

## Guide to instantiate two CloudLab servers connected by two experimental LANs

This guide assumes that the user has an active CloudLab account and has uploaded a public key. More information on these prerequisites can be found at <https://docs.cloudlab.us/users.html>.

**Step 1.** Log in to [cloudlab.us](https://cloudlab.us)

**Step 2.** Check for resource availability at <https://www.cloudlab.us/resinfo.php>. For the INSANE project, the user should ensure that there are currently at least two free nodes of suitable hardware types. Suitable hardware types are: d6515, c6525-100g, c6525-25g, r7525. The two nodes must be of the same type, but you can choose the most appropriate to your need. If there are enough of that type, the user can proceed to Step 3. Otherwise, there are two alternatives:

- a) Schedule the experiment for a later time. A graph on the resource information page shows when the desired resources will be available. This is usually the quickest way to get them: proceed to Step 3 and set the desired date and time at Step 6.
- b) Reserve resources at <https://www.cloudlab.us/resgroup.php>. This might take longer. More info on resource reservation can be found at <https://docs.cloudlab.us/reservations.html>. Here is an example:

Reservation Request

Project

INSANEProject

Cluster(s)

Cloudlab Utah

d6515 (28)

2

Number of Days  
(Optional)

0.0

Search

If you tell us how many days, we will find a slot and set the Start/End for you.

Reservation Start  
(Optional)

11/10/2023

9:00 AM

Reservation End

When CloudLab is busy, we give preference to reservation requests that use few resources or are short in duration.

11/10/2023

10:00 AM

Reason

Artifact evaluation for the ACM Middleware conference

Check

**Step 3.** Start the experiment by instantiating a profile. For the INSANE project, please use a profile that has Ubuntu 22.04 and meets the [requirements](#) describe in our repository. For your convenience, we have prepared a profile that is ready to use. Click on the following link, then click “next”:

<https://www.cloudlab.us/instantiate.php?project=INSANEProject&profile=Ubuntu22.04-TwoLANs>

1. Select a Profile

2. Parameterize

3. Finalize

4. Schedule

**Selected Profile:** Ubuntu22.04-TwoLANs:5

Node with Ubuntu 22.04, two experimental LANs and DPDK 22.11 installed.

Copy Profile

Show Profile

Previous

Next

**Step 4.** Parametrize the profile with 2 nodes and a hardware type and click “next”. For the INSANE evaluation, we suggest using one of the following hardware types: d6515, c6525-100g, c6525-25g, r7525. Choose the more appropriate type based on your need and the current availability. For the Middleware paper, we used the d6515 hardware type (default).

1. Select a Profile

2. Parameterize

3. Finalize

4. Schedule

**Selected Profile:** Ubuntu22.04-TwoLANs:6

Save/Load Parameters

Resource Availability

+ Show All Parameter Help

This profile is parameterized; please make your selections below, and then click **Next**.

Number of nodes

2

Physical node type ⓘ

d6515

Previous

Next

**Step 5.** Give a name to the experiment and associate it to a project, then click “next”

1. Select a Profile

2. Parameterize

3. Finalize

4. Schedule

**Selected Profile:** Ubuntu22.04-TwoLANs:5

Source

Please review the selections below and then click Next.

Name:

insane-test

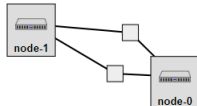
Project:

INSANEProject ▾

ⓘ Advanced Options

Check Resource Availability

Click Node to Select



Previous

Next

**Step 6.** Select either the duration of the experiment or a date/time for the experiment start, then click “finish”. If you do not specify a date/time but only the duration, the experiment will start immediately, provided that there are enough available resources.

Please select when you would like to start this experiment and then click Finish. hours

Start on date/time (optional) ⓘ

11/10/2023 9:00 AM

Experiment Duration ⓘ

2

[Previous](#) [Finish](#)

**Step 7.** Once the time comes from the experiment, the resources will be allocated. This process might take several minutes (even 10-15), so be patient. Eventually, this is the interface that appears in case of success:

▼ Your experiment is ready

Name: insane-test

State: ready

Profile: Ubuntu22.04-TwoLANs

Creator: lr499

Project: INSANEProject

Scheduled: Nov 10, 2023 10:00 AM

Started: Nov 10, 2023 10:00 AM

Expires: Nov 10, 2023 12:00 PM (in 2 hours)

[Logs](#) [Portal Log](#) [Share](#) [Save Parameters](#) [Modify](#) [Create Disk Image](#) [Copy](#) [Extend](#) [Terminate](#)

Topology View **List View** Manifest Graphs Bindings

ID	Node	Type	Cluster	Status	Startup	Image	SSH command (if you provided your own key)	<input type="checkbox"/>	
node-0	amd189	c6525-25g	Utah	<span>ready</span>	n/a	insaneproject-PG0/Ubuntu22.04-DPDK22.11	ssh lr499@amd189.utah.cloudlab.us	<input type="checkbox"/>	
node-1	amd203	c6525-25g	Utah	<span>ready</span>	n/a	insaneproject-PG0/Ubuntu22.04-DPDK22.11	ssh lr499@amd203.utah.cloudlab.us	<input type="checkbox"/>	

- In case of success, the status of the nodes is “running” (see picture above). The “SSH command” column reports the command to run to access the two nodes. If you saved your private key on a non-standard location or with a name different from “id\_rsa”, then you should also pass the command “-i <private\_key\_file>” to use the correct key. Then proceed to Step 8.
- If there are errors (either because of a wrong profile configuration or because there are no resources available, terminate the experiment and go back to Step 2.

