

MININET

Mininet è un emulatore di rete open source progettato per consentire la creazione, la configurazione e il test di reti di computer software-defined (SDN). SDN è un paradigma di rete che separa il controllo della rete dalla trasmissione dei dati, consentendo una maggiore programmabilità e flessibilità nelle reti.

Mininet crea una rete virtuale composta da host, switch e controller SDN all'interno di un singolo sistema Linux. Gli host sono rappresentati da processi Linux isolati, mentre gli switch e i controller SDN sono eseguiti come processi all'interno dello stesso sistema. Ciò consente agli sviluppatori e agli operatori di testare e sperimentare con reti SDN in un ambiente controllato e riproducibile, senza la necessità di hardware fisico.

Gli utenti possono definire topologie di rete personalizzate, specificando il numero di host, switch, router e collegamenti desiderati. Ciò consente di creare scenari di rete realistici e variabili.

Mininet è spesso utilizzato in combinazione con il protocollo OpenFlow, che permette la programmazione del comportamento dei switch di rete. Questo è particolarmente utile per testare e sperimentare con reti definite dal software (SDN).

Gli utenti possono creare script Python per automatizzare la creazione di topologie, l'avvio della rete, la configurazione degli host e l'esecuzione di test. Questo offre un alto grado di flessibilità e controllo.

Funzionamento:

- **Creazione della Topologia:**
Viene definita la struttura della rete specificando gli host, gli switch e i collegamenti desiderati attraverso script Python o interfacce utente dedicate.
- **Avvio della Rete:**
Una volta definita la topologia, Mininet avvia la rete virtuale emulando gli host e gli switch attraverso container leggeri.
- **Interazione con la Rete:**
Gli utenti possono quindi interagire con la rete emulata come se fosse una rete fisica. Possono eseguire comandi da terminale, configurare le interfacce di rete degli host e monitorare il traffico di rete.

Come installare Mininet.

Opzione 1: Installazione di Mininet tramite VM

Scarica un'immagine VM di Mininet da <https://github.com/mininet/mininet/releases/>

Scarica e installa un sistema di virtualizzazione, come Virtual Box.

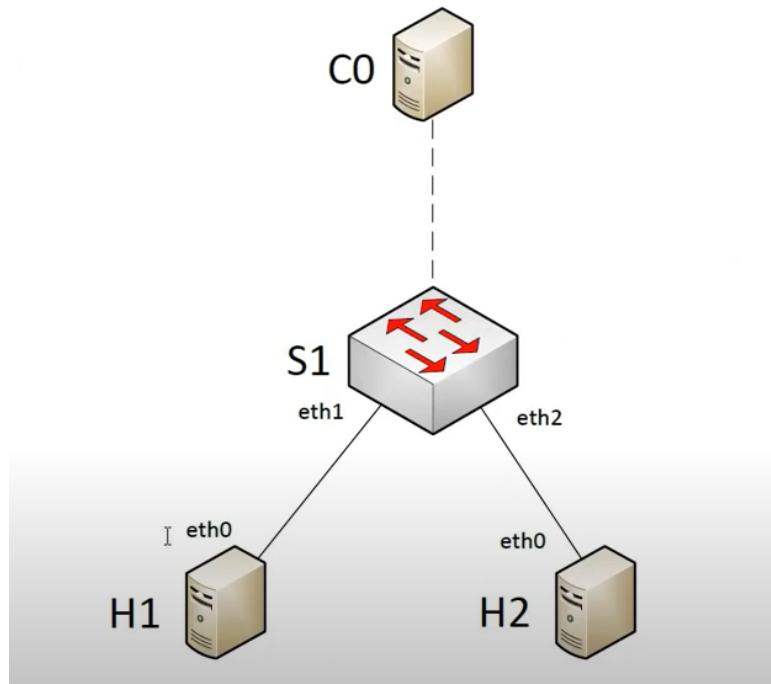
Opzione 2: Installazione nativa dal codice sorgente

```
git clone https://github.com/mininet/mininet
```

```
cd mininet
sudo ./util/install.sh -a
```

Dopo l'installazione, è possibile verificare se Mininet è stato installato correttamente eseguendo un esempio di topologia. Ad esempio:

```
sudo mn --test pingall
```



```
ilenia@ilenia-VirtualBox:~/mininet$ sudo mn --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.523 seconds
```

Creazione di una topologia.

Si può creare una topologia personalizzata utilizzando uno script Python che definisce gli host, gli switch e la loro connessione.

Esempio:

```
from mininet.topo import Topo
from mininet.net import Mininet
import pickle

class MyTopology(Topo):
    def __init__(self):
        super(MyTopology, self).__init__()

        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        s1 = self.addSwitch('s1')

        self.addLink(h1, s1)
        self.addLink(h2, s1)

if __name__ == '__main__':
    mytopo = MyTopology()

    net = Mininet(topo=mytopo)
    net.start()

    # Salvataggio della topologia in un file utilizzando pickle
    with open('mytopo.pkl', 'wb') as f:
        pickle.dump(mytopo, f)

    h1 = net.get('h1')
    h2 = net.get('h2')
    h1.cmd('ping -c 3', h2.IP())

    net.stop()
```

Viene definita una classe MyTopology che eredita dalla classe Topo di Mininet. All'interno del costruttore `__init__`, vengono aggiunti due host (h1 e h2) e uno switch (s1) alla topologia. Successivamente, vengono aggiunte due connessioni tra gli host e lo switch.

