Guide to instantiate two CloudLab servers connected by two experimental LANs

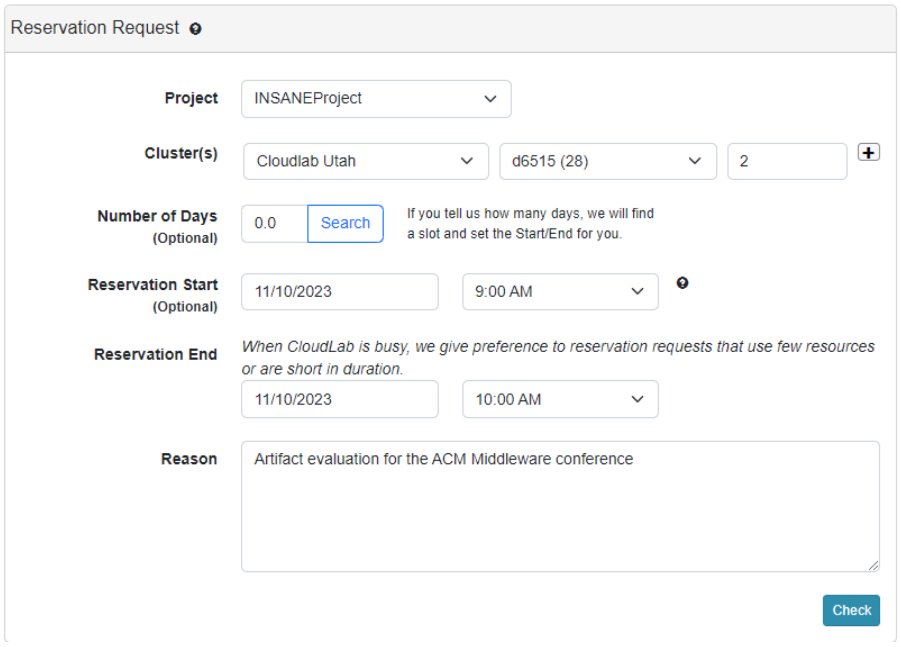
This guide assumes that the user has an active CloudLab account and has uploaded a public key. More information on these prerequisites can be found at https://docs.cloudlab.us/users.html.

Step 1. Log in to cloudlab.us

Step 2. Check for resource availability at https://www.cloudlab.us/resinfo.php. To evaluate INSANE on multiple network technologies, 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 available of that type, the user can proceed to Step 3. Otherwise, there are two alternatives:

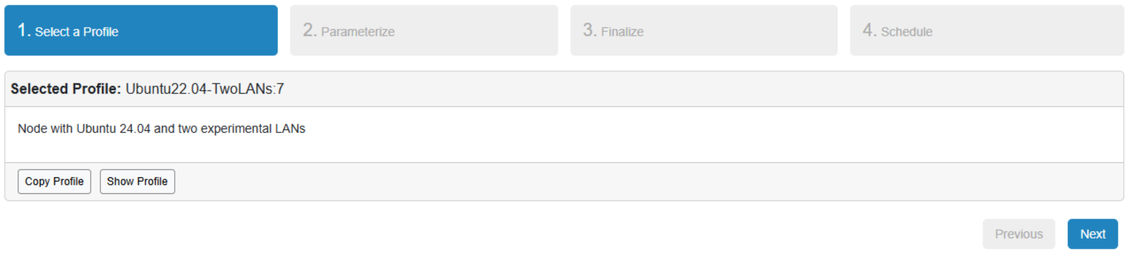
a) Schedule the experiment for a later time. A graph on the resource information page shows when the desired resources will be available. This is usually the quickest way to get them: proceed to Step 3 and set the desired date and time at Step 6.

