# LSF-CE/SLURM-CodeEngine-Gateway

Christof Westhues
IBM
cwesthues@de.ibm.com

2024/10/24

# What is it and why?

A gateway to run LSF-CE or SLURM workload (‚jobs‘) on (dynamic) IBM CodeEngine instances

Run workload on ‚**serverless computenodes**‘

No complex handling of (VPC) compute nodes, network, storage

For simple, basic workload

# Value proposition

- ✓ **dynamically scale compute resources** on-demand **for peak workloads**
- ✓ improve **cost efficiency** through Pay-Per-Use Model
- ✓ **reduce operational overhead** ("No Infrastructure Management")
- ✓ **secure** containerized job environments
- ✓ **easy Job Submission** and Management
- ✓ **enhanced Job Distribution** and **Fault Tolerance**
- ✓ stateful **handling of data** through COS mounts to container
- ✓ **easy to implement** (and demo)

# Potential applications

**EDA:**

Ansys HFSS

Cadence Xcelium

Keysight ADS

Mentor Graphics Calibre

Silvaco TCAD

Synopsys Sentaurus

Synopsys VCS

**Manufacturing:**

ABAQUS *

Altair HyperWorks *

ANSYS Additive

ANSYS Fluent *

ANSYS Mechanical

ANSYS Multiphysics

AutoForm

COMSOL Multiphysics *

OpenFOAM *

Star-CCM+ *

**Oil & Gas:**

ECLIPSE (by Schlumberger) *

Schlumberger Petrel

Schlumberger VISAGE *

**Lifesciences:**

AlphaFold *

AMBER *

AutoDock

BLAST

DOCK

GATK

GROMACS *

NAMD *

ROSETTA

**Others:**

Matlab

R

Tensorflow

Octave

PyTorch

* typicallly uses MPI/Infiniband

# Install your SW/apps (example) in docker container:

```
cat > Dockerfile <<EOF
FROM docker.io/library/ubuntu:latest
RUN apt-get -yq update ; \
    apt -y install curl git python3-pip unzip wget
COPY my_app /usr/bin/my_app
EOF
docker build -t docker.io/${DOCKERUSER}/ubuntu .
docker push docker.io/${DOCKERUSER}/ubuntu
```

# Access to data through COS

What

Where

```
s3fs ${COSBUCKET} ${MOUNTPOINT} \
-o url=https://s3.eu-de.cloud-object-storage.appdomain.cloud
mkdir -p ${MOUNTPOINT}/${JOBID}
cp INPUT.txt ${MOUNTPOINT}/${JOBID}
umount ${MOUNTPOINT}
```

1. Mount a COS bucket and copy input data

submitting/ master node

```
cat /mnt/${JOBID}/INPUT.txt | ... > /mnt/${JOBID}/OUTPUT.txt
```

2. Inside the container, „the job“

inside container

```
s3fs ${COSBUCKET} ${MOUNTPOINT} \
-o url=https://s3.eu-de.cloud-object-storage.appdomain.cloud
cp ${MOUNTPOINT}/${JOBID}/OUTPUT.txt ...
umount ${MOUNTPOINT}
```

3. Mount a COS bucket and copy output data

submitting/ master node

# Jumping straight into tech stuff…

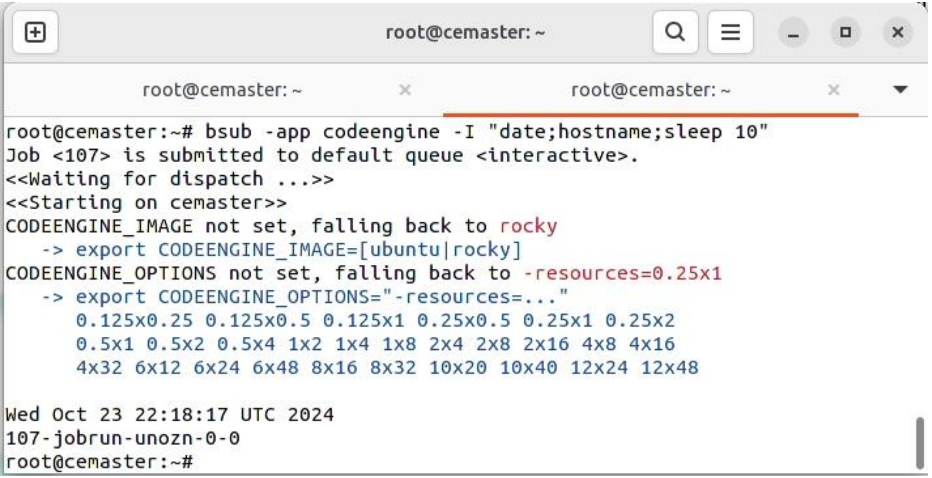
```
Begin Application
NAME                  = codeengine
RES_REQ               = select[codeengine
CONTAINER             = docker[image(${LSB_CODEENGINE_IMAGE}) \
                        options(${LSB_CODEENGINE_OPTIONS})]
DESCRIPTION           = Codeengine User Service
EXEC_DRIVER           = context[user(default)] \
                        starter[codeengine-starter.sh] \
                        controller[codeengine-control.sh]
PRE_EXEC              = codeengine-preexec.sh
POST_EXEC             = codeengine-postexec.sh
JOB_INCLUDE_POSTPROC = Y
End Application
```

# Some more details on codeengine-starter.sh (simplified)

```
#!/bin/sh

ibmcloud login ...

ibmcloud ce project select -n ...

ibmcloud ce job create --name ... --image ... --argument ${CMD}

add-volume-mount-to-job.sh ${LSB_JOBID} s3fstest2 /mnt

ibmcloud ce jobrun submit --job ${LSB_JOBID}

ibmcloud ce jobrun logs -f -r --jobrun ${JOBRUN_NAME}

ibmcloud ce jobrun delete --name ${JOBRUN_NAME} -f

ibmcloud ce job delete --name ${LSB_JOBID} -f
```

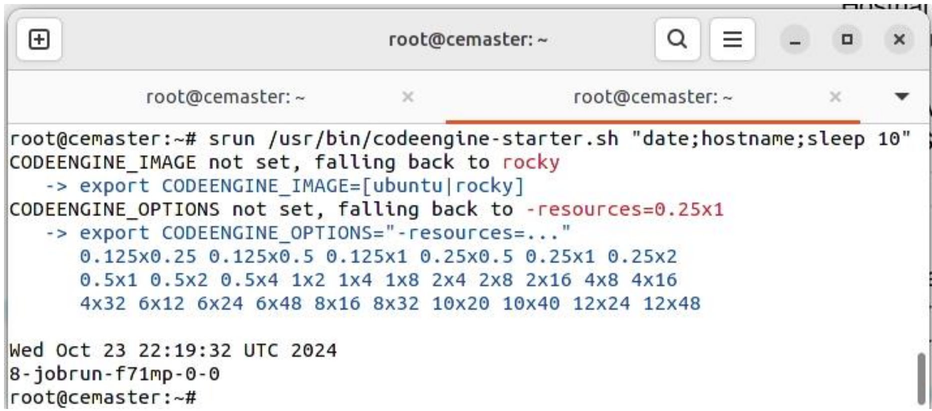
# Running LSF-CE and SLURM interactive jobs on CodeEngine

## LSF-CE



```
root@cemaster:~# bsub -app codeengine -I "date;hostname;sleep 10"
Job <107> is submitted to default queue <interactive>.
<<Waiting for dispatch ...>>
<<Starting on cemaster>>
CODEENGINE_IMAGE not set, falling back to rocky
   -> export CODEENGINE_IMAGE=[ubuntu|rocky]
CODEENGINE_OPTIONS not set, falling back to -resources=0.25x1
   -> export CODEENGINE_OPTIONS="-resources=..."
      0.125x0.25 0.125x0.5 0.125x1 0.25x0.5 0.25x1 0.25x2
      0.5x1 0.5x2 0.5x4 1x2 1x4 1x8 2x4 2x8 2x16 4x8 4x16
      4x32 6x12 6x24 6x48 8x16 8x32 10x20 10x40 12x24 12x48

Wed Oct 23 22:18:17 UTC 2024
107-jobrun-unozn-0-0
root@cemaster:~#
```

