

Reti di Calcolatori

Emulazione delle Reti – GNS3



Introduzione

GNS3 è un software open source che simula reti complesse in maniera realistica senza aver bisogno di hardware di rete dedicato. Possiede un'interfaccia grafica e funziona su più sistemi operativi, basandosi su altri software:

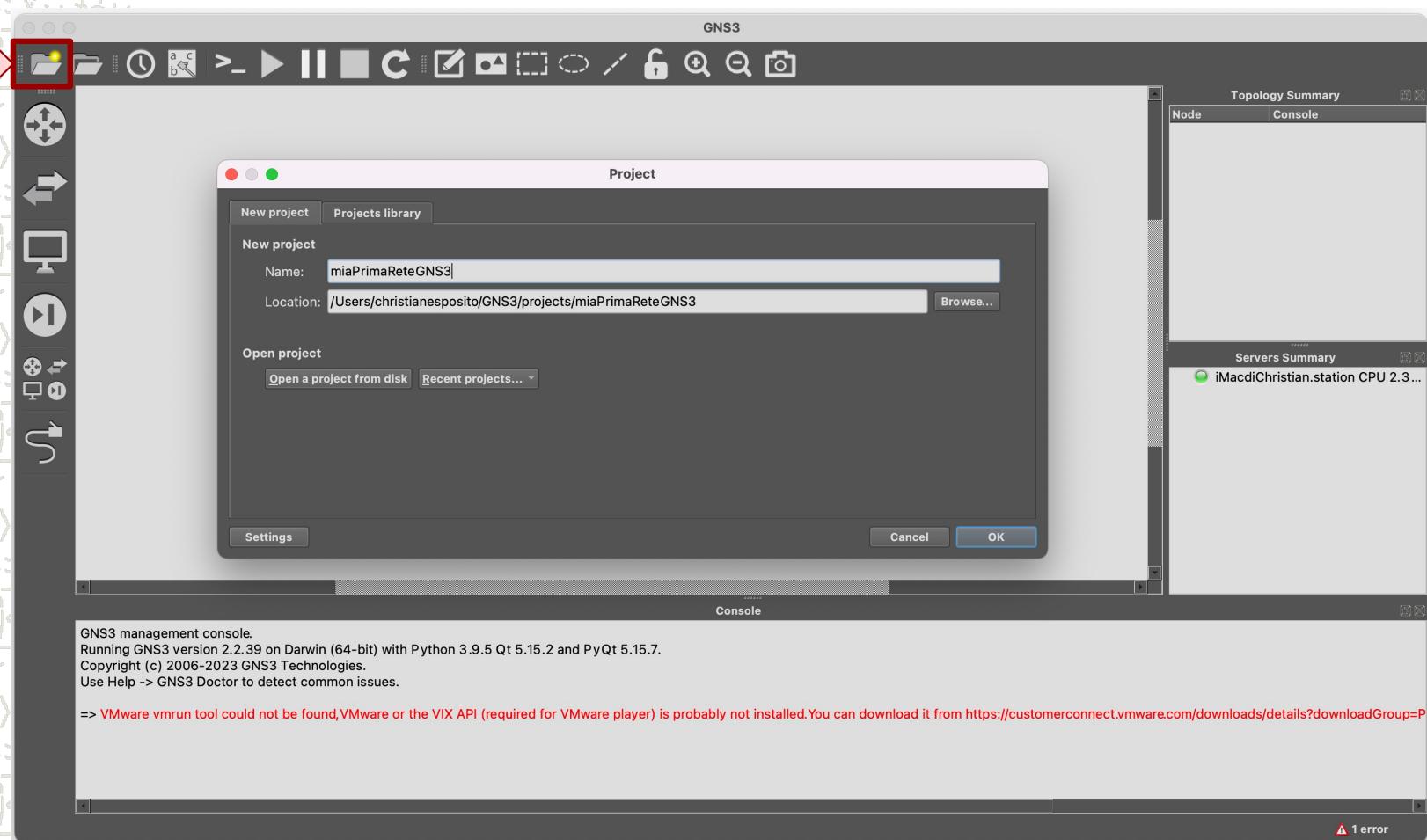
- Dynamips, che emula Cisco IOS,
- VirtualBox o Qemu, che emulano nodi server con vari SO.

GNS3 consiste di due componenti software: il software GNS3 all-in-one (GUI) per la definizione di topologie e la macchina virtuale GNS3 (VM), che consente di emulare, configurare, e testare reti in un ambiente di laboratorio virtuale.

Vi sono anche altri software a supporto che servono ad analizzare i pacchetti in entrata ed uscita, applicando quello che comunemente è definito Sniffing , cioè l'attività di intercettazione passiva dei dati che transitano in una rete per valutare il comportamento di un protocollo di rete come: NpCap, SolarWinds Response TimeViewer, WinPcap e Wireshark.

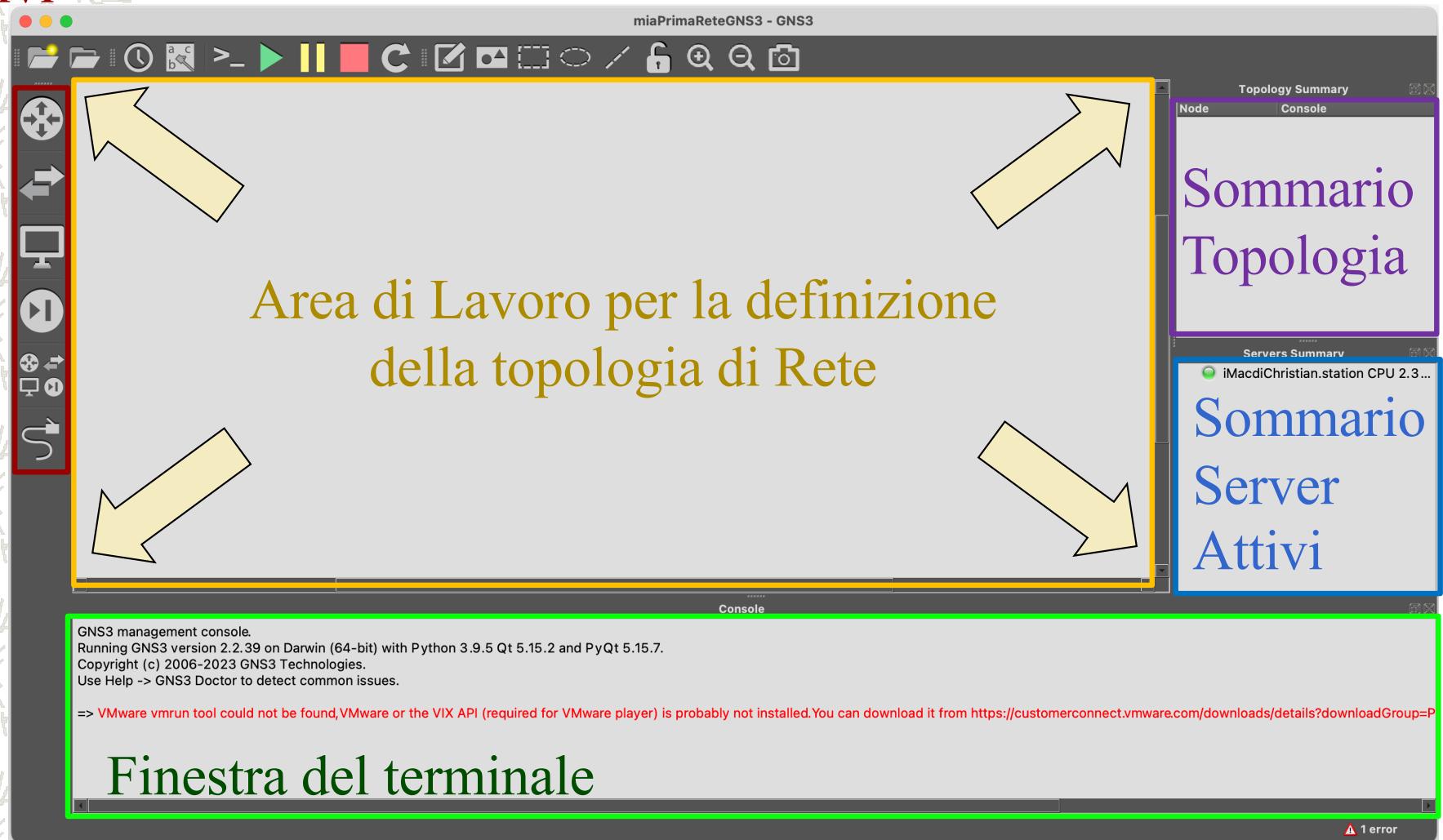
Esempio

Creiamo un progetto vuoto ed assegniamo un nome come miaPrimaReteGNS3:



Esempio

Toolbar dei dispositivi



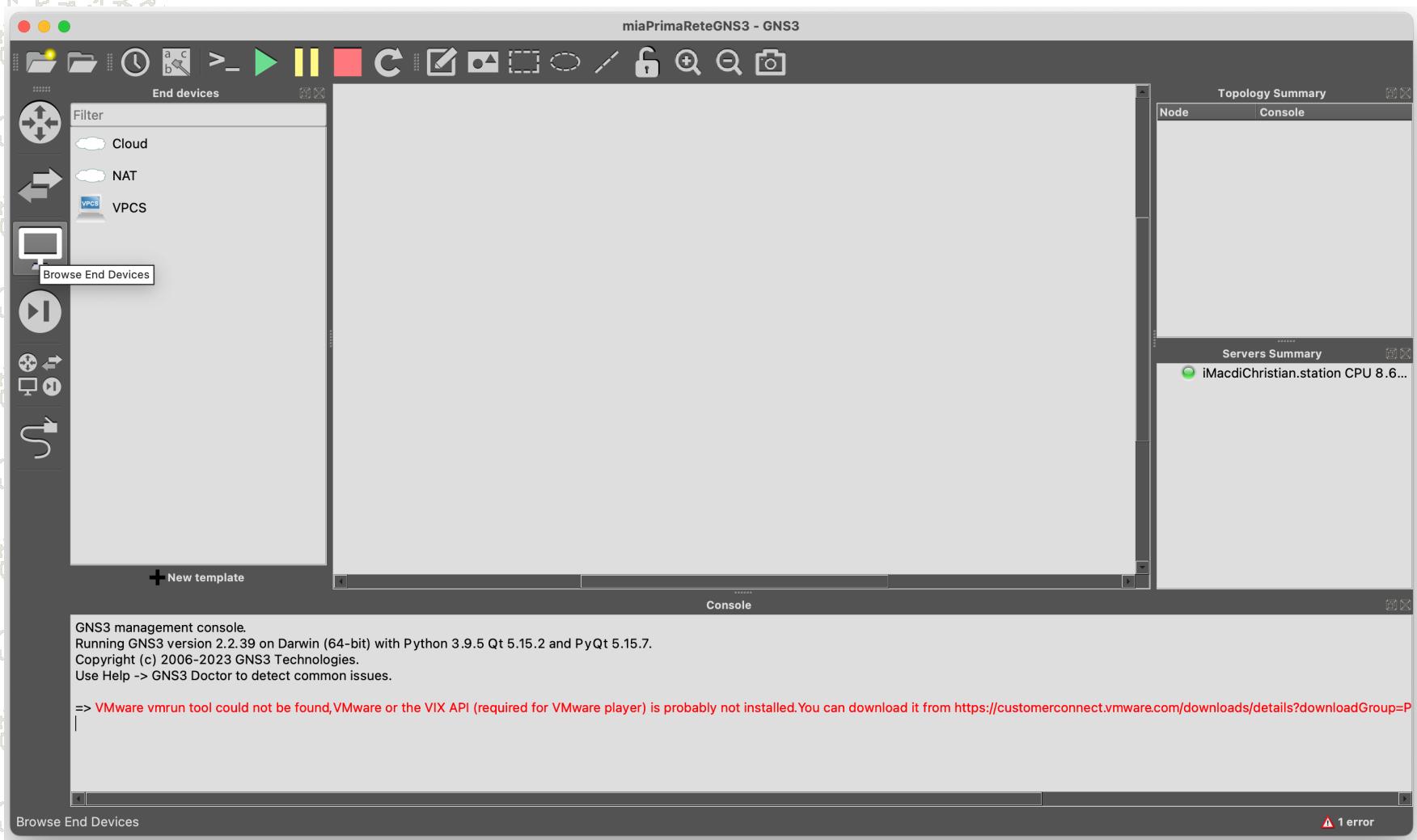
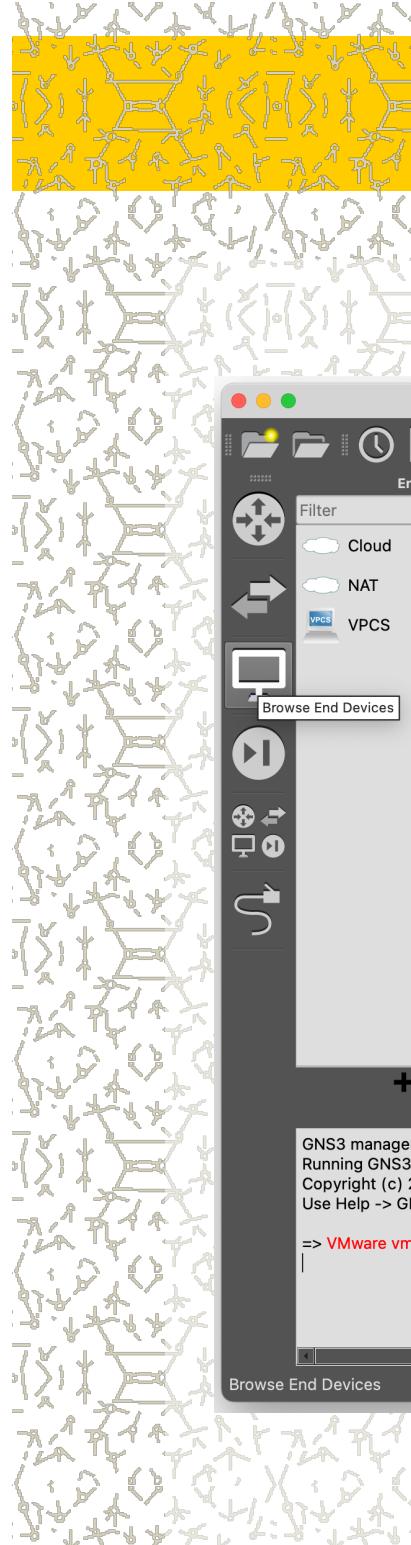
Esempio

Attualmente, la nostra rete non ha dispositivi, che vanno aggiunti. GNS3 supporta due tipi di dispositivi: dispositivi emulati e dispositivi simulati. L'installazione predefinita di GNS3 include dispositivi simulati.

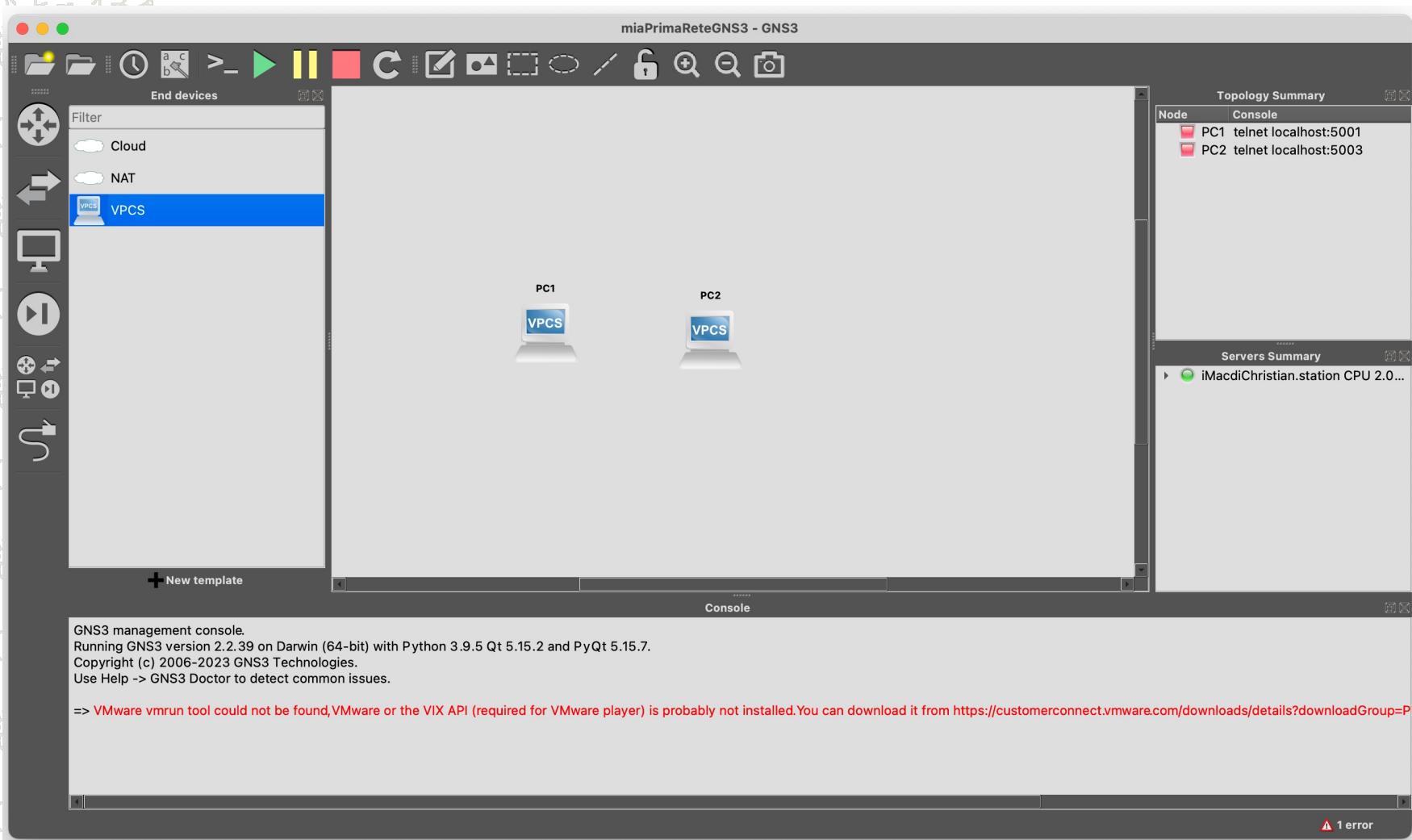
Costruiamo e testiamo una semplice rete che contiene due PC e uno switch Ethernet. Per costruire questa rete, per prima cosa dobbiamo aggiungere i PC e lo switch Ethernet all'area di lavoro.

Per aggiungere un dispositivo alla rete, seleziona il dispositivo dalla toolbar dei dispositivi e trascina il dispositivo nell'area di lavoro. Per disselezionare un'opzione selezionata sulla barra degli strumenti del dispositivo, fare clic sull'opzione selezionata o chiudere il sottoriquadro aperto.

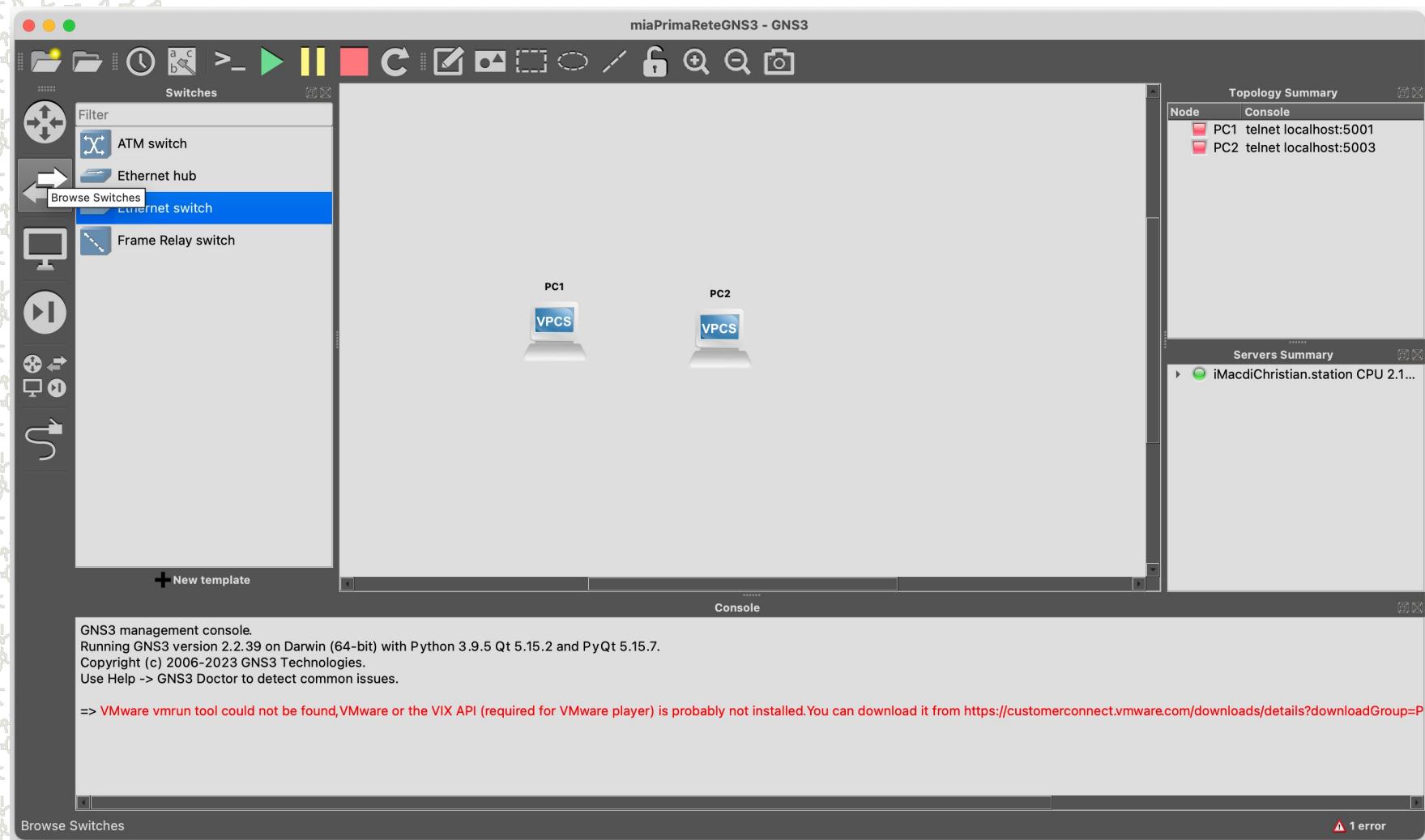
Esempio



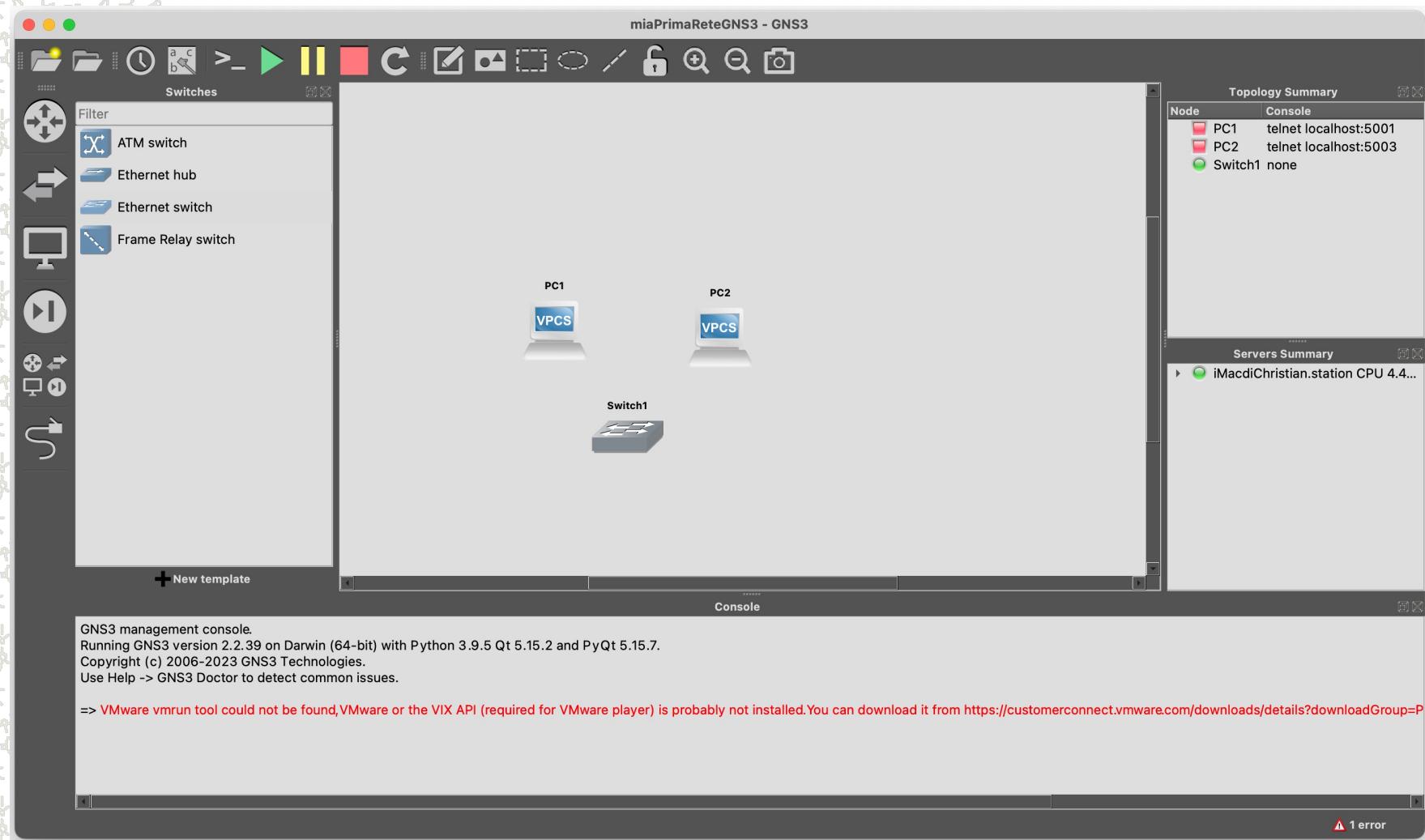
Esempio



Esempio



Esempio

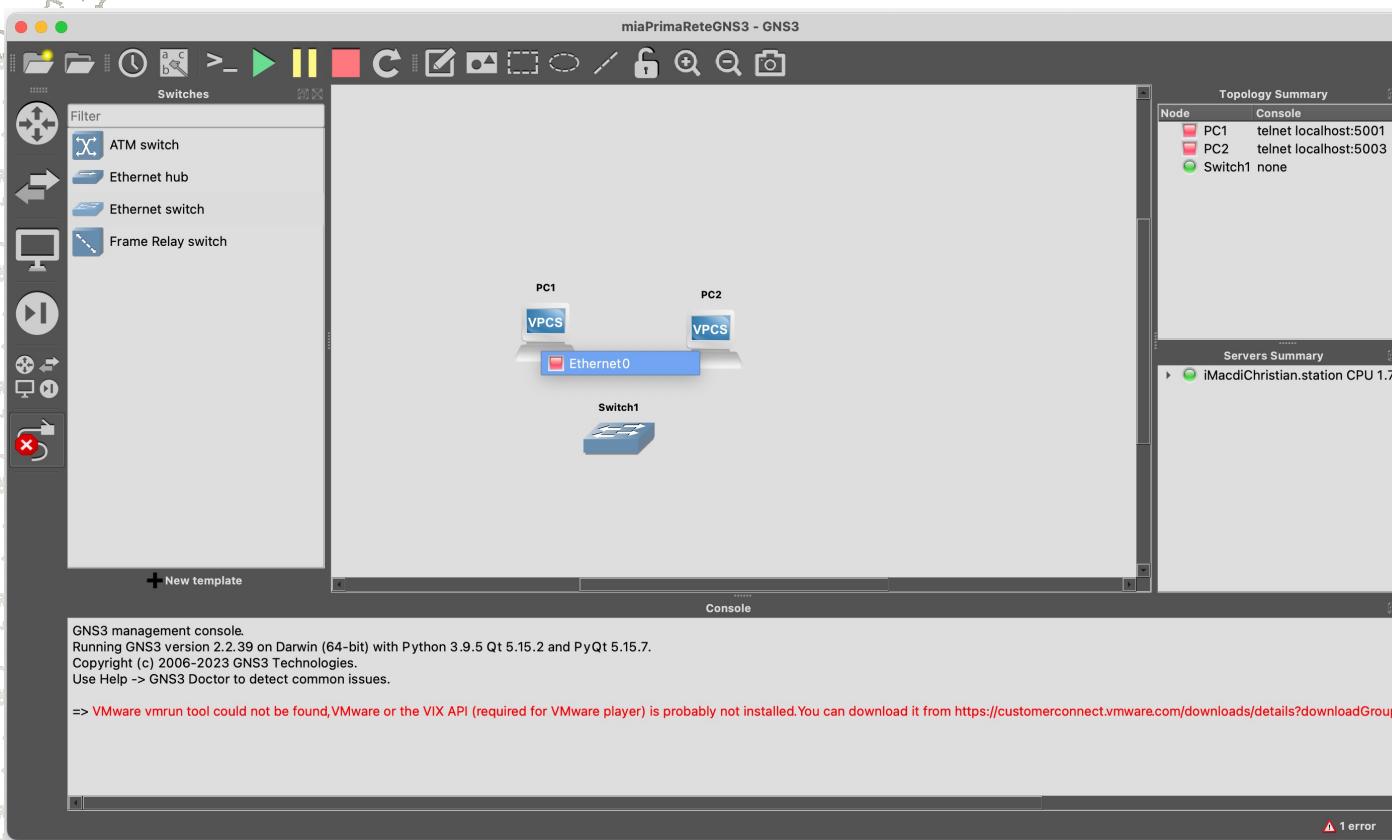


Esempio

Per connettere i dispositivi, fai clic su Aggiungi un collegamento.