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 — 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

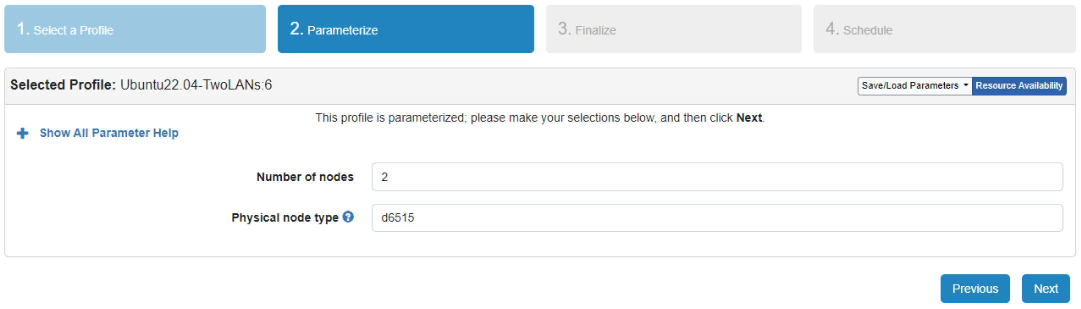
Step 3. Start the experiment by instantiating a profile. For the INSANE project, please use a profile that has Ubuntu 24.04 (although the name is still 22.04, which would work as well). For your convenience, we have prepared a profile that is ready to use. Click on the following link, then click "next":
https://www.cloudlab.us/instantiate.php?project=INSANEProject&profile=Ubuntu22.04-TwoLANs



Selected Profile: Ubuntu22.04-TwoLANs:7

Node with Ubuntu 24.04 and two experimental LANs

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 needs and the current availability. For the Middleware paper, we used the d6515 hardware type (default).