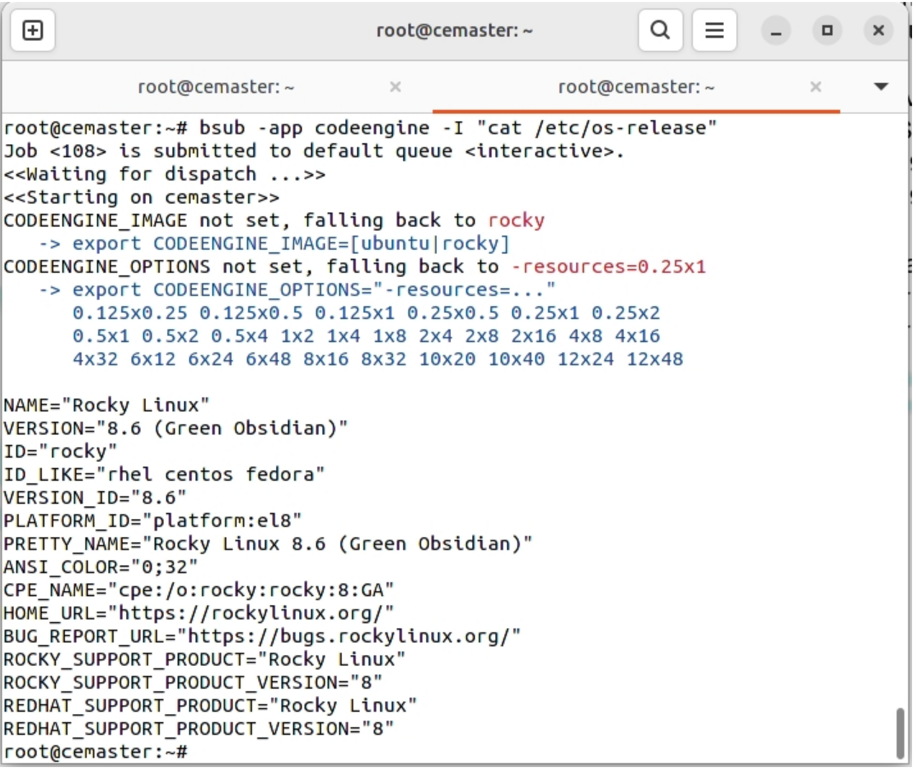
## SLURM



```
root@cemaster:~# srun /usr/bin/codeengine-starter.sh "date;hostname;sleep 10"
CODEENGINE_IMAGE not set, falling back to rocky
   -> export CODEENGINE_IMAGE=[ubuntu|rocky]
CODEENGINE_OPTIONS not set, falling back to -resources=0.25x1
   -> export CODEENGINE_OPTIONS="-resources=..."
      0.125x0.25 0.125x0.5 0.125x1 0.25x0.5 0.25x1 0.25x2
      0.5x1 0.5x2 0.5x4 1x2 1x4 1x8 2x4 2x8 2x16 4x8 4x16
      4x32 6x12 6x24 6x48 8x16 8x32 10x20 10x40 12x24 12x48

Wed Oct 23 22:19:32 UTC 2024
8-jobrun-f71mp-0-0
root@cemaster:~#
```

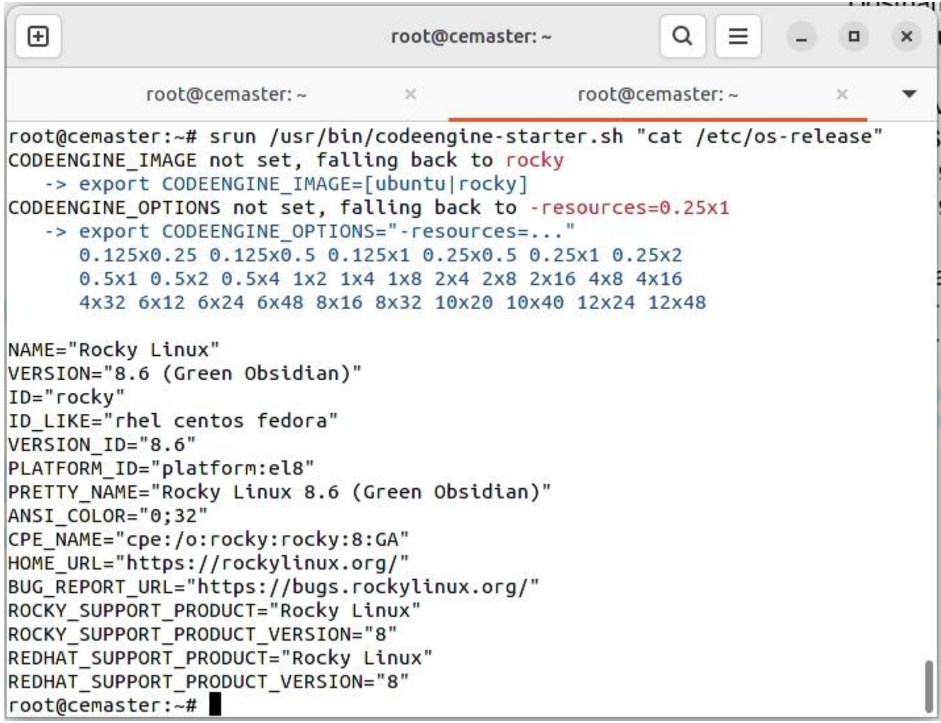
# Running LSF-CE and SLURM interactive jobs on CodeEngine

## LSF-CE



```
root@cemaster:~# bsub -app codeengine -I "cat /etc/os-release"
Job <108> is submitted to default queue <interactive>.
<<Waiting for dispatch ...>>
<<Starting on cemaster>>
CODEENGINE_IMAGE not set, falling back to rocky
   -> export CODEENGINE_IMAGE=[ubuntu|rocky]
CODEENGINE_OPTIONS not set, falling back to -resources=0.25x1
   -> export CODEENGINE_OPTIONS="-resources=..."
      0.125x0.25 0.125x0.5 0.125x1 0.25x0.5 0.25x1 0.25x2
      0.5x1 0.5x2 0.5x4 1x2 1x4 1x8 2x4 2x8 2x16 4x8 4x16
      4x32 6x12 6x24 6x48 8x16 8x32 10x20 10x40 12x24 12x48

NAME="Rocky Linux"
VERSION="8.6 (Green Obsidian)"
ID="rocky"
ID_LIKE="rhel centos fedora"
VERSION_ID="8.6"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Rocky Linux 8.6 (Green Obsidian)"
ANSI_COLOR="0;32"
CPE_NAME="cpe:/o:rocky:rocky:8:GA"
HOME_URL="https://rockylinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
ROCKY_SUPPORT_PRODUCT="Rocky Linux"
ROCKY_SUPPORT_PRODUCT_VERSION="8"
REDHAT_SUPPORT_PRODUCT="Rocky Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8"
root@cemaster:~#
```