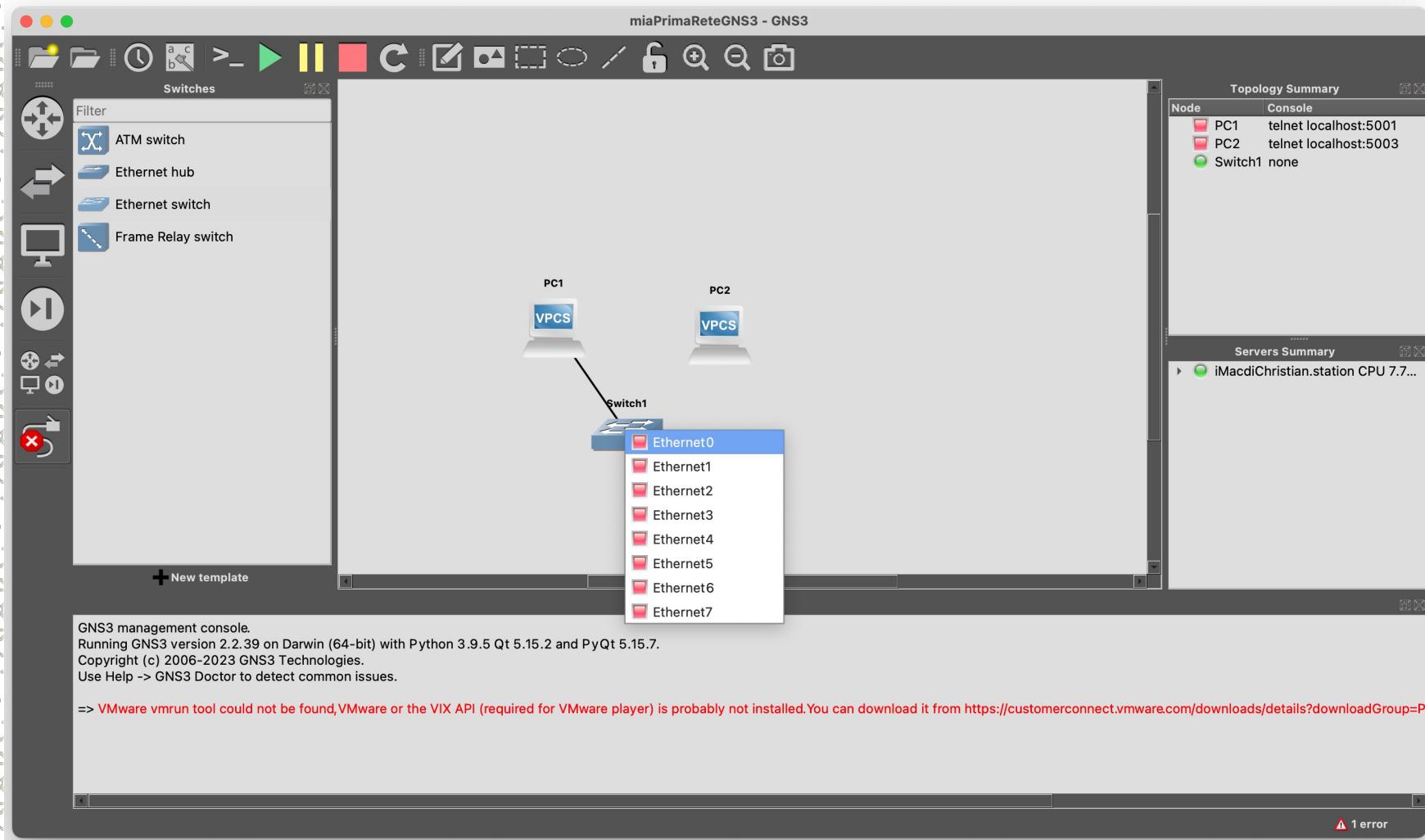
Esempio



Fare clic sul primo dispositivo (PC1). Il dispositivo visualizzerà tutte le interfacce disponibili. Seleziona l'interfaccia che desideri utilizzare per connettere questo dispositivo a un altro dispositivo.

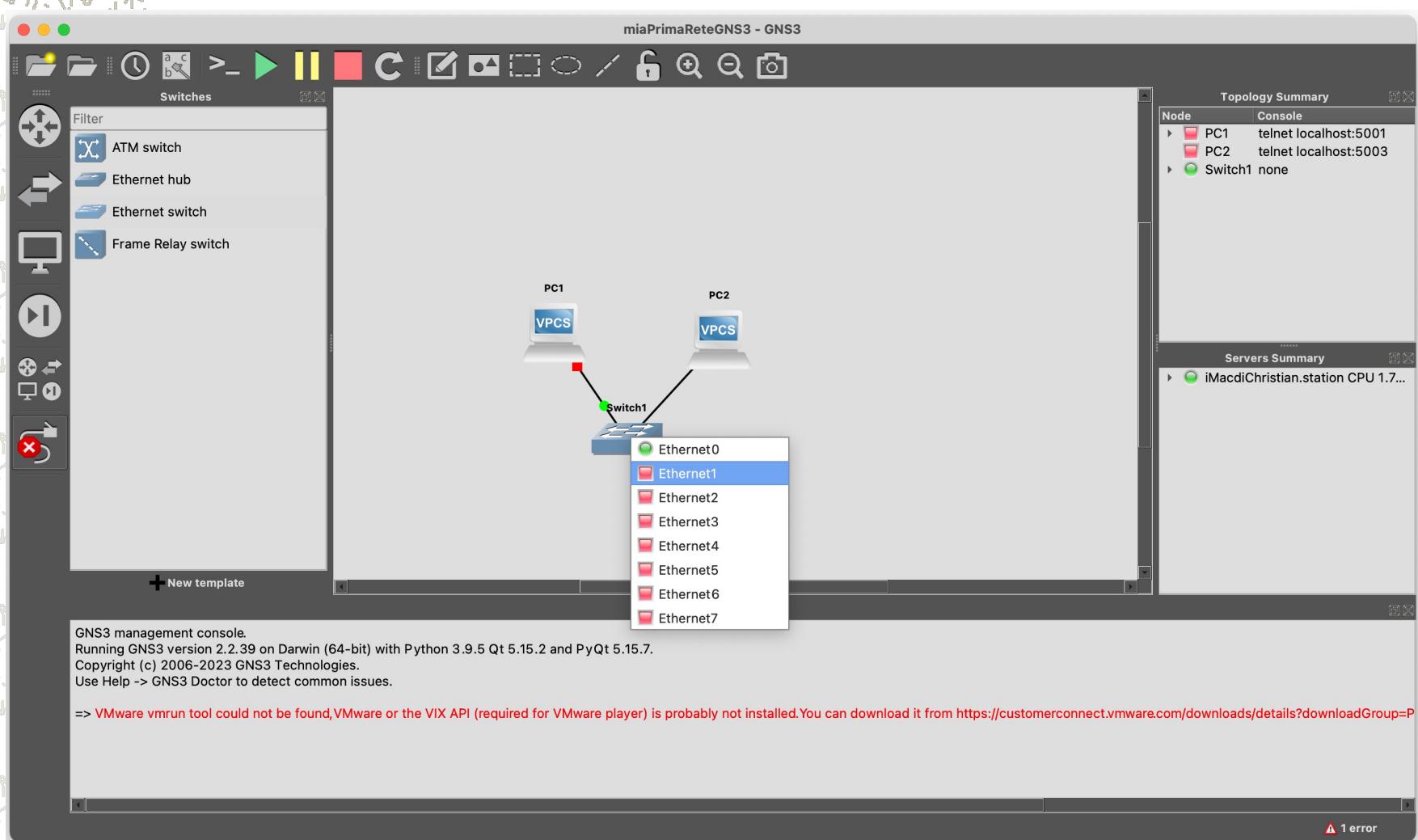
Ora, fai clic sul secondo dispositivo (Switch1) e seleziona l'interfaccia che desideri utilizzare per connetterti al primo dispositivo (PC1).

Esempio

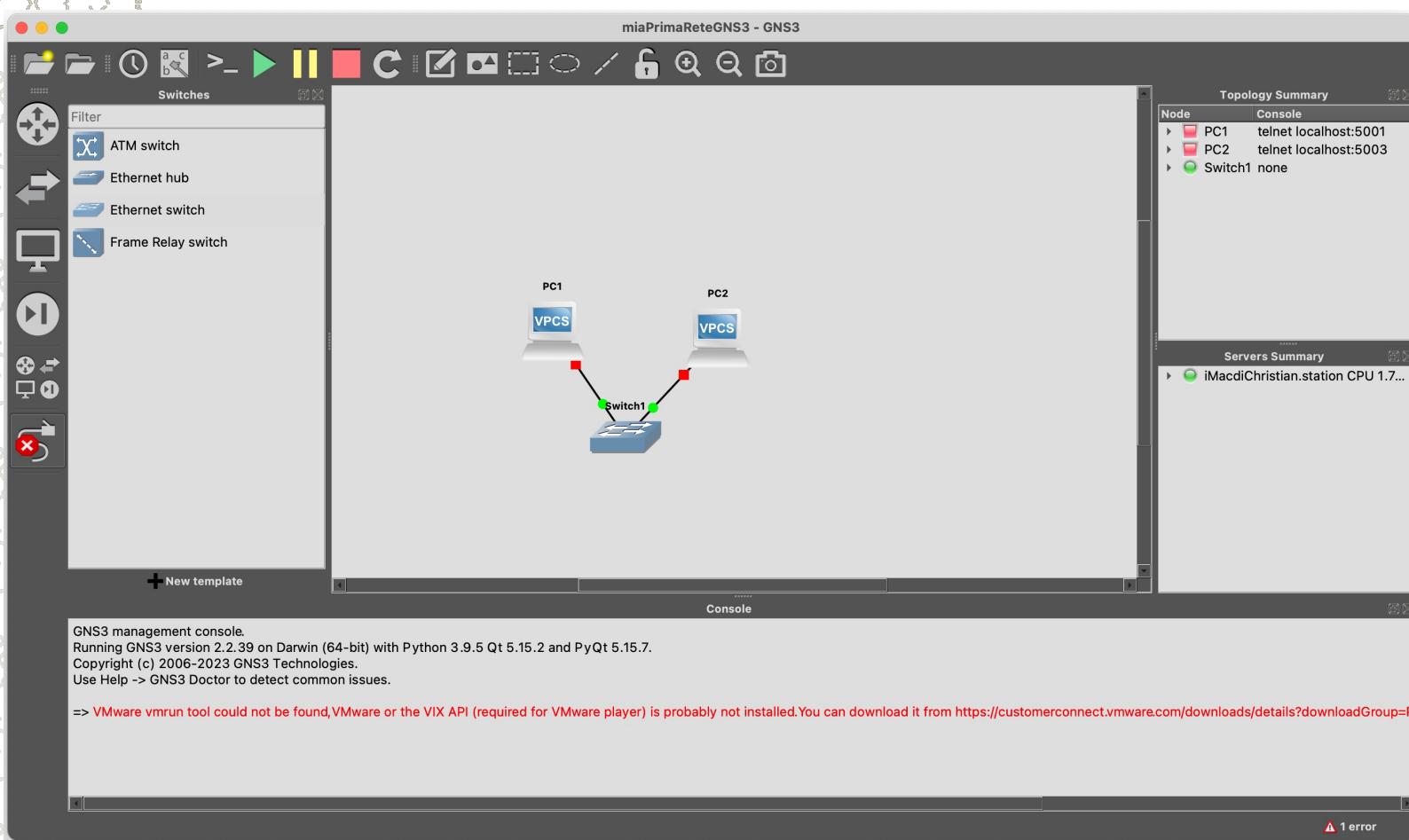


Ripetere l'operazione anche sul secondo dispositivo, facendo attenzione di non usare l'interfaccia già impegnata col primo dispositivo.

Esempio



Esempio

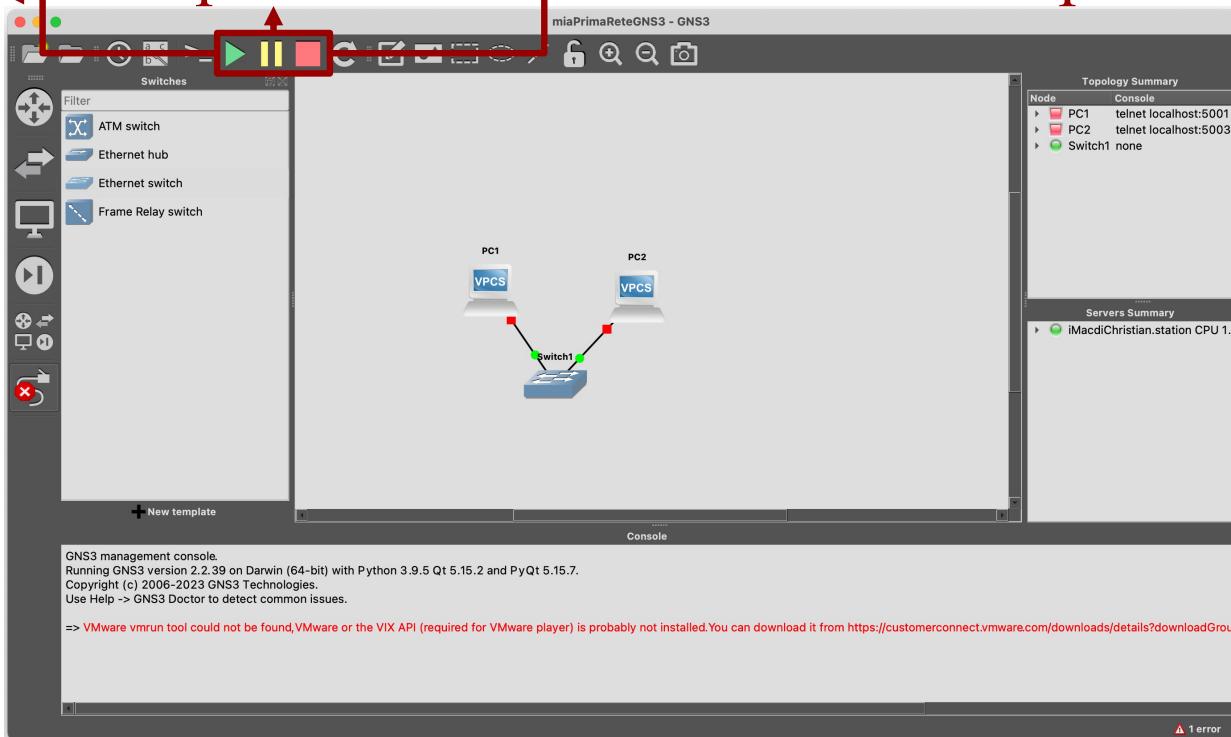


GNS3 aggiunge i PC nello stato spento mentre aggiunge gli interruttori nello stato acceso. Per mostrare lo stato di un dispositivo, utilizza tre colori diversi: Rosso per Spento, Verde per Acceso, e Giallo per Sospeso.

Esempio

Per modificare lo stato del dispositivo, selezionare il dispositivo e fare clic con il pulsante destro del mouse. Per modificare lo stato di tutti i dispositivi nell'area di lavoro, è possibile utilizzare le opzioni disponibili nella barra degli strumenti GNS3.

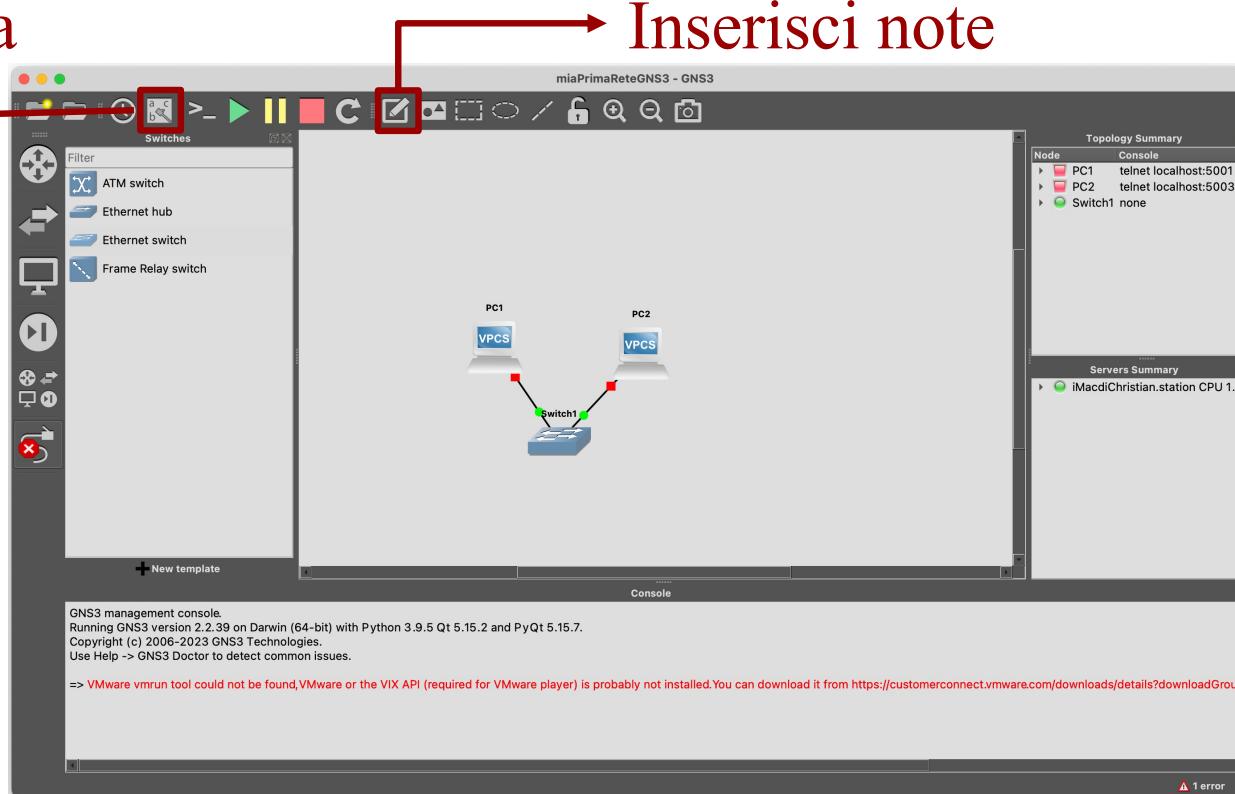
Avvia tutti i dispositivi Sospendi tutti Ferma tutti i dispositivi



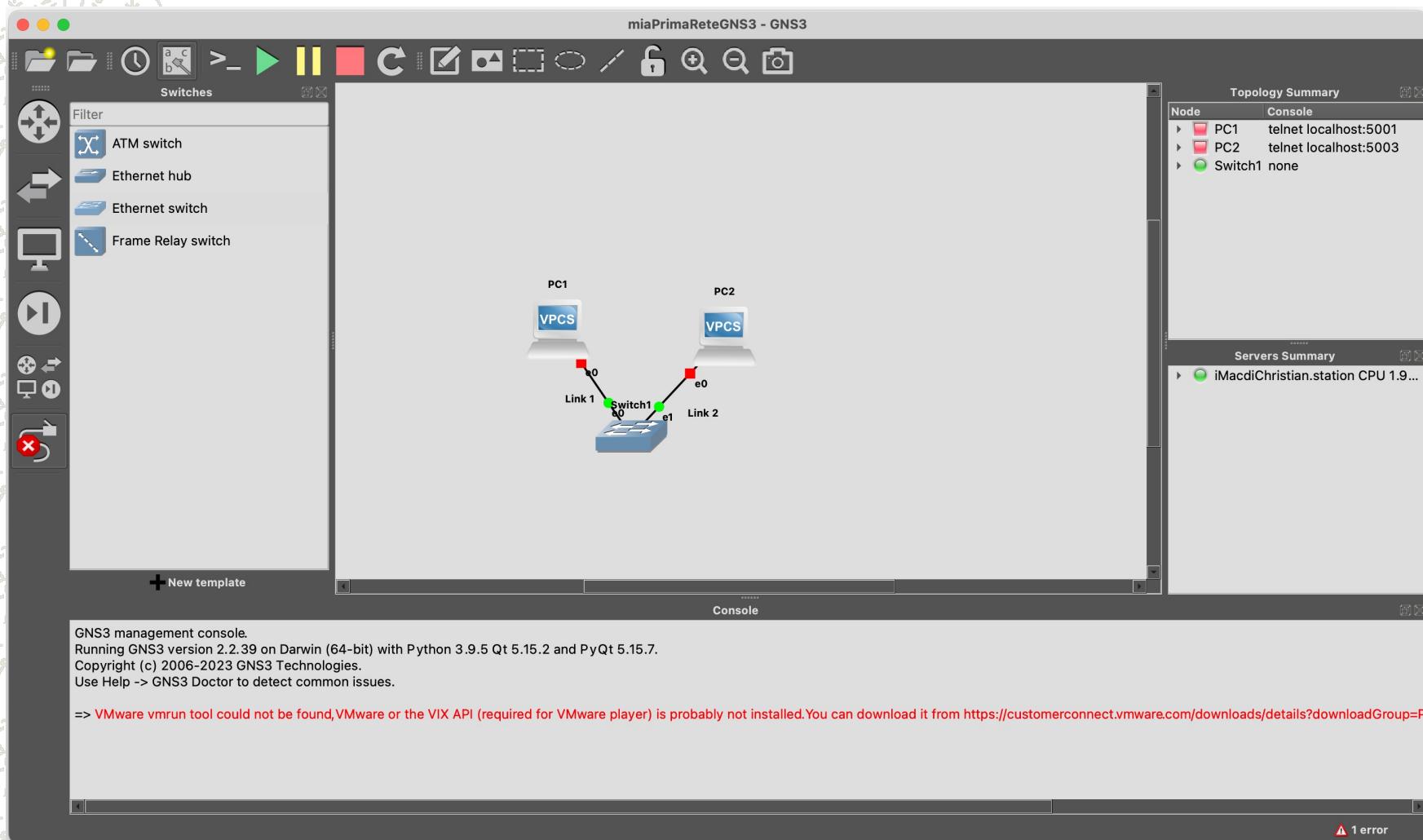
Esempio

Per impostazione predefinita, GNS3 non visualizza i nomi delle interfacce. Per mostrare i nomi delle interfacce, fare clic su Mostra/nascondi etichette interfaccia. È anche possibile aggiungere delle note.