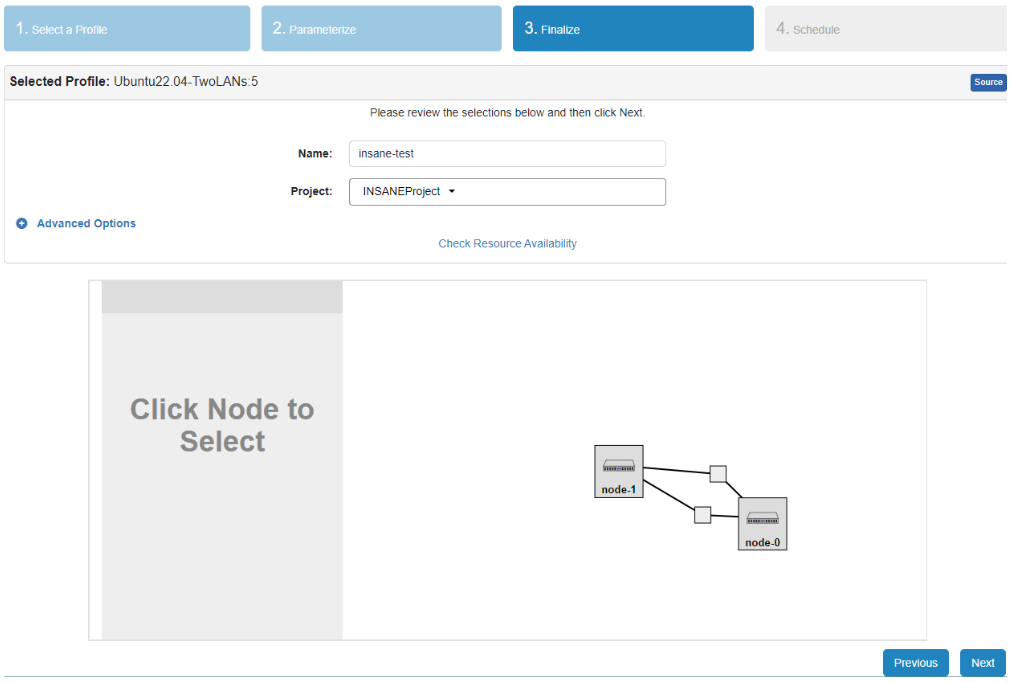


Selected Profile: Ubuntu22.04-TwoLANs:6

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

Number of nodes	2
Physical node type	d6515

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

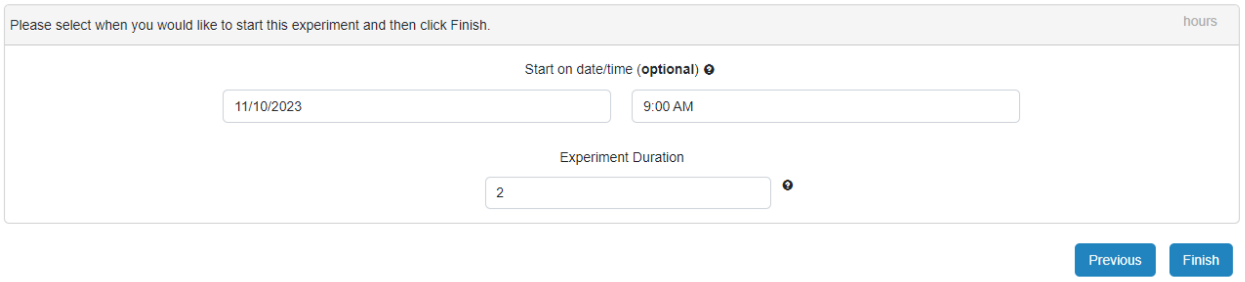


Selected Profile: Ubuntu22.04-TwoLANs:5

Please review the selections below and then click Next.

Name:	insane-test
Project:	INSANEProject

Step 6. Select the duration of the experiment <u>and</u>/<u>or</u> a date/time for the experiment beginning, 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 of the required hardware type.


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

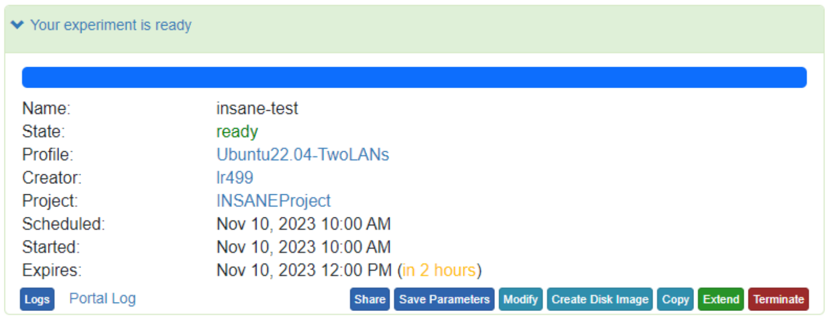
hours

Start on date/time (**optional**)

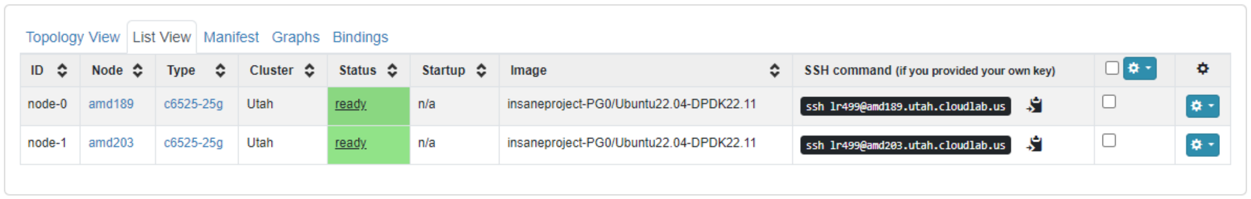
11/10/2023	9:00 AM

Experiment Duration: 2

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)



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

- In case of <u>success</u>, the status of the nodes is "ready" (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 argument "`-i <private_key_file>`" to the ssh command to use the correct key. Then proceed to Step 8.

- If there are <u>errors</u> (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 at the beginning of this document).