**Step 8.** Login to the two nodes using ssh. You must use the private key associated with the public key uploaded during the signup process (see prerequisites).

**Step 9.** Check the network configuration. Each node has three relevant network interfaces:

- a) Management (eno33np0). This is the interface used for ssh and should not be used for any experimental data traffic.
- b) Experimental 1 (enp65s0f0np0). This is the first experimental interface. On different machines and hardware, the name of the interface might change, but the IP address should be 192.168.0.1 on node 0, 192.168.0.2 on node 1, etc. **We recommend using this interface for the DPDK traffic.**
- c) Experimental 2 (enp65s0f1np1). This is the second experimental interface. On different machines and hardware, the name of the interface might change, but the IP address should be 10.0.0.1 on node 0, 10.0.0.2 on node 1, etc. **We recommend using this interface for the kernel UDP traffic.**

Here are some examples of two nodes:

#### Node 0:

```
lr499@node-0:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eno33np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 1c:34:da:6f:c6:66 brd ff:ff:ff:ff:ff:ff
    altname enp1s0f0np0
    inet 128.110.219.100/21 metric 1024 brd 128.110.223.255 scope global eno33np0
        valid_lft forever preferred_lft forever
    inet6 fe80::1e34:daff:fe6f:c666/64 scope link
        valid_lft forever preferred_lft forever
3: eno34np1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 1c:34:da:6f:c6:67 brd ff:ff:ff:ff:ff:ff
    altname enp1s0f1np1
4: enp65s0f0np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 0c:42:a1:dd:5f:74 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/16 brd 192.168.255.255 scope global enp65s0f0np0
        valid_lft forever preferred_lft forever
    inet6 fe80::e42:a1ff:fedd:5f74/64 scope link
        valid_lft forever preferred_lft forever
5: enp65s0f1np1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 0c:42:a1:dd:5f:75 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/16 brd 10.0.255.255 scope global enp65s0f1np1
        valid_lft forever preferred_lft forever
    inet6 fe80::e42:a1ff:fedd:5f75/64 scope link
        valid_lft forever preferred_lft forever
```

## Node 1:

```
lr499@node-1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eno33np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 1c:34:da:6f:cc:e6 brd ff:ff:ff:ff:ff:ff
    altname enp1s0f0np0
    inet 128.110.219.114/21 metric 1024 brd 128.110.223.255 scope global eno33np0
        valid_lft forever preferred_lft forever
    inet6 fe80::1e34:daff:fe6f:cce6/64 scope link
        valid_lft forever preferred_lft forever
3: eno34np1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 1c:34:da:6f:cc:e7 brd ff:ff:ff:ff:ff:ff
    altname enp1s0f1np1
4: enp65s0f0np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 0c:42:a1:dd:60:b0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.2/16 brd 192.168.255.255 scope global enp65s0f0np0
        valid_lft forever preferred_lft forever
    inet6 fe80::e42:a1ff:fedd:60b0/64 scope link
        valid_lft forever preferred_lft forever
5: enp65s0f1np1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 0c:42:a1:dd:60:b1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/16 brd 10.0.255.255 scope global enp65s0f1np1
        valid_lft forever preferred_lft forever
    inet6 fe80::e42:a1ff:fedd:60b1/64 scope link
        valid_lft forever preferred_lft forever
```

**Step 10.** Check prerequisites. There is no need to install the [INSANE prerequisites](#) as the profile already has DPDK installed. Nevertheless, users can still get through the various steps to make sure the environment is correctly configured.

**Step 11.** [Prepare the environment](#) by following the instructions on the repository. **Important:** as the NIC is a Mellanox card, in this case the user must not bind the interface to DPDK. The Mellanox driver is already capable of supporting userspace networking. Therefore, please **only setup hugepages**.

However, we still **need to know the PCI address** of the interface we will use with DPDK. We can obtain it by using the `dpdk-devbind.py` script and looking at the first experimental interface. In this case, the experimental interface for DPDK is `enp65s0f0np0` (see Step 9) and thus the corresponding PCI address is `0000:41:00.0`. Please copy this address as it will be needed later.

```
lr499@node-0:~/INSANE/build$ sudo dpdk-devbind.py -s

Network devices using kernel driver
=====
0000:01:00.0 'MT27800 Family [ConnectX-5] 1017' if=en033np0 drv=mlx5_core unused=vfio-pci *Active*
0000:01:00.1 'MT27800 Family [ConnectX-5] 1017' if=en034np1 drv=mlx5_core unused=vfio-pci
0000:41:00.0 'MT27800 Family [ConnectX-5] 1017' if=enp65s0f0np0 drv=mlx5_core unused=vfio-pci *Active*
0000:41:00.1 'MT27800 Family [ConnectX-5] 1017' if=enp65s0f1np1 drv=mlx5_core unused=vfio-pci *Active*
```

**Step 12.** Clone and [build](#) the INSANE repository.

**Step 13.** Launch the INSANE runtime. In CloudLab, it will be necessary to pass the DPDK interface address as an environment variable called “DPDK\_PCI”.