Mostrare/Na
scondere
nomi delle
Interfacce



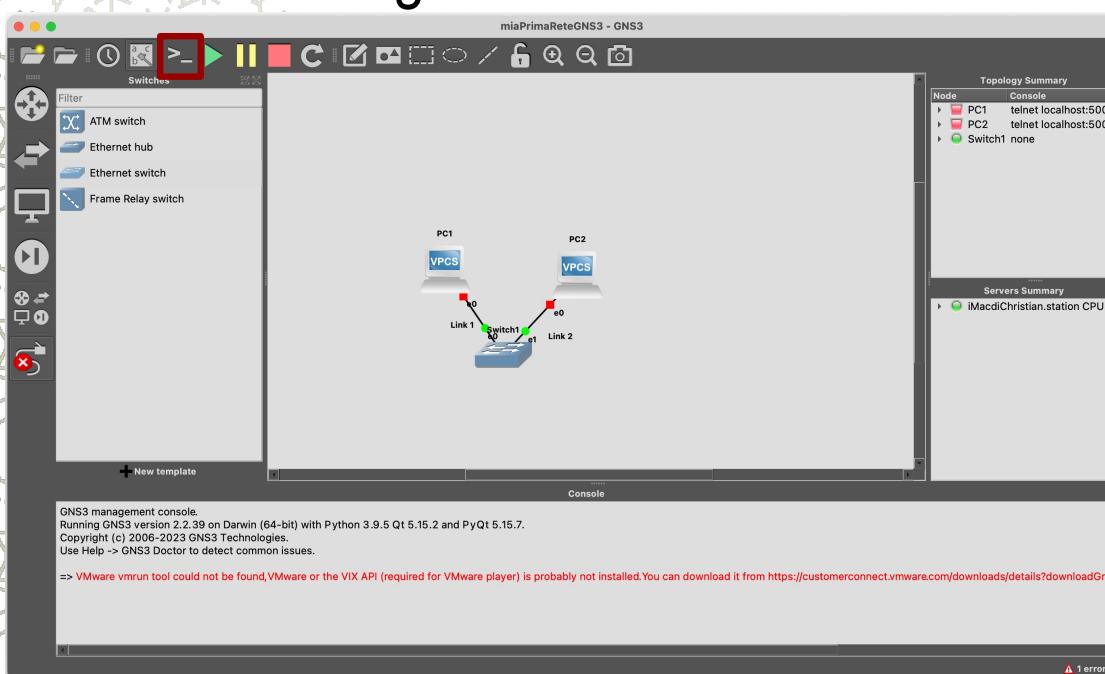
Esempio



Esempio

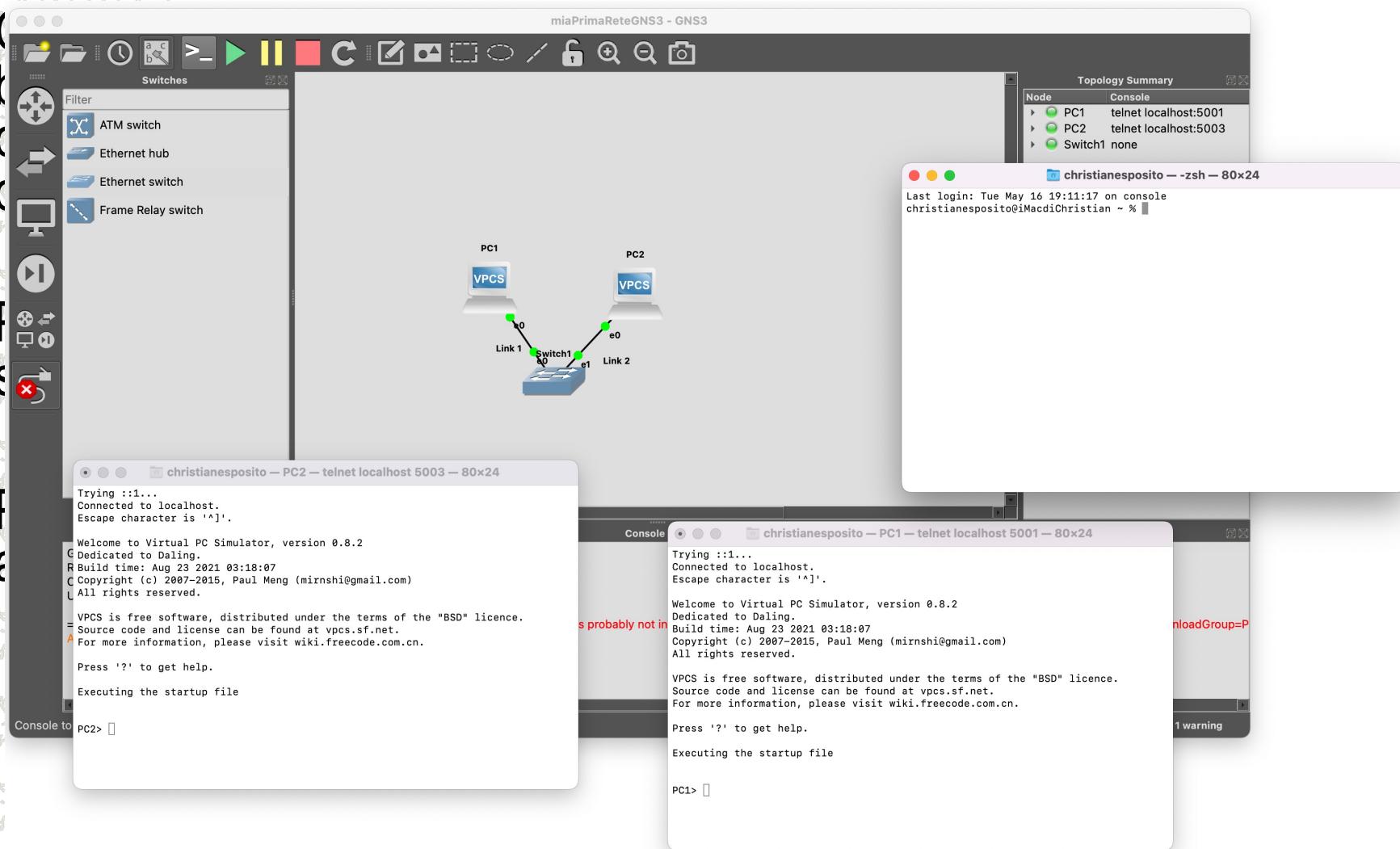
GNS3 non fornisce un'utilità grafica per configurare i dispositivi, ma bisogna utilizzare il prompt della console. Per aprire il prompt della console di un dispositivo, fare clic con il pulsante destro del mouse sul dispositivo e fare clic su Console.

Per aprire il prompt della console di tutti i dispositivi, fare clic su Console sulla barra degli strumenti GNS3.



Per avviare il prompt della console, tutti i dispositivi devono essere avviati.

Esempio



Esempio

Per assegnare un indirizzo IP al PC, utilizzare il seguente comando.

ip [indirizzo IP] [maschera di sottorete]

GNS3 controlla l'indirizzo IP fornito sulla rete. Se l'IP fornito non è in uso, assegna l'indirizzo IP fornito al PC.

```
christianesposito — PC2 — telnet localhost 5003 — 80x24
Trying ::1...
Connected to localhost.
Escape character is '^]'.

Welcome to Virtual PC Simulator, version 0.8.2
Dedicated to Daling.
Build time: Aug 23 2021 03:18:07
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

[PC2> ip 10.0.0.2 255.0.0.0
Checking for duplicate address...
PC2 : 10.0.0.2 255.0.0.0

PC2> ]
```

```
christianesposito — PC1 — telnet localhost 5001 — 80x24
Trying ::1...
Connected to localhost.
Escape character is '^]'.

Welcome to Virtual PC Simulator, version 0.8.2
Dedicated to Daling.
Build time: Aug 23 2021 03:18:07
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

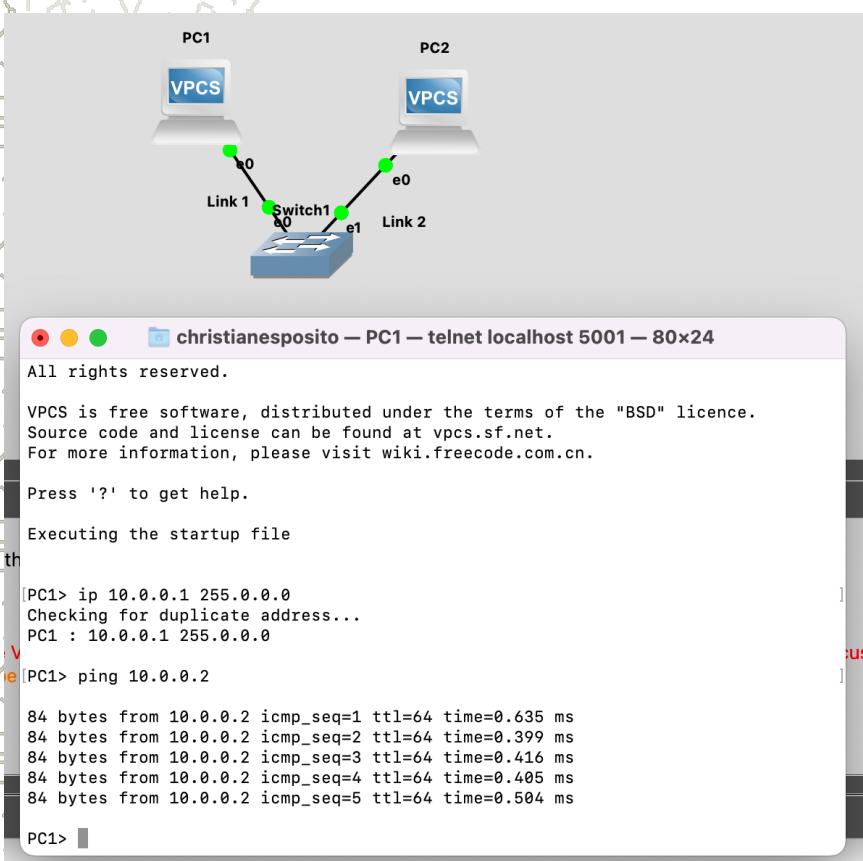
Executing the startup file

[PC1> ip 10.0.0.1 255.0.0.0
Checking for duplicate address...
PC1 : 10.0.0.1 255.0.0.0

PC1> ]
```

Esempio

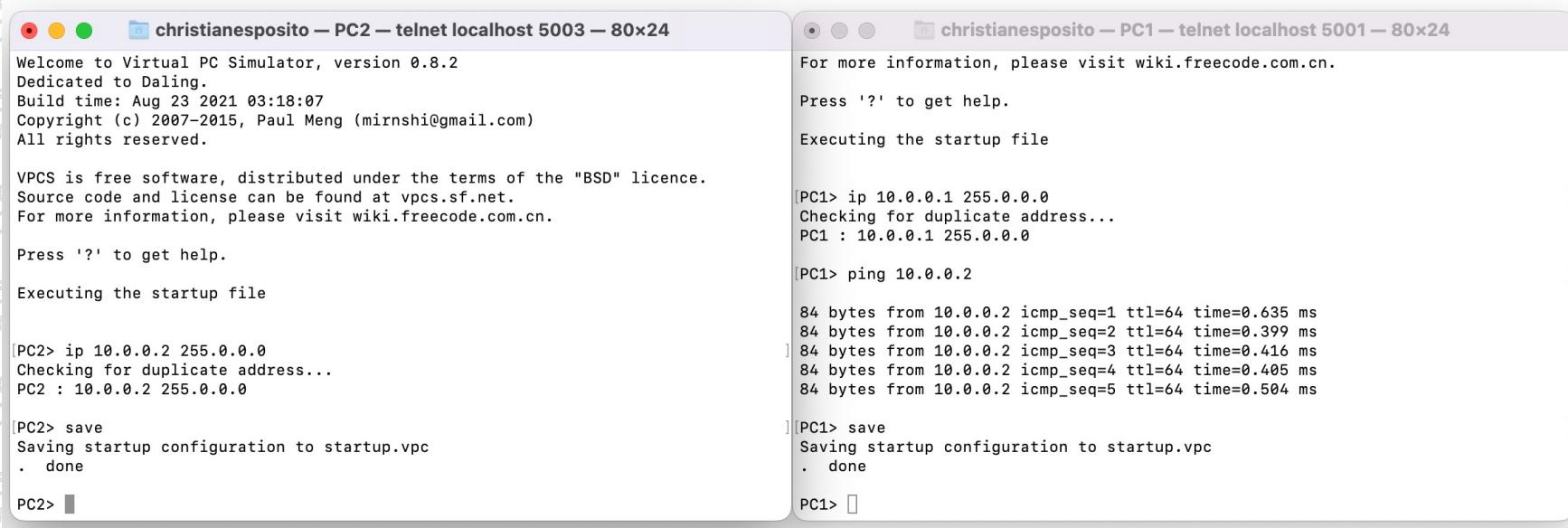
Per testare la connettività tra dispositivi, è possibile utilizzare il comando ping, che invia piccoli pacchetti di dati di prova al dispositivo di destinazione in sequenza. Se il dispositivo di destinazione è connesso al dispositivo di origine, il dispositivo di destinazione risponde.



Se vengono visualizzati messaggi di risposta nell'output di questo comando, verifica che il dispositivo di origine e il dispositivo di destinazione dispongano della connettività corretta.

Esempio

Per impostazione predefinita, GNS3 non salva la configurazione in esecuzione sul dispositivo. Significa che se viene ricaricato il progetto o il dispositivo, la configurazione in esecuzione sul dispositivo andrà persa. Per salvare la configurazione in esecuzione sul dispositivo, è necessario utilizzare la funzione di salvataggio del dispositivo. Per salvare la configurazione in esecuzione su un PC, utilizzare il comando save.



The screenshot shows two terminal windows side-by-side. The left window, titled "christianesposito — PC2 — telnet localhost 5003 — 80x24", displays the startup configuration for a Virtual PC Simulator version 0.8.2. It includes details about the build date (Aug 23 2021 03:18:07), copyright information (Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)), and a note that all rights reserved. It also mentions the "BSD" license and where to find source code and license. The user then types "ip 10.0.0.2 255.0.0.0" and "save". The right window, titled "christianesposito — PC1 — telnet localhost 5001 — 80x24", shows the configuration for PC1. It starts by executing the startup file, then performs an "ip 10.0.0.1 255.0.0.0" command, followed by a ping to 10.0.0.2. Finally, it saves the configuration with the "save" command. Both windows show the configuration being saved to "startup.vpc".

```
christianesposito — PC2 — telnet localhost 5003 — 80x24
Welcome to Virtual PC Simulator, version 0.8.2
Dedicated to Daling.
Build time: Aug 23 2021 03:18:07
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

[PC2> ip 10.0.0.2 255.0.0.0
Checking for duplicate address...
PC2 : 10.0.0.2 255.0.0.0

[PC2> save
Saving startup configuration to startup.vpc
. done

PC2>

christianesposito — PC1 — telnet localhost 5001 — 80x24
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

[PC1> ip 10.0.0.1 255.0.0.0
Checking for duplicate address...
PC1 : 10.0.0.1 255.0.0.0

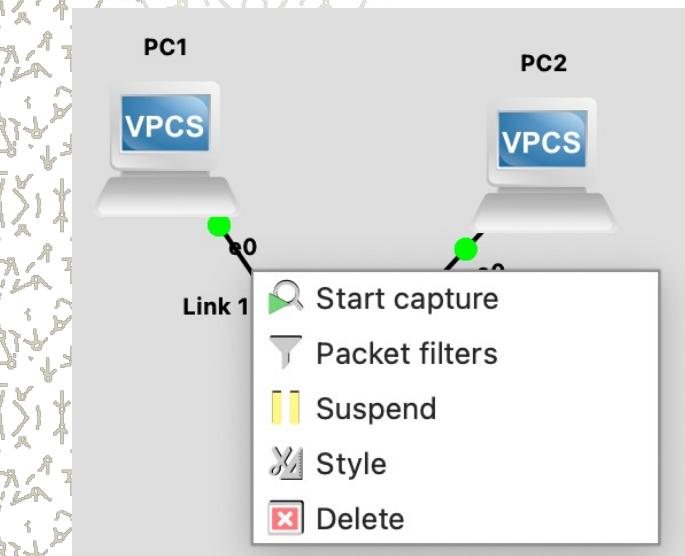
[PC1> ping 10.0.0.2
84 bytes from 10.0.0.2 icmp_seq=1 ttl=64 time=0.635 ms
84 bytes from 10.0.0.2 icmp_seq=2 ttl=64 time=0.399 ms
84 bytes from 10.0.0.2 icmp_seq=3 ttl=64 time=0.416 ms
84 bytes from 10.0.0.2 icmp_seq=4 ttl=64 time=0.405 ms
84 bytes from 10.0.0.2 icmp_seq=5 ttl=64 time=0.584 ms

[PC1> save
Saving startup configuration to startup.vpc
. done

PC1>
```

Esempio

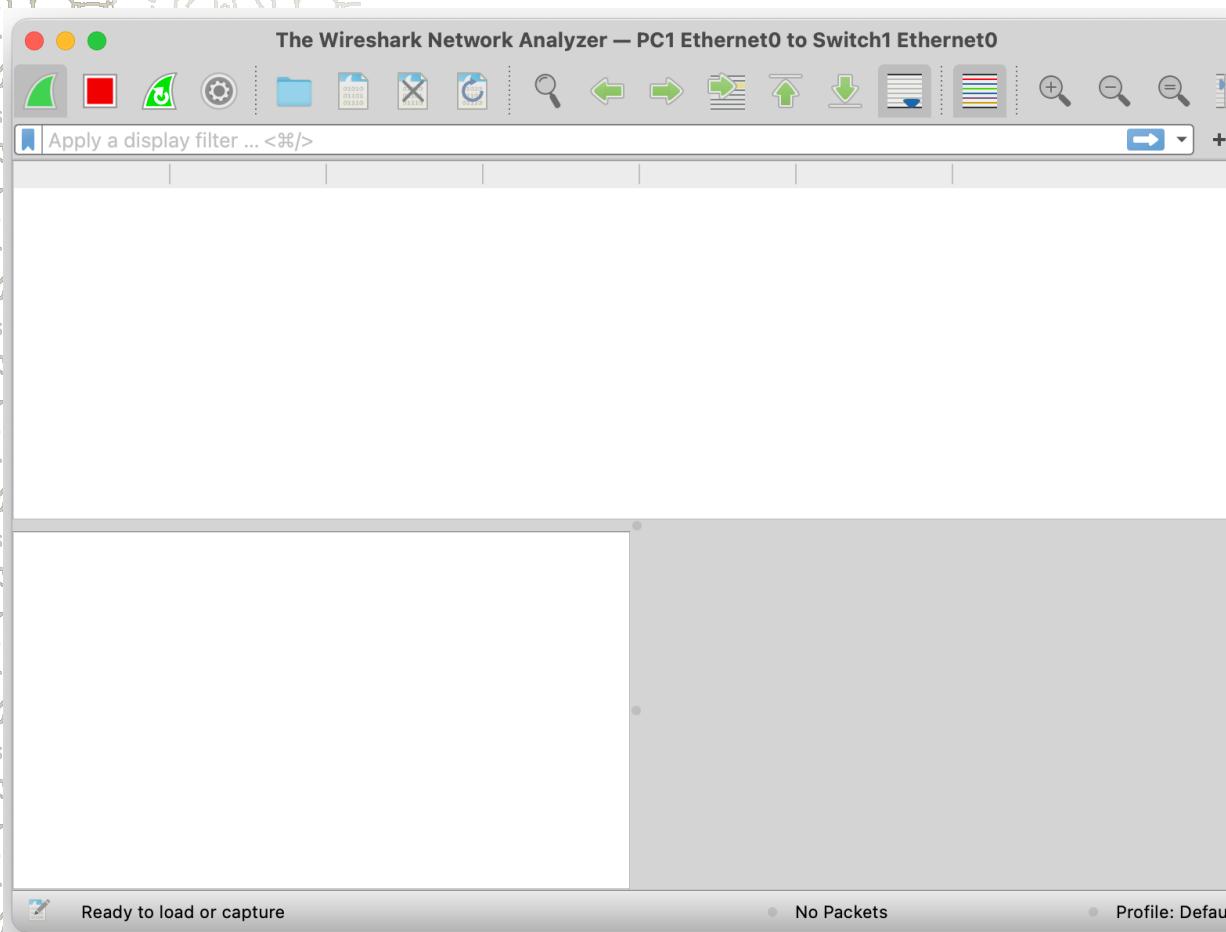
È possibile utilizzare l'acquisizione dei pacchetti sul diagramma di rete del progetto GNS3. L'acquisizione dei pacchetti consente di visualizzare in dettaglio il formato e il flusso comportamentale di vari protocolli.



Per avviare, selezionare un link, tasto destro e cliccare su Start capture. Per poter utilizzare questa operazione è necessario installare Wireshark.

Esempio

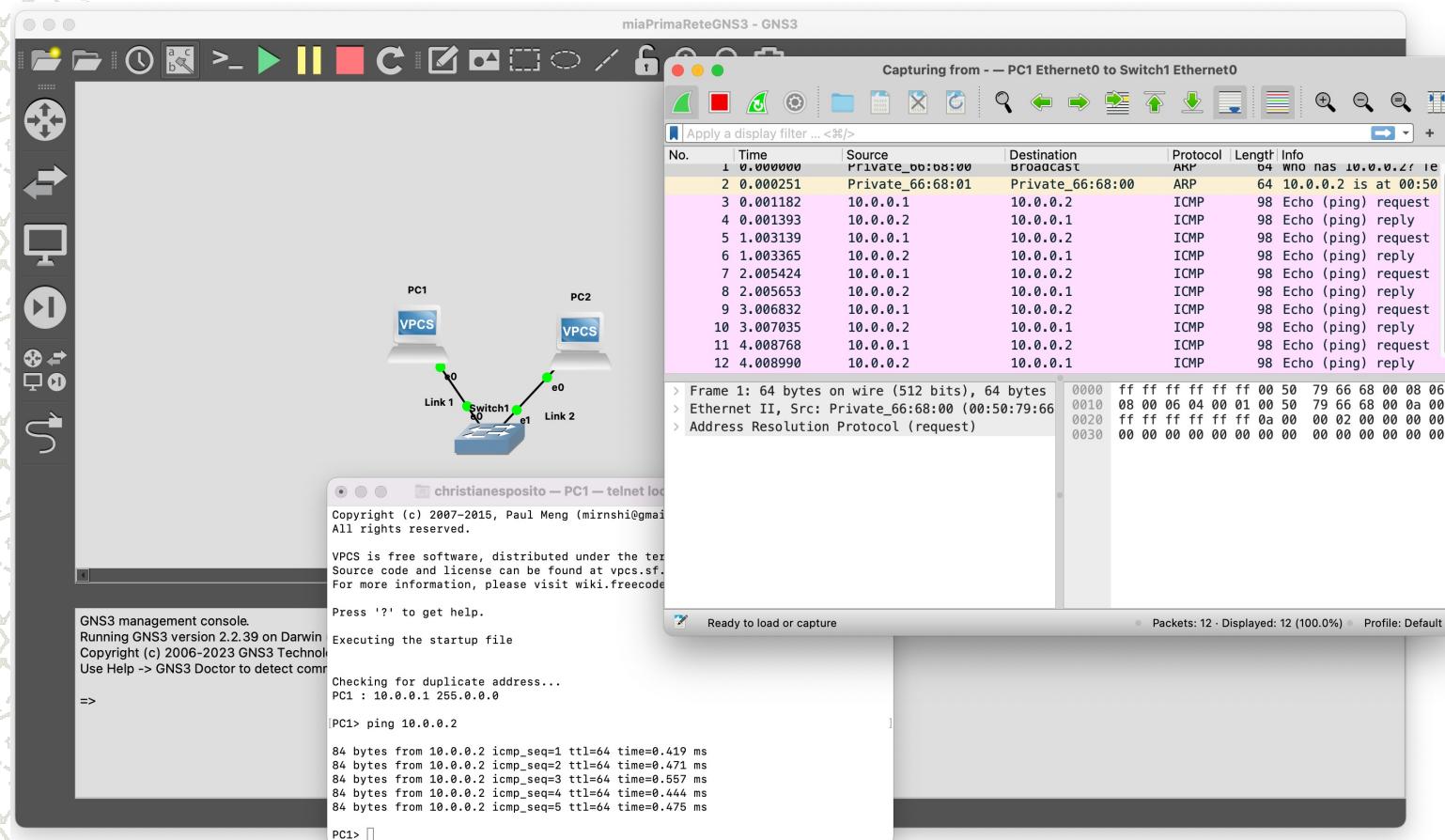
Wireshark sarà avviato indicando le interfacce del link su cui si stanno catturando i pacchetti.



A questo punto riproviamo ad eseguire il comando ping sulla console del PC1.

Esempio

Su Wireshark verranno visualizzati i vari pacchetti man a mano che saranno generati dal comando ping.



Esempio

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.2? Tell 10.0.0.1
2	0.000251	Private_66:68:01	Private_66:68:00	ARP	64	10.0.0.2 is at 00:50:79:66:68:01
3	0.001182	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x21cd, seq=1/256, ttl=64 (reply in 4)
4	0.001393	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x21cd, seq=1/256, ttl=64 (request in 3)
5	1.003139	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x22cd, seq=2/512, ttl=64 (reply in 6)
6	1.003365	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x22cd, seq=2/512, ttl=64 (request in 5)
7	2.005424	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x23cd, seq=3/768, ttl=64 (reply in 8)
8	2.005653	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x23cd, seq=3/768, ttl=64 (request in 7)
9	3.006832	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x24cd, seq=4/1024, ttl=64 (reply in 10)
10	3.007035	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x24cd, seq=4/1024, ttl=64 (request in 9)
11	4.008768	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x25cd, seq=5/1280, ttl=64 (reply in 12)
12	4.008990	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x25cd, seq=5/1280, ttl=64 (request in 11)

```
> Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)
```

0000	ff	ff	ff	ff	ff	ff	00	50	79	66	68	00	08	06	00	01P yfh.....
0010	08	00	06	04	00	01	00	50	79	66	68	00	0a	00	00	01P yfh.....
0020	ff	ff	ff	ff	ff	ff	0a	00	00	02	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Il primo pacchetto catturato è in broadcast, ed è del protocollo ARP e rappresenta una ARP Request, al fine di ottenere l'indirizzo fisico della macchina che ha l'indirizzo logico passato al comando ping. Nel campo info abbiamo il contenuto dell'ARP Request.

Nelle aree al di sotto della lista di pacchetti abbiamo le informazioni di dettaglio del pacchetto (a sinistra) e il contenuto dello stesso (a destra).

