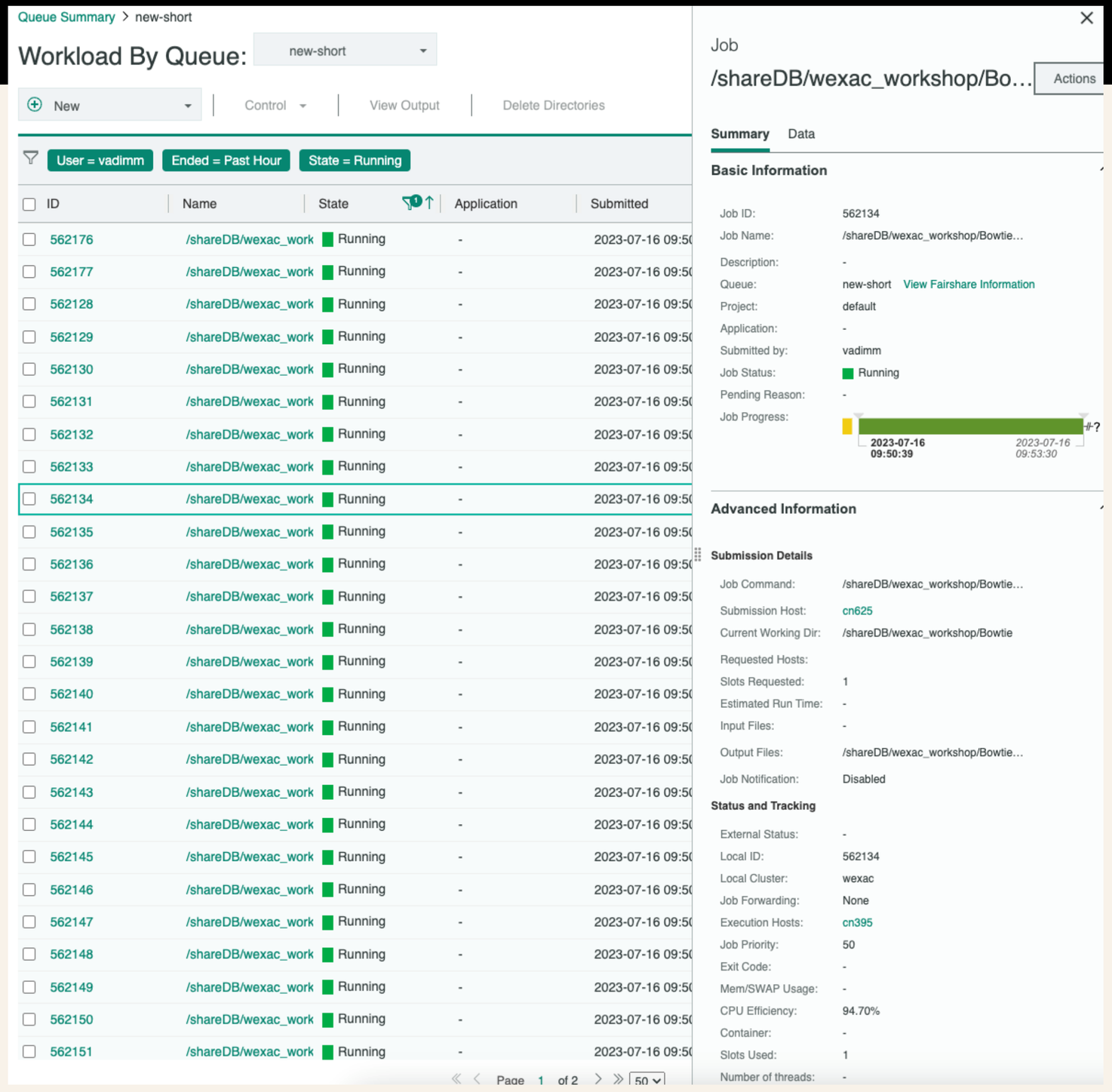
Notification

- [x] Notify me:
  - [x] When workload starts
  - [x] When workload ends
  - [x] If workload exits
  - [x] If workload is suspended

Notify via:

Browser | Email


Workload
New Workload
Workload
By Queue
By Group
Data
Definitions
VNC Consoles
Add Custom Page
Queue Summary > new-short
Workload By Queue: new-short
New
Control
View Output
Delete Directories
Search
User = vadimm
Ended = Past Hour
State = Running
ID Name State Application Submitted Start Time Ended User
562176 /shareDB/wexac_work Running - 2023-07-16 09:50:41 2023-07-16 09:50:42 - vadimm
562177 /shareDB/wexac_work Running - 2023-07-16 09:50:41 2023-07-16 09:50:42 - vadimm
562128 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm
562129 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm
562130 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm
562131 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm
562132 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm
Running
562133 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm
562134 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm
562135 /shareDB/wexac_work Running - 2023-07-16 09:50:38 2023-07-16 09:50:39 - vadimm



LSF PAC - Platform Application Center

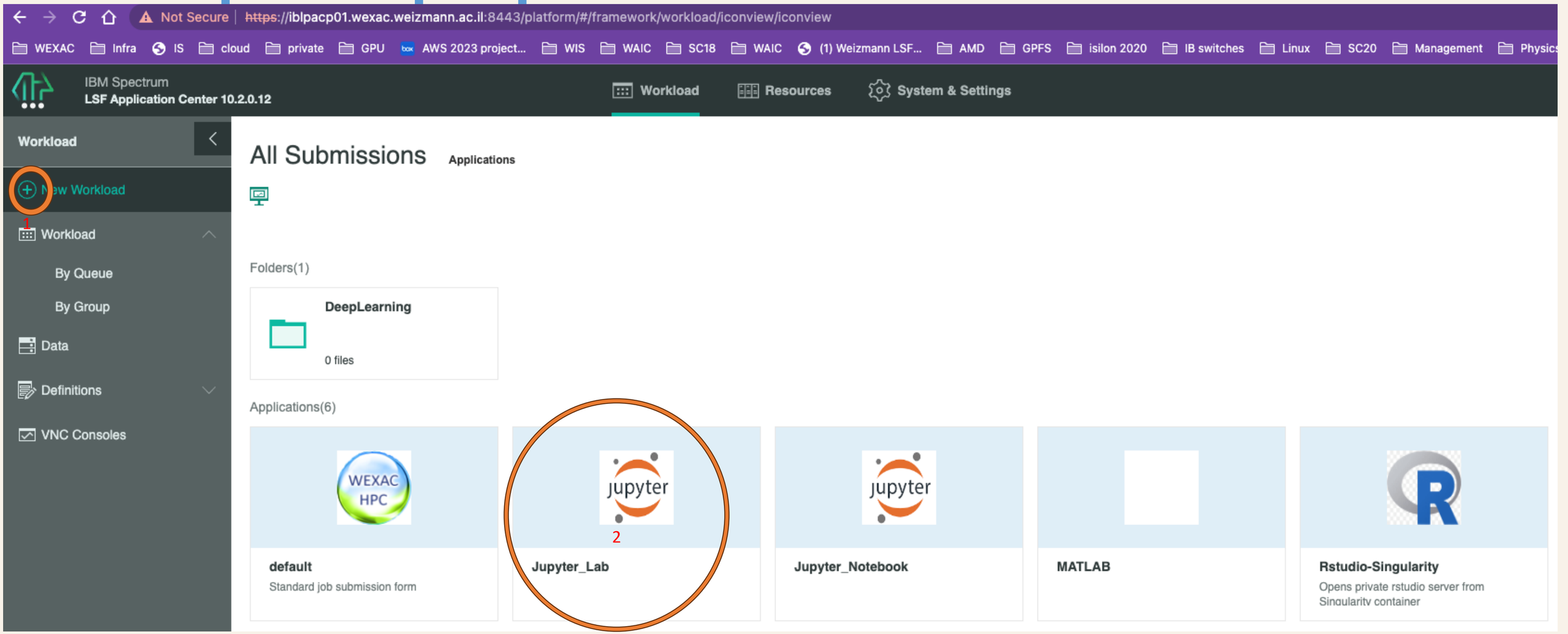
https://iblpacp01.wexac.weizmann.ac.il:8443/

```
[vadimm@access output]$ ls -la
total 20930720
drwxrwxrwx 2 root   root                 8192 Jul 16 09:17 .
drwxrwxrwx 7 root   root                16384 Jun 13 11:29 ..
-rw-r--r-- 1 vadimm testing-wx-grp 759274937 Jul 16 09:23 Exp1-outbe.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759495212 Jul 16 09:23 Exp1-outbf.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759293416 Jul 16 09:23 Exp1-outbg.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759344489 Jul 16 09:23 Exp1-outbh.sam
-rw-r--r-- 1 vadimm testing-wx-grp 760894038 Jul 16 09:23 Exp1-outbi.sam
-rw-r--r-- 1 vadimm testing-wx-grp 760653451 Jul 16 09:23 Exp1-outbj.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759836464 Jul 16 09:23 Exp1-outbk.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759183533 Jul 16 09:21 Exp1-outbl.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759169655 Jul 16 09:21 Exp1-outbm.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759138626 Jul 16 09:21 Exp1-outbn.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759582466 Jul 16 09:22 Exp1-outbo.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759170838 Jul 16 09:23 Exp1-outbp.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759237494 Jul 16 09:23 Exp1-outbq.sam
-rw-r--r-- 1 vadimm testing-wx-grp 760350597 Jul 16 09:23 Exp1-outbr.sam
-rw-r--r-- 1 vadimm testing-wx-grp 761008311 Jul 16 09:23 Exp1-outbs.sam
-rw-r--r-- 1 vadimm testing-wx-grp 760287262 Jul 16 09:23 Exp1-outbt.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759705155 Jul 16 09:22 Exp1-outbu.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759567102 Jul 16 09:22 Exp1-outbv.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759252744 Jul 16 09:22 Exp1-outbw.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759232941 Jul 16 09:22 Exp1-outbx.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759656301 Jul 16 09:22 Exp1-outby.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759269421 Jul 16 09:22 Exp1-outbz.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759701559 Jul 16 09:21 Exp1-outca.sam
-rw-r--r-- 1 vadimm testing-wx-grp 761172986 Jul 16 09:21 Exp1-outcb.sam
-rw-r--r-- 1 vadimm testing-wx-grp 760700650 Jul 16 09:21 Exp1-outcc.sam
-rw-r--r-- 1 vadimm testing-wx-grp 759748746 Jul 16 09:21 Exp1-outcd.sam
-rw-r--r-- 1 vadimm testing-wx-grp 186778087 Jul 16 09:18 Exp1-outce.sam
```