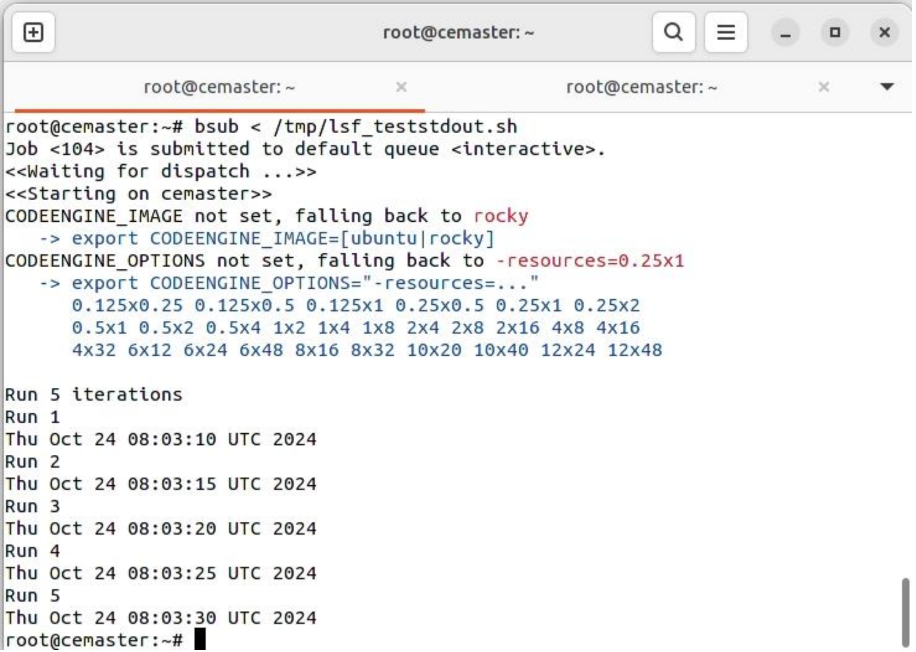
## SLURM



```
root@cemaster:~# srun /usr/bin/codeengine-starter.sh "cat /etc/os-release"
CODEENGINE_IMAGE not set, falling back to rocky
   -> export CODEENGINE_IMAGE=[ubuntu|rocky]
CODEENGINE_OPTIONS not set, falling back to -resources=0.25x1
   -> export CODEENGINE_OPTIONS="-resources=..."
      0.125x0.25 0.125x0.5 0.125x1 0.25x0.5 0.25x1 0.25x2
      0.5x1 0.5x2 0.5x4 1x2 1x4 1x8 2x4 2x8 2x16 4x8 4x16
      4x32 6x12 6x24 6x48 8x16 8x32 10x20 10x40 12x24 12x48

NAME="Rocky Linux"
VERSION="8.6 (Green Obsidian)"
ID="rocky"
ID_LIKE="rhel centos fedora"
VERSION_ID="8.6"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Rocky Linux 8.6 (Green Obsidian)"
ANSI_COLOR="0;32"
CPE_NAME="cpe:/o:rocky:rocky:8:GA"
HOME_URL="https://rockylinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
ROCKY_SUPPORT_PRODUCT="Rocky Linux"
ROCKY_SUPPORT_PRODUCT_VERSION="8"
REDHAT_SUPPORT_PRODUCT="Rocky Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8"
root@cemaster:~# █
```

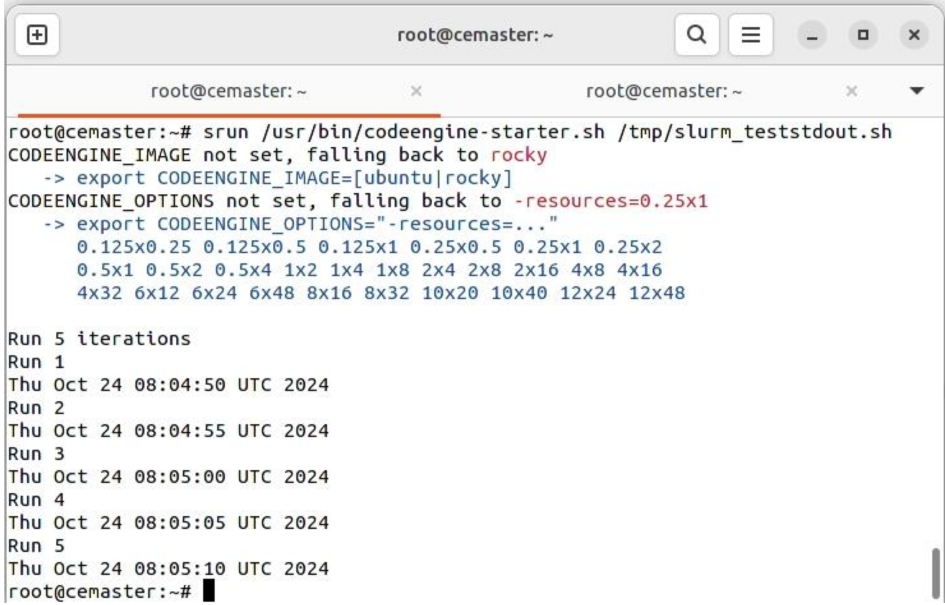
# Running LSF-CE and SLURM interactive jobs on CodeEngine

## LSF-CE



```
root@cemaster:~# bsub < /tmp/lsf_teststdout.sh
Job <104> is submitted to default queue <interactive>.
<<Waiting for dispatch ...>>
<<Starting on cemaster>>
CODEENGINE_IMAGE not set, falling back to rocky
   -> export CODEENGINE_IMAGE=[ubuntu|rocky]
CODEENGINE_OPTIONS not set, falling back to -resources=0.25x1
   -> export CODEENGINE_OPTIONS="-resources=..."
      0.125x0.25 0.125x0.5 0.125x1 0.25x0.5 0.25x1 0.25x2
      0.5x1 0.5x2 0.5x4 1x2 1x4 1x8 2x4 2x8 2x16 4x8 4x16
      4x32 6x12 6x24 6x48 8x16 8x32 10x20 10x40 12x24 12x48

Run 5 iterations
Run 1
Thu Oct 24 08:03:10 UTC 2024
Run 2
Thu Oct 24 08:03:15 UTC 2024
Run 3
Thu Oct 24 08:03:20 UTC 2024
Run 4
Thu Oct 24 08:03:25 UTC 2024
Run 5
Thu Oct 24 08:03:30 UTC 2024
root@cemaster:~# 
```

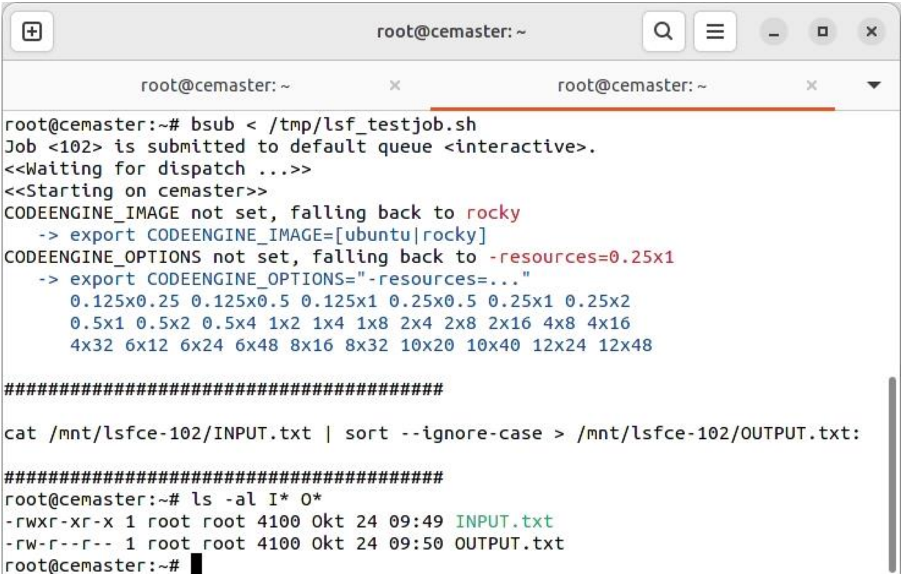
## SLURM



```
root@cemaster:~# srun /usr/bin/codeengine-starter.sh /tmp/slurm_teststdout.sh
CODEENGINE_IMAGE not set, falling back to rocky
   -> export CODEENGINE_IMAGE=[ubuntu|rocky]
CODEENGINE_OPTIONS not set, falling back to -resources=0.25x1
   -> export CODEENGINE_OPTIONS="-resources=..."
      0.125x0.25 0.125x0.5 0.125x1 0.25x0.5 0.25x1 0.25x2
      0.5x1 0.5x2 0.5x4 1x2 1x4 1x8 2x4 2x8 2x16 4x8 4x16
      4x32 6x12 6x24 6x48 8x16 8x32 10x20 10x40 12x24 12x48

Run 5 iterations
Run 1
Thu Oct 24 08:04:50 UTC 2024
Run 2
Thu Oct 24 08:04:55 UTC 2024
Run 3
Thu Oct 24 08:05:00 UTC 2024
Run 4
Thu Oct 24 08:05:05 UTC 2024
Run 5
Thu Oct 24 08:05:10 UTC 2024
root@cemaster:~# 
```