Esempio

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.2? Tell 10.0.0.1
2	0.000251	Private_66:68:01	Private_66:68:00	ARP	64	10.0.0.2 is at 00:50:79:66:68:01
3	0.001182	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x21cd, seq=1/256, ttl=64 (reply in 4)
4	0.001393	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x21cd, seq=1/256, ttl=64 (request in 3)
5	1.003139	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x22cd, seq=2/512, ttl=64 (reply in 6)
6	1.003365	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x22cd, seq=2/512, ttl=64 (request in 5)
7	2.005424	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x23cd, seq=3/768, ttl=64 (reply in 8)
8	2.005653	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x23cd, seq=3/768, ttl=64 (request in 7)
9	3.006832	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x24cd, seq=4/1024, ttl=64 (reply in 10)
10	3.007035	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x24cd, seq=4/1024, ttl=64 (request in 9)
11	4.008768	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x25cd, seq=5/1280, ttl=64 (reply in 12)
12	4.008990	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x25cd, seq=5/1280, ttl=64 (request in 11)

```
> Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request)
```

0000 ff ff ff ff ff ff 00 50 79 66 68 00 08 06 00 01	P yfh.....
0010 08 00 06 04 00 01 00 50 79 66 68 00 0a 00 00 01	P yfh.....
0020 ff ff ff ff ff 0a 00 00 02 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Il secondo pacchetto è un ARP Response da parte del secondo dispositivo che comunica il suo indirizzo fisico.

Esempio

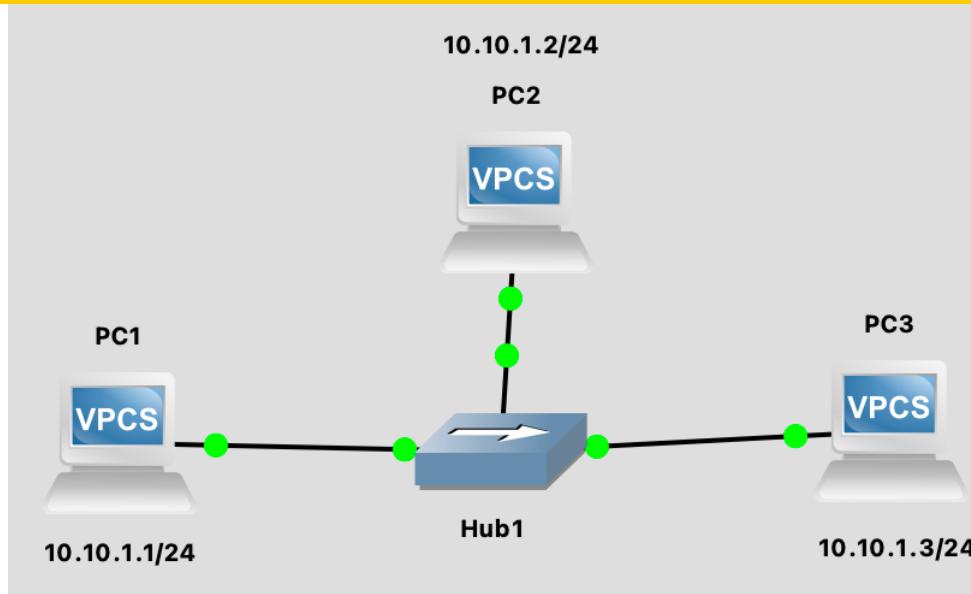
Successivamente si ha una sequenza della coppia Echo request e Echo reply del protocollo ICMP.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.2? Tell 10.0.0.1
2	0.000251	Private_66:68:01	Private_66:68:00	ARP	64	10.0.0.2 is at 00:50:79:66:68:01
3	0.001182	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x21cd, seq=1/256, ttl=64 (reply in 4)
4	0.001393	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x21cd, seq=1/256, ttl=64 (request in 3)
5	1.003139	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x22cd, seq=2/512, ttl=64 (reply in 6)
6	1.003365	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x22cd, seq=2/512, ttl=64 (request in 5)
7	2.005424	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x23cd, seq=3/768, ttl=64 (reply in 8)
8	2.005653	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x23cd, seq=3/768, ttl=64 (request in 7)
9	3.006832	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x24cd, seq=4/1024, ttl=64 (reply in 10)
10	3.007035	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x24cd, seq=4/1024, ttl=64 (request in 9)
11	4.008768	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0x25cd, seq=5/1280, ttl=64 (reply in 12)
12	4.008990	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0x25cd, seq=5/1280, ttl=64 (request in 11)

> Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Private_66:68:01 (00:50:79:66:68:01)
> Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
> Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xfe3d [correct]
[Checksum Status: Good]
Identifier (BE): 8653 (0x21cd)
Identifier (LE): 52513 (0xcd21)
Sequence Number (BE): 1 (0x0001)
Sequence Number (LE): 256 (0x0100)
[Response frame: 4]
> Data (56 bytes)

0000 00 50 79 66 68 01 00 50 79 66 68 00 08 00 45 00 ·Pyfh·P yfh··E·
0010 00 54 cd 21 00 00 40 01 99 85 0a 00 00 01 0a 00 ·T·!·@·
0020 00 02 08 00 fe 3d 21 cd 00 01 08 09 0a 0b 0c 0d ···=!·
0030 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d ···· ···· ···· ····
0040 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d ··· !#\$% &'()*,,-
0050 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d ./012345 6789:;<=·
0060 3e 3f >?

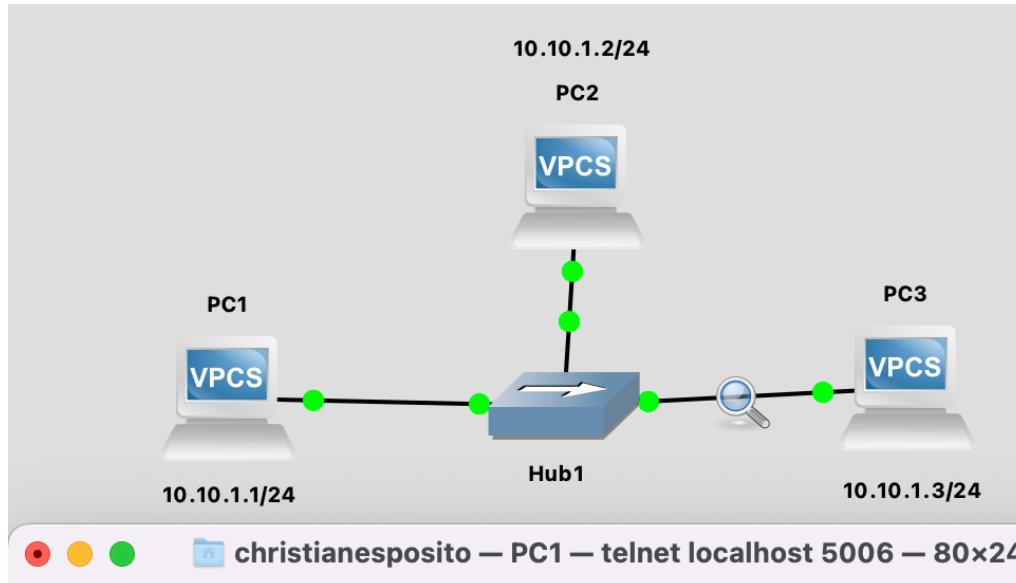
Esercizio - 1



Ricreare la seguente topologia in GNS3. Avviare tutti i dispositivi e catturare i pacchetti tra Hub1 e PC3. Aprire il terminale per PC1 ed eseguire i seguenti comandi:

- da PC1 ping PC2 (arresto del ping dopo aver ricevuto la seconda risposta),
- da PC1 ping PC3 (arresto del ping dopo aver ricevuto la seconda risposta), quindi
- dal PC1 eseguire il ping dell'indirizzo IP non assegnato 10.10.1.4 e verificare che non venga ricevuta alcuna risposta.

Esercizio - 1



christianesposito — PC1 — telnet localhost 5006 — 80x24

```
Trying ::1...
Connected to localhost.
Escape character is '^['.

[PC1> ping 10.10.1.2

84 bytes from 10.10.1.2 icmp_seq=1 ttl=64 time=0.474 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=64 time=0.446 ms
^C
[PC1> ping 10.10.1.3

84 bytes from 10.10.1.3 icmp_seq=1 ttl=64 time=0.428 ms
84 bytes from 10.10.1.3 icmp_seq=2 ttl=64 time=0.507 ms
^C
[PC1> ping 10.10.1.4

host (10.10.1.4) not reachable

PC1>
```

Esercizio - 2

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
2	1.001294	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
3	2.001323	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
4	26.932357	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
5	27.933124	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
6	28.933949	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
7	48.308337	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
8	49.308849	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
9	50.309167	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
10	81.011332	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.2? Tell 10.10.1.1
11	81.011560	Private_66:68:01	Private_66:68:00	ARP	64	10.10.1.2 is at 00:50:79:66:68:01
12	81.012596	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7dd1, seq=1/256, ttl=64 (reply in 13)
13	81.012829	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7dd1, seq=1/256, ttl=64 (request in 12)
14	82.014044	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7ed1, seq=2/512, ttl=64 (reply in 15)
15	82.014239	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7ed1, seq=2/512, ttl=64 (request in 14)
16	91.935242	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.1
17	91.935470	Private_66:68:02	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:02
18	91.936419	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x88d1, seq=1/256, ttl=64 (reply in 19)
19	91.936624	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x88d1, seq=1/256, ttl=64 (request in 18)
20	92.938057	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x89d1, seq=2/512, ttl=64 (reply in 21)
21	92.938297	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x89d1, seq=2/512, ttl=64 (request in 20)
22	99.845634	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
23	100.847058	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
24	101.847447	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1

Le configurazioni iniziali del comando ip sono inviate in broadcast.

Inoltre, i pacchetti tra PC1 e PC2 vengono trasmessi anche sul collegamento che collega PC3 all'hub. Tutti questi pacchetti possono essere catturati da Wireshark.

Esercizio - 1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
2	1.001294	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
3	2.001323	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
4	26.932357	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
5	27.933124	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
6	28.933949	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
7	48.308337	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
8	49.308849	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
9	50.309167	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
10	81.011332	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.2? Tell 10.10.1.1
11	81.011560	Private_66:68:01	Private_66:68:00	ARP	64	10.10.1.2 is at 00:50:79:66:68:01
12	81.012596	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7dd1, seq=1/256, ttl=64 (reply in 13)
13	81.012829	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7dd1, seq=1/256, ttl=64 (request in 12)
14	82.014044	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7ed1, seq=2/512, ttl=64 (reply in 15)
15	82.014239	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7ed1, seq=2/512, ttl=64 (request in 14)
16	91.935242	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.1
17	91.935470	Private_66:68:02	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:02
18	91.936419	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x88d1, seq=1/256, ttl=64 (reply in 19)
19	91.936624	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x88d1, seq=1/256, ttl=64 (request in 18)
20	92.938057	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x89d1, seq=2/512, ttl=64 (reply in 21)
21	92.938297	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x89d1, seq=2/512, ttl=64 (request in 20)
22	99.845634	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
23	100.847058	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
24	101.847447	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1

Si noti che PC1, prima di trasmettere una richiesta echo ICMP a PC2 [frame #12], deve apprendere l'indirizzo MAC di PC2 emettendo una richiesta broadcast ARP (Who has 10.10.1.2? Tell 10.10.1.1) [frame #10]. PC1 apprende che l'indirizzo IP 10.10.1.2 è associato all'indirizzo MAC 00:50:79:66:68:01 tramite la risposta ARP di PC2 [frame #11].

Esercizio - 1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
2	1.001294	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
3	2.001323	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
4	26.932357	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
5	27.933124	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
6	28.933949	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
7	48.308337	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
8	49.308849	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
9	50.309167	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
10	81.011332	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.2? Tell 10.10.1.1
11	81.011560	Private_66:68:01	Private_66:68:00	ARP	64	10.10.1.2 is at 00:50:79:66:68:01
12	81.012596	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7dd1, seq=1/256, ttl=64 (reply in 13)
13	81.012829	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7dd1, seq=1/256, ttl=64 (request in 12)
14	82.014044	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7ed1, seq=2/512, ttl=64 (reply in 15)
15	82.014239	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7ed1, seq=2/512, ttl=64 (request in 14)
16	91.935242	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.1
17	91.935470	Private_66:68:02	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:02
18	91.936419	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x88d1, seq=1/256, ttl=64 (reply in 19)
19	91.936624	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x88d1, seq=1/256, ttl=64 (request in 18)
20	92.938057	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x89d1, seq=2/512, ttl=64 (reply in 21)
21	92.938297	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x89d1, seq=2/512, ttl=64 (request in 20)
22	99.845634	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
23	100.847058	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
24	101.847447	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1

PC2, invece, può inviare direttamente una risposta echo ICMP a PC1 [frame #13], poiché ha appreso l'indirizzo MAC di PC1 direttamente dalla precedente richiesta ARP. Allo stesso modo, PC1, prima di trasmettere una richiesta echo ICMP a PC3 [frame #18], deve apprendere l'indirizzo MAC di PC3 emettendo una richiesta broadcast ARP (Who has 10.10.1.2? Tell 10.10.1.1) [frame #16].

Esercizio - 1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
2	1.001294	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
3	2.001323	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
4	26.932357	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
5	27.933124	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
6	28.933949	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
7	48.308337	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
8	49.308849	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
9	50.309167	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
10	81.011332	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.2? Tell 10.10.1.1
11	81.011560	Private_66:68:01	Private_66:68:00	ARP	64	10.10.1.2 is at 00:50:79:66:68:01
12	81.012596	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7dd1, seq=1/256, ttl=64 (reply in 13)
13	81.012829	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7dd1, seq=1/256, ttl=64 (request in 12)
14	82.014044	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7ed1, seq=2/512, ttl=64 (reply in 15)
15	82.014239	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7ed1, seq=2/512, ttl=64 (request in 14)
16	91.935242	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.1
17	91.935470	Private_66:68:02	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:02
18	91.936419	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x88d1, seq=1/256, ttl=64 (reply in 19)
19	91.936624	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x88d1, seq=1/256, ttl=64 (request in 18)
20	92.938057	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x89d1, seq=2/512, ttl=64 (reply in 21)
21	92.938297	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x89d1, seq=2/512, ttl=64 (request in 20)
22	99.845634	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
23	100.847058	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
24	101.847447	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1

PC1 apprende che l'indirizzo IP 10.10.1.3 è associato all'indirizzo MAC 00:50:79:66:68:02 tramite la risposta ARP di PC3 [frame #17].

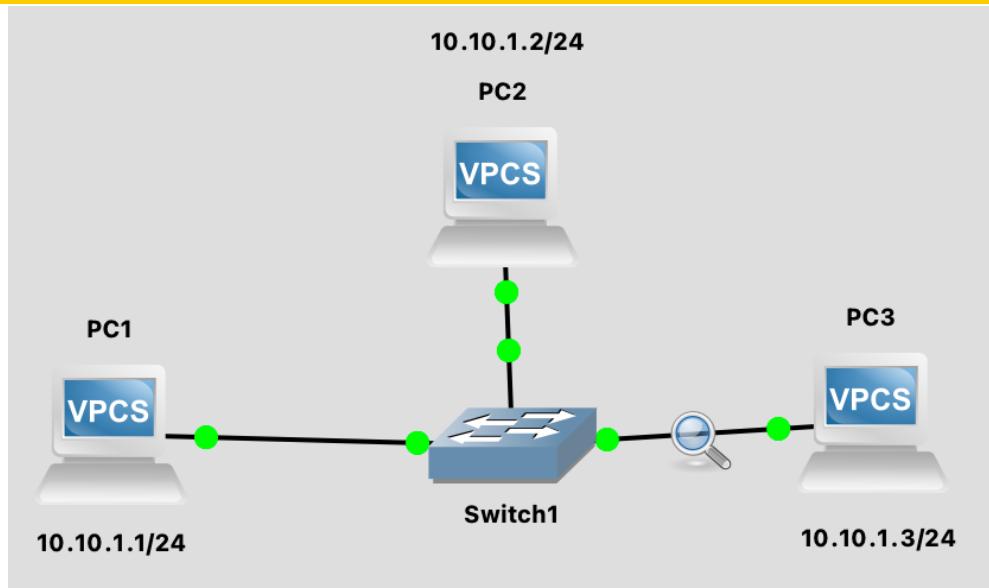
PC3, invece, può inviare direttamente una risposta echo ICMP a PC1 [frame #10], poiché ha appreso l'indirizzo MAC di PC1 direttamente dalla precedente richiesta ARP.

Esercizio - 1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
2	1.001294	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
3	2.001323	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
4	26.932357	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
5	27.933124	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
6	28.933949	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
7	48.308337	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
8	49.308849	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
9	50.309167	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
10	81.011332	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.2? Tell 10.10.1.1
11	81.011560	Private_66:68:01	Private_66:68:00	ARP	64	10.10.1.2 is at 00:50:79:66:68:01
12	81.012596	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7dd1, seq=1/256, ttl=64 (reply in 13)
13	81.012829	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7dd1, seq=1/256, ttl=64 (request in 12)
14	82.014044	10.10.1.1	10.10.1.2	ICMP	98	Echo (ping) request id=0x7ed1, seq=2/512, ttl=64 (reply in 15)
15	82.014239	10.10.1.2	10.10.1.1	ICMP	98	Echo (ping) reply id=0x7ed1, seq=2/512, ttl=64 (request in 14)
16	91.935242	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.1
17	91.935470	Private_66:68:02	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:02
18	91.936419	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x88d1, seq=1/256, ttl=64 (reply in 19)
19	91.936624	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x88d1, seq=1/256, ttl=64 (request in 18)
20	92.938057	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x89d1, seq=2/512, ttl=64 (reply in 21)
21	92.938297	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x89d1, seq=2/512, ttl=64 (request in 20)
22	99.845634	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
23	100.847058	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
24	101.847447	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1

Quando esegue l'ultimo ping, PC1 tenta di scoprire l'indirizzo MAC associato all'indirizzo IP 10.10.1.4 inviando una richiesta ARP broadcast (Who has 10.10.1.4? Tell 10.10.1.1) [frame #22]. Dopo un timeout di 1 secondo, PC1 invia la stessa richiesta ARP una seconda e una terza volta. Non ricevendo alcuna risposta, PC1 conclude che l'indirizzo 10.10.1.4 è irraggiungibile.

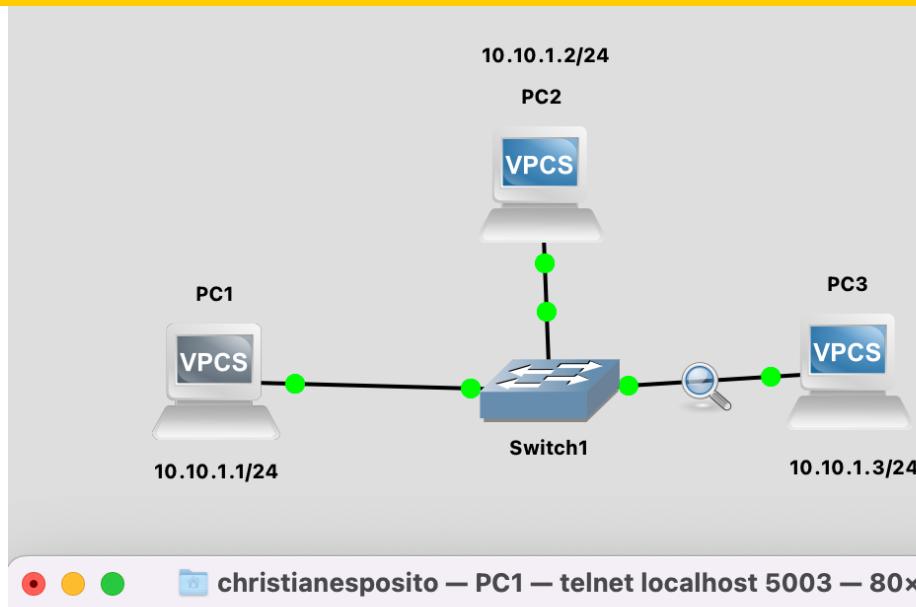
Esercizio - 2



Ricreare la seguente topologia in GNS3. Avviare tutti i dispositivi e catturare i pacchetti tra Switch1 e PC3. Aprire il terminale per PC1 ed eseguire i seguenti comandi:

- da PC1 ping PC2 (arresto del ping dopo aver ricevuto la seconda risposta),
- da PC1 ping PC3 (arresto del ping dopo aver ricevuto la seconda risposta), quindi
- dal PC1 eseguire il ping dell'indirizzo IP non assegnato 10.10.1.4 e verificare che non venga ricevuta alcuna risposta.

Esercizio - 2



christianesposito — PC1 — telnet localhost 5003 — 80x24

Press '?' to get help.

Executing the startup file

Checking for duplicate address...
PC1 : 10.10.1.1 255.255.255.0

[PC1]> ping 10.10.1.2

```
84 bytes from 10.10.1.2 icmp_seq=1 ttl=64 time=0.688 ms
84 bytes from 10.10.1.2 icmp_seq=2 ttl=64 time=0.447 ms
^C
```

[PC1]> ping 10.10.1.3

```
84 bytes from 10.10.1.3 icmp_seq=1 ttl=64 time=0.280 ms
84 bytes from 10.10.1.3 icmp_seq=2 ttl=64 time=0.425 ms
^C
```

[PC1]> ping 10.10.1.4

```
host (10.10.1.4) not reachable
```

Esercizio - 2

No.	Time	Source	Destination	Protocol	Length	Info
8	1.059593	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
9	1.063368	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
10	2.057775	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
11	2.060241	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
12	2.064015	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
13	15.724506	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.2? Tell 10.10.1.1
14	25.311897	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.1
15	25.312056	Private_66:68:02	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:02
16	25.313027	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x3eda, seq=1/256, ttl=64 (reply in 17)
17	25.313163	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x3eda, seq=1/256, ttl=64 (request in 16)
18	26.315314	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x3fda, seq=2/512, ttl=64 (reply in 19)
19	26.315492	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x3fda, seq=2/512, ttl=64 (request in 18)
20	33.344423	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
21	34.345592	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
22	35.346686	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1

Si noti che ora Wireshark in esecuzione sul collegamento di PC3 verso lo switch può vedere solo due tipi di pacchetti:

- frame broadcast, come le richieste ARP [frame #13, #14, #20, #21 e #22];
- frame unicast trasmessi da o verso il MAC del PC3, ovvero frame con un MAC sorgente o destinazione 00:50:79:66:68:02 [frame #15, #16, #17, #18 e #19].

Tutti gli altri frame catturati da Wireshark nell'esercizio precedente vengono effettivamente trasmessi e scambiati da PC1 e PC2 ma non vengono visti da Wireshark.

Esercizio - 2

No.	Time	Source	Destination	Protocol	Length	Info
8	1.059593	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
9	1.063368	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
10	2.057775	Private_66:68:00	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.1 (Request)
11	2.060241	Private_66:68:01	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.2 (Request)
12	2.064015	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.10.1.3 (Request)
13	15.724506	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.2? Tell 10.10.1.1
14	25.311897	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.1
15	25.312056	Private_66:68:02	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:02
16	25.313027	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x3eda, seq=1/256, ttl=64 (reply in 17)
17	25.313163	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x3eda, seq=1/256, ttl=64 (request in 16)
18	26.315314	10.10.1.1	10.10.1.3	ICMP	98	Echo (ping) request id=0x3fda, seq=2/512, ttl=64 (reply in 19)
19	26.315492	10.10.1.3	10.10.1.1	ICMP	98	Echo (ping) reply id=0x3fda, seq=2/512, ttl=64 (request in 18)
20	33.344423	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
21	34.345592	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1
22	35.346686	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.4? Tell 10.10.1.1

Questo è esattamente coerente con il comportamento di uno switch Ethernet. Questi pacchetti apparentemente "mancanti" includono:

- Risposta ARP trasmessa da PC2 a PC1;
- Richieste echo ICMP trasmesse da PC1 a PC2;
- Risposte echo ICMP trasmesse da PC2 a PC1.

GNS3 VM

Per alcune topologie di rete è necessario adoperare GNS3 VM per l'emulazione di appositi dispositivi di rete. A tal file è necessario scaricare GNS3VM per VMware ed importarlo.