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

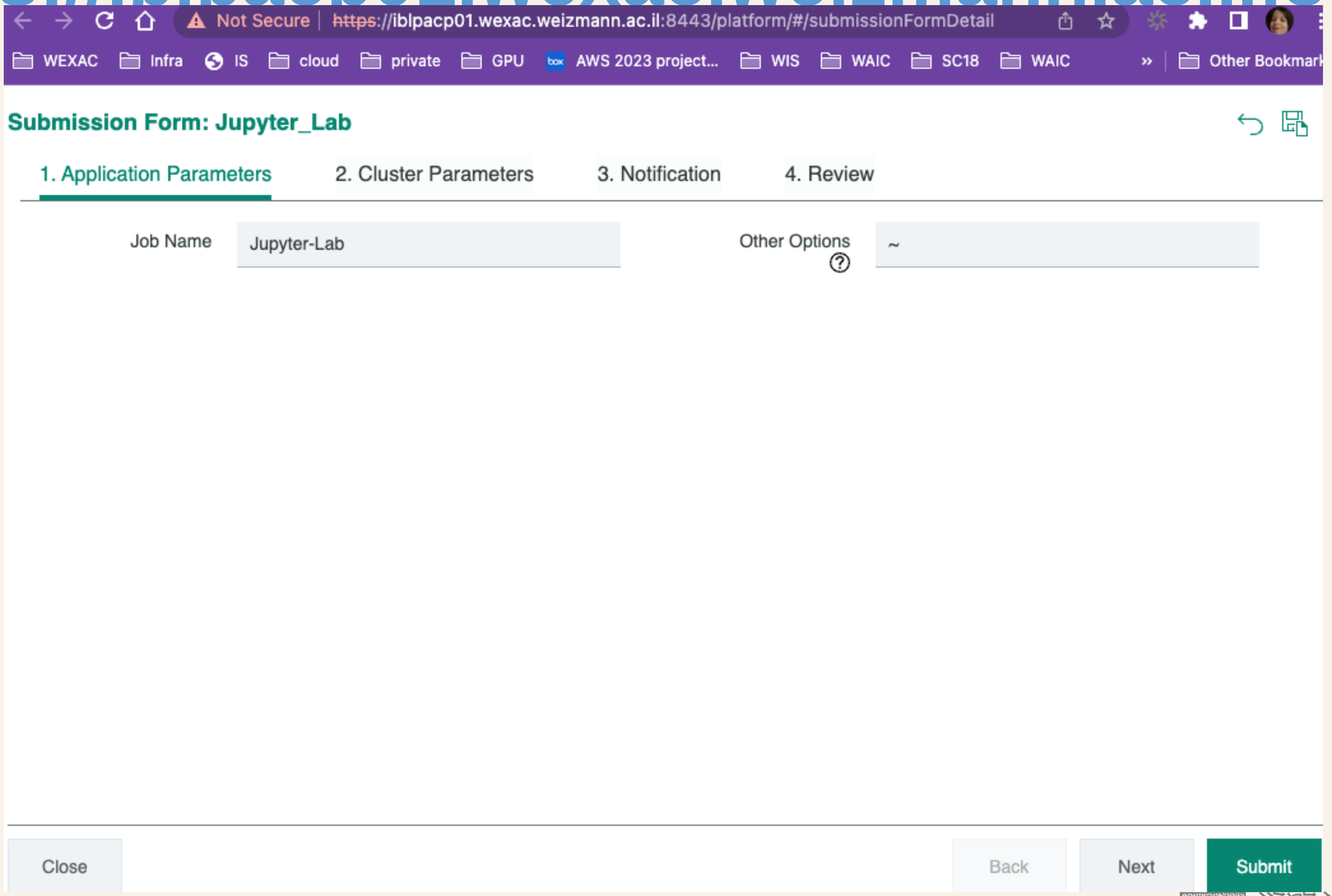
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



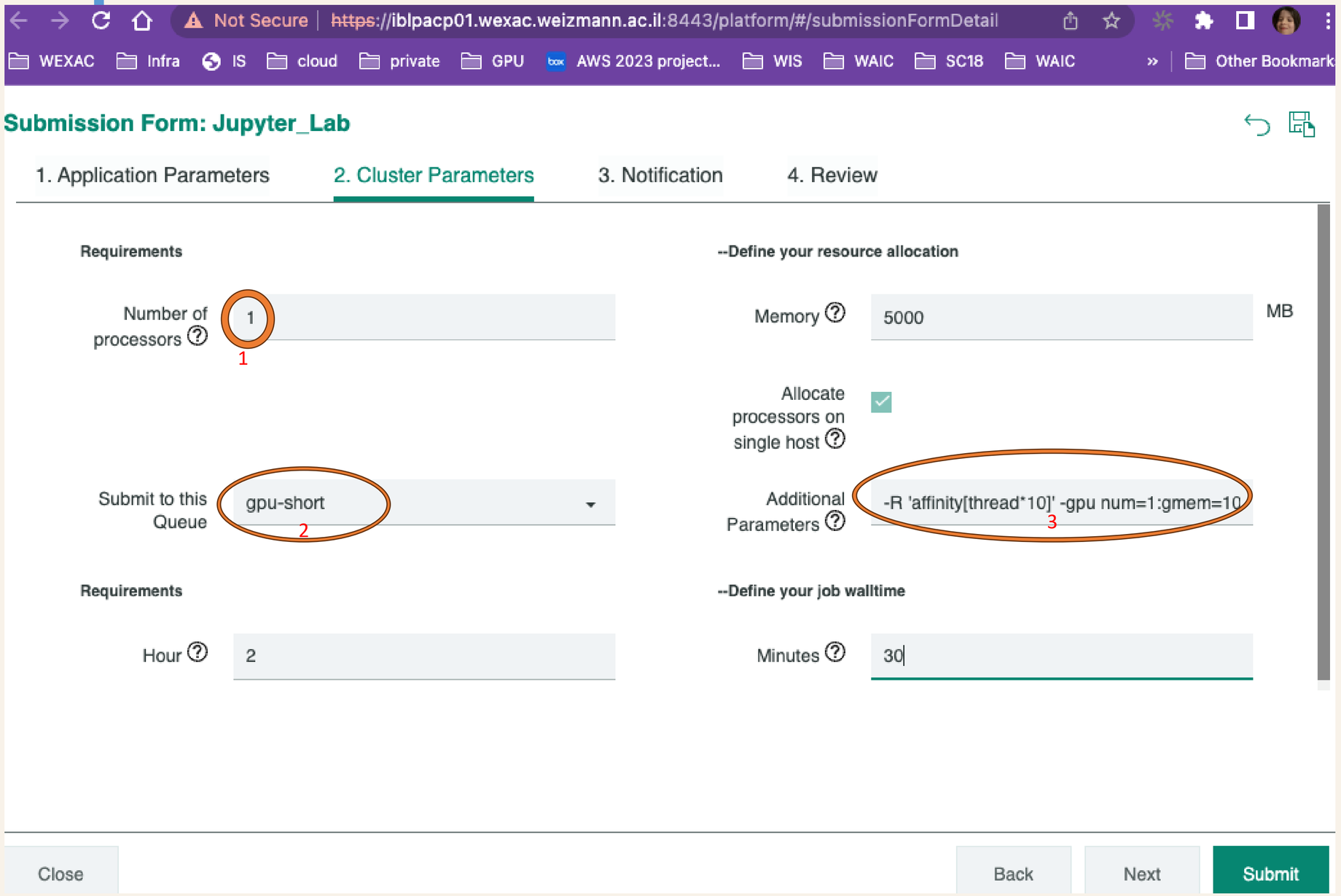
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



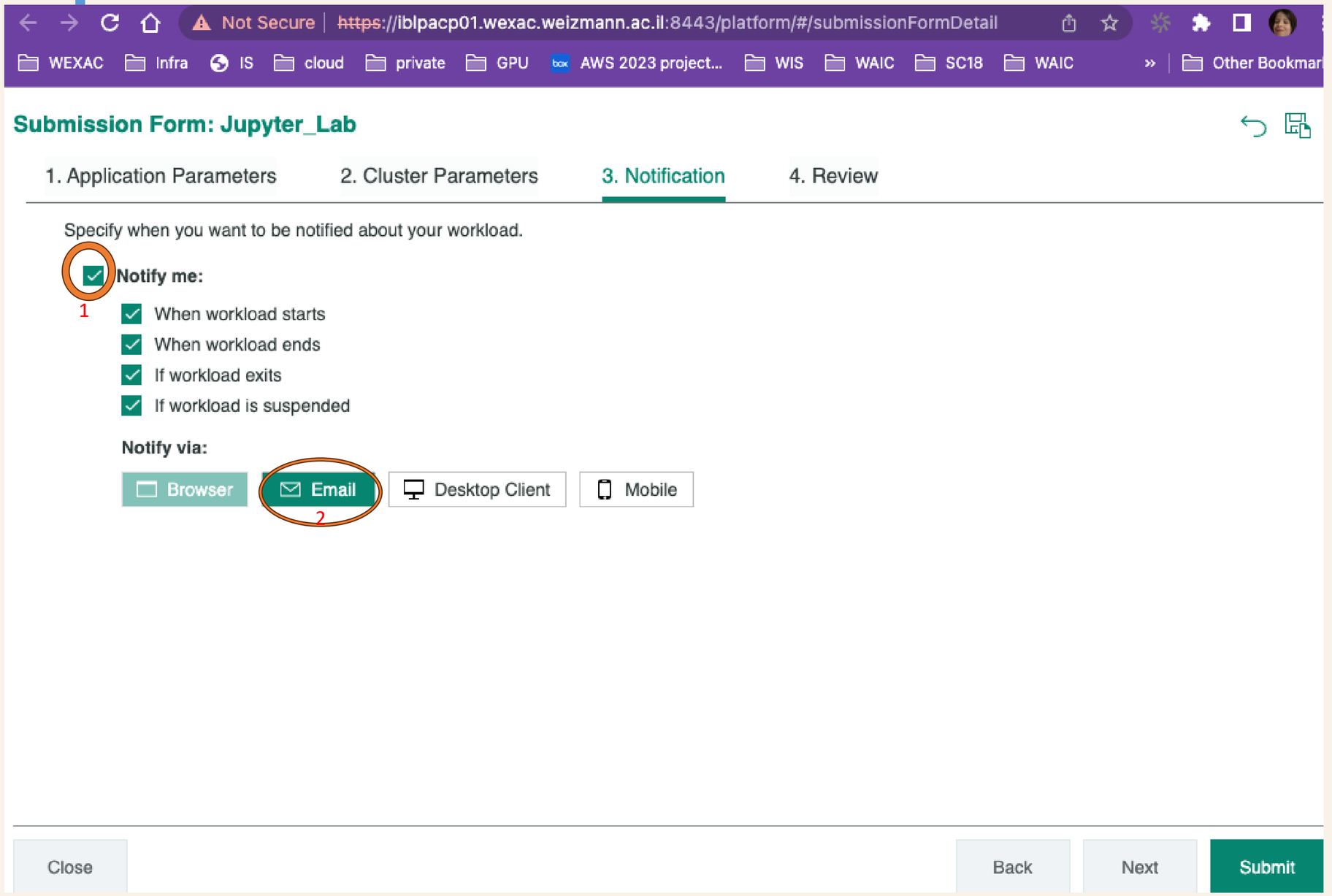
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