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

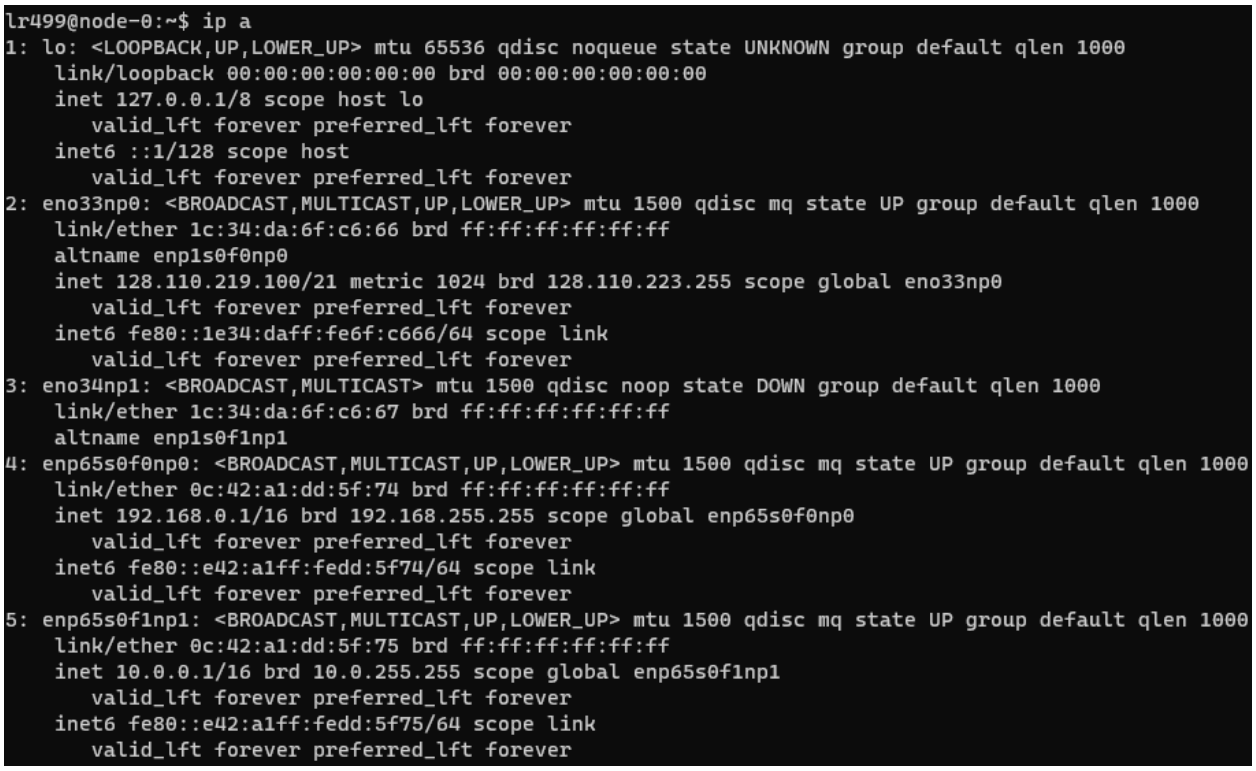
a) Management (`eno33np0`). This is the interface used for ssh and should not be used for any experimental data traffic. On different machines and hardware, the name of the interface might change.

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**.