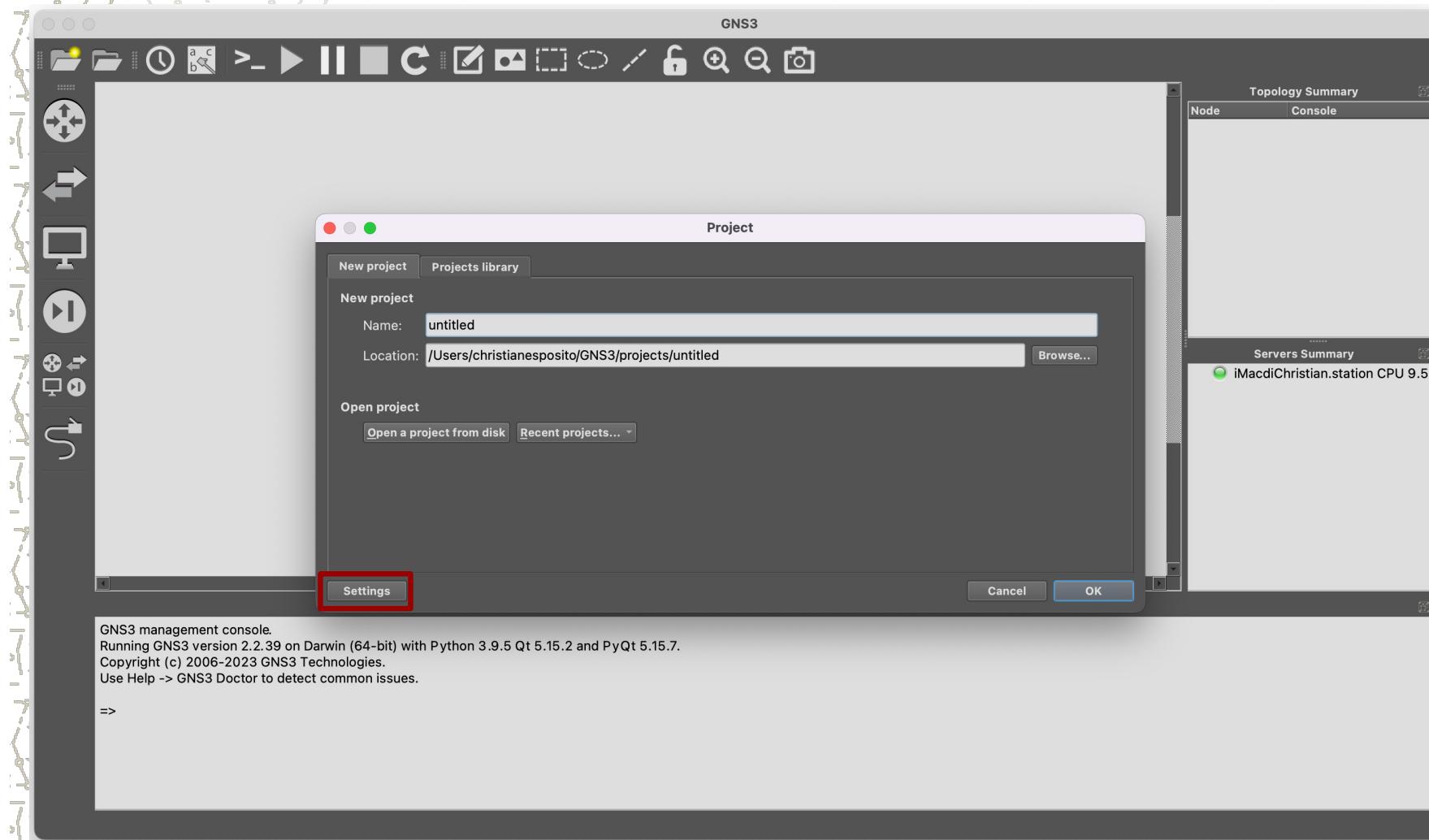
The screenshot shows the 'Download GNS3 VM' section of the GNS3 website. It features a dark background with white text and icons. At the top, there's a navigation bar with links to Software, Documentation, Community, Marketplace, and Academy. Below the navigation, the title 'Download GNS3 VM' is centered. A descriptive paragraph explains that the VM is recommended for Windows or Mac OS. Below this, four download options are listed, each with an icon, version number, and a 'Download' button:

- VirtualBox**
Version 2.2.39
[Download](#)
- VMware Workstation and Fusion**
Version 2.2.39
[Download](#)
- VMware ESXi**
Version 2.2.39
[Download](#)
- Microsoft Hyper-V**
Version 2.2.39
[Download](#)

At the bottom of the section, there are links to 'Learn more about the GNS3 VM' and a note stating 'GNS3 is a Free and Open Source software under GPL v3 licensing'.

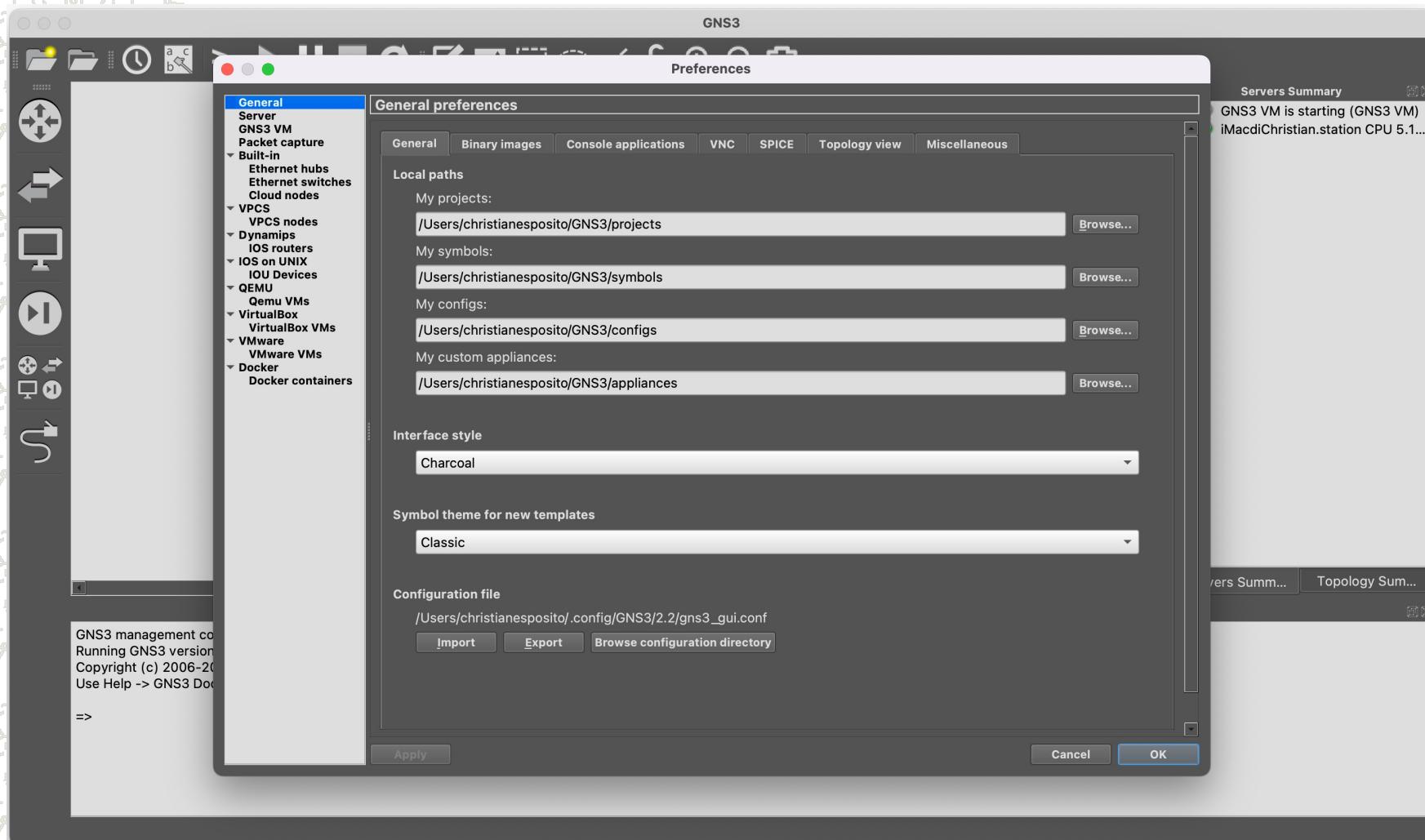
GNS3 VM

Successivamente bisogna andare in settings e configurare GNS3 VM.

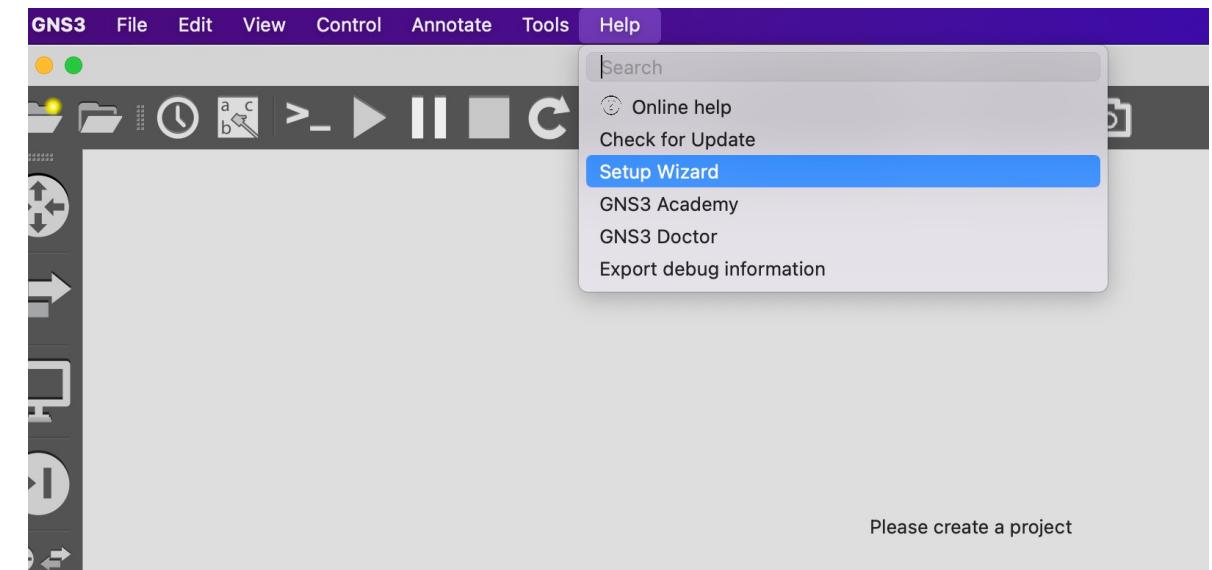


GNS3 VM

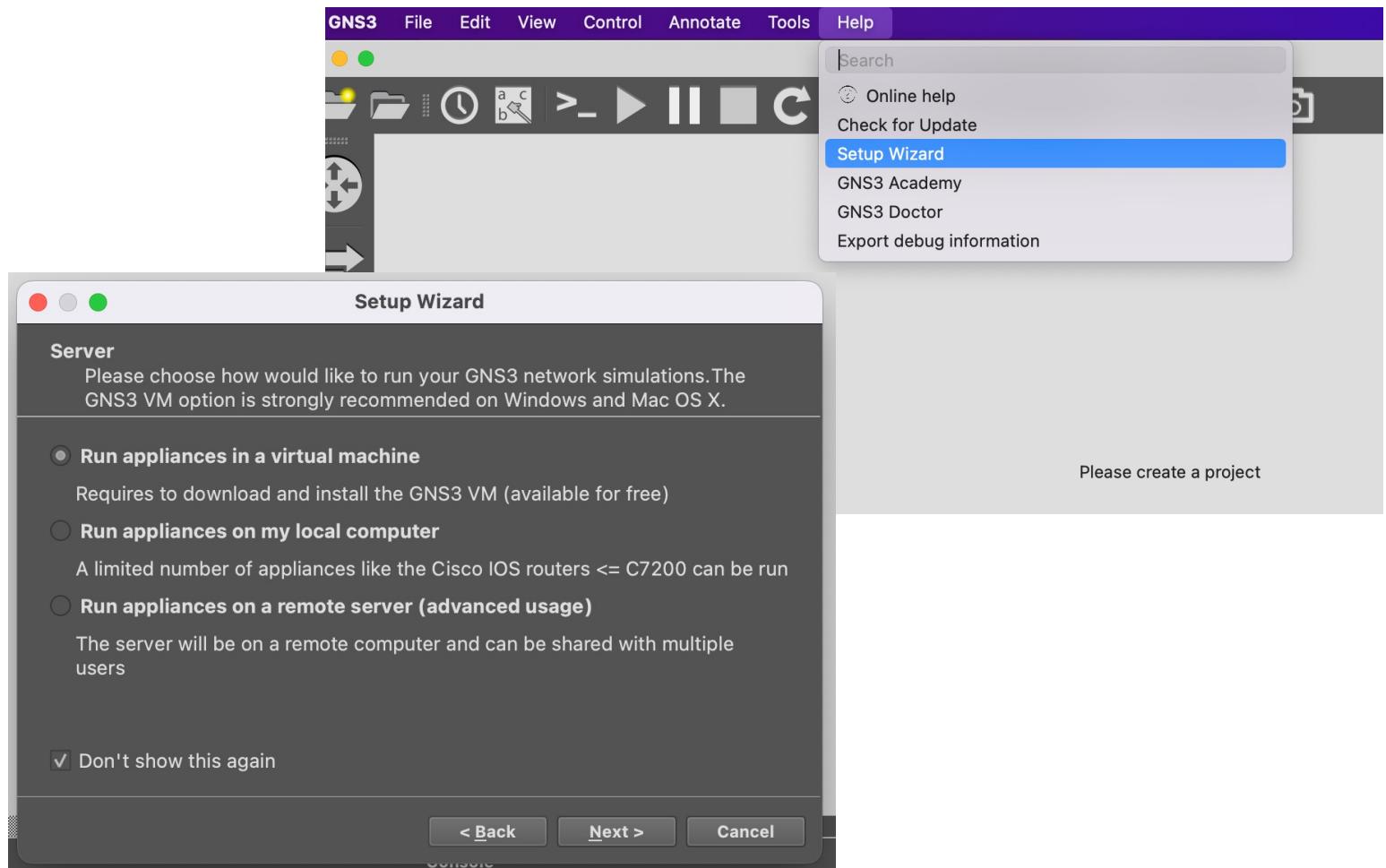
Successivamente bisogna andare in settings e configurare GNS3 VM.



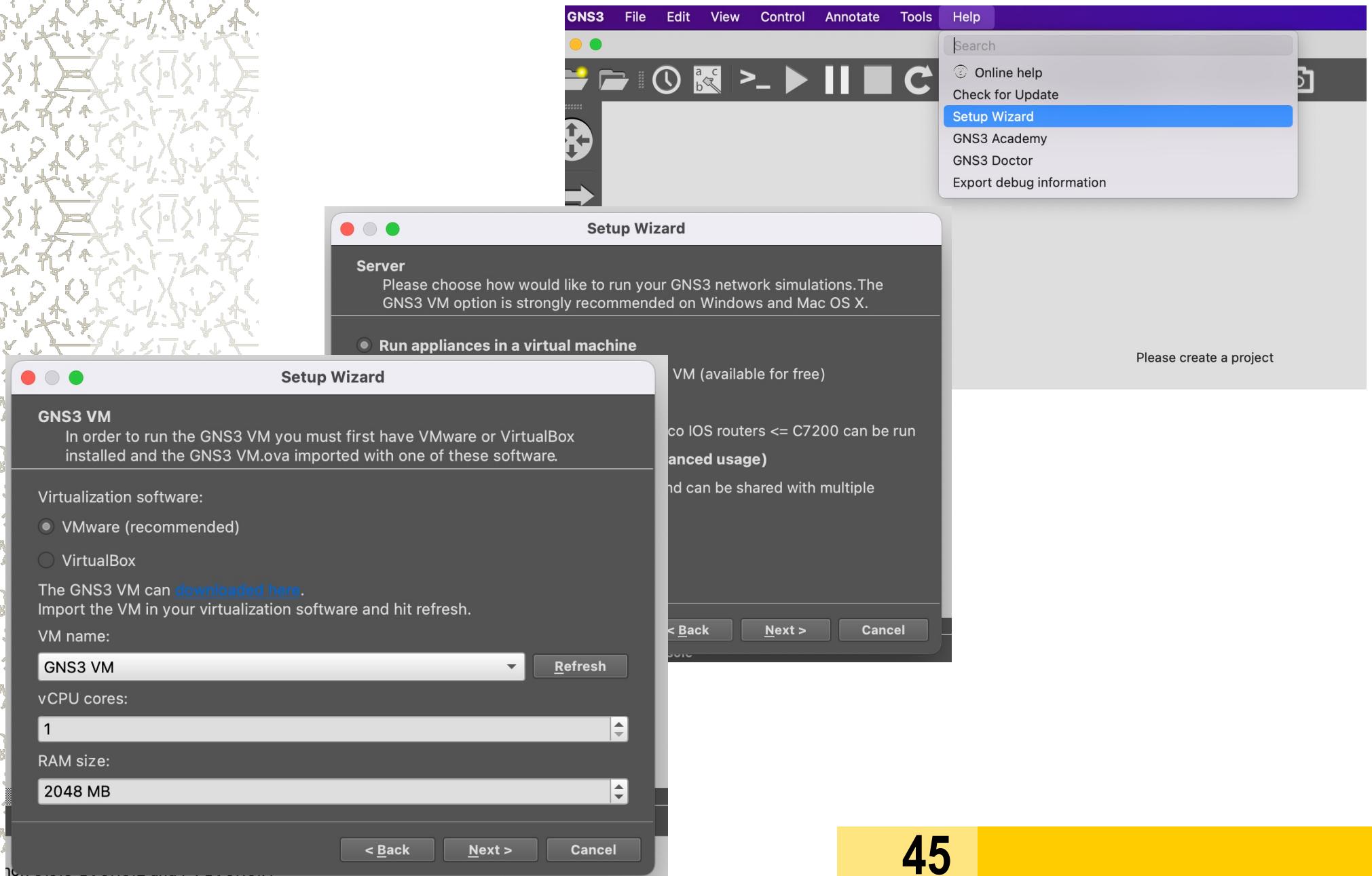
GNS3 VM



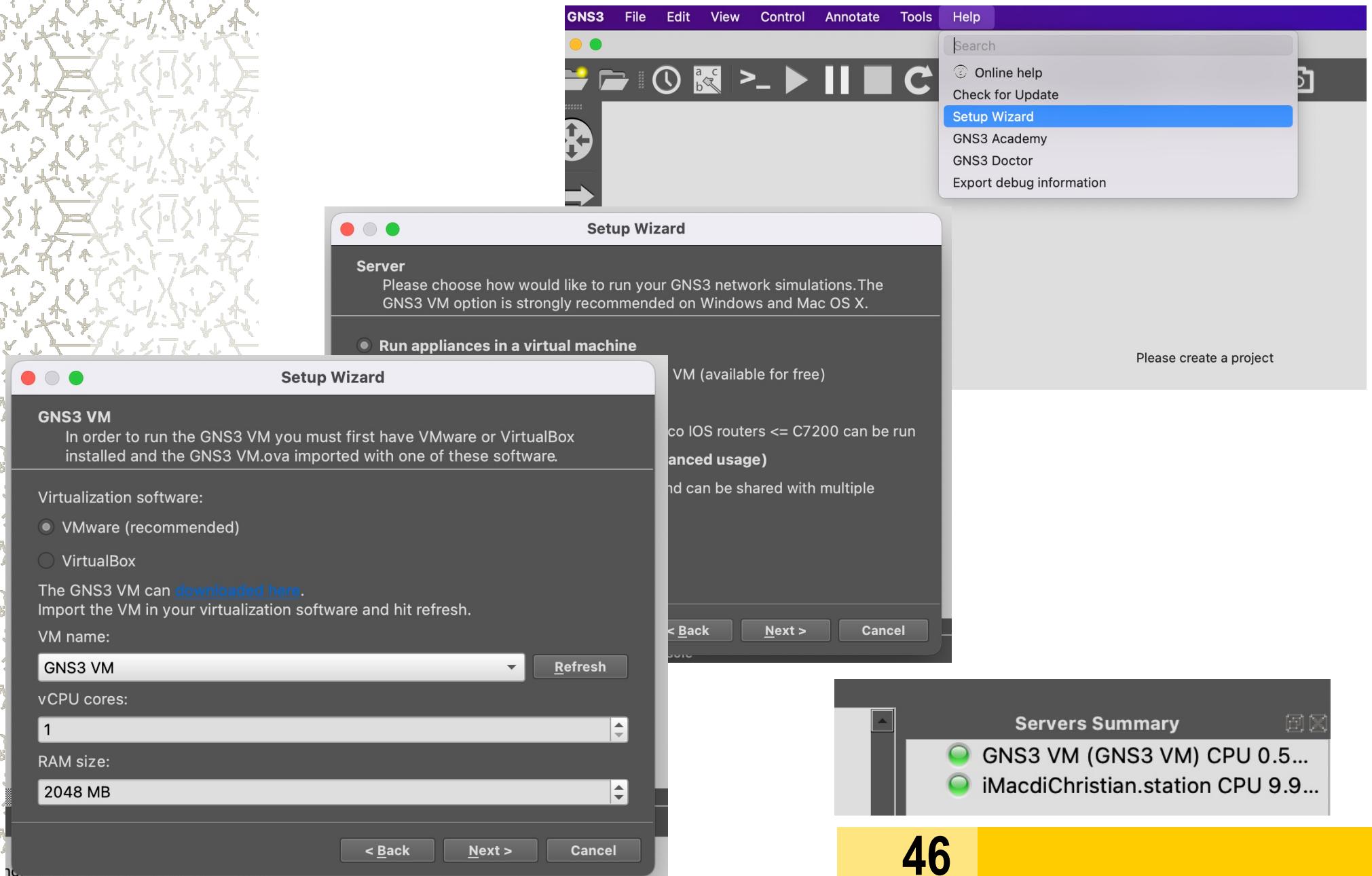
GNS3 VM



GNS3 VM



GNS3 VM

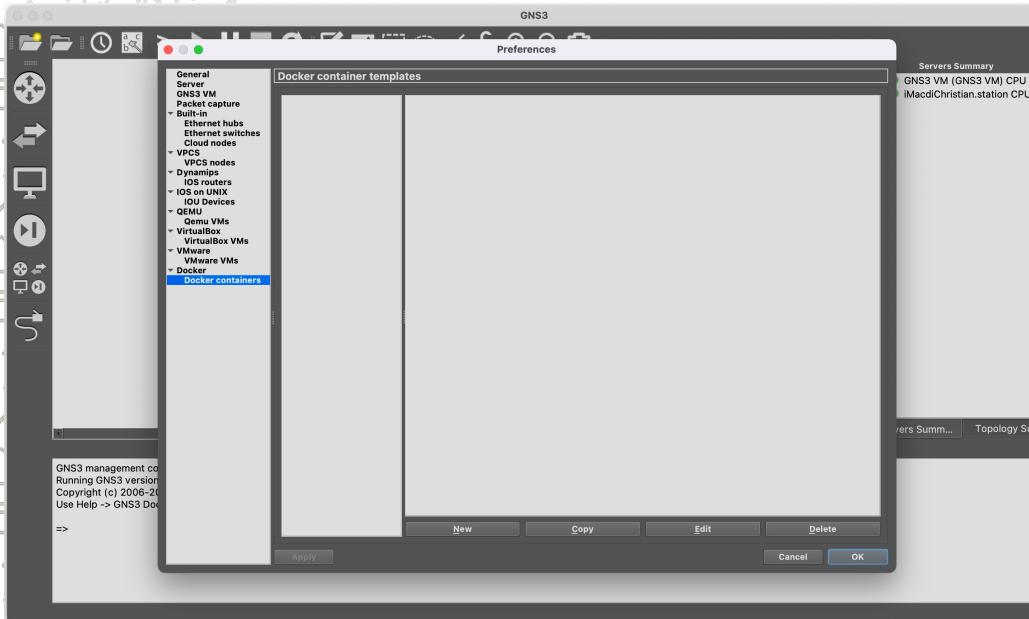


Esercizio - 3

Fasi di preparazione

La procedura qui descritta verrà eseguita una sola volta. Questo esercizio richiede alla macchina virtuale GNS3 di creare un'istanza di un contenitore Docker di un router basato su Linux basato sulla suite di software FRR (Free Range Routing).

- In GNS3 vai su Modifica/Preferenze... e seleziona Modelli di contenitore Docker dal pannello a sinistra della finestra Preferenze.

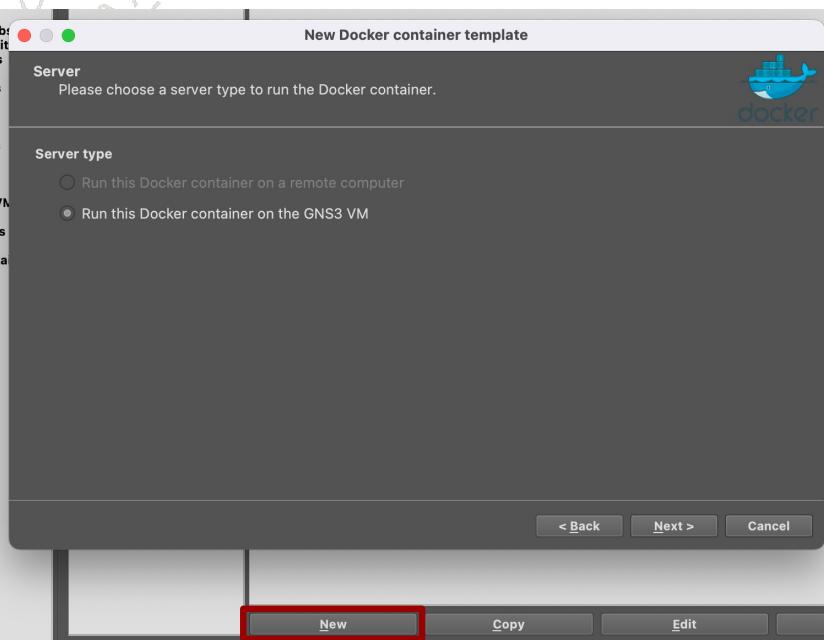


Esercizio - 3

Fasi di preparazione

La procedura qui descritta verrà eseguita una sola volta. Questo esercizio richiede alla macchina virtuale GNS3 di creare un'istanza di un contenitore Docker di un router basato su Linux basato sulla suite di software FRR (Free Range Routing).

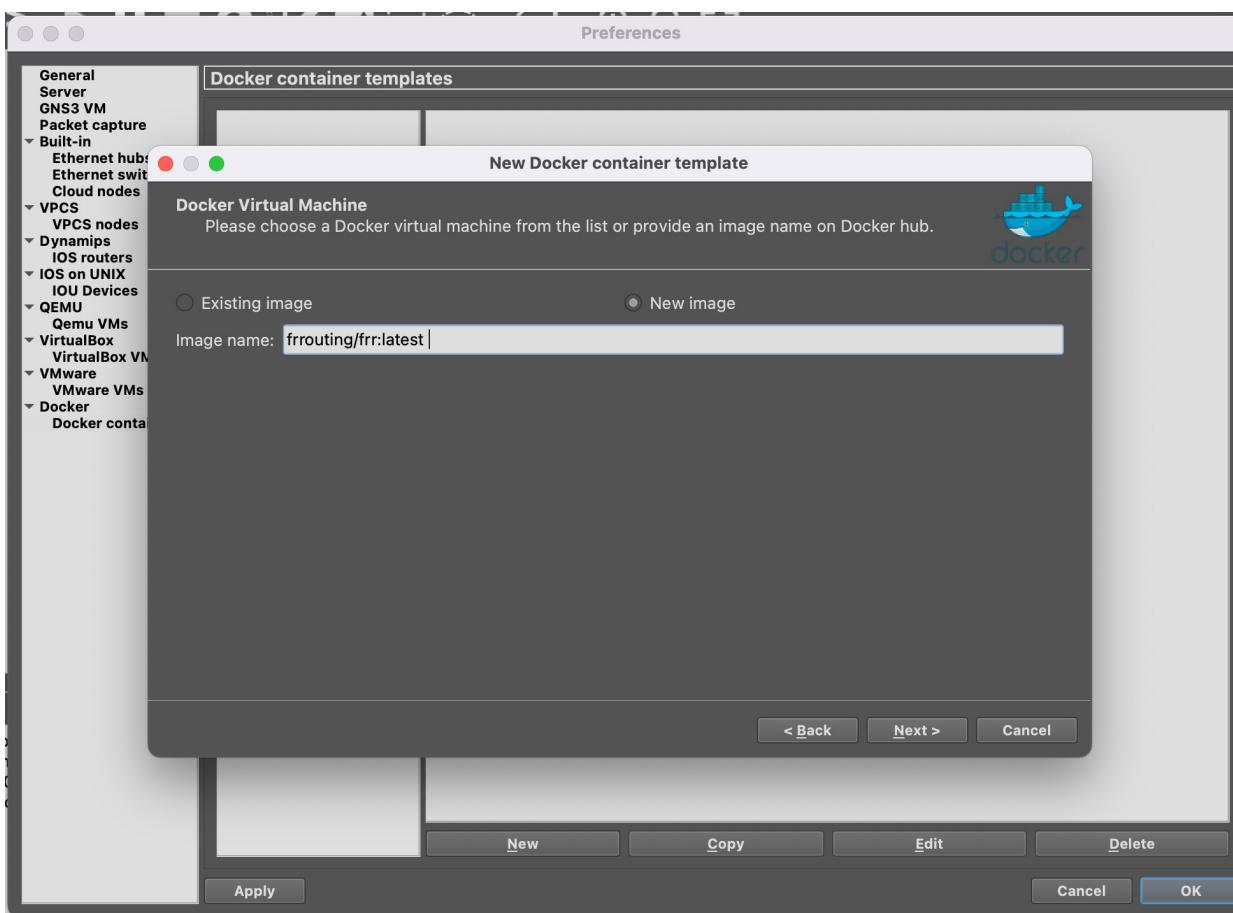
- In GNS3 vai su Modifica/Preferenze... e seleziona Modelli di contenitore Docker dal pannello a sinistra della finestra Preferenze.