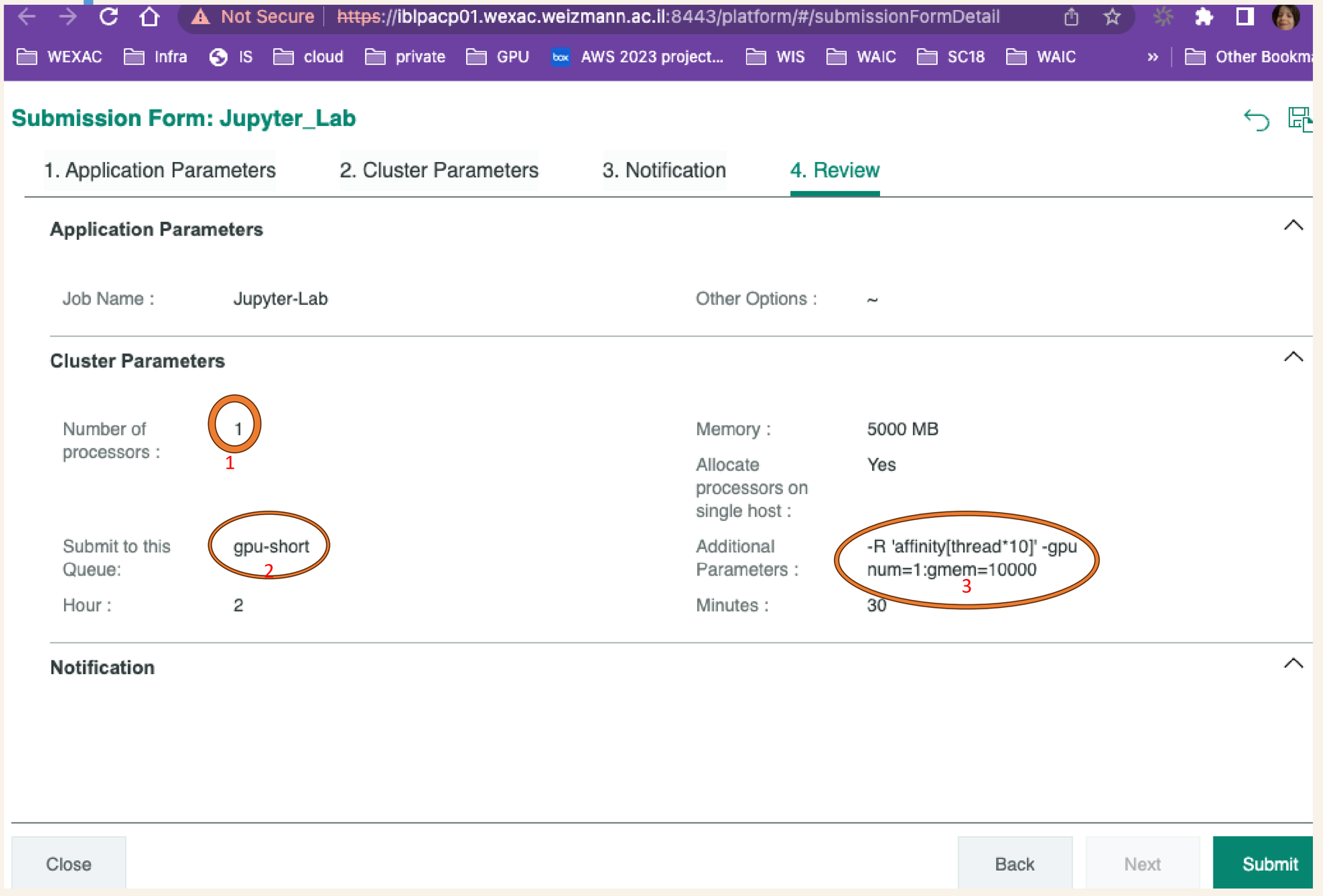
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



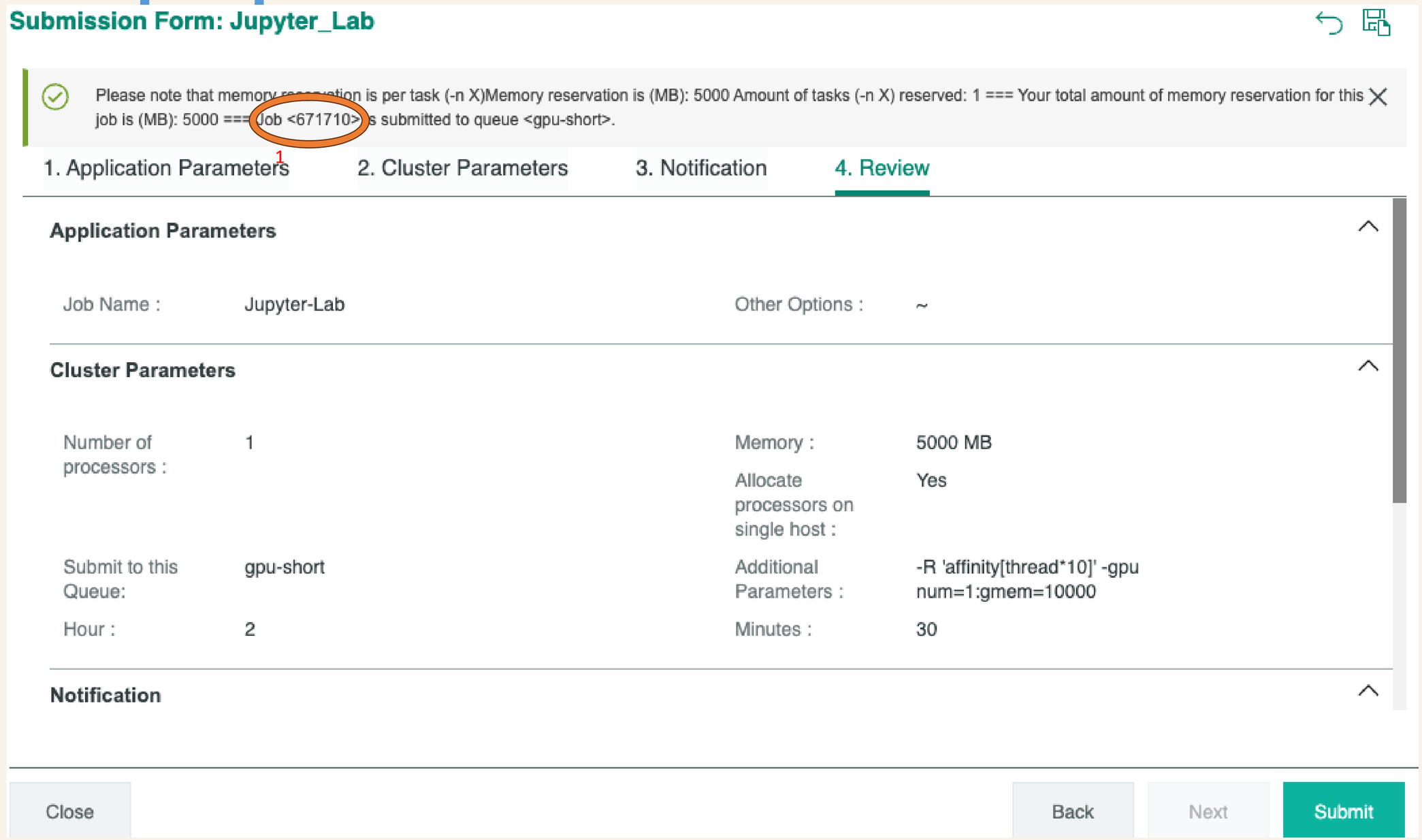
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