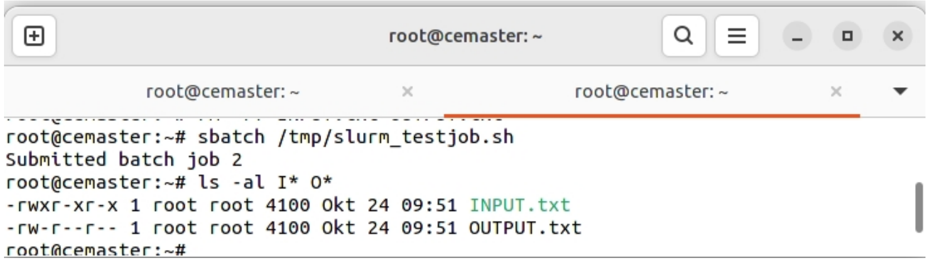
# Running LSF-CE and SLURM batch jobs on CodeEngine

## LSF-CE



```
root@cemaster:~# bsub < /tmp/lsf_testjob.sh
Job <102> is submitted to default queue <interactive>.
<<Waiting for dispatch ...>>
<<Starting on cemaster>>
CODEENGINE_IMAGE not set, falling back to rocky
   -> export CODEENGINE_IMAGE=[ubuntu|rocky]
CODEENGINE_OPTIONS not set, falling back to -resources=0.25x1
   -> export CODEENGINE_OPTIONS="-resources=..."
      0.125x0.25 0.125x0.5 0.125x1 0.25x0.5 0.25x1 0.25x2
      0.5x1 0.5x2 0.5x4 1x2 1x4 1x8 2x4 2x8 2x16 4x8 4x16
      4x32 6x12 6x24 6x48 8x16 8x32 10x20 10x40 12x24 12x48

########################################

cat /mnt/lsfce-102/INPUT.txt | sort --ignore-case > /mnt/lsfce-102/OUTPUT.txt:

########################################
root@cemaster:~# ls -al I* O*
-rwxr-xr-x 1 root root 4100 Okt 24 09:49 INPUT.txt
-rw-r--r-- 1 root root 4100 Okt 24 09:50 OUTPUT.txt
root@cemaster:~# 
```

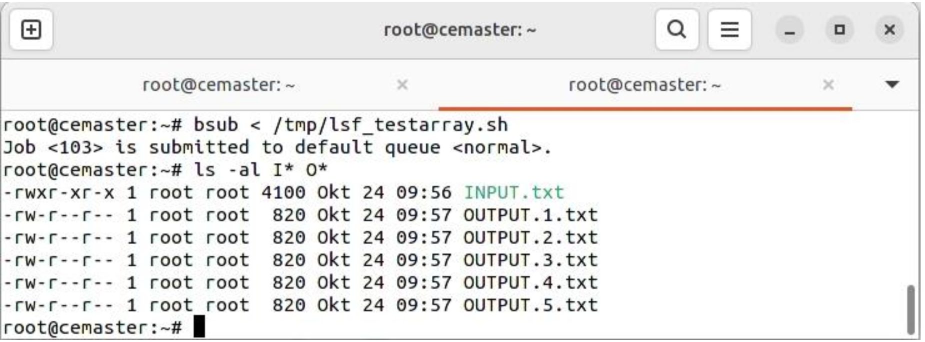
## SLURM



```
root@cemaster:~# sbatch /tmp/slurm_testjob.sh
Submitted batch job 2
root@cemaster:~# ls -al I* O*
-rwxr-xr-x 1 root root 4100 Okt 24 09:51 INPUT.txt
-rw-r--r-- 1 root root 4100 Okt 24 09:51 OUTPUT.txt
root@cemaster:~#
```

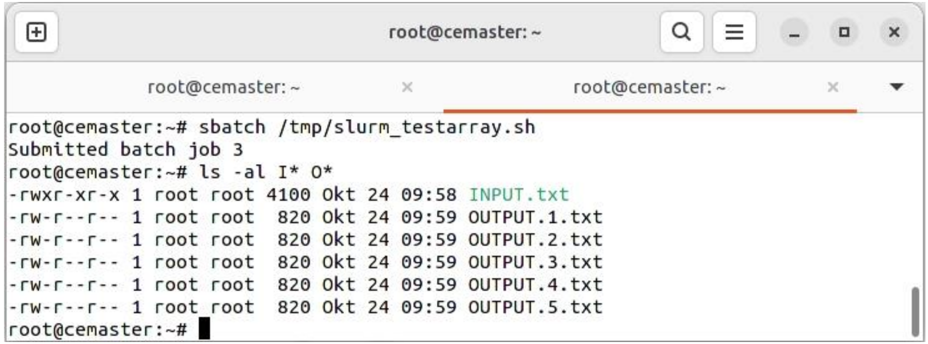
# Running LSF-CE and SLURM array jobs on CodeEngine

## LSF-CE



```
root@cemaster:~# bsub < /tmp/lsf_testarray.sh
Job <103> is submitted to default queue <normal>.
root@cemaster:~# ls -al I* O*
-rwxr-xr-x 1 root root 4100 Okt 24 09:56 INPUT.txt
-rw-r--r-- 1 root root  820 Okt 24 09:57 OUTPUT.1.txt
-rw-r--r-- 1 root root  820 Okt 24 09:57 OUTPUT.2.txt
-rw-r--r-- 1 root root  820 Okt 24 09:57 OUTPUT.3.txt
-rw-r--r-- 1 root root  820 Okt 24 09:57 OUTPUT.4.txt
-rw-r--r-- 1 root root  820 Okt 24 09:57 OUTPUT.5.txt
root@cemaster:~# 
```