- Fare clic su Nuovo.
- Selezionare l'opzione Esegui questo contenitore Docker sulla macchina virtuale GNS3 e premere Avanti.

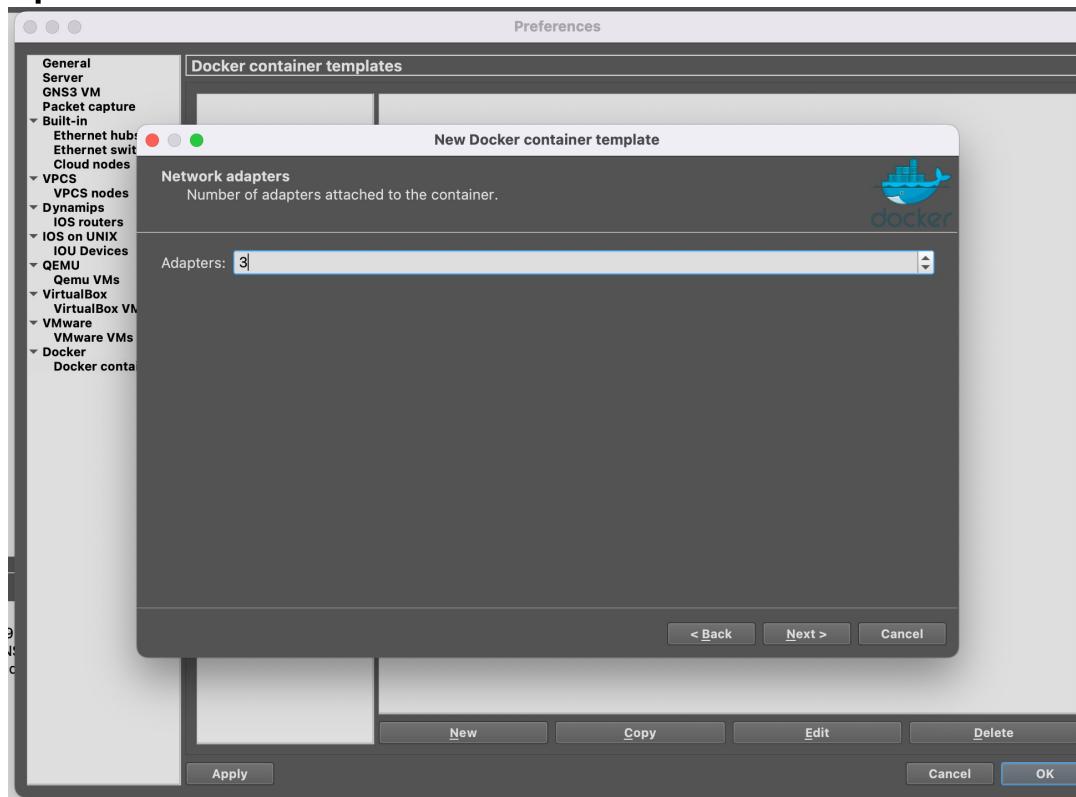
Esercizio - 3

- Scegli l'azione Nuova immagine.
- Digita frrouting/frr:latest nella casella di testo Nome immagine e premi Avanti.



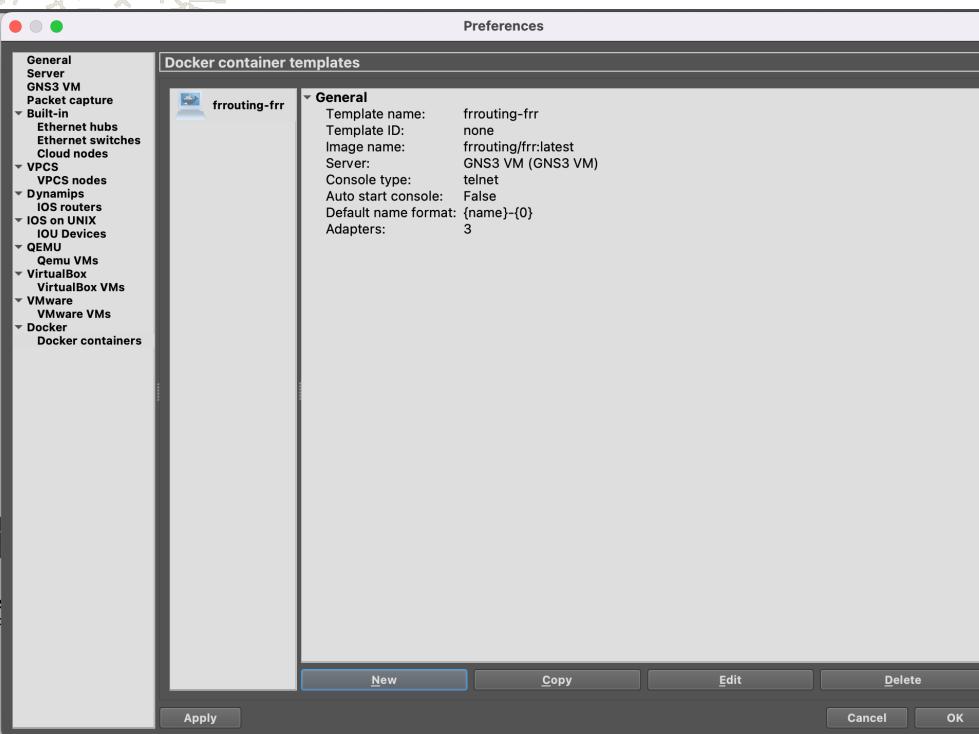
Esercizio - 3

- Confermare il nome predefinito frrouting-frr per il modello e premere Avanti.
- Specificare 3 come numero di adattatori che si desidera vengano utilizzati da questo contenitore e fare clic su Avanti.



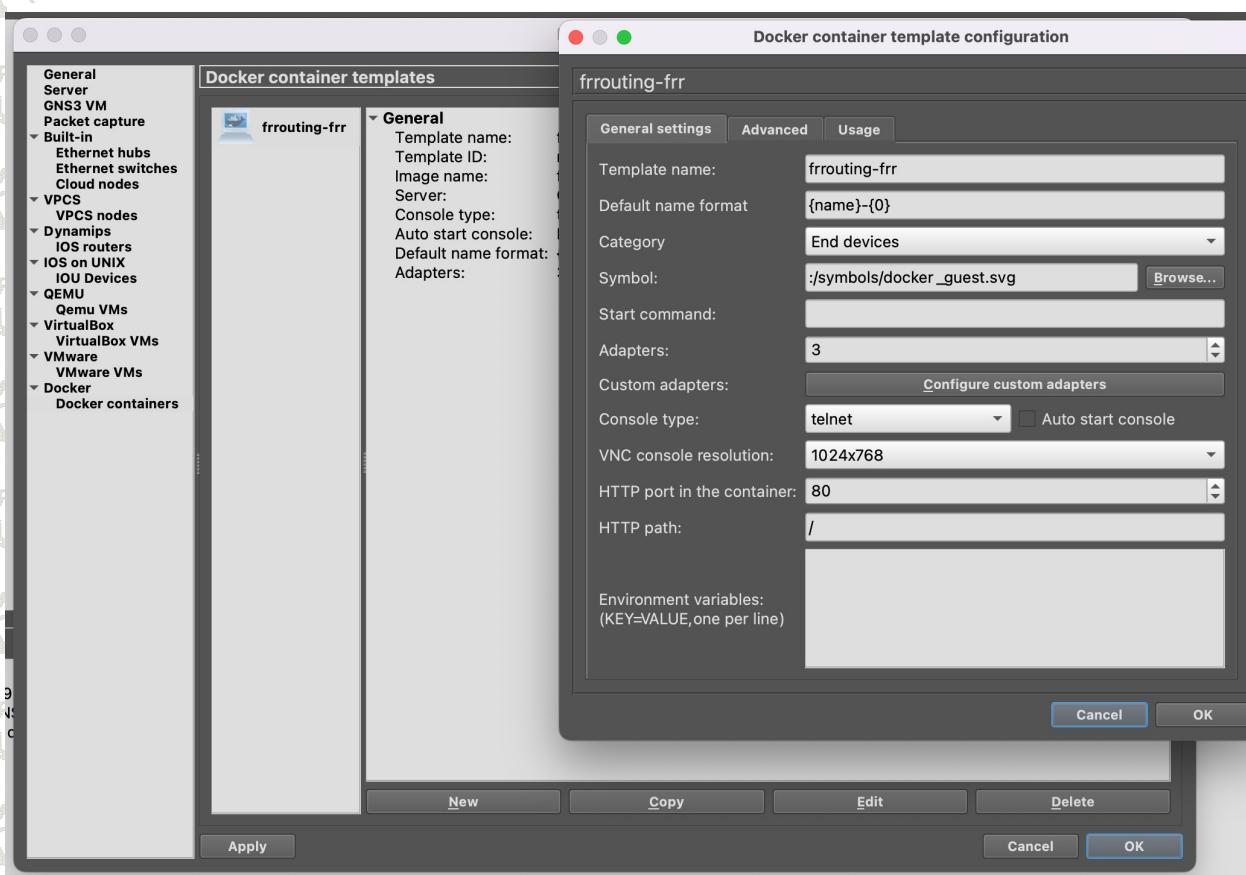
Esercizio - 3

- Lasciare vuota la casella di testo del comando Avvia e fare clic su Avanti.
- Lasciare il tipo di console impostato su telnet e fare clic su Avanti.
- Fare clic su Fine per completare questo processo di configurazione.
frrouting-frr verrà ora visualizzato nell'elenco dei modelli di contenitore Docker disponibili.



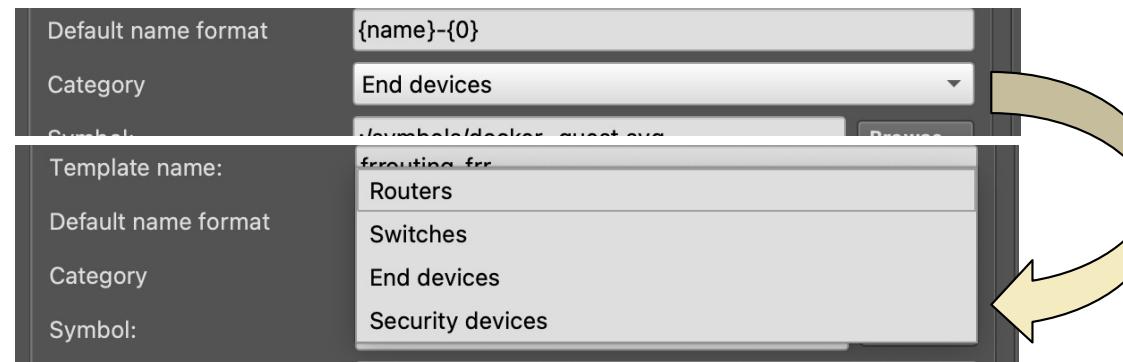
Esercizio - 3

- Nella finestra Preferenze, seleziona frrouting-frr dall'elenco dei modelli di contenitore Docker disponibili e fai clic su Modifica.



Esercizio - 3

- Nella finestra di configurazione del modello di contenitore Docker visualizzata, scegli la categoria Router.

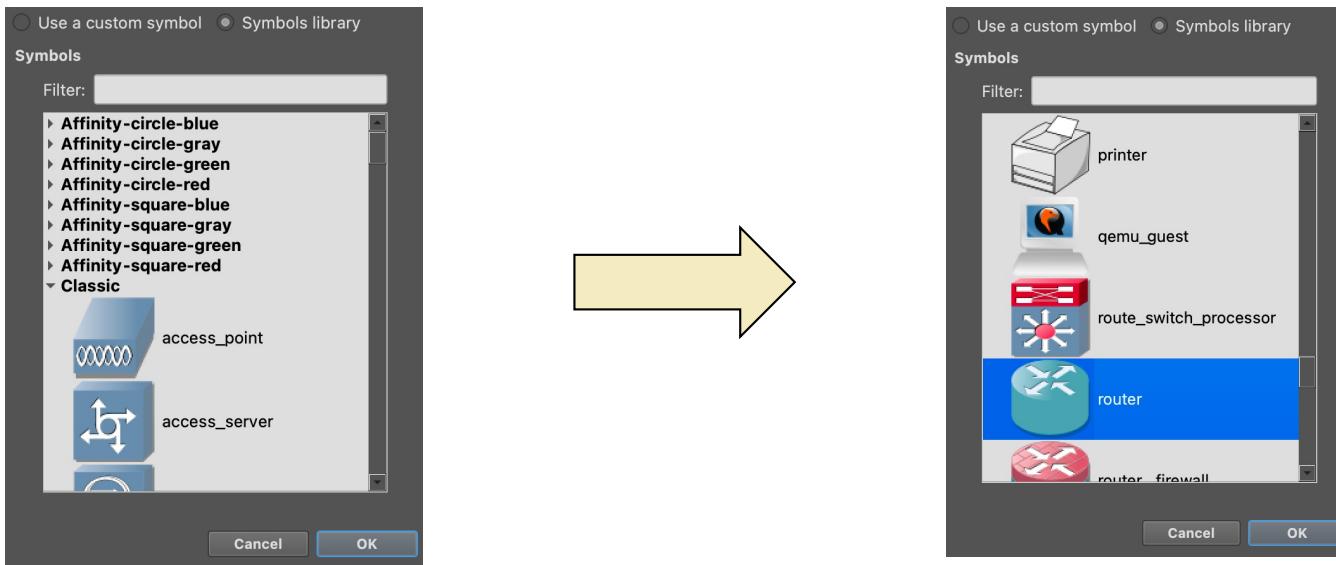


- Nella stessa finestra, scegli il pulsante Sfoglia...,



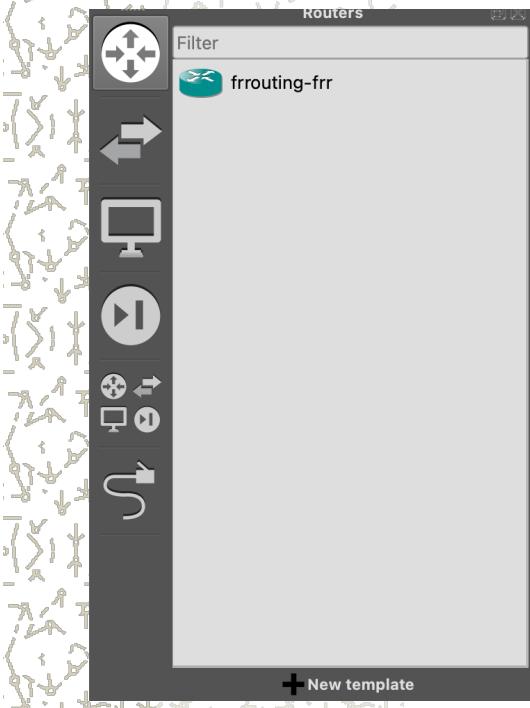
seleziona il simbolo del router dal set di icone Classico e premi OK.

Esercizio - 3



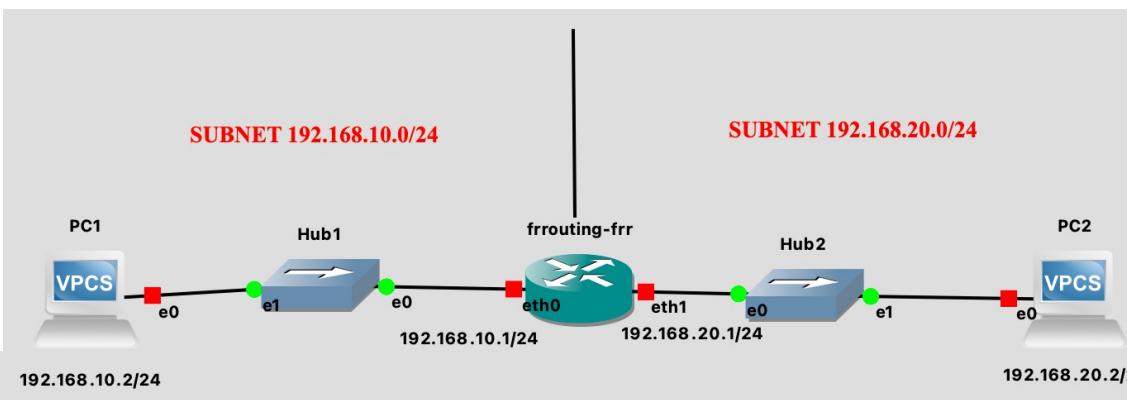
- Premere OK per chiudere la finestra di configurazione del modello di contenitore Docker.

Si noti che la prima volta che il componente frrouting-frr verrà aggiunto a un progetto GNS3, l'immagine del contenitore Docker verrà estratta dal repository Docker Hub. Ciò richiederà alcuni secondi.

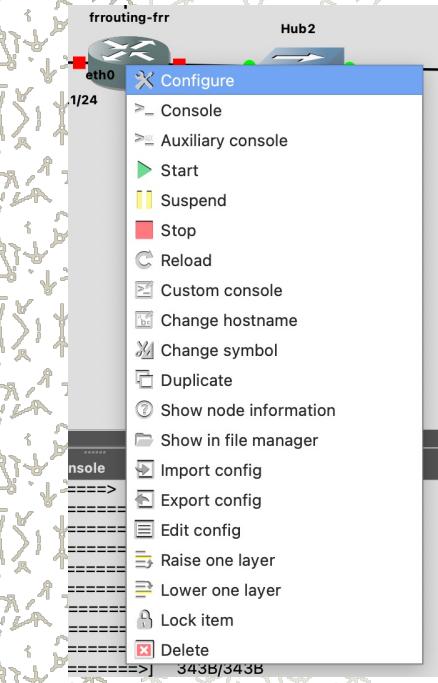


Esercizio - 3

Ricrea la seguente topologia in GNS3. Scegli il server "GNS3 VM" per istanziare tutti i dispositivi di questo laboratorio.

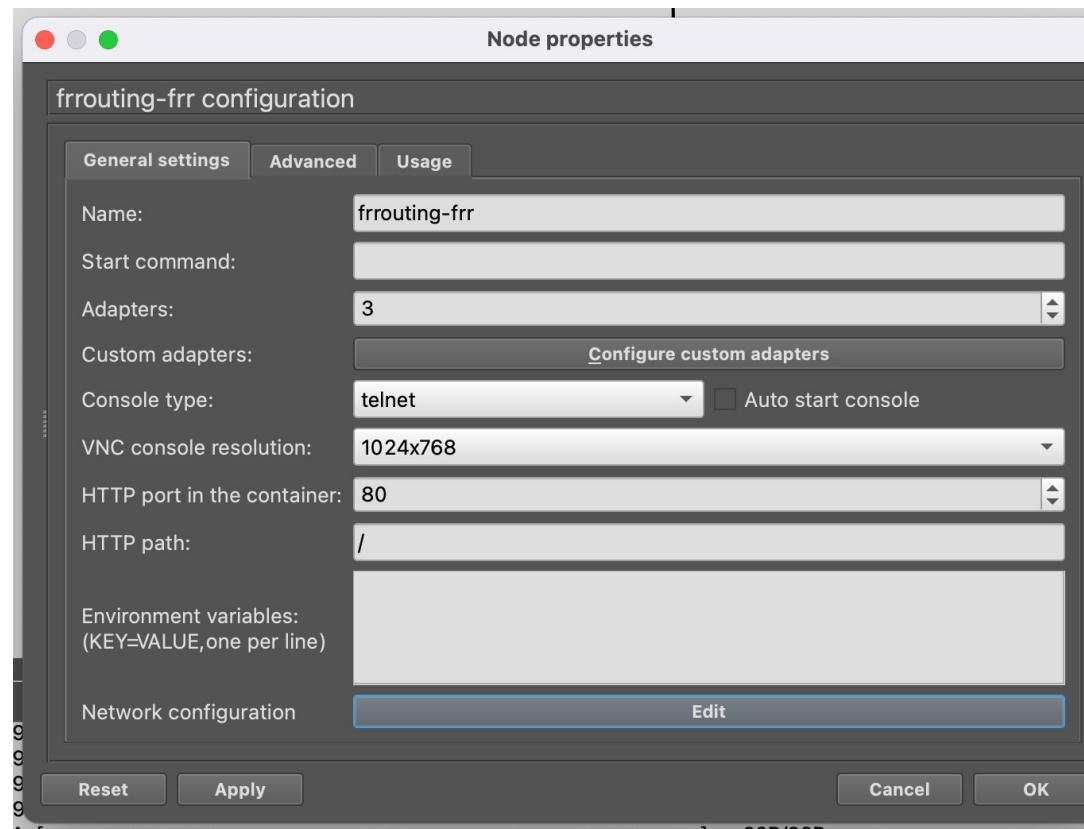


Quando i dispositivi sono ancora inattivi, fare clic con il tasto destro sull'icona del router e selezionare l'opzione Configura.



Esercizio - 3

Premere Modifica per modificare la configurazione di rete del router.



Esercizio - 3

Modificare la configurazione delle interfacce del router come segue:

```
# Static config for eth0
#auto eth0
#iface eth0 inet static
#           address 192.168.0.2
#           netmask 255.255.255.0
#           gateway 192.168.0.1
#           up echo nameserver 192.168.0.1 > /etc/resolv.conf

# DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
#           hostname frrouting-frr-2

# Static config for eth1
#auto eth1
#iface eth1 inet static
#           address 192.168.1.2
#           netmask 255.255.255.0
#           gateway 192.168.1.1
#           up echo nameserver 192.168.1.1 > /etc/resolv.conf
```

Esercizio - 3

Modificare la configurazione delle interfacce del router come segue:

```
# Static config for eth0
#auto eth0
#iface eth0 inet static
#           address 192.168.0.2
#           netmask 255.255.255.0
#           gateway 192.168.0.1      #
#           up echo nameserver 192.168.0.1#
```

```
# DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
#           hostname frrouting-frr-2
```

```
# Static config for eth1
#auto eth1
#iface eth1 inet static
#           address 192.168.1.2
#           netmask 255.255.255.0
#           gateway 192.168.1.1      #
#           up echo nameserver 192.168.1.1#
```

```
# Static config for eth0
auto eth0
iface eth0 inet static
           address 192.168.10.1
           netmask 255.255.255.0
           gateway 192.168.0.1
           up echo nameserver 192.168.0.1 > /etc/resolv.conf
```

```
# DHCP config for eth0
#auto eth0
#iface eth0 inet dhcp
#           hostname frrouting-frr-2
```

```
# Static config for eth1
auto eth1
iface eth1 inet static
           address 192.168.20.1
           netmask 255.255.255.0
           gateway 192.168.1.1
           up echo nameserver 192.168.1.1 > /etc/resolv.conf
```

Esercizio - 3

Avvia tutti i dispositivi.

Apri il terminale PC1 ed esegui quanto segue:

- assegnare l'indirizzo IP 192.168.10.2 con netmask /24 e l'indirizzo gateway predefinito 192.168.10.1 al PC1 con il comando:
`ip 192.168.10.2/24 192.168.10.1`
- rendere questa la configurazione predefinita automatica all'avvio per PC1 emettendo il comando save

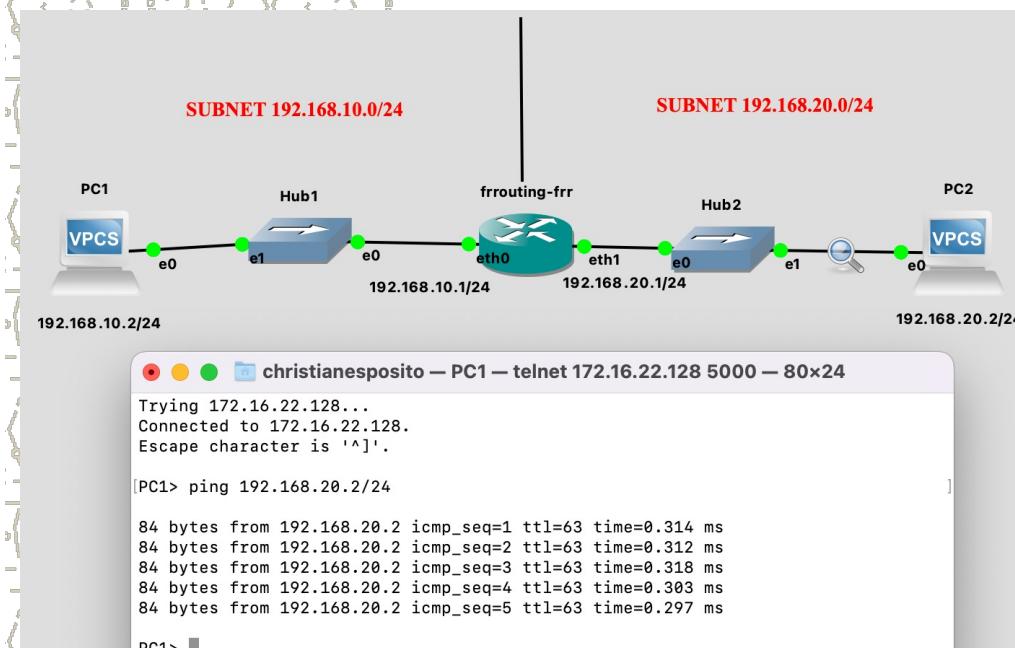
Apri il terminale PC2 ed esegui quanto segue:

- assegnare l'indirizzo IP 192.168.20.2 con netmask /24 e l'indirizzo gateway predefinito 192.168.20.1 al PC2 con il comando:
`ip 192.168.20.2/24 192.168.20.1`
- rendere questa la configurazione predefinita automatica all'avvio per PC2 emettendo il comando save.