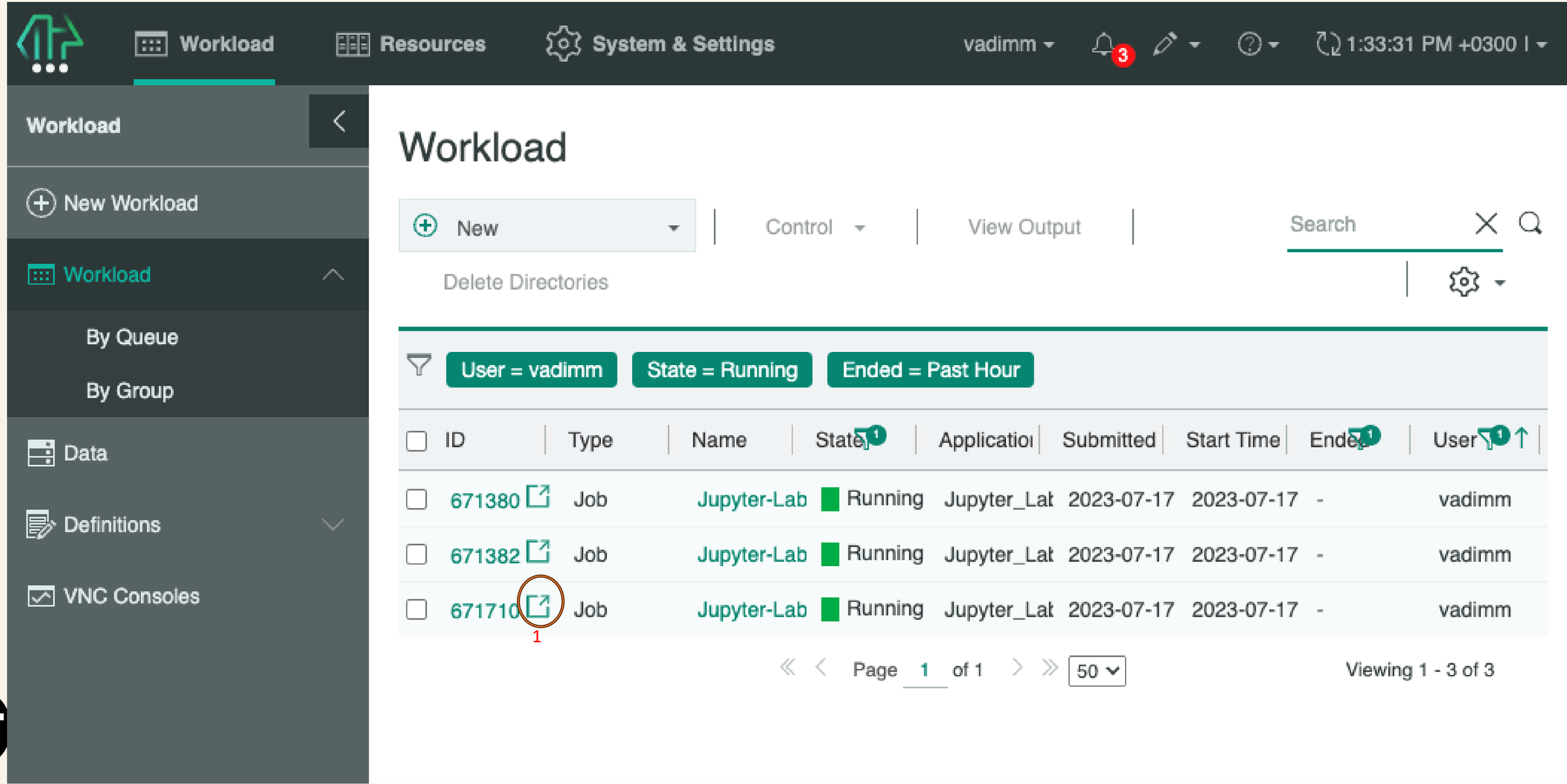
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



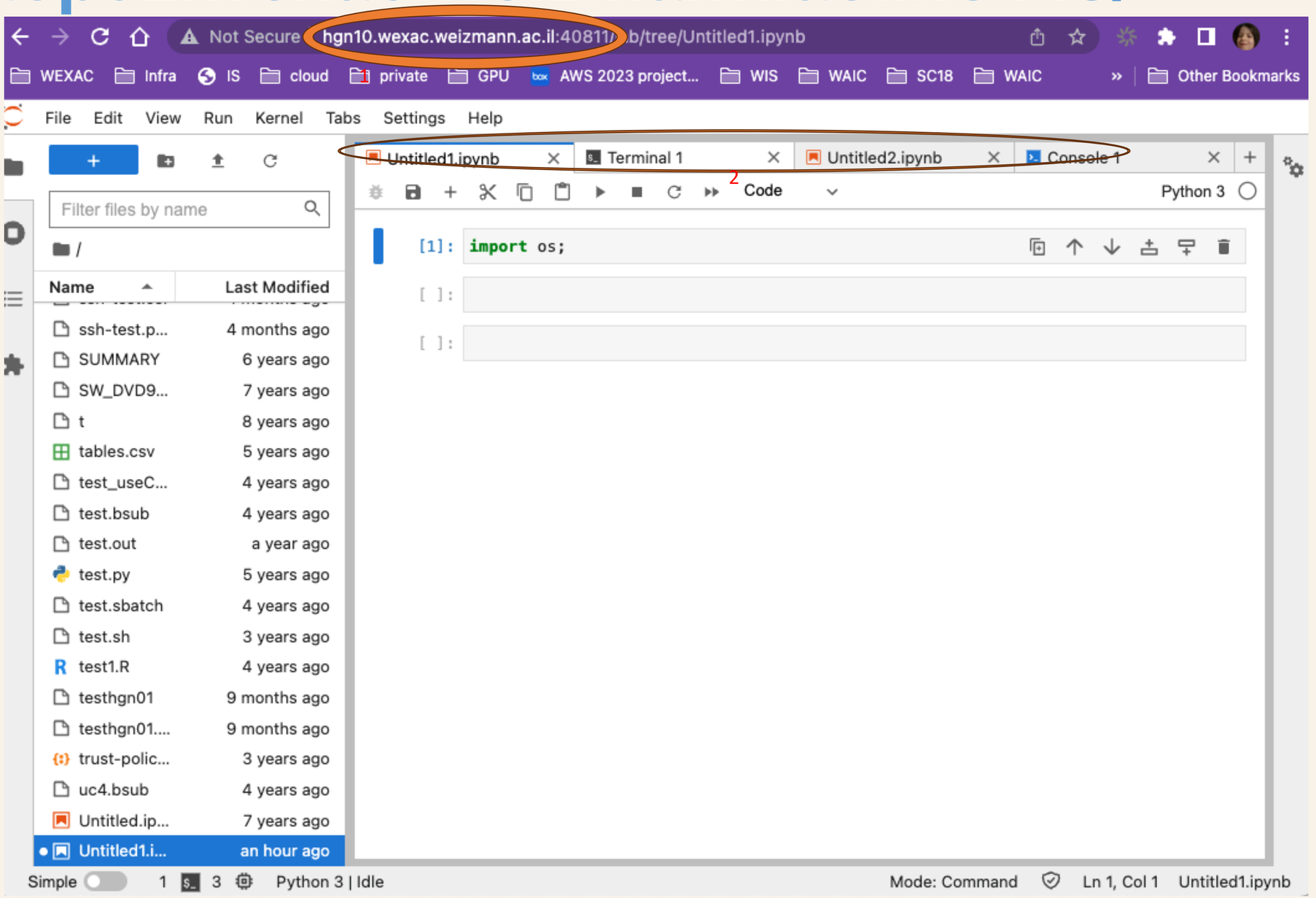
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