Salvare lo script come topo2.py

- Alternativa

Si può creare una topologia anche attraverso riga di comando.

Il seguente comando:

- **sudo mn -topo linear, 3**

Crea una topologia lineare, con 3 switch connessi tra loro con un single host.
Mentre il seguente:

- **sudo mn -topo single, 3**

Crea un singolo switch con 3 host connessi ad esso.

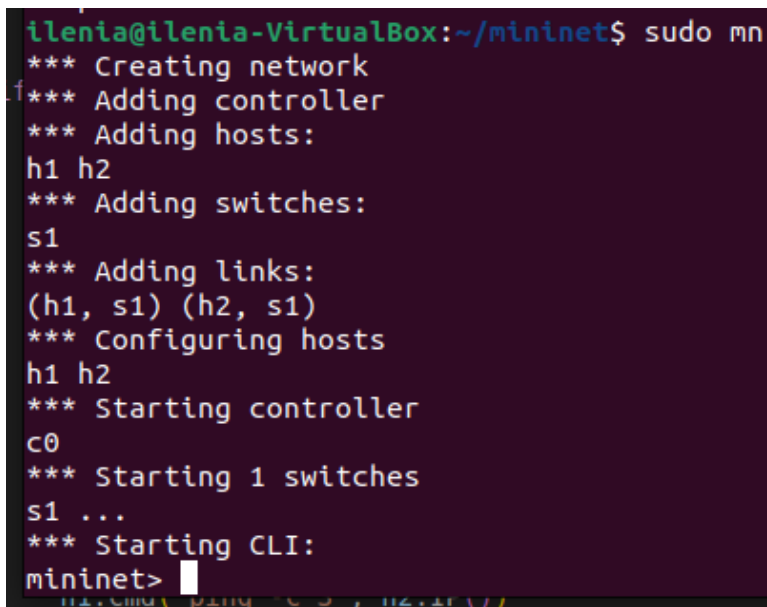
Avvio rete con Mininet.

Esguire:

sudo python mytopo.py

Verificare che la rete sia stata creata correttamente eseguendo il comando:

sudo mn



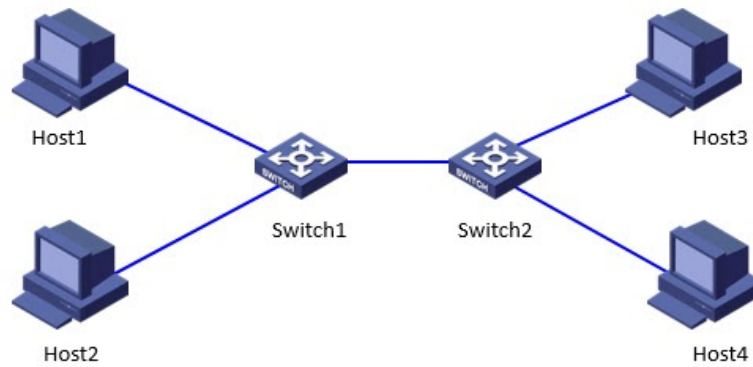
```
ilenia@ilenia-VirtualBox:~/mininet$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Adesso si può interagire con la rete direttamente dalla console di Mininet, come ad esempio eseguendo il comando di ping tra gli host:

```
mininet> h1 ping -c 3 h2
```

Altri esempi che si possono fare:

Per generare la rete riportata in figura



Codice:

```

from mininet.topo import Topo
class MyTopo( Topo ):
    "Simple topology example."
    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        Host1 = self.addHost( 'h1' )
        Host2 = self.addHost( 'h2' )
        Host3 = self.addHost( 'h3' )
        Host4 = self.addHost( 'h4' )
        Switch1 = self.addSwitch('s1')
        Switch2 = self.addSwitch('s2')
        # Add links
        self.addLink( Host1, Switch1 )
        self.addLink( Host2, Switch1 )
        self.addLink( Host3, Switch2 )
        self.addLink( Host4, Switch2 )
        self.addLink( Switch1, Switch2 )
topos = { 'mytopo': ( lambda: MyTopo() ) }

```

Una volta salvato lo script, per esempio 'topo3.py'
 eseguire il comando:

```
sudo mn --custom .topo3.py --topo=mytopo
```

E con il comando '*nodes*' possiamo vedere un elenco dei nodi presenti nella topologia della rete.

```
ilenia@ilenia-VirtualBox:~/mininet_scripts$ sudo mn --custom ./topo3.py --topo=mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 s1 s2
mininet>
```

Usare il comando *'links'* per ottenere informazioni sul collegamento, come i nodi collegati e altre proprietà del collegamento stesso.

```
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s2-eth1 (OK OK)
h4-eth0<->s2-eth2 (OK OK)
s1-eth3<->s2-eth3 (OK OK)
```