

This document presents steps necessary to run fio tool tests to verify Ceph cluster performance.

This procedure will perform exactly the same storage benchmarking as it was done on the cloud right after its deployment.

```
# create test project or use existing one
# if you create a new project make sure the networking, routing and security groups in the
openstack project allow for ssh connection to VMs and that VMs are able to get package updates
from the internet/local landscape.

# create volumes (make sure they get created using the Ceph backend in multiple backends are
implemented)
for i in b c d; do openstack volume create --size 10 vol-vd${i}; done

# launch instance
# fill in appropriate values for the variables below
image=<name or id of the latest generic Ubuntu Focal image (without any customizations)>
flavor=<flavor name; min 4G of memory and 4vCPUs>
keypair=<keypair used to connect to the VM>
netid=<private network id>
openstack server create --image ${image} --flavor ${flavor} --key-name ${keypair} \
  --nic net-id=${netid} \
  --block-device-mapping vdb=vol-vdb \
  --block-device-mapping vdc=vol-vdc \
  --block-device-mapping vdd=vol-vdd \
  cephbench

# create floating ip on the public network
# this step might be omitted if the VM is accessible from other VM in the same network and the VM
can pull Ubuntu packages for install/update
os floating ip create <public network id>

# assign floating ip to the benchmarking VM
os server add floating ip cephbench <floating ip id from the previous step>

# login to the benchmarking VM
ssh -i ~/<private key> ubuntu@<floating IP>

# verify the volumes have been attached (devices /dev/vdb, /dev/vdc, and /dev/vdd should be
present)
lsblk -fp

# update the packages and install the fio tool
sudo apt update && sudo apt -y dist-upgrade && sudo apt install -y fio

# reboot the VM and reconnect
sudo reboot
ssh -i ~/<private key> ubuntu@<floating IP>

# create the run_fio.sh script
cat << EOF > run_fio.sh
#!/bin/bash

set -x

OPS_LIST="randwrite randread write read randrw"
BS_LIST="4k 4M"

# All in seconds
TEST_DURATION=300 # must divide by 30 without reminder
RUNS=$((\${TEST_DURATION} / 30 ))
```

```

set +x

for bs in \${BS_LIST}; do
    for ops in \${OPS_LIST}; do
        for run in \$(seq 1 \${RUNS}); do
            outfile="\${bs}_\${ops}_\${run}.json"
            echo "Running \${bs}: \${ops}. Run: \${run}. Results will be saved to: \${outfile}"
            sudo fio --rw=\${ops} --bs=\${bs} --output-format=json --output=\${outfile} disk.fio
        done
    done
done

echo "Done"
EOF

# make the run_fio.sh script executable
chmod +x ./run_fio.sh

# create the fio config file
cat << EOF > disk.fio
[global]
ioengine=libaio
iodepth=32
direct=1
random_generator=lfsr
numjobs=1
group_reporting=1
runtime=30
time_based=1

[job 1]
filename=/dev/vdb

[job 2]
filename=/dev/vdc

[job 3]
filename=/dev/vdd
EOF

# run the benchmarking
./run_fio.sh

# As a result a number of .json output files will be created in the current directory.
Please save all of them to tar.gz/tar.xz archive and send them to Canonical support.

```