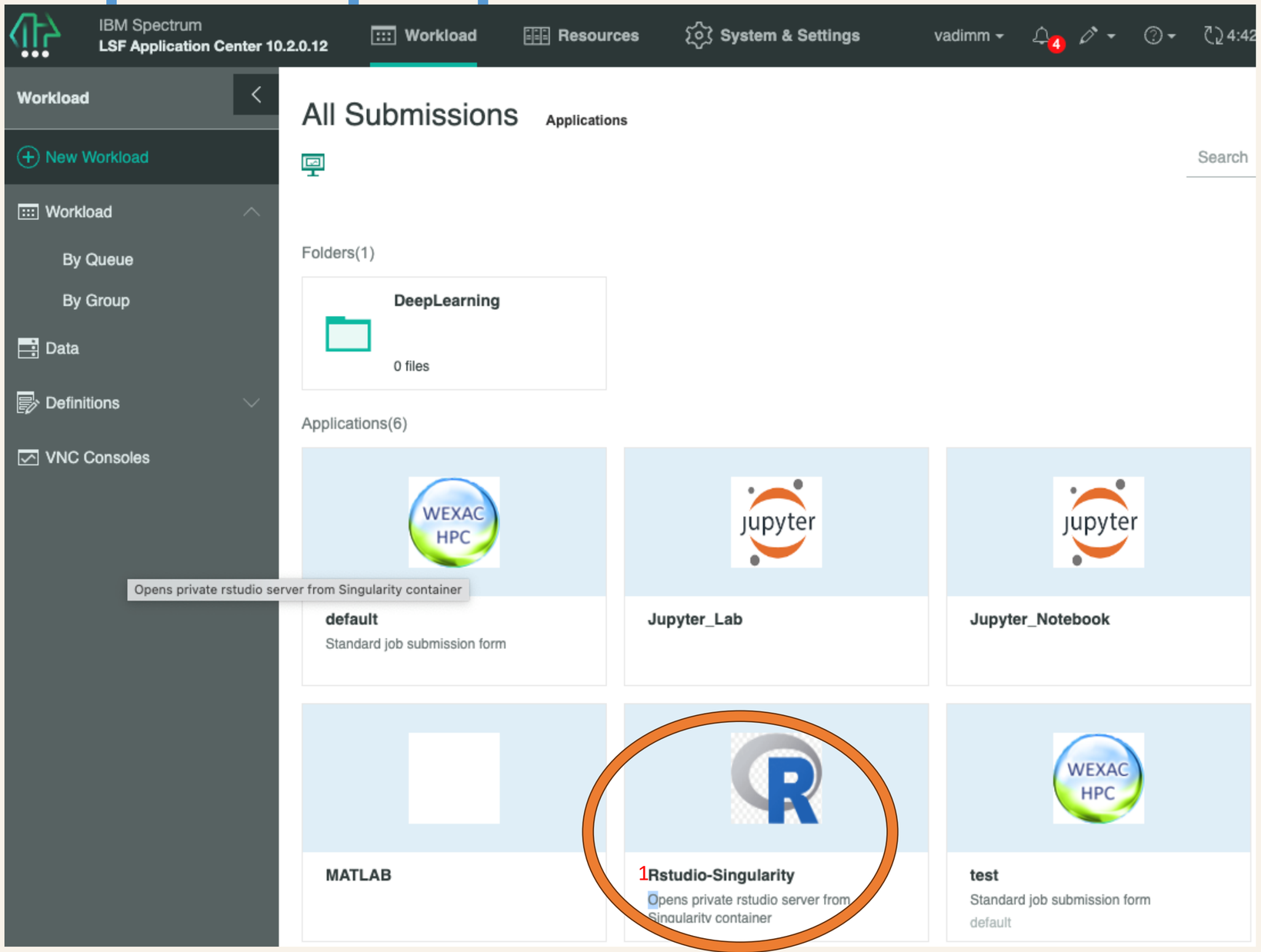
LSF PAC - Interactive Jupyter Lab

https://iblpacp01.wexac.weizmann.ac.il:8443/



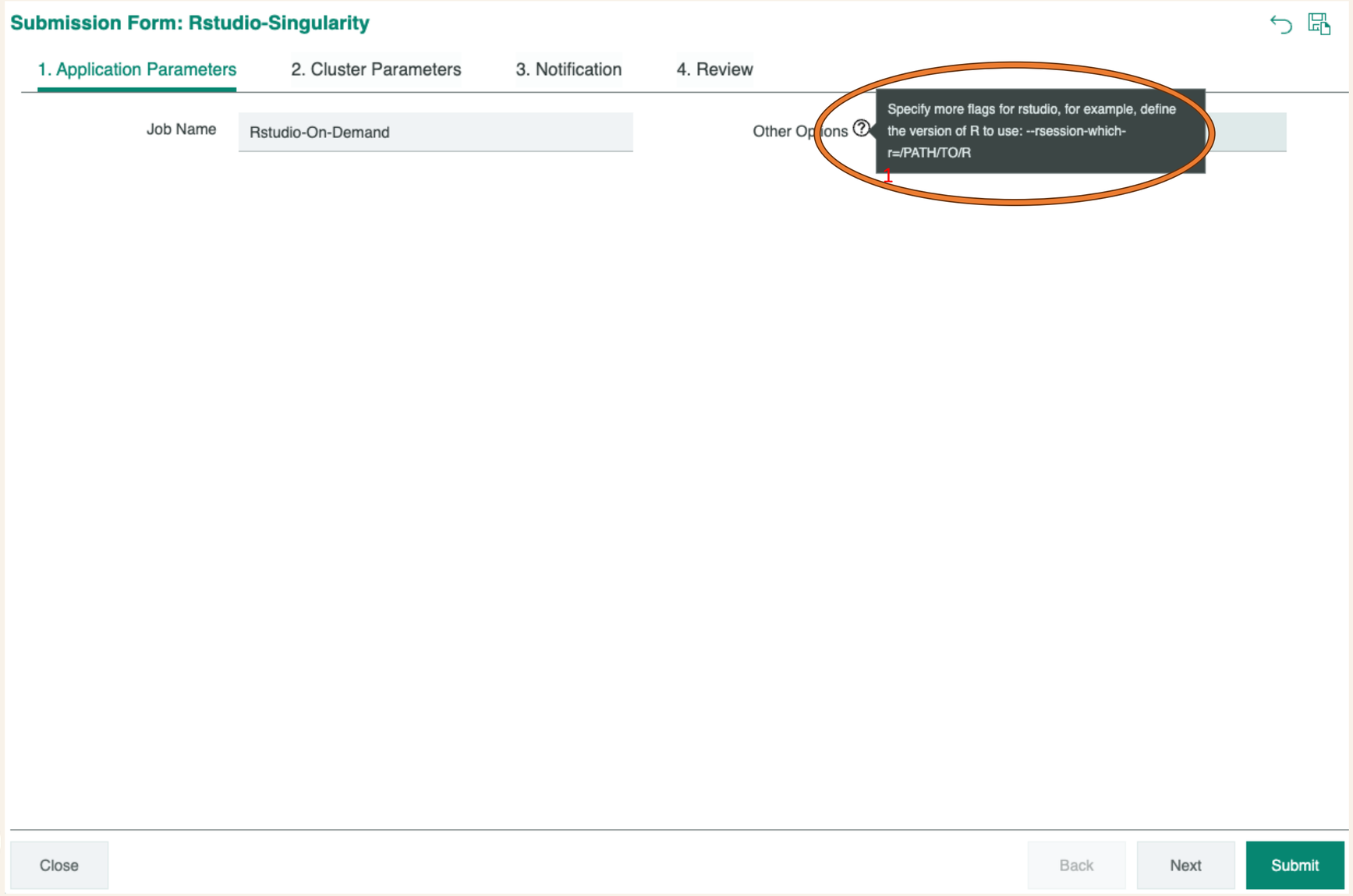
LSF PAC - Interactive Rstudio

https://iblpacp01.wexac.weizmann.ac.il:8443/



LSF PAC - Interactive Rstudio

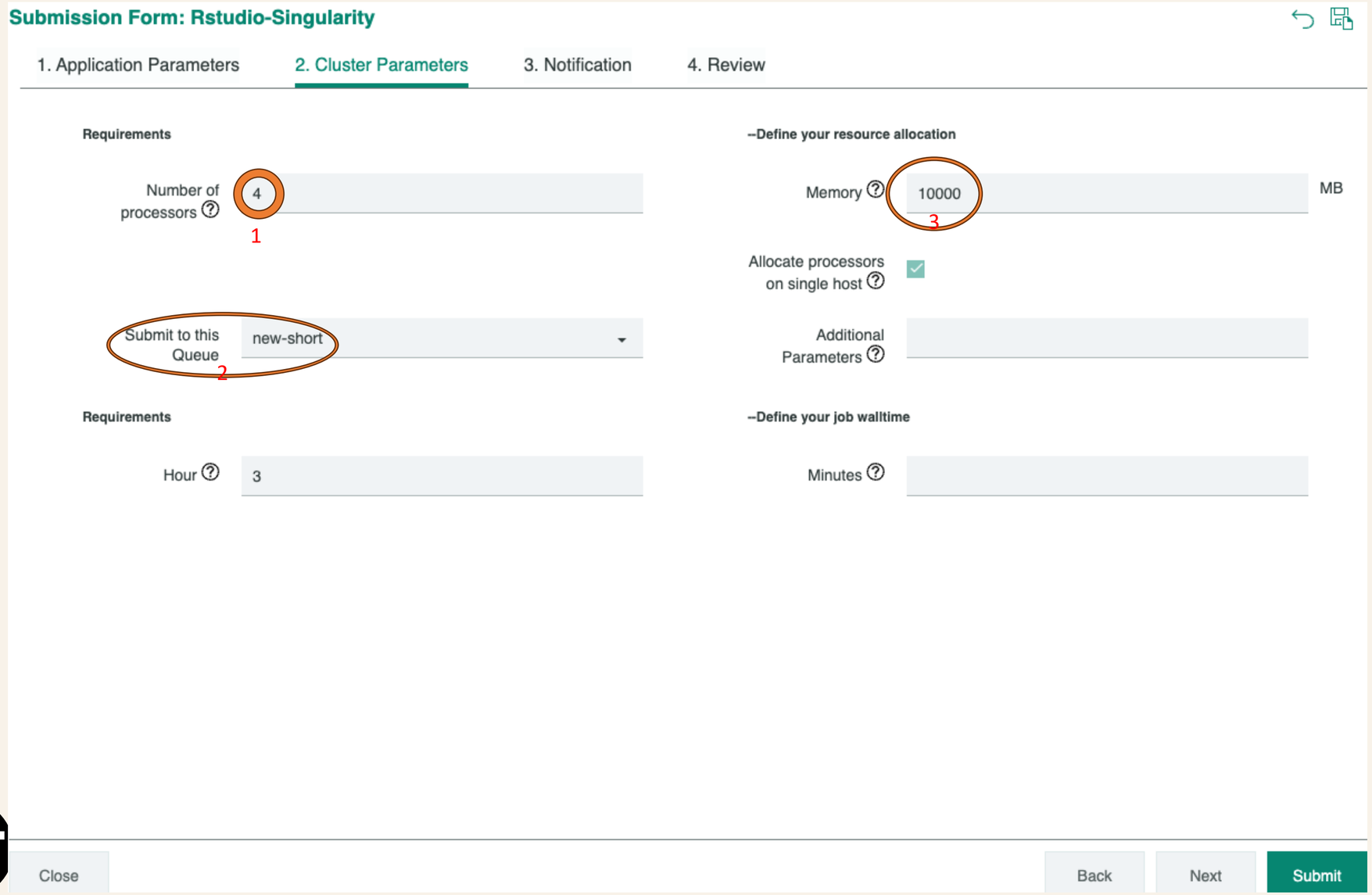
https://iblpacp01.wexac.weizmann.ac.il:8443/



מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

LSF PAC - Interactive Rstudio

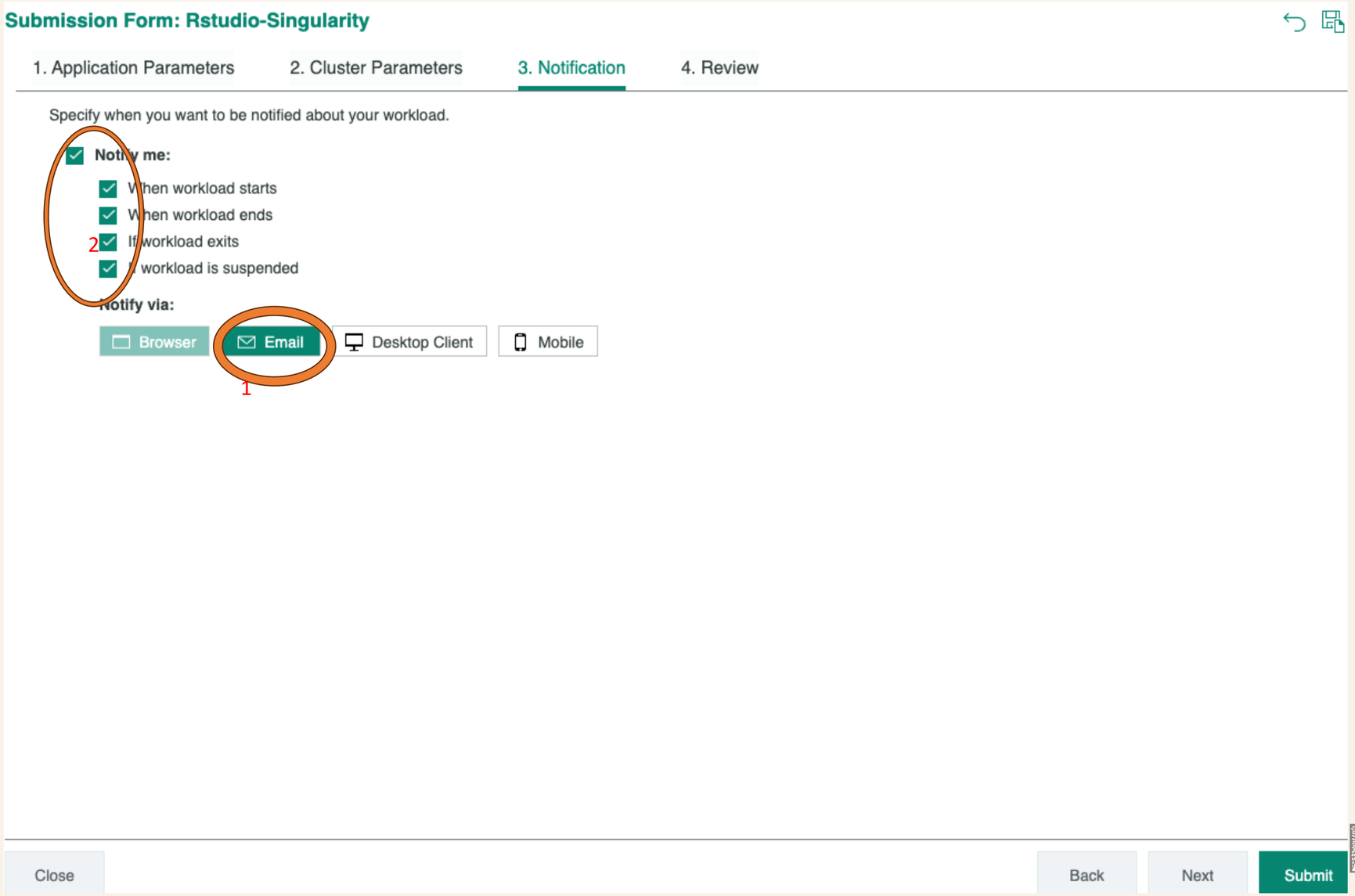
https://iblpacp01.wexac.weizmann.ac.il:8443/



מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

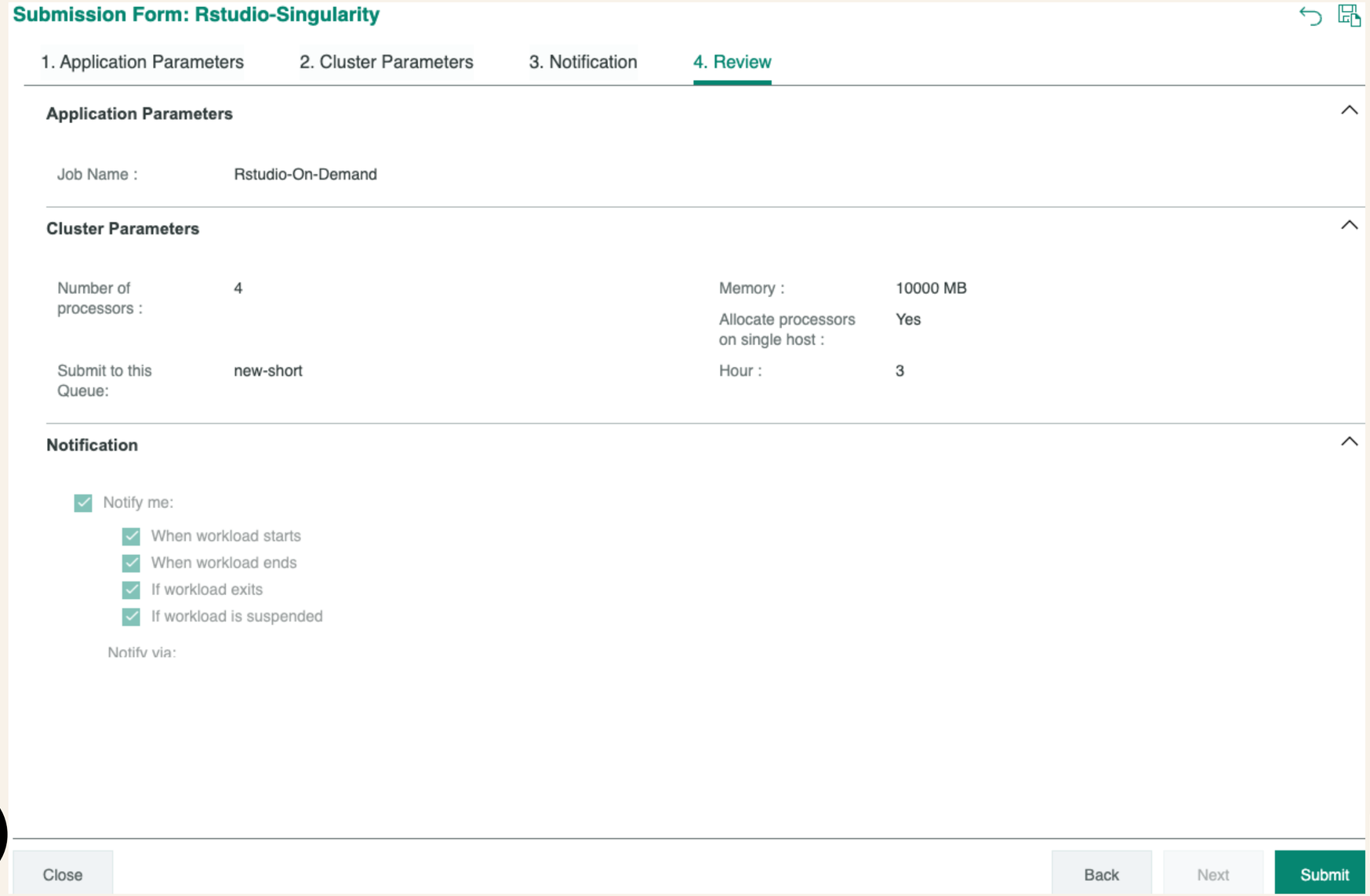
LSF PAC - Interactive Rstudio

https://iblpacp01.wexac.weizmann.ac.il:8443/



LSF PAC - Interactive Rstudio

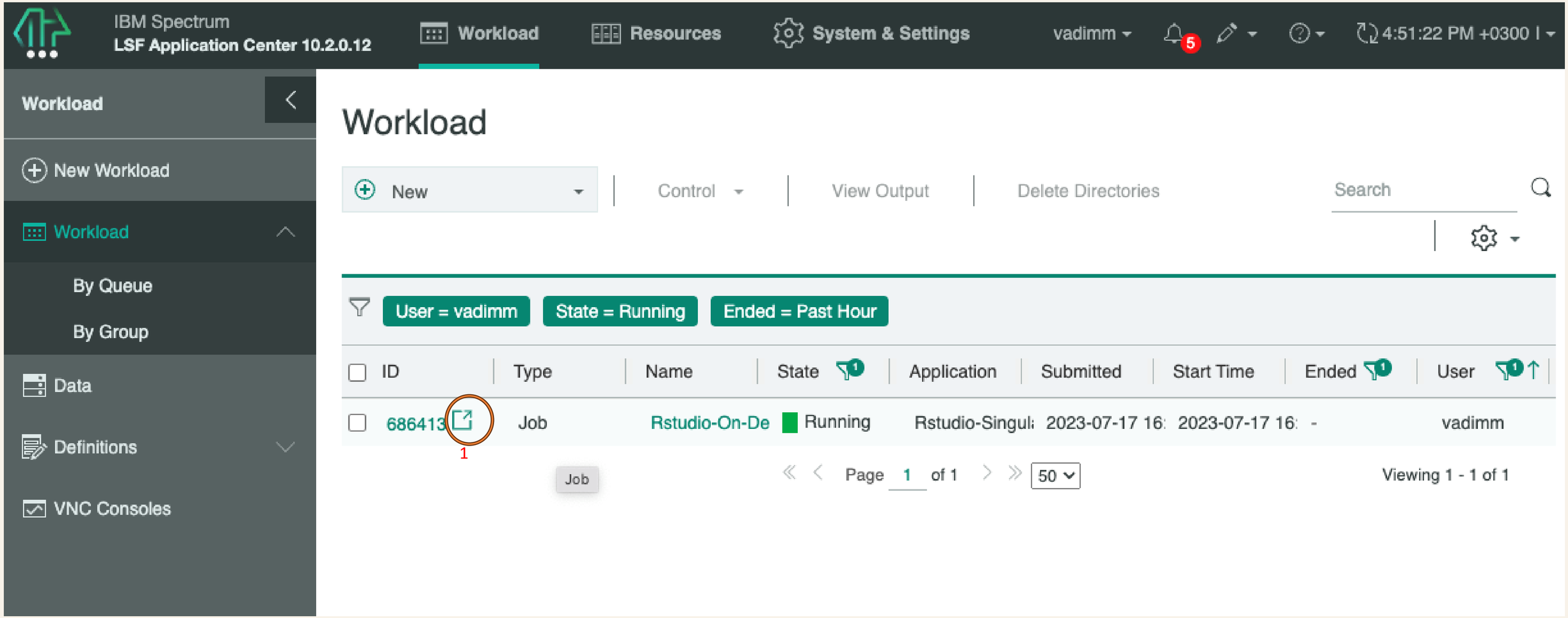
https://iblpacp01.wexac.weizmann.ac.il:8443/



מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

LSF PAC - Interactive Rstudio

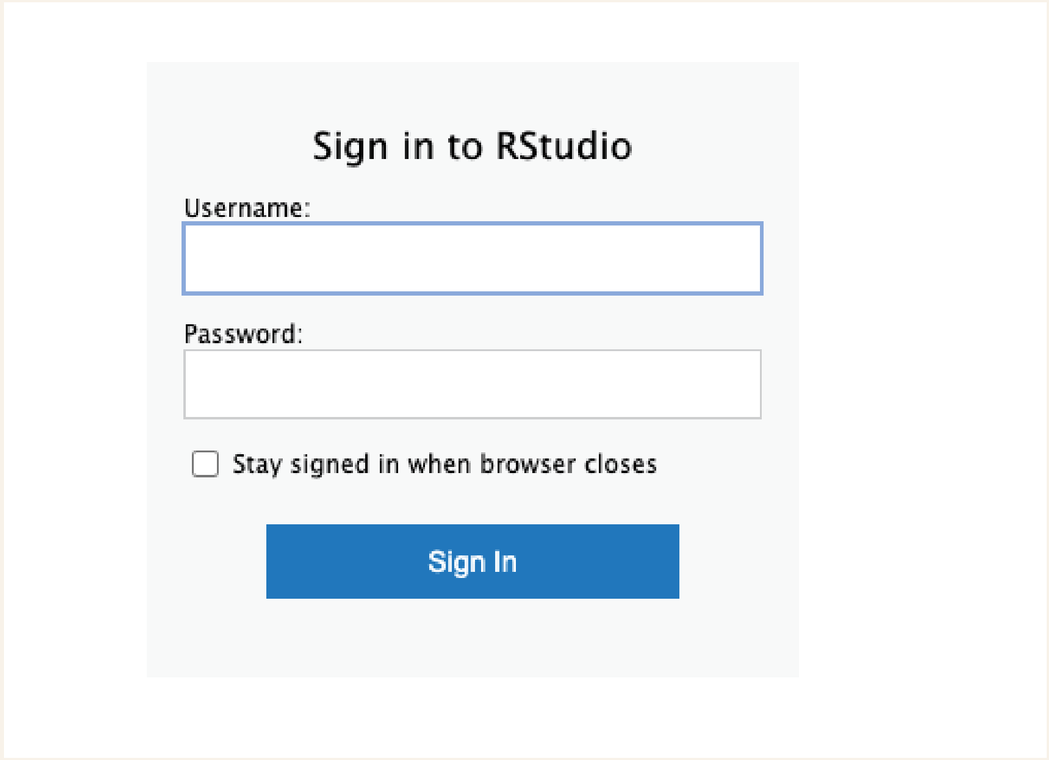
https://iblpacp01.wexac.weizmann.ac.il:8443/



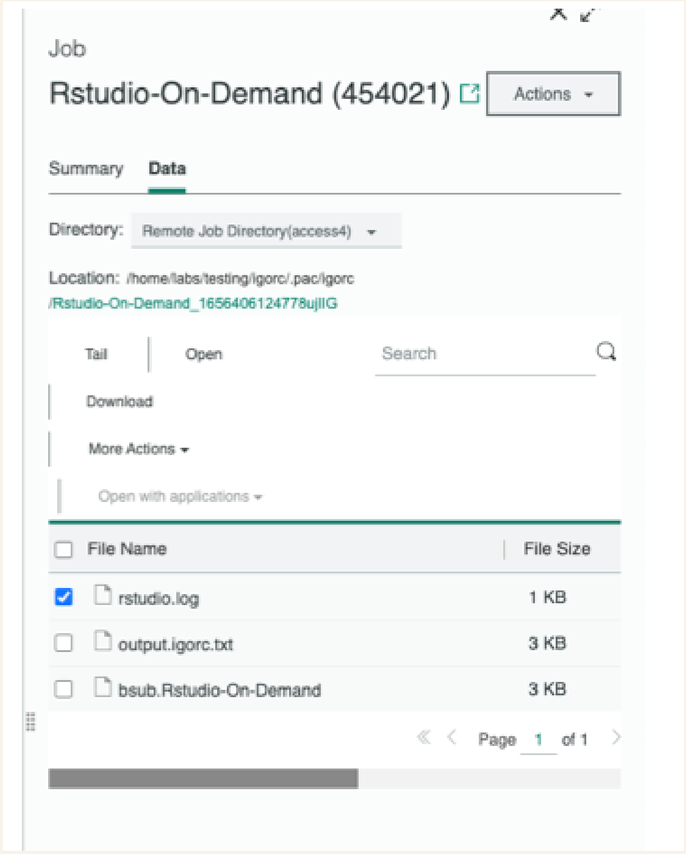
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Example of Rstudio application

This will open the RSTUDIO login page. Use your username and TEMPORARY PASSWORD received in the rstudio.log file.



To open rstudio.log file you can press the JOBID and then choose the rstudio.log file and press OPEN:



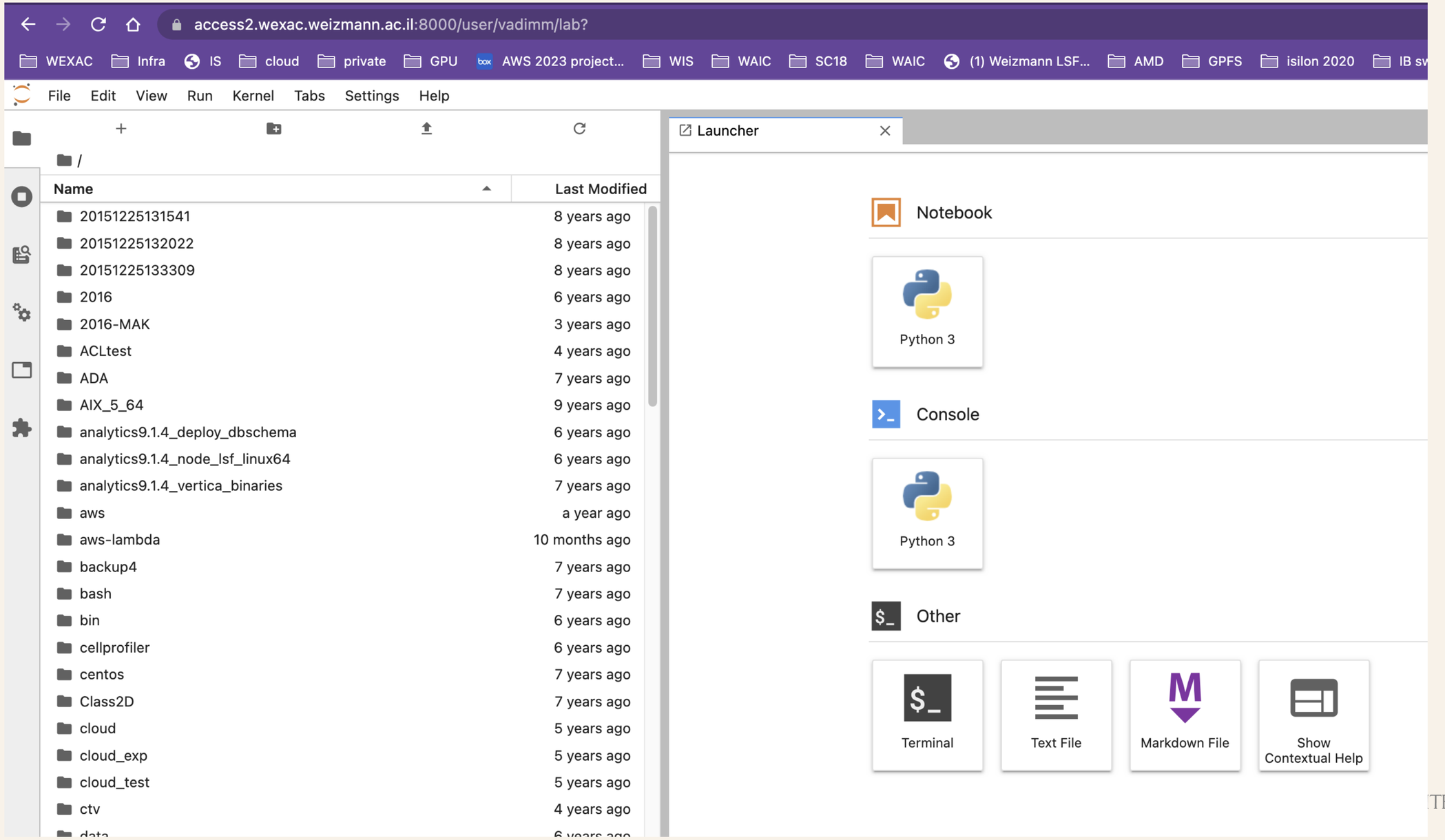
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Example of Rstudio application - Summary

- Visit the LSF PAC web interface: https://iblpacp01.wexac.weizmann.ac.il:8443/
- Login with your wexac username and password.
- On the RESOURCE TAB you may overview cluster statistics such as amount of available and closed nodes.
- In order to run an interactive job, navigate to WORKLOAD tab.
- On the NEW WORKLOAD page, you can execute available applications.
- On the WORKLOAD page you may overview details of your jobs.

WEXAC Jupyterhub

https://access2.wexac.weizmann.ac.il:8000/hub/login



WEXAC Private cloud
Current situation:

WEXAC members have private access nodes.

WEXAC members have private compute nodes.

WEXAC members uses only their own private hosts in private queues.

WEXAC guests and members use both public and private resources.

hpc@weizmann.ac.il

Problems with current situation:

A lot of guest's jobs preempted due to high priority members starting in the middle of guest's jobs.

Significant overhead in management and maintenance of >20 private access servers.

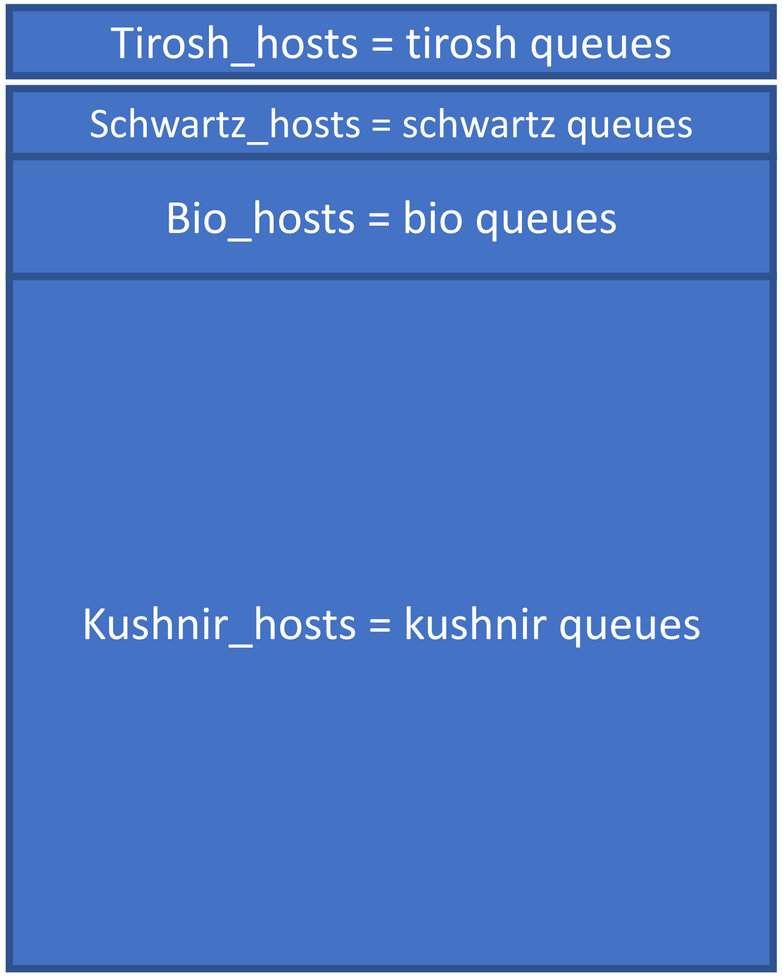
Silos of compute power: >20 different hardware configurations use different WEXAC members queues.