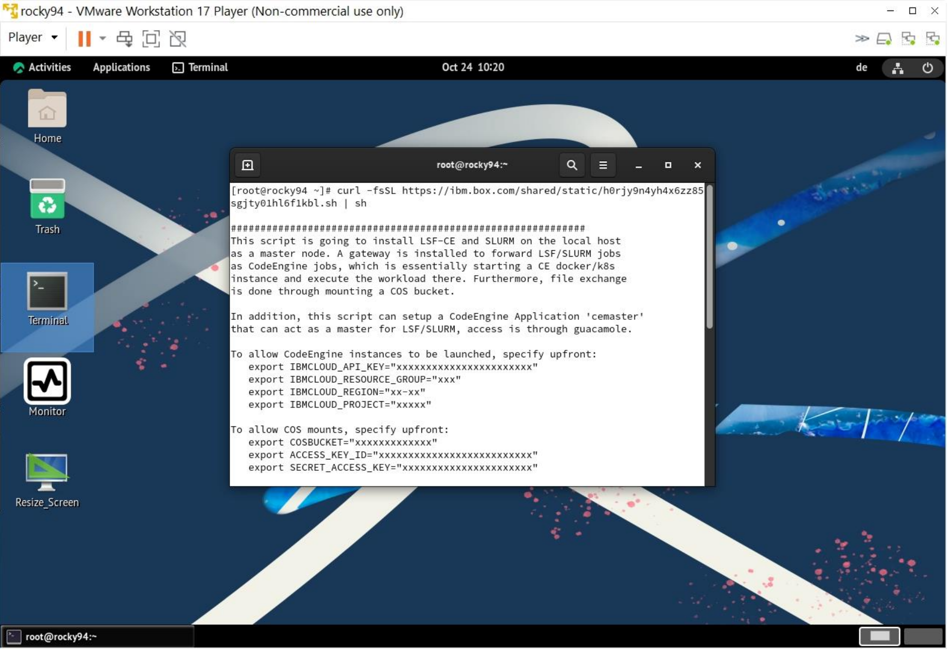
## SLURM



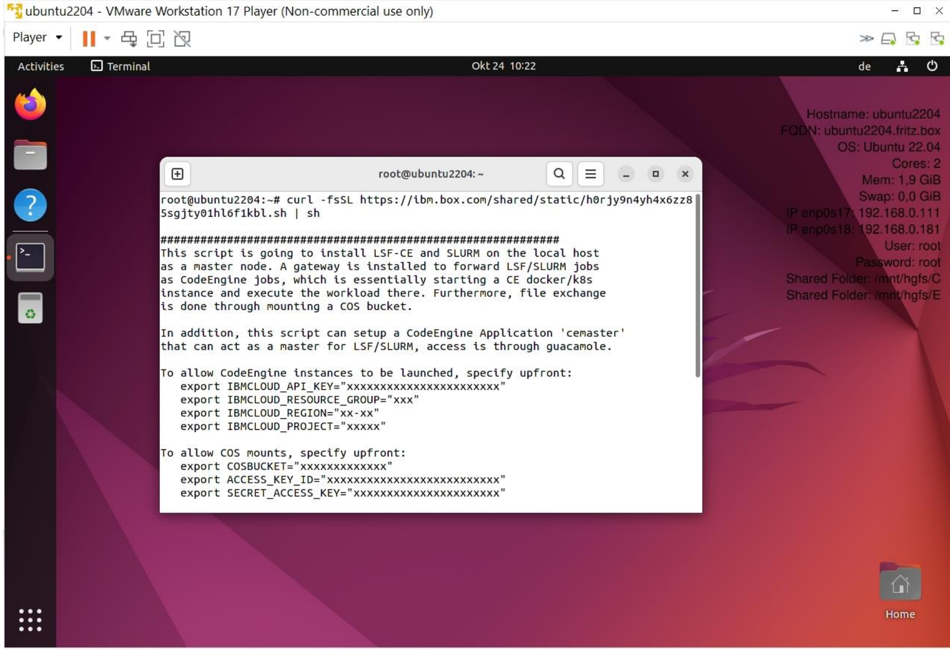
```
root@cemaster:~# sbatch /tmp/slurm_testarray.sh
Submitted batch job 3
root@cemaster:~# ls -al I* O*
-rwxr-xr-x 1 root root 4100 Okt 24 09:58 INPUT.txt
-rw-r--r-- 1 root root  820 Okt 24 09:59 OUTPUT.1.txt
-rw-r--r-- 1 root root  820 Okt 24 09:59 OUTPUT.2.txt
-rw-r--r-- 1 root root  820 Okt 24 09:59 OUTPUT.3.txt
-rw-r--r-- 1 root root  820 Okt 24 09:59 OUTPUT.4.txt
-rw-r--r-- 1 root root  820 Okt 24 09:59 OUTPUT.5.txt
root@cemaster:~# 
```

# Install on Rocky and Ubuntu

curl -fsSL https://ibm.box.com/shared/static/h0rjy9n4yh4x6zz85sgjty01hl6f1kbl.sh | sh



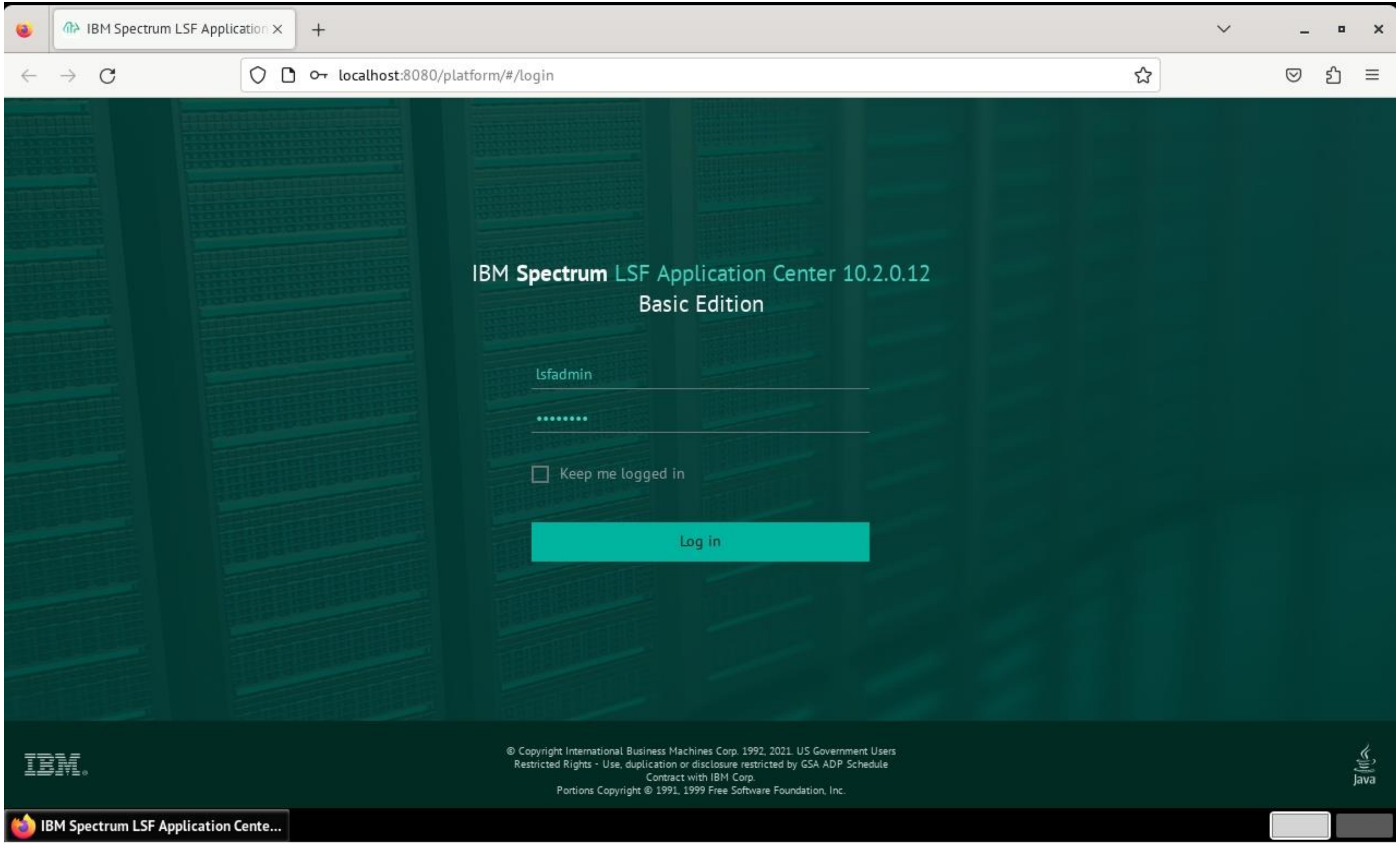
```
[root@rocky94 ~]# curl -fsSL https://ibm.box.com/shared/static/h0rjy9n4yh4x6zz85
sgjty01hl6f1kbl.sh | sh

############################################################
This script is going to install LSF-CE and SLURM on the local host
as a master node. A gateway is installed to forward LSF/SLURM jobs
as CodeEngine jobs, which is essentially starting a CE docker/k8s
instance and execute the workload there. Furthermore, file exchange
is done through mounting a COS bucket.

In addition, this script can setup a CodeEngine Application 'cemaster'
that can act as a master for LSF/SLURM, access is through guacamole.

To allow CodeEngine instances to be launched, specify upfront:
   export IBMCLOUD_API_KEY="xxxxxxxxxxxxxxxxxxxxxxxx"
   export IBMCLOUD_RESOURCE_GROUP="xxx"
   export IBMCLOUD_REGION="xx-xx"
   export IBMCLOUD_PROJECT="xxxxx"

To allow COS mounts, specify upfront:
   export COSBUCKET="xxxxxxxxxxxxx"
   export ACCESS_KEY_ID="xxxxxxxxxxxxxxxxxxxxxxxxxxx"
   export SECRET_ACCESS_KEY="xxxxxxxxxxxxxxxxxxxxxx"
```


ubuntu2204 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Terminal
Okt 24 10:22
de
root@ubuntu2204: ~
Home