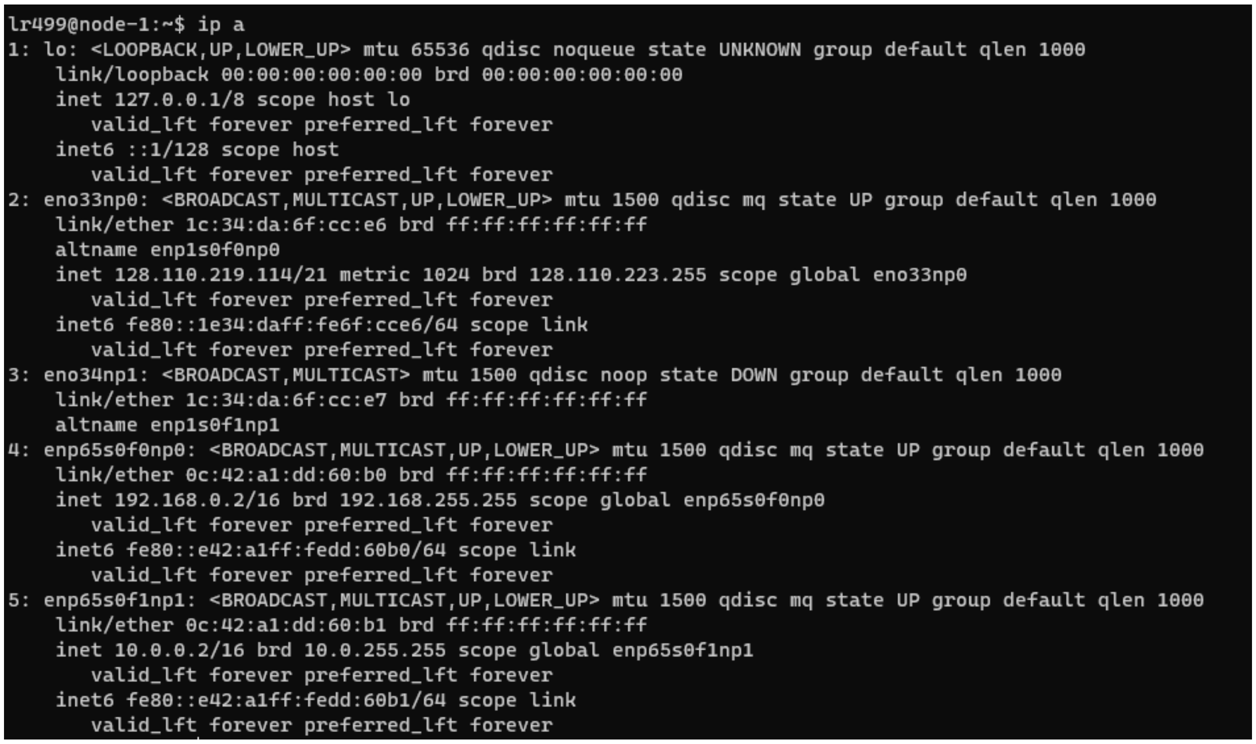
Here is an example on the two nodes:

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:

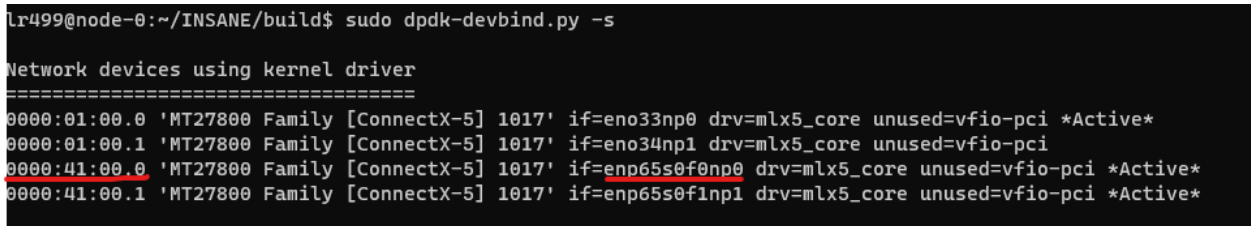


```
lr499@node-1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft 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 and install them: INSANE prerequisites. For civetweb, cJSON, DPDK, TLDK, we recommend installing them into a user-level folder (e.g., $INSANE_DIR/deps). This ensures that everything is under control and it is easier in case of need for re-installations. You can install these using the scripts provided in the INSANE repository.

Step 11. If using the DPDK network plugins: **Important:** if 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. In that case, please **only set up hugepages**. You can understand that a NIC is a Mellanox card by using the `dpdk-devbind.py` script and checking if the associated driver mlx5_core and mlx4_core.

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.



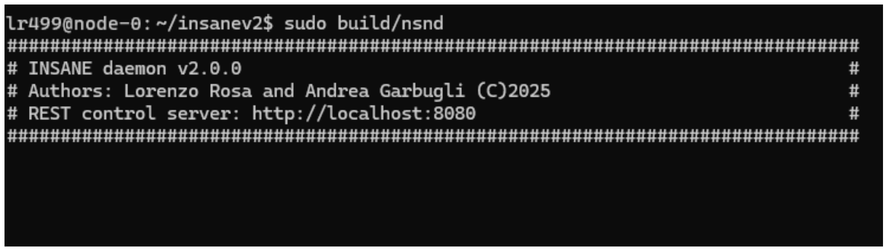
```
lr499@node-0:~/INSANE/build$ sudo dpdk-devbind.py -s