On node0:

```
sudo DPDK_PCI=0000:41:00.0 taskset -c 0-2 ./nsnd 192.168.0.1
192.168.0.2 10.0.0.1 10.0.0.2
```

and on node1:

```
sudo DPDK_PCI=0000:41:00.0 taskset -c 0-2 ./nsnd 192.168.0.2
192.168.0.1 10.0.0.2 10.0.0.1
```

```
lr499@node-0:~/INSANE/build$ sudo DPDK_PCI=0000:41:00.0 taskset -c 0-2 ./nsnd 192.168.0.1 192.168.0.2 10
.0.0.1 10.0.0.2
EAL: Detected CPU lcores: 32
EAL: Detected NUMA nodes: 1
EAL: Detected shared linkage of DPDK
EAL: Multi-process socket /var/run/dpdk/rte/mp_socket
EAL: Selected IOVA mode 'VA'
EAL: VFIO support initialized
EAL: Probe PCI driver: mlx5_pci (15b3:1017) device: 0000:41:00.0 (socket -1)
TELEMETRY: No legacy callbacks, legacy socket not created
##### PORTS: 1

WARNING: Too many lcores enabled. Only 1 used.

Core 1 forwarding packets. [Ctrl+C to quit]
```

```
lr499@node-1:~/INSANE/build$ sudo DPDK_PCI=0000:41:00.0 taskset -c 0-2 ./nsnd 192.168.0.2 192.168.0.1 10
.0.0.2 10.0.0.1
EAL: Detected CPU lcores: 32
EAL: Detected NUMA nodes: 1
EAL: Detected shared linkage of DPDK
EAL: Multi-process socket /var/run/dpdk/rte/mp_socket
EAL: Selected IOVA mode 'VA'
EAL: VFIO support initialized
EAL: Probe PCI driver: mlx5_pci (15b3:1017) device: 0000:41:00.0 (socket -1)
TELEMETRY: No legacy callbacks, legacy socket not created
##### PORTS: 1

WARNING: Too many lcores enabled. Only 1 used.

Core 1 forwarding packets. [Ctrl+C to quit]
```

Troubleshoot. If the INSANE runtime does not exit normally (e.g., if killed or crashed) it does not correctly clean the environment. That might prevent further executions to succeed. In that case, please remove the following files and try again:

```
sudo rm -rf /dev/shm/insane
sudo rm -rf /tmp/insane_control.socket
```

**Step 14.** Launch the INSANE test applications. Note. Please stop the INSANE runtime every time. We will solve this issue in the next release. For instance, here are the steps to launch a latency test (ping-pong):

a) Launch the pong application on node1:

```
lr499@node-1:~/INSANE/build$ sudo DPDK_PCI=0000:41:00.0 taskset -c 0-2 examples/nsn-perf pong -s 64 -n 1000 -q fast
Welcome to the test application of the INSANE middleware
Running with the following arguments:
  Role..... : PONG
  Payload size.... : 64
  Max messages.... : 1000
  Datapath QoS.... : Fast
  Source id..... : 0
  Sleep time..... : 0

EAL: Detected CPU lcores: 32
EAL: Detected NUMA nodes: 1
EAL: Detected shared linkage of DPDK
EAL: Multi-process socket /var/run/dpdk/rte/mp_socket_5390_3add856b7e0
EAL: Selected IOVA mode 'VA'
EAL: VFIO support initialized
EAL: Probe PCI driver: mlx5_pci (15b3:1017) device: 0000:41:00.0 (socket -1)
```

b) Launch the ping application on node0:

```
lr499@node-0:~/INSANE/build$ sudo DPDK_PCI=0000:41:00.0 taskset -c 0-2 examples/nsn-perf ping -s 64 -n 1000 -q fast
Welcome to the test application of the INSANE middleware
Running with the following arguments:
  Role..... : PING
  Payload size.... : 64
  Max messages.... : 1000
  Datapath QoS.... : Fast
  Source id..... : 0
  Sleep time..... : 0

EAL: Detected CPU lcores: 32
EAL: Detected NUMA nodes: 1
EAL: Detected shared linkage of DPDK
EAL: Multi-process socket /var/run/dpdk/rte/mp_socket_5004_4d4f01dc99a
EAL: Selected IOVA mode 'VA'
EAL: VFIO support initialized
EAL: Probe PCI driver: mlx5_pci (15b3:1017) device: 0000:41:00.0 (socket -1)
28.494
8.926
8.857
8.135
```

Please stop and restart the daemon after each experiment. This is an issue we are working to solve.