Hostname: ubuntu2204
FQDN: ubuntu2204.fritz.box
OS: Ubuntu 22.04
Cores: 2
Mem: 1,9 GiB
Swap: 0,0 GiB
IP enp0s17: 192.168.0.111
IP enp0s18: 192.168.0.181
User: root
Password: root
Shared Folder: /mnt/hgfs/C
Shared Folder: /mnt/hgfs/E

```
root@ubuntu2204:~# curl -fsSL https://ibm.box.com/shared/static/h0rjy9n4yh4x6zz8
5sgjty01hl6f1kbl.sh | sh

############################################################
This script is going to install LSF-CE and SLURM on the local host
as a master node. A gateway is installed to forward LSF/SLURM jobs
as CodeEngine jobs, which is essentially starting a CE docker/k8s
instance and execute the workload there. Furthermore, file exchange
is done through mounting a COS bucket.

In addition, this script can setup a CodeEngine Application 'cemaster'
that can act as a master for LSF/SLURM, access is through guacamole.

To allow CodeEngine instances to be launched, specify upfront:
   export IBMCLOUD_API_KEY="xxxxxxxxxxxxxxxxxxxxxxxx"
   export IBMCLOUD_RESOURCE_GROUP="xxx"
   export IBMCLOUD_REGION="xx-xx"
   export IBMCLOUD_PROJECT="xxxxx"

To allow COS mounts, specify upfront:
   export COSBUCKET="xxxxxxxxxxxxx"
   export ACCESS_KEY_ID="xxxxxxxxxxxxxxxxxxxxxxxxxxx"
   export SECRET_ACCESS_KEY="xxxxxxxxxxxxxxxxxxxxxx"
```

# Application Center Basic Edition is part of LSF-CE



IBM Spectrum LSF Application
localhost:8080/platform/#/login

IBM **Spectrum** LSF Application Center 10.2.0.12
Basic Edition

lsfadmin

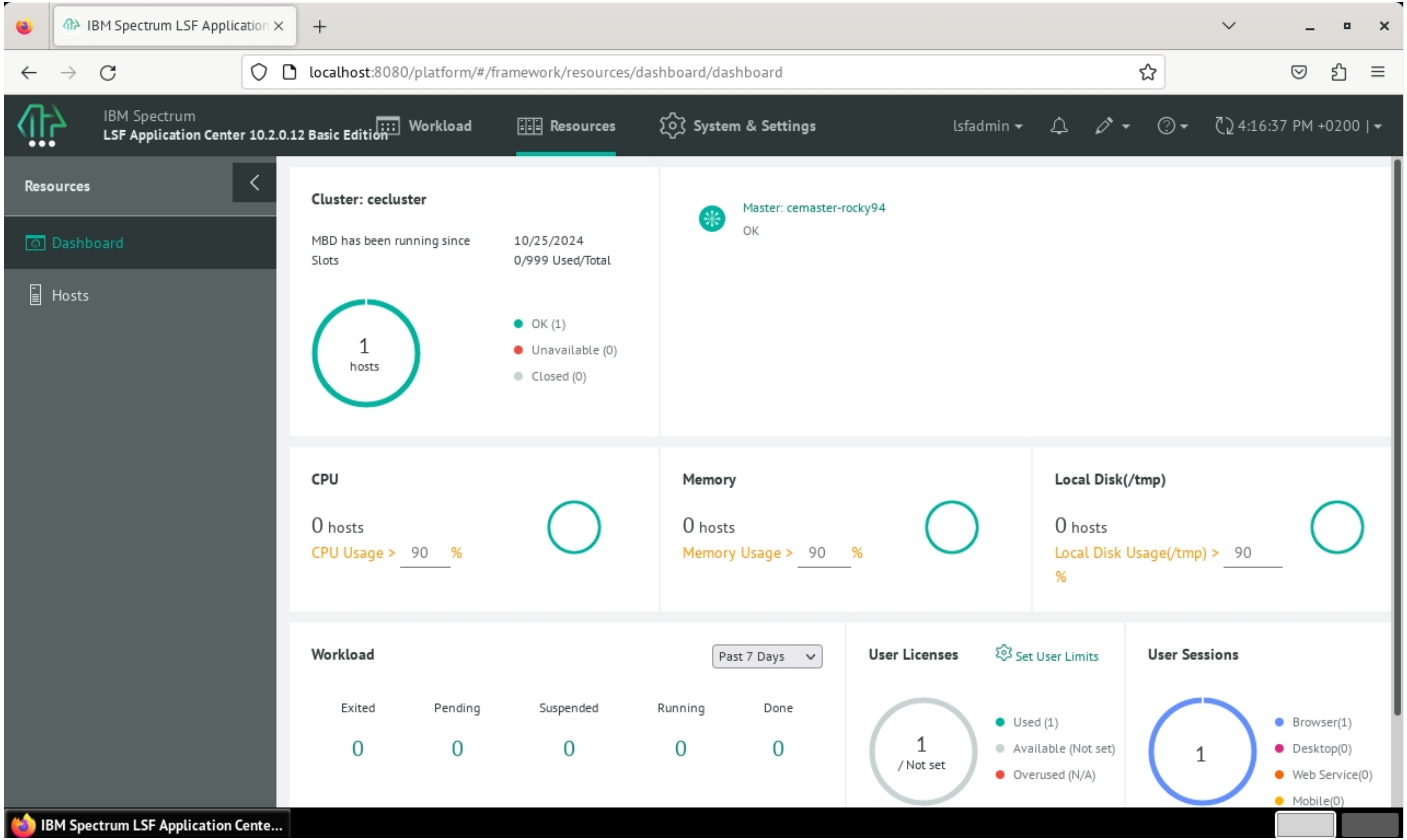
•••••••••

□ Keep me logged in

Log in

Java
IBM Spectrum LSF Application Cente...

# A one-node (cemaster) cluster…


IBM Spectrum LSF Application
localhost:8080/platform/#/framework/resources/dashboard/dashboard
IBM Spectrum
LSF Application Center 10.2.0.12 Basic Edition
Workload
Resources
System & Settings
lsfadmin
4:16:37 PM +0200
Resources
Dashboard
Hosts

Cluster: cecluster

MBD has been running since 10/25/2024
Slots 0/999 Used/Total

● OK (1)
● Unavailable (0)
● Closed (0)

Master: cemaster-rocky94
OK

CPU
0 hosts
CPU Usage > 90 %

Memory
0 hosts
Memory Usage > 90 %

Local Disk(/tmp)
0 hosts
Local Disk Usage(/tmp) > 90 %

Workload    Past 7 Days

| Exited | Pending | Suspended | Running | Done |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |

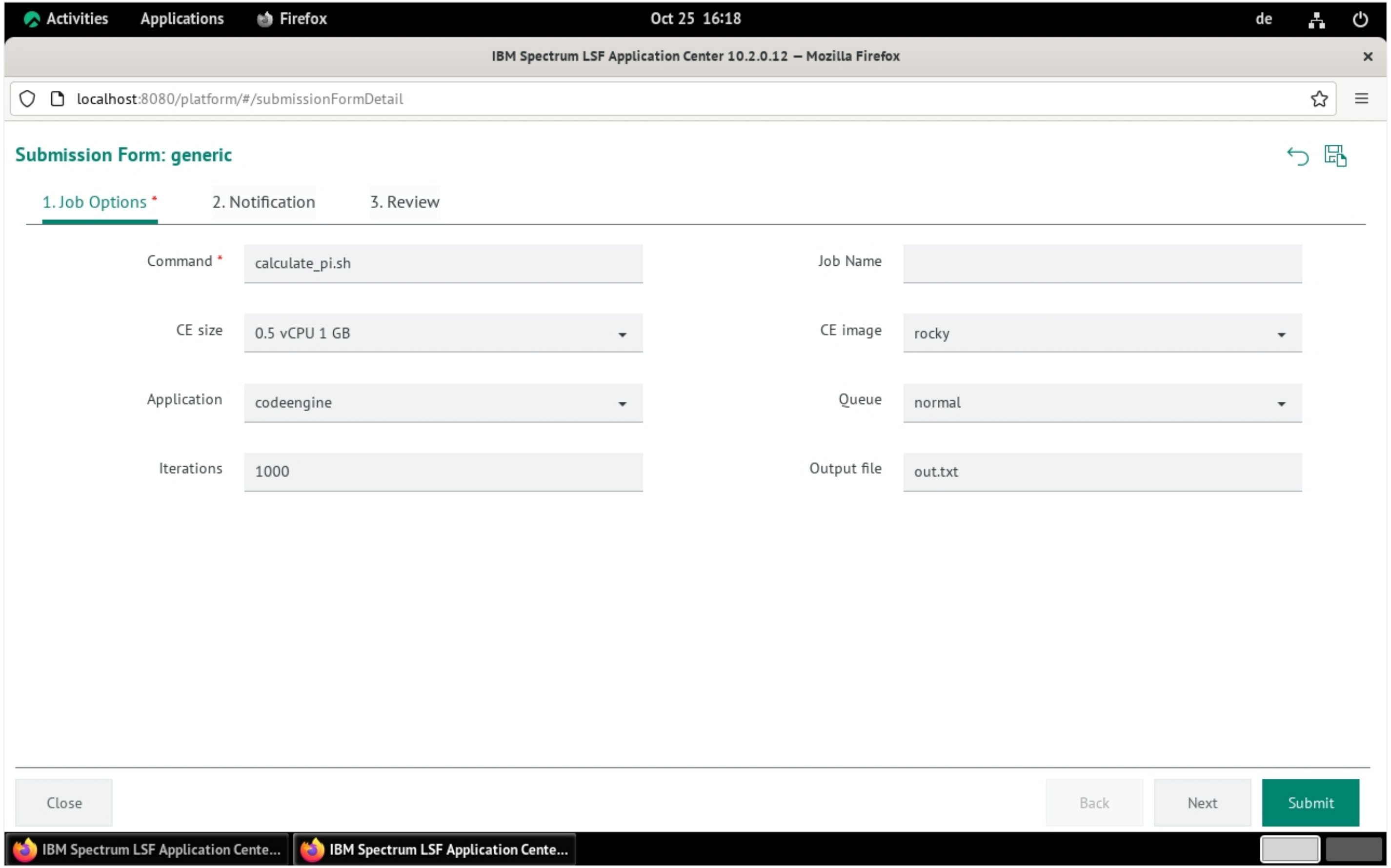
User Licenses    Set User Limits

● Used (1)
● Available (Not set)
● Overused (N/A)

User Sessions

● Browser(1)
● Desktop(0)
● Web Service(0)
● Mobile(0)

# Template for calculate-pi, to be executed on CodeEngine

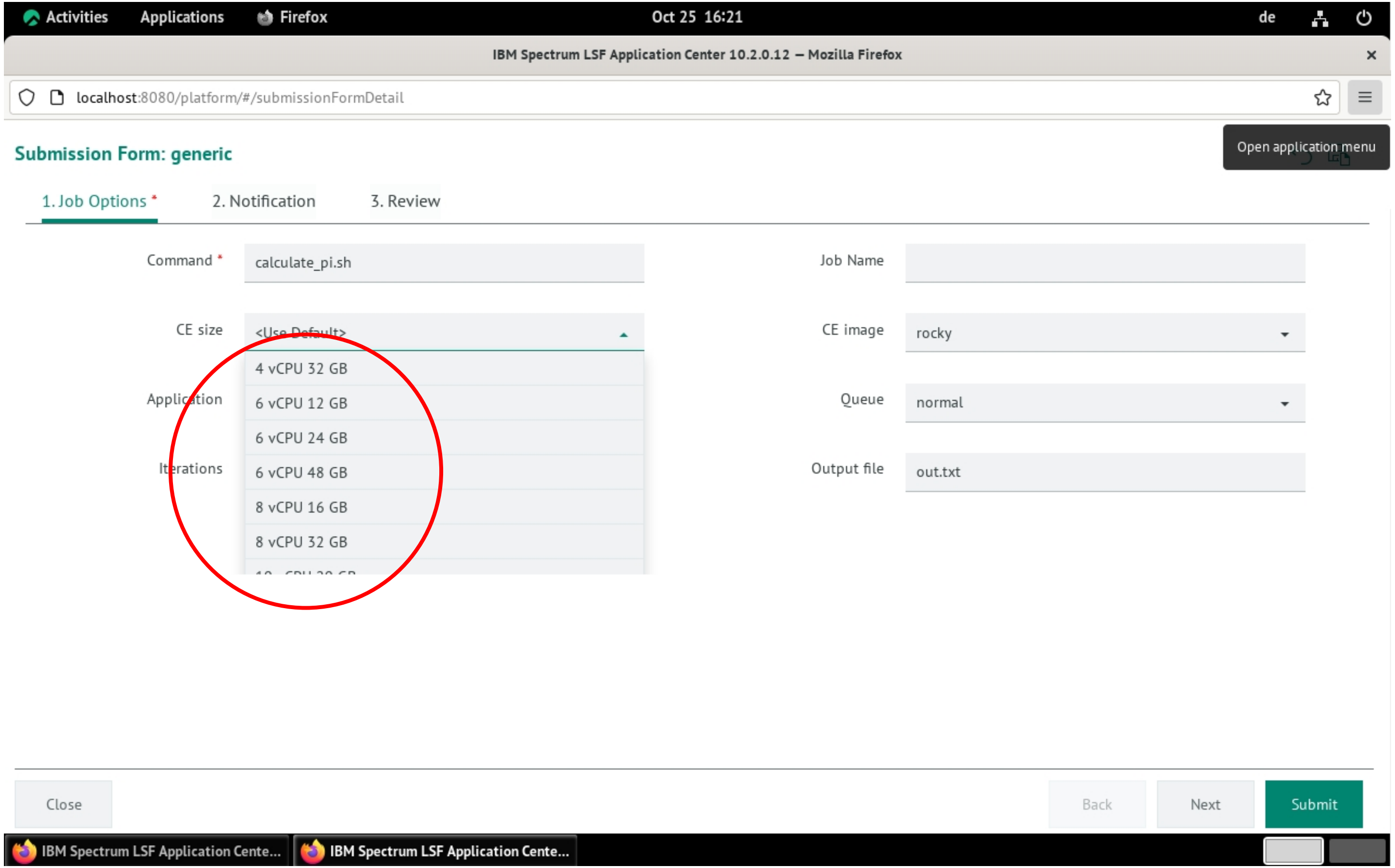

Activities
Applications
Firefox
Oct 25 16:18
de
IBM Spectrum LSF Application Center 10.2.0.12 — Mozilla Firefox
localhost:8080/platform/#/submissionFormDetail
Close
Back
Next
Submit
IBM Spectrum LSF Application Cente...
IBM Spectrum LSF Application Cente...

**Submission Form: generic**

1. Job Options * 2. Notification 3. Review

| | | | |
|---|---|---|---|
| Command * | calculate_pi.sh | Job Name | |
| CE size | 0.5 vCPU 1 GB | CE image | rocky |
| Application | codeengine | Queue | normal |
| Iterations | 1000 | Output file | out.txt |

# Select the size of the CodeEngine instance



Activities
Applications
Firefox
Oct 25 16:21
de
IBM Spectrum LSF Application Center 10.2.0.12 — Mozilla Firefox
localhost:8080/platform/#/submissionFormDetail
Open application menu

Submission Form: generic

1. Job Options *
2. Notification
3. Review

Command *: calculate_pi.sh

Job Name:

CE size: <Use Default>

- 4 vCPU 32 GB
- 6 vCPU 12 GB
- 6 vCPU 24 GB
- 6 vCPU 48 GB
- 8 vCPU 16 GB
- 8 vCPU 32 GB

CE image: rocky

Application

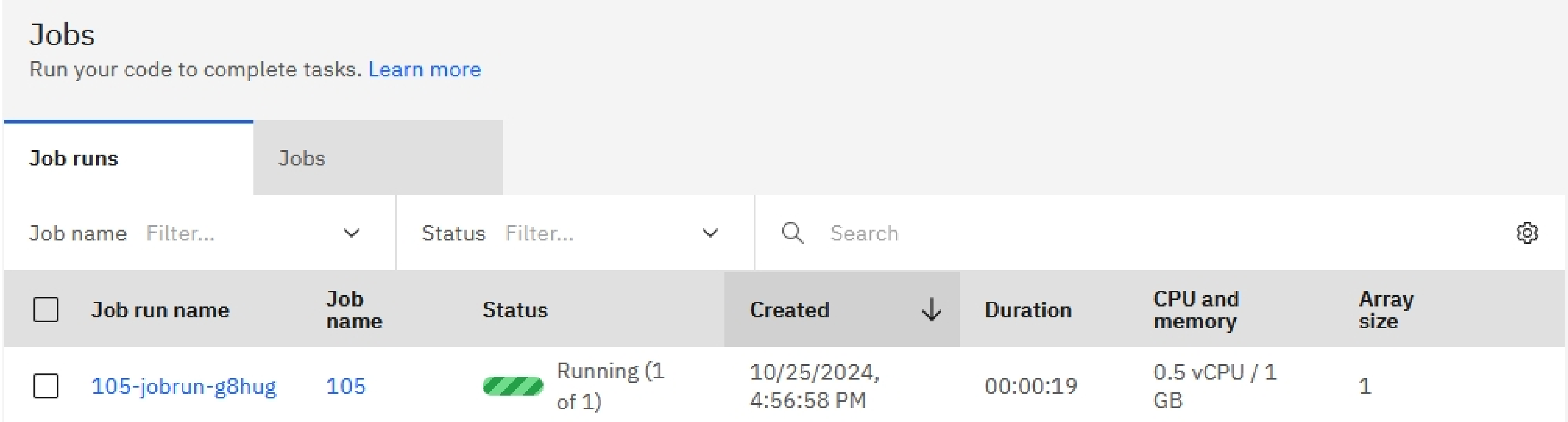
Queue: normal

Iterations

Output file: out.txt

Close
Back
Next
Submit
IBM Spectrum LSF Application Cente...
IBM Spectrum LSF Application Cente...