Network devices using kernel driver
===================================
0000:01:00.0 'MT27800 Family [ConnectX-5] 1017' if=eno33np0 drv=mlx5_core unused=vfio-pci *Active*
0000:01:00.1 'MT27800 Family [ConnectX-5] 1017' if=eno34np1 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.

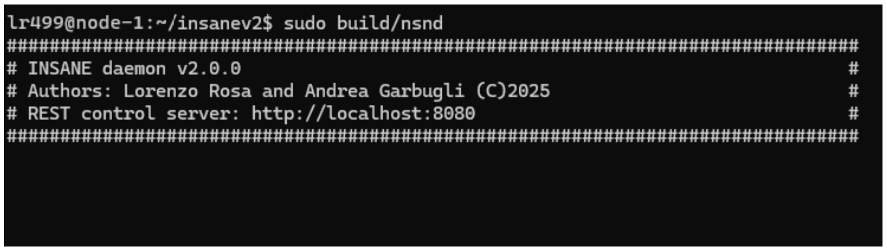
Step 13. Create a configuration file for the INSANE runtime. An example can be found in the configs/ folder. You must pass the remote IPs (for each peer, for each plugin) and the local IPs (for each plugin). We recommend placing this file in the same directory where you will invoke the daemon, although the --config flag of the daemon allows to specify alternative locations.

Step 14. Launch the INSANE runtime. We assume our working directory is the root of the repository.
On node0 **and** on node1:

```
sudo build/nsnd
```



```
lr499@node-0:~/insanev2$ sudo build/nsnd
################################################################################
# INSANE daemon v2.0.0                                                         #
# Authors: Lorenzo Rosa and Andrea Garbugli (C)2025                            #
# REST control server: http://localhost:8080                                   #
################################################################################
```

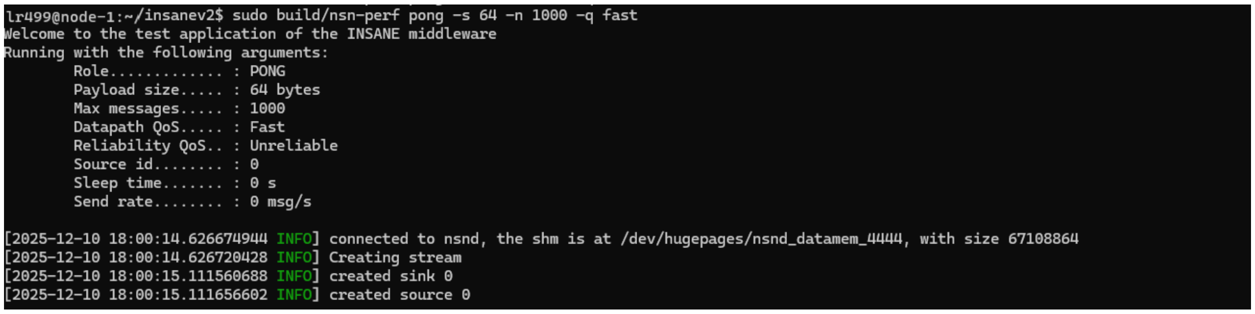


```
lr499@node-1:~/insanev2$ sudo build/nsnd
################################################################################
# INSANE daemon v2.0.0                                                         #
# Authors: Lorenzo Rosa and Andrea Garbugli (C)2025                            #
# REST control server: http://localhost:8080                                   #
################################################################################
```