Esercizio - 3

Avvia l'acquisizione sul collegamento che collega Hub2 a PC2.
Nel terminale PC1 eseguire il comando per eseguire il ping su PC2:
`ping 192.168.20.2`
e verificare che le risposte vengano ricevute.

Esercizio - 3



I dispositivi correttamente si vedono.

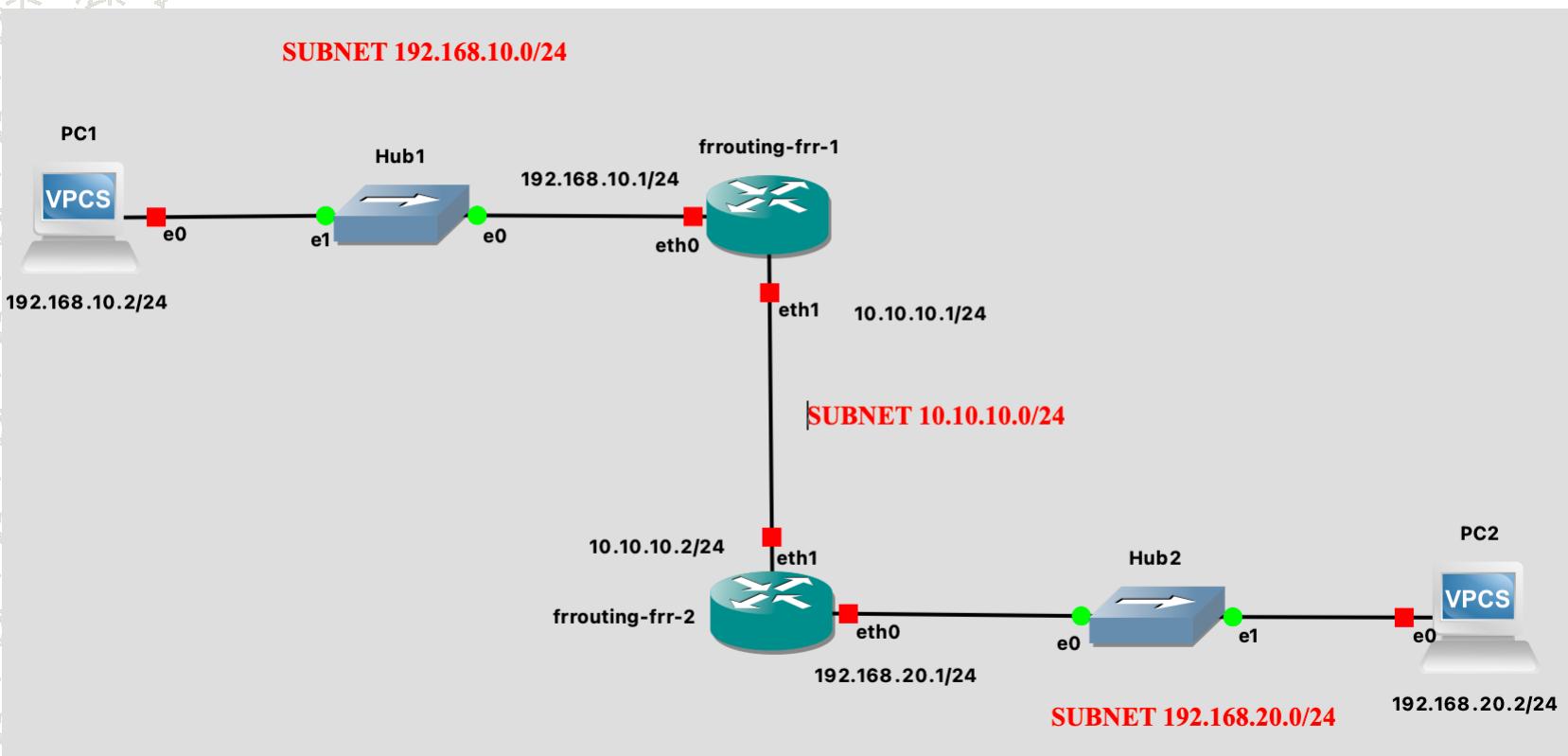
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	4e:31:06:be:c7:25	Broadcast	ARP	42	Who has 192.168.20.2? Tell 192.168.20.1
2	0.000162	Private_66:68:01	4e:31:06:be:c7:25	ARP	42	192.168.20.2 is at 00:50:79:66:68:01
3	0.000226	192.168.10.2	192.168.20.2	ICMP	98	Echo (ping) request id=0x36fb, seq=1/256, ttl=63 (reply in 4)
4	0.000296	192.168.20.2	192.168.10.2	ICMP	98	Echo (ping) reply id=0x36fb, seq=1/256, ttl=64 (request in 3)
5	1.001390	192.168.10.2	192.168.20.2	ICMP	98	Echo (ping) request id=0x37fb, seq=2/512, ttl=63 (reply in 6)
6	1.001497	192.168.20.2	192.168.10.2	ICMP	98	Echo (ping) reply id=0x37fb, seq=2/512, ttl=64 (request in 5)
7	2.003516	192.168.10.2	192.168.20.2	ICMP	98	Echo (ping) request id=0x38fb, seq=3/768, ttl=63 (reply in 8)
8	2.003622	192.168.20.2	192.168.10.2	ICMP	98	Echo (ping) reply id=0x38fb, seq=3/768, ttl=64 (request in 7)
9	3.005660	192.168.10.2	192.168.20.2	ICMP	98	Echo (ping) request id=0x39fb, seq=4/1024, ttl=63 (reply in 10)
10	3.005739	192.168.20.2	192.168.10.2	ICMP	98	Echo (ping) reply id=0x39fb, seq=4/1024, ttl=64 (request in 9)
11	4.007596	192.168.10.2	192.168.20.2	ICMP	98	Echo (ping) request id=0x3afb, seq=5/1280, ttl=63 (reply in 12)
12	4.007676	192.168.20.2	192.168.10.2	ICMP	98	Echo (ping) reply id=0x3afb, seq=5/1280, ttl=64 (request in 11)
13	6.186669	fe80::4c31:6ff:feb...	ff02::2	ICMPv6	70	Router Solicitation from 4e:31:06:be:c7:25

```
> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface -, id 0
Ethernet II, Src: 4e:31:06:be:c7:25 (4e:31:06:be:c7:25), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
```

```
0000 ff ff ff ff ff ff 4e 31 06 be c7 25 08 06 00 01 .....N1 ..%  
0010 08 00 06 04 00 01 0c 31 06 be c7 25 c0 a8 14 01 .....N1 ..%  
0020 00 00 00 00 00 00 c0 a8 14 02 ..... .
```

Esercizio - 4

Lo scopo di questo laboratorio è mostrare come due LAN possano essere collegate tramite due router. Ricrea la seguente topologia in GNS3. Scegli il server "GNS3 VM" per istanziare tutti i dispositivi di questo laboratorio.



Esercizio - 4

Configurare i due router come svolto nel precedente esercizio.

frrouting-frr-1 interfaces

```
#  
# This is a sample network config,please uncomment lines to configure the network  
  
# Uncomment this line to load custom interface files  
# source /etc/network/interfaces.d/*  
  
# Static config for eth0  
auto eth0  
iface eth0 inet static  
    address 192.168.10.1  
    netmask 255.255.255.0  
    # gateway 192.168.0.1  
    # up echo nameserver 192.168.0.1 > /etc/resolv.conf  
  
# DHCP config for eth0  
#auto eth0  
#iface eth0 inet dhcp  
#    hostname frrouting-frr-1  
  
# Static config for eth1  
auto eth1  
iface eth1 inet static  
    address 10.10.10.1  
    netmask 255.255.255.0  
    # gateway 192.168.1.1  
    # up echo nameserver 192.168.1.1 > /etc/resolv.conf
```

frrouting-frr-2 interfaces

```
#  
# This is a sample network config,please uncomment lines to configure the network  
  
# Uncomment this line to load custom interface files  
# source /etc/network/interfaces.d/*  
  
# Static config for eth0  
auto eth0  
iface eth0 inet static  
    address 192.168.20.1  
    netmask 255.255.255.0  
    # gateway 192.168.0.1  
    # up echo nameserver 192.168.0.1 > /etc/resolv.conf  
  
# DHCP config for eth0  
#auto eth0  
#iface eth0 inet dhcp  
#    hostname frrouting-frr-2  
  
# Static config for eth1  
auto eth1  
iface eth1 inet static  
    address 10.10.10.2  
    netmask 255.255.255.0  
    # gateway 192.168.1.1  
    # up echo nameserver 192.168.1.1 > /etc/resolv.conf
```

Esercizio - 4

Avvia tutti i dispositivi.

Apri il terminale PC1 ed esegui i seguenti comandi:

```
ip 192.168.10.2/24 192.168.10.1
```

```
save
```

Apri il terminale PC2 ed esegui i seguenti comandi:

```
ip 192.168.20.2/24 192.168.20.1
```

```
save
```

Avvia l'acquisizione sul collegamento che collega frouting-frr-2 a PC2.

Nel terminale PC1 eseguire il comando:

```
ping 192.168.10.1
```

e verificare che le risposte vengano ricevute dal gateway predefinito.

Nel terminale PC1 eseguire il comando:

```
ping 192.168.20.2
```

e verificare che NON si ricevano risposte dal PC2.

Esercizio - 4

```
[PC1]> ping 192.168.10.1
```

```
84 bytes from 192.168.10.1 icmp_seq=1 ttl=64 time=0.202 ms
84 bytes from 192.168.10.1 icmp_seq=2 ttl=64 time=0.173 ms
84 bytes from 192.168.10.1 icmp_seq=3 ttl=64 time=0.314 ms
84 bytes from 192.168.10.1 icmp_seq=4 ttl=64 time=0.176 ms
84 bytes from 192.168.10.1 icmp_seq=5 ttl=64 time=0.240 ms
```

```
[PC1]> ping 192.168.20.2
```

```
*192.168.10.1 icmp_seq=1 ttl=64 time=0.116 ms (ICMP type:3, code:0, Destination
network unreachable)
*192.168.10.1 icmp_seq=2 ttl=64 time=0.128 ms (ICMP type:3, code:0, Destination
network unreachable)
*192.168.10.1 icmp_seq=3 ttl=64 time=0.176 ms (ICMP type:3, code:0, Destination
network unreachable)
*192.168.10.1 icmp_seq=4 ttl=64 time=0.180 ms (ICMP type:3, code:0, Destination
network unreachable)
192.168.20.2 icmp_seq=5 timeout
```

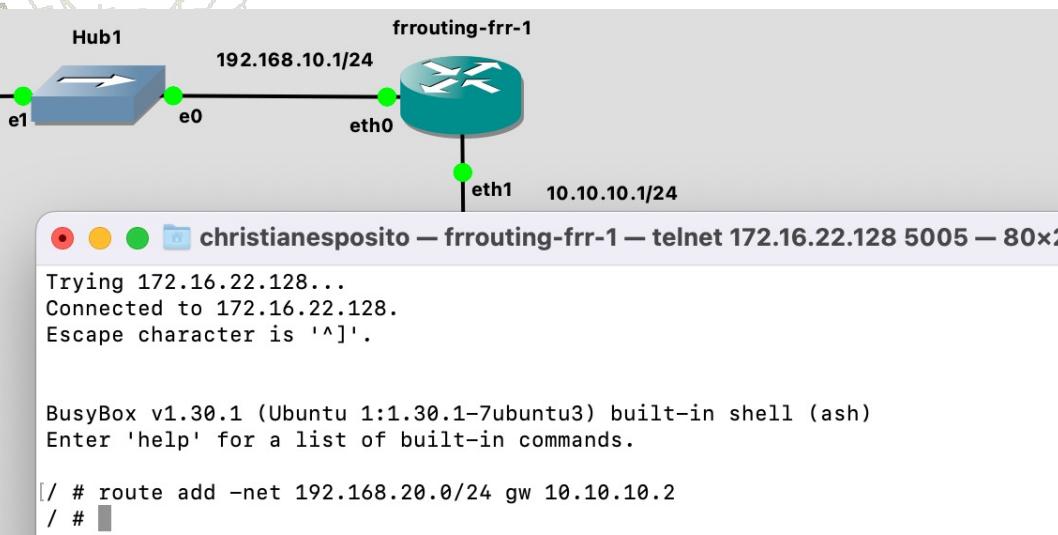
```
PC1>
```

Esercizio - 4

Fare clic con il tasto destro sull'icona del router frr-routing-2 e selezionare l'opzione Auxiliary Console.

Nel terminale frr-routing-1, digita il comando:

```
route add -net 192.168.20.0/24 gw 10.10.10.2
```



Nel terminale PC1 eseguire nuovamente il comando:

```
ping 192.168.20.2
```

e verificare che le risposte dal PC2 siano ora ricevute.

Esercizio - 4

Fare clic con il tasto destro sull'icona del router frr-routing-2 e selezionare l'opzione Auxiliary Console.

Nel terminale frr-routing-1, digita il comando:

```
route add -net 192.168.20.0/24 gw 10.10.10.2
```



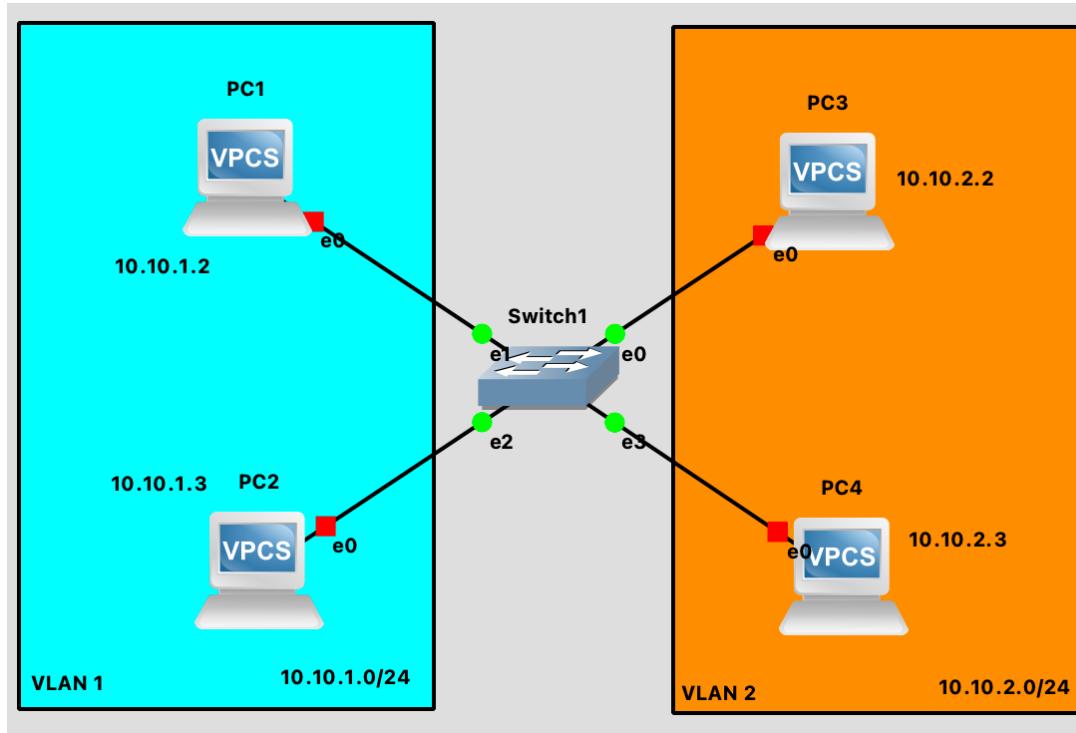
Nel terminale PC1 eseguire
ping 192.168.20.2
e verificare che le risposte d

Esercizio - 4

Svolgere gli stessi comandi su PC2 per il ping verso PC1 senza e con la regola di routing statico impostato nel router frr-routing-2.

Esercizio - 5

Ricrea la seguente topologia in GNS3. Scegli il server "GNS3 VM" per istanziare tutti i dispositivi di questo laboratorio.

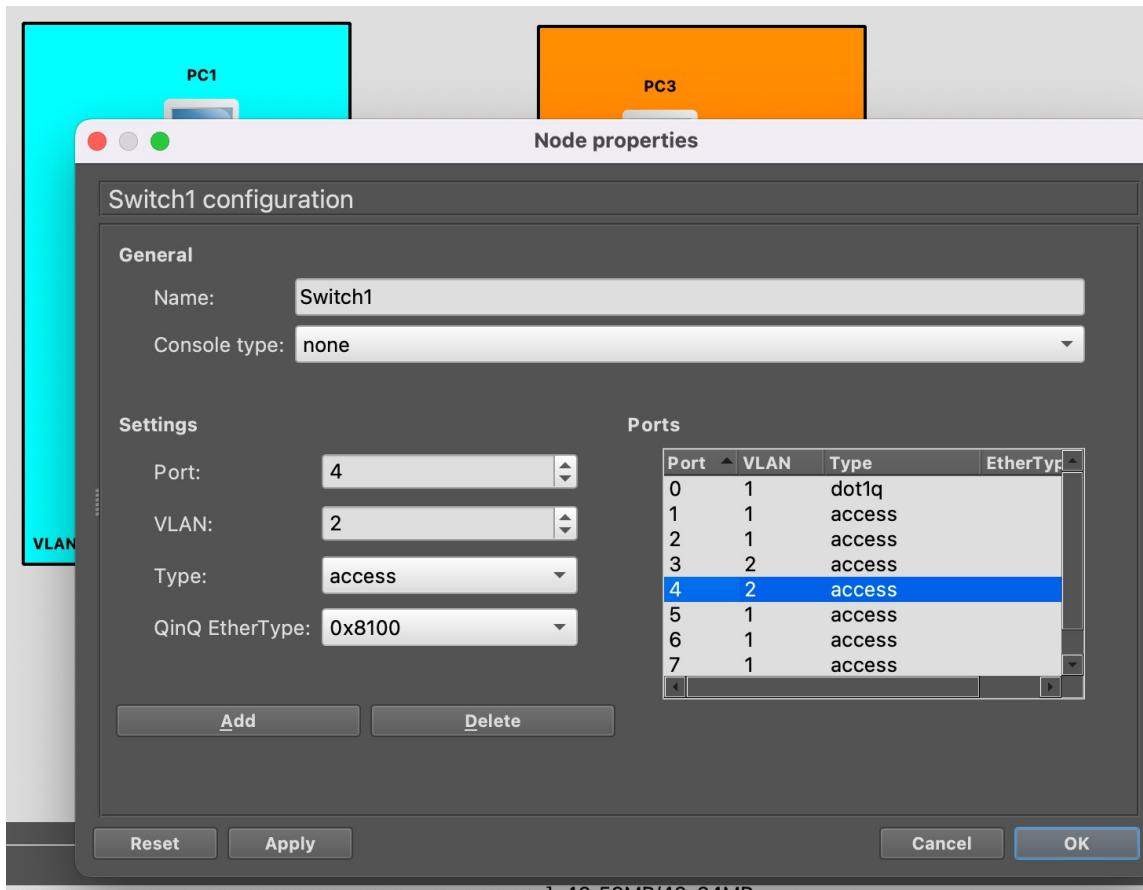


Quando i dispositivi sono ancora inattivi, fare clic con il tasto destro sull'icona dello switch e configurarlo come segue:

- porta 0 configurata in modalità trunk (802.1q) e assegnata nativamente alla VLAN 1 (in realtà questa porta non è utilizzata in questo esercizio);

Esercizio - 5

- porte 1, 2 configurate in modalità accesso e assegnate alla VLAN 1 (VLAN di default);
- porte 3,4 configurate in modalità di accesso e assegnate alla VLAN 2.



Esercizio - 5

Avvia tutti i dispositivi.

Apri il terminale PC1 ed esegui il comando:

```
ip 10.10.1.2/24 10.10.1.1
```

Apri il terminale PC2 ed esegui il comando:

```
ip 10.10.1.3/24 10.10.1.1
```

Apri il terminale PC3 ed esegui il comando:

```
ip 10.10.2.2/24 10.10.2.1
```

Apri il terminale PC4 ed esegui il comando:

```
ip 10.10.2.3/24 10.10.2.1
```

Si noti che PC1 e PC2 sono configurati con un indirizzo gateway predefinito 10.10.1.1, ma, di fatto, questo indirizzo non è associato ad alcun dispositivo. Allo stesso modo, PC3 e PC4 sono configurati con un indirizzo gateway predefinito 10.10.2.1, ma, di fatto, questo indirizzo non è associato ad alcun dispositivo.

Rendere le configurazioni permanenti col comando save.

Esercizio - 5

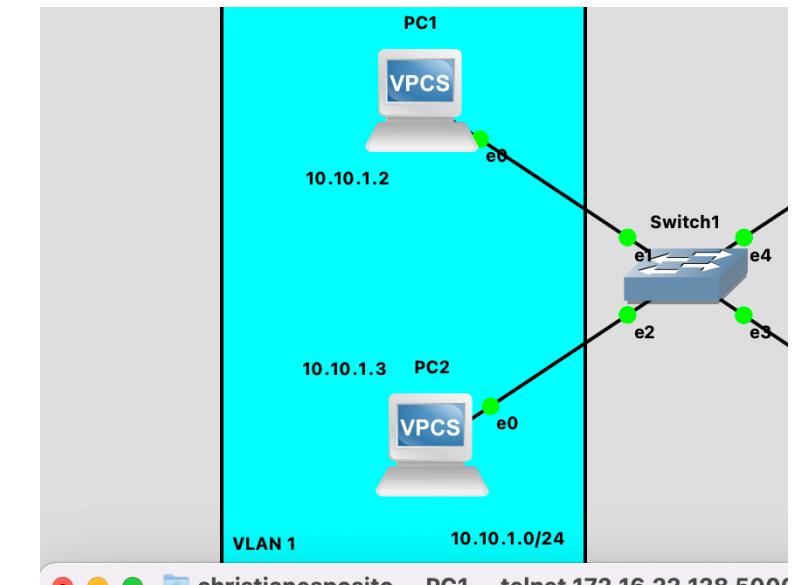
Nel terminale PC1 eseguire il comando:

ping 10.10.1.3

e verificare che le risposte vengano ricevute dal PC2.

ping 10.10.2.2

e verificare che NON si ricevano risposte dal PC3.



christianesposito — PC1 — telnet 172.16.22.128 5000
Executing the startup file

```
[PC1]> ip 10.10.1.2/24 10.10.1.1
Checking for duplicate address...
PC1 : 10.10.1.2 255.255.255.0 gateway 10.10.1.1
```

```
[PC1]> save
Saving startup configuration to startup.vpc
. done
```

```
[PC1]> ping 10.10.1.3
```

```
84 bytes from 10.10.1.3 icmp_seq=1 ttl=64 time=0.260 ms
84 bytes from 10.10.1.3 icmp_seq=2 ttl=64 time=0.455 ms
84 bytes from 10.10.1.3 icmp_seq=3 ttl=64 time=0.200 ms
84 bytes from 10.10.1.3 icmp_seq=4 ttl=64 time=0.120 ms
84 bytes from 10.10.1.3 icmp_seq=5 ttl=64 time=0.149 ms
```

```
[PC1]> ping 10.10.2.2
```

```
host (10.10.1.1) not reachable
```

```
[PC1]>
```

Esercizio - 5

Nel terminale PC1 eseguire il comando:

ping 10.10.1.3

e verificare che le risposte vengano ricevute dal PC2.

ping 10.10.2.2

e verificare che NON si ricevano risposte dal PC3.