# LSF job 105 sumitted via Application Center on CodeEngine

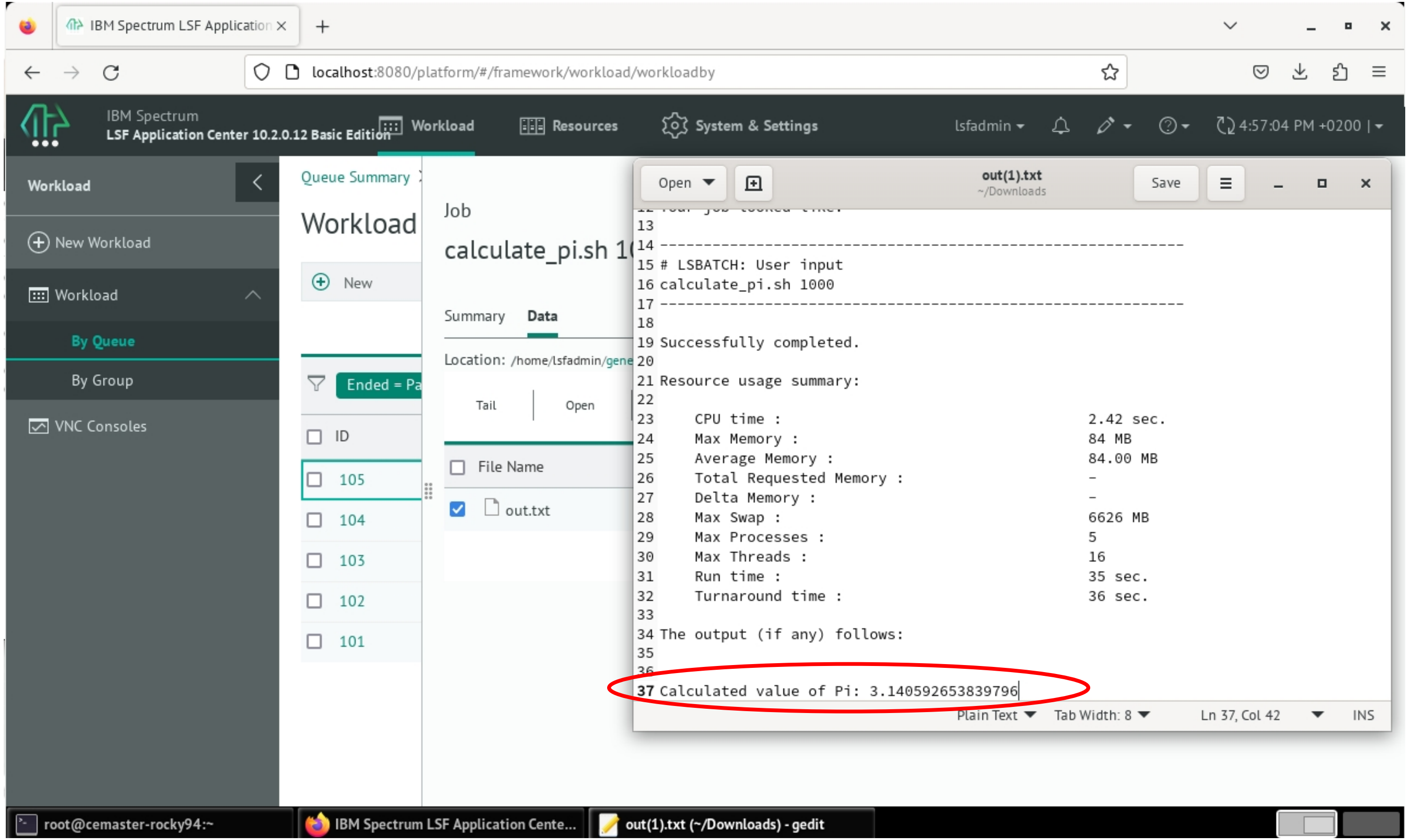


## Jobs

Run your code to complete tasks. Learn more

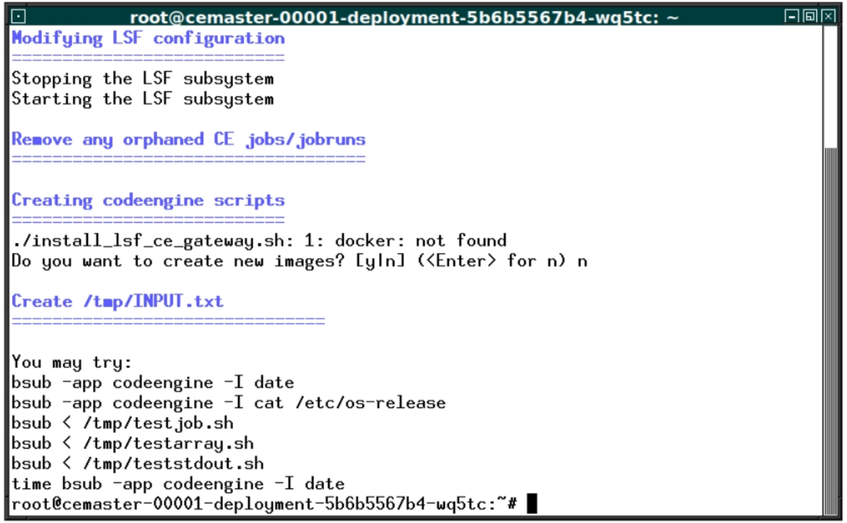
| □ | Job run name | Job name | Status | Created | Duration | CPU and memory | Array size |
|---|---|---|---|---|---|---|---|
| □ | 105-jobrun-g8hug | 105 | Running (1 of 1) | 10/25/2024, 4:56:58 PM | 00:00:19 | 0.5 vCPU / 1 GB | 1 |

# Result/output of LSF job 105…



```
Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
calculate_pi.sh 1000
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                   2.42 sec.
    Max Memory :                                 84 MB
    Average Memory :                             84.00 MB
    Total Requested Memory :                     -
    Delta Memory :                               -
    Max Swap :                                   6626 MB
    Max Processes :                              5
    Max Threads :                                16
    Run time :                                   35 sec.
    Turnaround time :                            36 sec.

The output (if any) follows:


Calculated value of Pi: 3.140592653839796
```

# One step further, run LSF/SLURM master as CE instance… (using guacamole/fvwm2/xterm), due to container restrictions…



```
Modifying LSF configuration
===========================
Stopping the LSF subsystem
Starting the LSF subsystem

Remove any orphaned CE jobs/jobruns
===================================

Creating codeengine scripts
===========================
./install_lsf_ce_gateway.sh: 1: docker: not found
Do you want to create new images? [y|n] (<Enter> for n) n

Create /tmp/INPUT.txt
===============================

You may try:
bsub -app codeengine -I date
bsub -app codeengine -I cat /etc/os-release
bsub < /tmp/testjob.sh
bsub < /tmp/testarray.sh
bsub < /tmp/teststdout.sh
time bsub -app codeengine -I date
root@cemaster-00001-deployment-5b6b5567b4-wq5tc:~#
```