Couldn’t tackle the low cluster utilization due to straight connection between private queues and specific HW configuration.

No overprovisioning option

Current hosts groups Design

Multiple heterogeneous hosts groups



GSLA Queue Conceptual Design

Single homogeneous host group

Single GSLA hosts group
Will serve Schwartz, bio, Tirosh and kushnir queues

hpc@weizmann.ac.il

The goals:

Replace private compute servers with GSLA hardware pool.

Reduce amount of cluster queues.

Reduce preemption percentage of WEXAC guests and members.

Introduce overprovisioning option.

hpc@weizmann.ac.il

How we plan to achieve these goals (Phase I):

In phase 1 we'll introduce shared GSLA hardware pool.

2 new GSLA queues deployment replacing existing member queues.

Low priority GSLA queue will be deployed for member risk and all public users. Deploy policy of low slot guarantee per user. 2 no-preemption duration GSLA risk queues: gsla-risk-short and gsla-risk-long, 4 hours and 7 days respectfully, based on fair share policy.

High priority GSLA queue for members only will be deployed. 30% oversubscription of selected GSLA resources.

Each group will have his own maximum slot allocation in shared GSLA pool

hpc@weizmann.ac.il

Phase I benefits:

Reduce public workloads preemption percentage.

Increase hardware utilization by unification of cluster hardware.

Old members servers' hardware refresh.

hpc@weizmann.ac.il

Example of HPC available resources

	Resource type	Billing type	Minimum allocation	Price (NIS) Given 5 years lifecycle
1	Interactive CPU physical server 52 cores, 256 GB RAM	Reserved	1 day	17.71
2	CPU cores – single CPU core 2 GHz frequency *, 4GB RAM	Reserved	1 day	0.34
3	GPU cards – single A40, 12 cores, 80 GB RAM	Reserved	1 day	16.05
4	Interactive GPU server	Reserved	1 day	160.47

* We commit to 2 GHz frequency, but CPU might belong to different generations

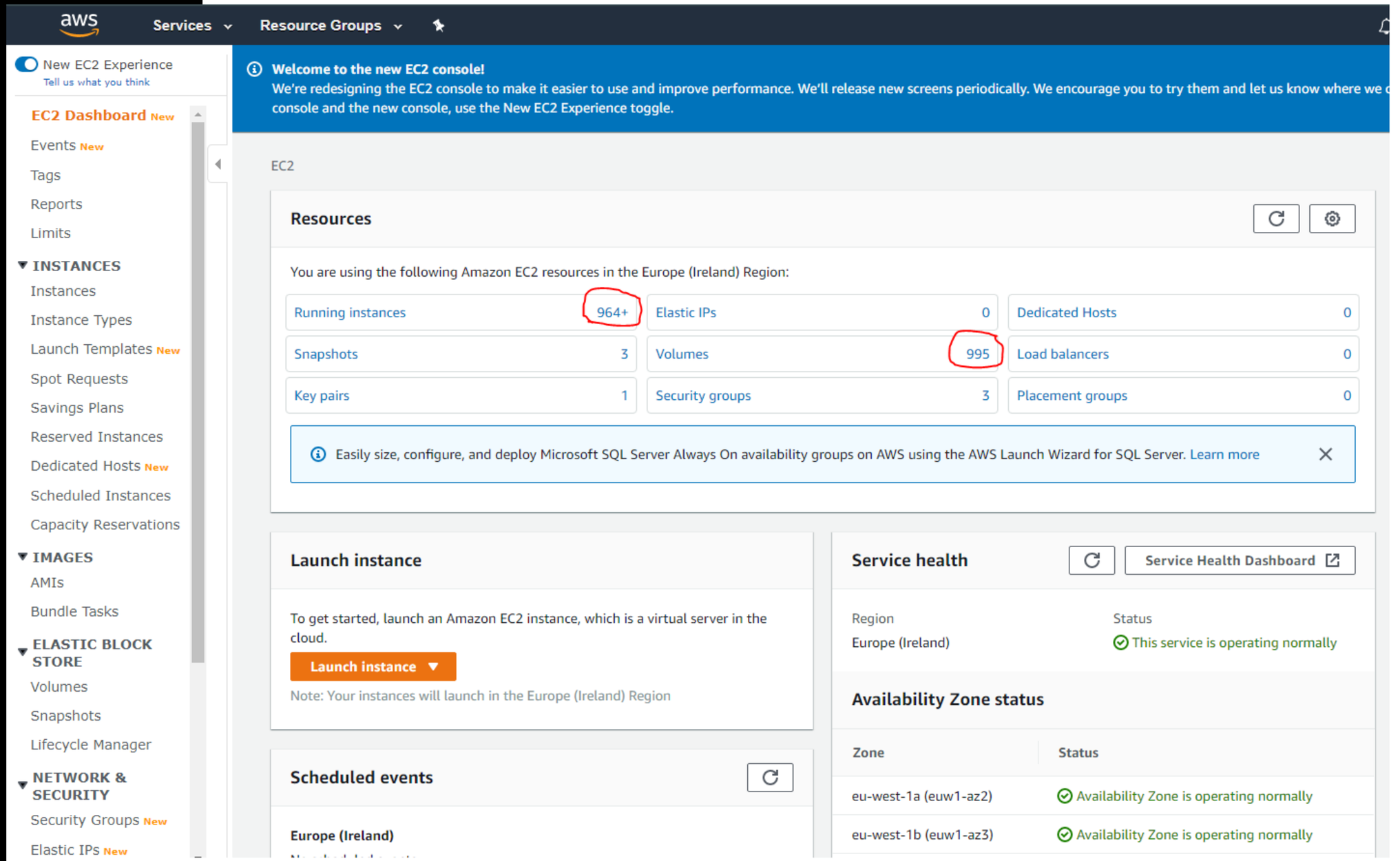
Storage available resources and costs

	Resource type	Billing type	Minimum allocation	Monthly price Given 5 years lifecycle
1	GPFS storage	Actual usage	1 month*	60 NIS per 1TB**
2	StorWIS	Actual usage	1 month	10 NIS per 1TB**
3	Archive	Actual usage	1 month	15 NIS per 1TB

* 1 month equal system price divide by 60 months

** Backup charged separately

WEXAC and AWS cloud integration



The AWS HPC Platform

The HPC (High Performance Computing) section’s objective is to speed up your scientific work and make it more efficient. This includes making AWS (Amazon Web Services) resources easy for you to utilize for HPC workloads.

To that end, we have established a AWS service platform facilitating easy onboarding of scientists to the WIS network-integrated AWS resources.

Service benefits include:

- Easy registration through Internal Services
- Billing via internal WIS systems
- The ability to specify a monthly budget, to control AWS resources assigned to your account
- DevOps services deployment of your scientific solutions on the cloud

The AWS HPC Platform

AWS provides the most elastic and scalable cloud infrastructure for execution of your HPC applications. With virtually unlimited capacity, researchers and HPC system owners can innovate without being burdened by the limitations of on-premises HPC infrastructure.

- The platform delivers an integrated suite of services providing everything required to quickly and easily build and manage HPC clusters on the cloud.
- These workloads span traditional HPC applications, such as genomics, computational chemistry and weather prediction, as well as such emerging applications as machine learning/deep learning.
- Flexible configuration and practically unlimited scalability allow you to grow or scale back your infrastructure as per workload requirements, rather than the other way around.
- Additionally, WEXAC integration allows you to easily migrate data back and forth between WEXAC and AWS infrastructures.

WEXAC Web Resources

https://hpcwiki.weizmann.ac.il/

https://appsrv.wexac.weizmann.ac.il/

https://www.weizmann.ac.il/hpc

https://www.weizmann.ac.il/DIS/sites/DIS/files/uploads/it/wexac_training_session.pdf

https://insightiq.weizmann.ac.il

https://wiki.weizmann.ac.il/ai_wiki/index.php/WAIC_cluster

https://iblmonitor01.wexac.weizmann.ac.il:3001/d/kjsnPAF4k/gpu-usage-by-groups?orgId=2

Join the WEXAC WhatsApp group!

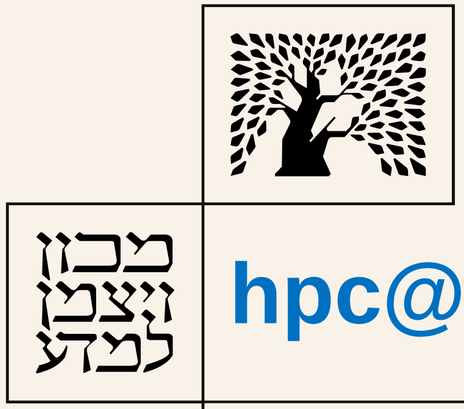
https://chat.whatsapp.com/05qVVCCcR8v9234vit28gc

hpc@weizmann.ac.il

Questions ?

hpc@weizmann.ac.il

WEIZMANN INSTITUTE OF SCIENCE



hpc@weizmann.ac.il

WEIZMANN
INSTITUTE
OF SCIENCE

July 2023 WEXAC workshop

Thank you