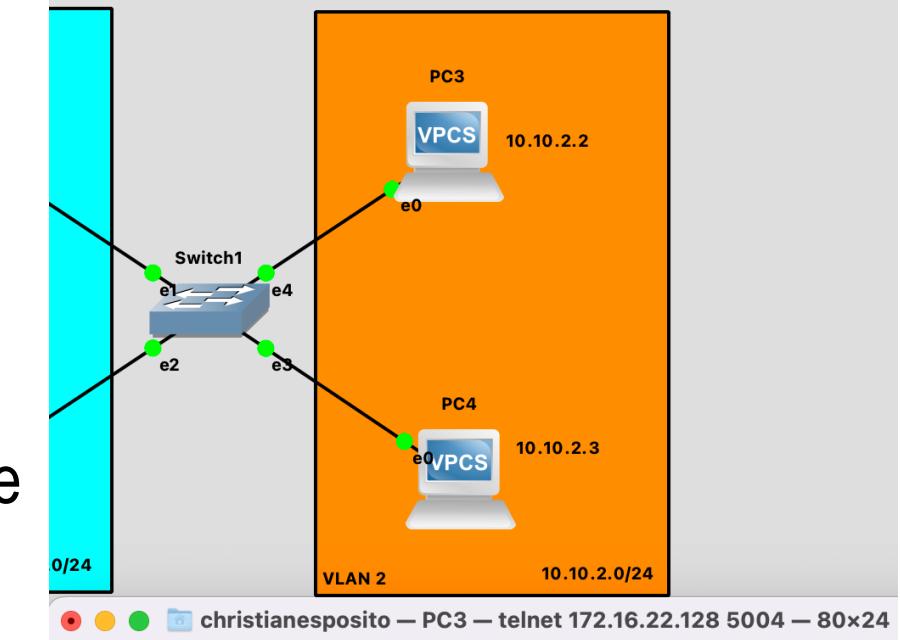
Nel terminale PC3 eseguire il comando:

ping 10.10.2.3

e verificare che le risposte vengano ricevute da PC4.

ping 10.10.1.2

e verificare che NON si ricevano risposte dal PC1.



```
[PC3]> ip 10.10.2.2/24 10.10.2.1
Checking for duplicate address...
PC3 : 10.10.2.2 255.255.255.0 gateway 10.10.2.1

[PC3]> save
Saving startup configuration to startup.vpc
. done

[PC3]> ping 10.10.2.3

84 bytes from 10.10.2.3 icmp_seq=1 ttl=64 time=0.187 ms
84 bytes from 10.10.2.3 icmp_seq=2 ttl=64 time=0.196 ms
84 bytes from 10.10.2.3 icmp_seq=3 ttl=64 time=0.120 ms
84 bytes from 10.10.2.3 icmp_seq=4 ttl=64 time=0.148 ms
84 bytes from 10.10.2.3 icmp_seq=5 ttl=64 time=0.119 ms

[PC3]> ping 10.10.2.1

host (10.10.2.1) not reachable

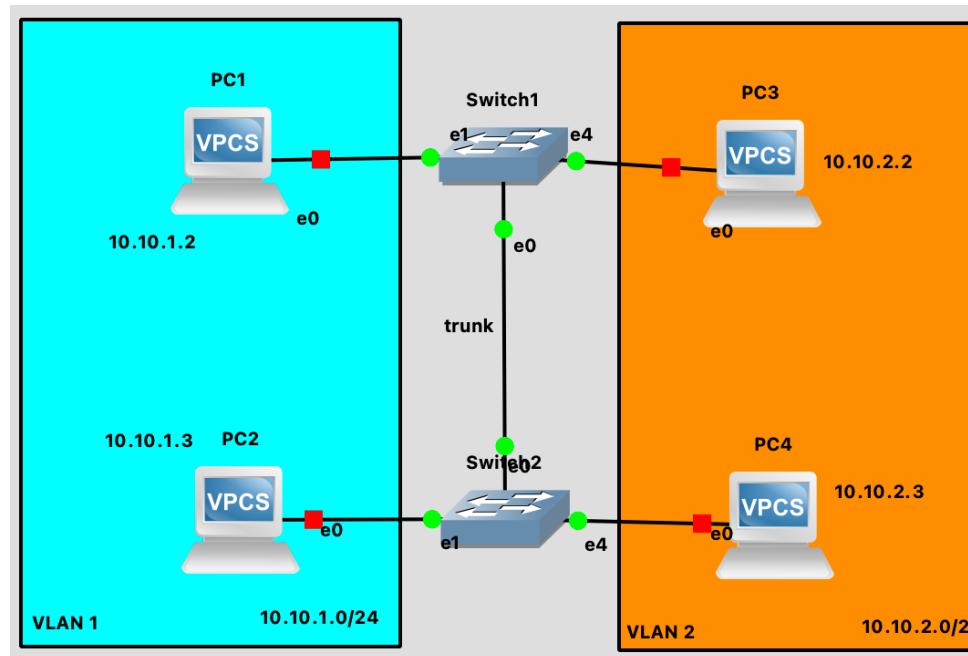
PC3>
```

Esercizio 5

Poiché non esiste un routing tra VLAN, le due VLAN sono isolate.
Quindi, PC1 può eseguire il ping solo su PC2, mentre PC3 può eseguire solo il ping su PC4.

Esercizio - 6

Ricrea la seguente topologia in GNS3. Scegli il server "GNS3 VM" per istanziare tutti i dispositivi di questo laboratorio.

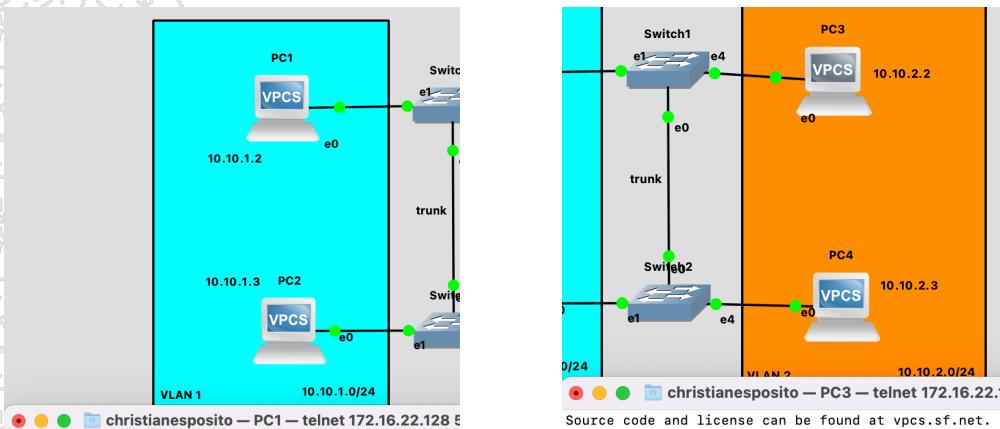


Quando i dispositivi sono ancora inattivi, fare clic con il tasto destro sull'icona del primo switch e configurarlo come fatto nel precedente esercizio. Replicare la configurazione anche con il secondo switch.

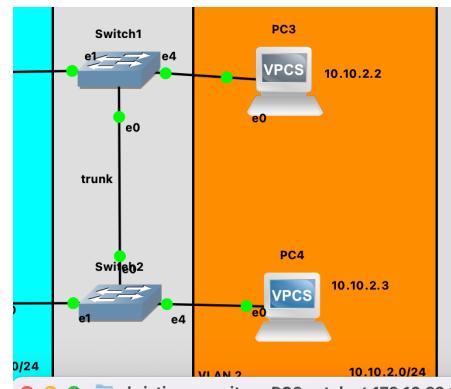
Esercizio - 6

Si noti che il collegamento trunk è connesso alla porta 0 sia di SW1 che di SW2. Il traffico VLAN 1 viene trasmesso come frame senza tag sul collegamento trunk. Il traffico VLAN 2 viene trasmesso come frame con tag 802.1q con VLAN ID 2.

Replichiamo la configurazione dei dispositivi come prima.



```
Press '?' to get help.  
Executing the startup file  
  
Checking for duplicate address...  
PC1 : 10.10.1.2 255.255.255.0 gateway 10.10.1.1  
  
[PC1]> ping 10.10.1.3  
84 bytes from 10.10.1.3 icmp_seq=1 ttl=64 time=0.127 ms  
84 bytes from 10.10.1.3 icmp_seq=2 ttl=64 time=0.245 ms  
84 bytes from 10.10.1.3 icmp_seq=3 ttl=64 time=0.224 ms  
84 bytes from 10.10.1.3 icmp_seq=4 ttl=64 time=0.234 ms  
84 bytes from 10.10.1.3 icmp_seq=5 ttl=64 time=0.376 ms  
  
[PC1]> ping 10.10.2.2  
host (10.10.1.1) not reachable  
  
[PC1]>
```



```
Press '?' to get help.  
Executing the startup file  
  
Checking for duplicate address...  
PC3 : 10.10.2.2 255.255.255.0 gateway 10.10.2.1  
  
[PC3]> ping 10.10.2.3  
84 bytes from 10.10.2.3 icmp_seq=1 ttl=64 time=0.208 ms  
84 bytes from 10.10.2.3 icmp_seq=2 ttl=64 time=0.228 ms  
84 bytes from 10.10.2.3 icmp_seq=3 ttl=64 time=0.244 ms  
84 bytes from 10.10.2.3 icmp_seq=4 ttl=64 time=0.226 ms  
84 bytes from 10.10.2.3 icmp_seq=5 ttl=64 time=0.240 ms  
  
[PC3]> ping 10.10.1.2  
host (10.10.2.1) not reachable  
  
[PC3]>
```

Replichiamo i ping come nell'esercizio precedente e avremo gli stessi risultati.

Esercizio - 6

The screenshot shows a packet capture in Wireshark with the following details:

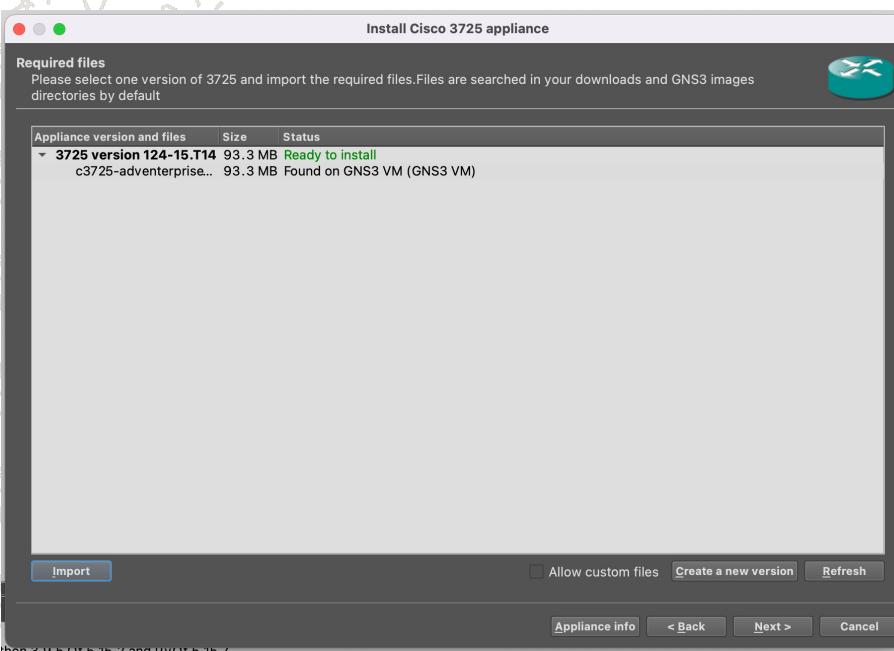
- Title:** VLAN
- Type:** 802.1Q VLAN id
- Selected Column:** Protocol (highlighted in blue)
- Context Menu (Open at row 0.001447):**
 - Align Left
 - Align Center
 - Align Right
 - Column Preferences...
 - Edit Column** (highlighted with a red arrow)
 - Resize to Contents
 - Resize Column to Width...
 - Resolve Names
- Table Headers:** VLAN Tag, Time, Source, Destination, Protocol, Length, Info
- Sampled Data:** The table shows several rows of network traffic, primarily ICMP echo requests and replies between hosts 10.10.1.2 and 10.10.1.3.

VLAN Tag	Time	Source	Destination	Protocol	Length	Info
0.000000	0.000000	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.2
0.000178	0.000178	Private_66:68:01	Private_66:68:00	ARP	64	10.10.1.3 is at 00:50:79:66:68:01
0.001253	0.001253	10.10.1.2	10.10.1.3	ICMP	98	Echo (ping) request id=0x1408, seq=1/256, ttl=64 (reply in 4)
0.001447	0.001447	10.10.1.3	10.10.1.2	ICMP	98	Echo (ping) reply id=0x1408, seq=1/256, ttl=64 (request in 3)
1.002810	1.002810	10.10.1.2	10.10.1.3	ICMP	98	Echo (ping) request id=0x1508, seq=2/512, ttl=64 (reply in 6)
1.002946	1.002946	10.10.1.3	10.10.1.2	ICMP	98	Echo (ping) reply id=0x1508, seq=2/512, ttl=64 (request in 5)
2.004186	2.004186	10.10.1.2	10.10.1.3	ICMP	98	Echo (ping) request id=0x1608, seq=3/768, ttl=64 (reply in 8)
2.004324	2.004324	10.10.1.3	10.10.1.2	ICMP	98	Echo (ping) reply id=0x1608, seq=3/768, ttl=64 (request in 7)
3.005880	3.005880	10.10.1.2	10.10.1.3	ICMP	98	Echo (ping) request id=0x1708, seq=4/1024, ttl=64 (reply in 10)
3.006016	3.006016	10.10.1.3	10.10.1.2	ICMP	98	Echo (ping) reply id=0x1708, seq=4/1024, ttl=64 (request in 9)
4.007417	4.007417	10.10.1.2	10.10.1.3	ICMP	98	Echo (ping) request id=0x1808, seq=5/1280, ttl=64 (reply in 12)
4.007556	4.007556	10.10.1.3	10.10.1.2	ICMP	98	Echo (ping) reply id=0x1808, seq=5/1280, ttl=64 (request in 11)
	10.395957	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.1? Tell 10.10.1.2
	11.396022	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.1? Tell 10.10.1.2
	12.397186	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.1? Tell 10.10.1.2
2	17.667897	Private_66:68:02	Broadcast	ARP	68	Who has 10.10.2.3? Tell 10.10.2.2
2	17.668038	Private_66:68:03	Private_66:68:02	ARP	68	10.10.2.3 is at 00:50:79:66:68:03
2	17.669225	10.10.2.2	10.10.2.3	ICMP	102	Echo (ping) request id=0x2508, seq=1/256, ttl=64 (reply in 19)
2	17.669342	10.10.2.3	10.10.2.2	ICMP	102	Echo (ping) reply id=0x2508, seq=1/256, ttl=64 (request in 18)
2	18.670813	10.10.2.2	10.10.2.3	ICMP	102	Echo (ping) request id=0x2608, seq=2/512, ttl=64 (reply in 21)
2	18.670944	10.10.2.3	10.10.2.2	ICMP	102	Echo (ping) reply id=0x2608, seq=2/512, ttl=64 (request in 20)
2	19.672822	10.10.2.2	10.10.2.3	ICMP	102	Echo (ping) request id=0x2708, seq=3/768, ttl=64 (reply in 23)
2	19.672960	10.10.2.3	10.10.2.2	ICMP	102	Echo (ping) reply id=0x2708, seq=3/768, ttl=64 (request in 22)
2	20.674399	10.10.2.2	10.10.2.3	ICMP	102	Echo (ping) request id=0x2808, seq=4/1024, ttl=64 (reply in 25)
2	20.674493	10.10.2.3	10.10.2.2	ICMP	102	Echo (ping) reply id=0x2808, seq=4/1024, ttl=64 (request in 24)
2	21.675996	10.10.2.2	10.10.2.3	ICMP	102	Echo (ping) request id=0x2908, seq=5/1280, ttl=64 (reply in 27)
2	21.676139	10.10.2.3	10.10.2.2	ICMP	102	Echo (ping) reply id=0x2908, seq=5/1280, ttl=64 (request in 26)
2	24.887921	Private_66:68:02	Broadcast	ARP	68	Who has 10.10.2.1? Tell 10.10.2.2
2	25.887962	Private_66:68:02	Broadcast	ARP	68	Who has 10.10.2.1? Tell 10.10.2.2
2	26.889186	Private_66:68:02	Broadcast	ARP	68	Who has 10.10.2.1? Tell 10.10.2.2

Esercizio - 7

Definiamo un meccanismo di routing per l'interconnessione tra 2 VLAN. Allo scopo useremo un router in configurazione router-on-stick, che permette di usare un'unica interfaccia fisica del router sviluppando molteplici interfacce logiche da utilizzare come default gateway per le varie VLAN da far comunicare.

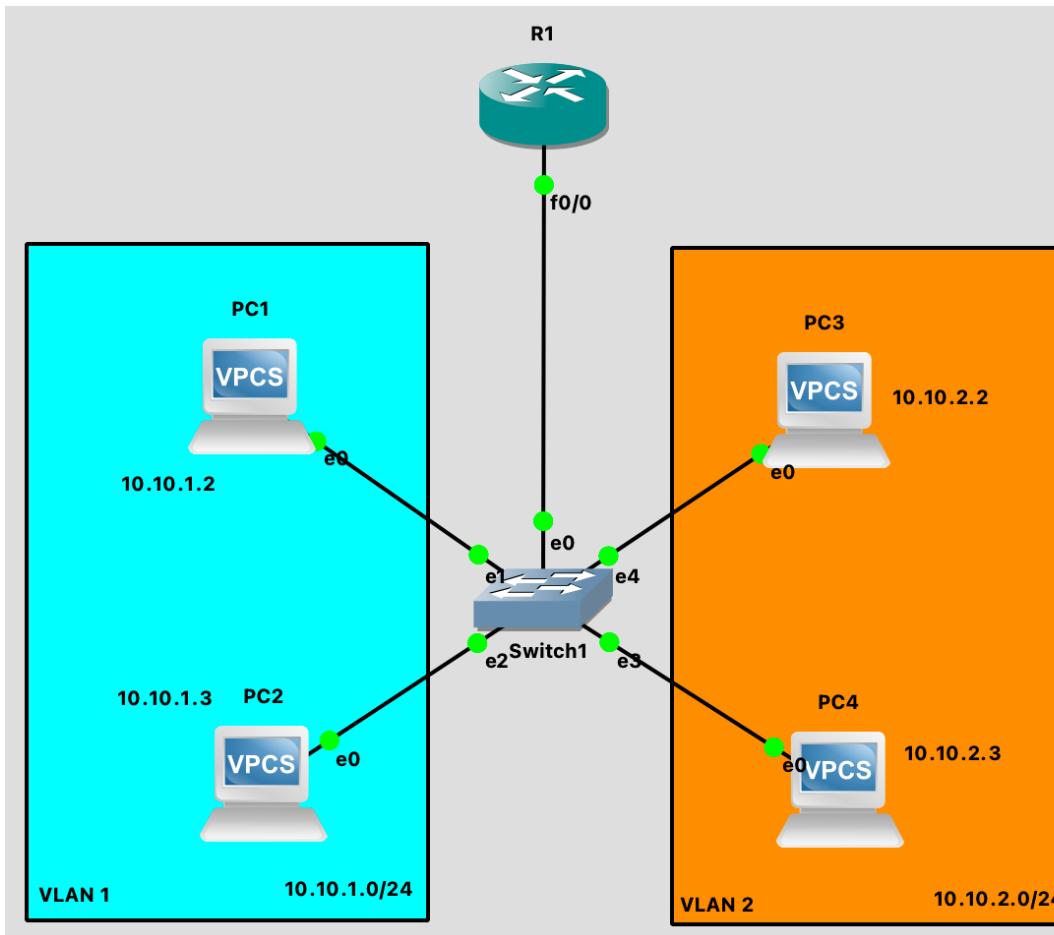
Allo scopo bisogna installare il terminale CISCO 3725 in GNS3. Ciò è possibile andando su File -> import appliance e selezionando il file cisco-3725.gns3a.



Successivamente, sarà chiesto un file mancante c3725-adventurek9-mz.124-15.T14.image, che può essere selezionato andando su Import. Alla conclusione della configurazione un nuovo elemento di routing è presente nel menù.

Esercizio - 7

Ricrea la seguente topologia in GNS3. Scegli il server "GNS3 VM" per istanziare tutti i dispositivi di questo laboratorio.



Tutti i dispositivi vanno configurati come prima.

Esercizio - 7

Il router va configurato da Console, andando a definire le due interfacce logiche, come segue:

```
|R1#config
|Configuring from terminal, memory, or network [terminal]? t
Enter configuration commands, one per line. End with CNTL/Z.
|R1(config)#interface fastEthernet 0/0
|R1(config-if)#no shutdown
|R1(config-if)#exit
|R1(config)#
 *Mar 1 00:04:49.255: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
 *Mar 1 00:04:50.255: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
|R1(config)#interface fastEthernet 0/0.1
|R1(config-subif)#encapsulation dot1Q 1
|R1(config-subif)#ip address
 *Mar 1 00:05:15.499: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
|R1(config-subif)#ip address 10.10.1.1 255.255.255.0
|R1(config-subif)#exit
|R1(config)#interface fastEthernet 0/0.2
|R1(config-subif)#encapsulation dot1Q 2
|R1(config-subif)#ip address 10.10.2.1 255.255.255.0
|R1(config-subif)#exit

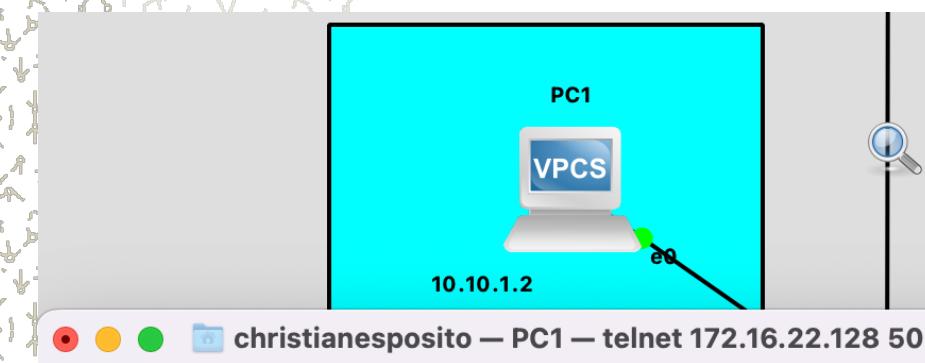
|R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

          10.0.0.0/24 is subnetted, 2 subnets
C            10.10.1.0 is directly connected, FastEthernet0/0.1
C            10.10.2.0 is directly connected, FastEthernet0/0.2
R1#
```

Esercizio - 7

Adesso, si provi il ping di PC1 verso PC2 e PC3, abilitando la cattura sul collegamento tra SW1 e R1.



Executing the startup file

Checking for duplicate address...
PC1 : 10.10.1.2 255.255.255.0 gateway 10.10.1.1

[PC1]> ping 10.10.1.3

```
84 bytes from 10.10.1.3 icmp_seq=1 ttl=64 time=0.287 ms
84 bytes from 10.10.1.3 icmp_seq=2 ttl=64 time=0.179 ms
84 bytes from 10.10.1.3 icmp_seq=3 ttl=64 time=0.284 ms
84 bytes from 10.10.1.3 icmp_seq=4 ttl=64 time=0.207 ms
84 bytes from 10.10.1.3 icmp_seq=5 ttl=64 time=0.201 ms
```

[PC1]> ping 10.10.2.2

```
10.10.2.2 icmp_seq=1 timeout
84 bytes from 10.10.2.2 icmp_seq=2 ttl=63 time=16.953 ms
84 bytes from 10.10.2.2 icmp_seq=3 ttl=63 time=13.079 ms
84 bytes from 10.10.2.2 icmp_seq=4 ttl=63 time=20.470 ms
84 bytes from 10.10.2.2 icmp_seq=5 ttl=63 time=21.246 ms
```

PC1>

Si vede che PC1 riesce a comunicare con PC3, ma dopo un pacchetto in timeout riuscirà anche a comunicare con PC2.

Esercizio - 7

Sulla cattura fatta da Wireshark vediamo che il primo messaggio va in timeout per la risoluzione dell'indirizzo fisico con ARP, mentre negli altri casi il ping si propaga dalla VLAN 1 a quella 2.

VLAN Tag	Time	Source	Destination	Protocol	Length	Info
	0.000000	c2:01:05:cd:00:00	c2:01:05:cd:00:00	LOOP	60	Reply
	7.990363	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.3? Tell 10.10.1.2
	10.003014	c2:01:05:cd:00:00	c2:01:05:cd:00:00	LOOP	60	Reply
	13.557267	c2:01:05:cd:00:00	CDP/VT/PDTP/PAgP/U...	CDP	363	Device ID: R1 Port ID: FastEthernet0/0.1
	19.227386	Private_66:68:00	Broadcast	ARP	64	Who has 10.10.1.1? Tell 10.10.1.2
	19.234560	c2:01:05:cd:00:00	Private_66:68:00	ARP	60	10.10.1.1 is at c2:01:05:cd:00:00
	19.235820	10.10.1.2	10.10.2.2	ICMP	98	Echo (ping) request id=0x5a1f, seq=1/256, ttl=64 (no response found!)
2	19.244426	c2:01:05:cd:00:00	Broadcast	ARP	64	Who has 10.10.2.2? Tell 10.10.2.1
2	19.244730	Private_66:68:02	c2:01:05:cd:00:00	ARP	64	10.10.2.2 is at 00:50:79:66:68:02
	20.002962	c2:01:05:cd:00:00	c2:01:05:cd:00:00	LOOP	60	Reply
	21.236911	10.10.1.2	10.10.2.2	ICMP	98	Echo (ping) request id=0x5c1f, seq=2/512, ttl=64 (no response found!)
2	21.243321	10.10.1.2	10.10.2.2	ICMP	102	Echo (ping) request id=0x5c1f, seq=2/512, ttl=63 (reply in 13)
2	21.243515	10.10.2.2	10.10.1.2	ICMP	102	Echo (ping) reply id=0x5c1f, seq=2/512, ttl=64 (request in 12)
	21.253713	10.10.2.2	10.10.1.2	ICMP	98	Echo (ping) reply id=0x5c1f, seq=2/512, ttl=63
	22.254112	10.10.1.2	10.10.2.2	ICMP	98	Echo (ping) request id=0x5d1f, seq=3/768, ttl=64 (no response found!)
2	22.256720	10.10.1.2	10.10.2.2	ICMP	102	Echo (ping) request id=0x5d1f, seq=3/768, ttl=63 (reply in 17)
2	22.256882	10.10.2.2	10.10.1.2	ICMP	102	Echo (ping) reply id=0x5d1f, seq=3/768, ttl=64 (request in 16)
	22.267008	10.10.2.2	10.10.1.2	ICMP	98	Echo (ping) reply id=0x5d1f, seq=3/768, ttl=63
	23.268135	10.10.1.2	10.10.2.2	ICMP	98	Echo (ping) request id=0x5e1f, seq=4/1024, ttl=64 (no response found!)
2	23.278226	10.10.1.2	10.10.2.2	ICMP	102	Echo (ping) request id=0x5e1f, seq=4/1024, ttl=63 (reply in 21)
2	23.278323	10.10.2.2	10.10.1.2	ICMP	102	Echo (ping) reply id=0x5e1f, seq=4/1024, ttl=64 (request in 20)
	23.288390	10.10.2.2	10.10.1.2	ICMP	98	Echo (ping) reply id=0x5e1f, seq=4/1024, ttl=63
	24.290070	10.10.1.2	10.10.2.2	ICMP	98	Echo (ping) request id=0x5f1f, seq=5/1280, ttl=64 (no response found!)
2	24.300944	10.10.1.2	10.10.2.2	ICMP	102	Echo (ping) request id=0x5f1f, seq=5/1280, ttl=63 (reply in 25)
2	24.301036	10.10.2.2	10.10.1.2	ICMP	102	Echo (ping) reply id=0x5f1f, seq=5/1280, ttl=64 (request in 24)
	24.311109	10.10.2.2	10.10.1.2	ICMP	98	Echo (ping) reply id=0x5f1f, seq=5/1280, ttl=63
	30.004971	c2:01:05:cd:00:00	c2:01:05:cd:00:00	LOOP	60	Reply