Step 15. Create a configuration file for each of the INSANE application. An example can be found in the configs/ folder. The file **must be located** in the working directory and **must be called** nsn-app.cfg. The application ID must be a valid TCP or UDP port. Only apps with the same ID can communicate. Currently, we do not allow more than one app to have the same ID if they are attaching to the same daemon (local communication).

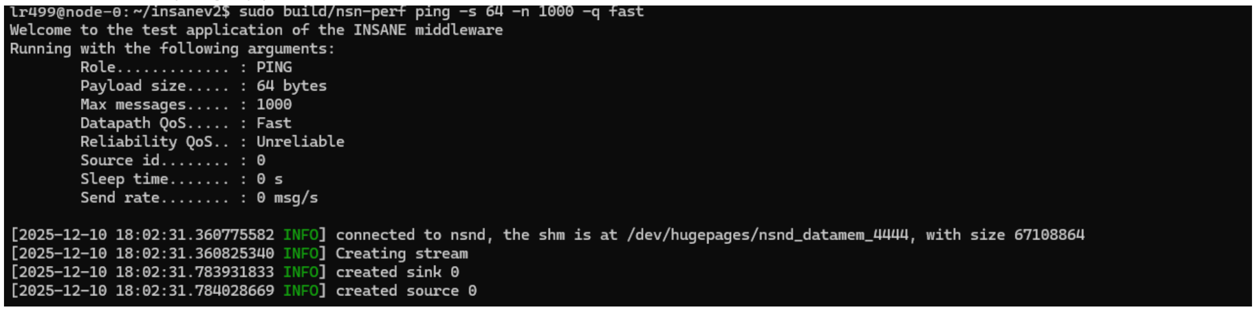
Step 16. In separate terminals, launch the INSANE test applications. For instance, here are the steps to launch a latency test (ping-pong):

a) Launch the pong application on node1:



```
lr499@node-1:~/insanev2$ sudo build/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 bytes
	Max messages..... : 1000
	Datapath QoS..... : Fast
	Reliability QoS.. : Unreliable
	Source id........ : 0
	Sleep time....... : 0 s
	Send rate........ : 0 msg/s

[2025-12-10 18:00:14.626674944 INFO] connected to nsnd, the shm is at /dev/hugepages/nsnd_datamem_4444, with size 67108864
[2025-12-10 18:00:14.626720428 INFO] Creating stream
[2025-12-10 18:00:15.111560688 INFO] created sink 0
[2025-12-10 18:00:15.111656602 INFO] created source 0
```

b) Launch the ping application on node0:



```
lr499@node-0:~/insanev2$ sudo build/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 bytes
        Max messages..... : 1000
        Datapath QoS..... : Fast
        Reliability QoS.. : Unreliable
        Source id........ : 0
        Sleep time....... : 0 s
        Send rate........ : 0 msg/s

[2025-12-10 18:02:31.360775582 INFO] connected to nsnd, the shm is at /dev/hugepages/nsnd_datamem_4444, with size 67108864
[2025-12-10 18:02:31.360825340 INFO] Creating stream
[2025-12-10 18:02:31.783931833 INFO] created sink 0
[2025-12-10 18:02:31.784028669 INFO] created source 0
```