



# AVALANCHE

## Contship - Subnet - Nodo

Questa documentazione mira a delineare i procedimenti necessari per configurare un nodo validatore sulla rete di test Fudji di Avalanche. Saranno trattati vari argomenti :

- La definizione e ruolo dei nodi
- L'installazione dei requisiti preliminari
- L'avvio del processo di bootstrap
- La conversione di un nodo standard in un nodo validatore
- Aggiunta del nodo al subnet

### La definizione e ruolo dei nodi

La blockchain è sostenuta da operatori noti come **"nodi"**. Questi elementi sono fondamentali per la decentralizzazione di una blockchain, in quanto ogni istanza della blockchain è replicata in un nodo. Esistono due tipologie principali di nodi: **nodi normali e nodi validatori**. Un nodo normale ha la capacità di leggere le informazioni presenti sulla blockchain. In contrasto, un nodo validatore partecipa attivamente all'algoritmo di consenso della blockchain, assumendosi la responsabilità di validare i blocchi.

### L'installazione dei requisiti preliminari

Per iniziare l'implementazione dei nodi, utilizzeremo i **binari precompilati** forniti da Avalanche. È cruciale mantenere i binari **aggiornati** per assicurare l'efficienza e la sicurezza del sistema. Per procedere con l'installazione, è possibile visitare il seguente indirizzo: [Releases · a](https://releases.lvlabs.io/avalanche) [va-labs/avalanchego](https://releases.lvlabs.io/avalanche) . Qui, è necessario selezionare la versione desiderata cliccando su di essa. Successivamente, scorrendo fino alla fine della pagina, sotto la sezione "Assets", è possibile scaricare i binari adeguati per **Linux, macOS e Windows**. Nel contesto del nostro progetto, abbiamo già effettuato il download dei binari nella versione **v1.11.0**, attualmente la più recente, per i sistemi operativi macOS e Linux. Per le fasi successive, si dovrà accedere alla cartella denominata `avalanche-node` .

### L'avvio del processo di bootstrap

Il processo di bootstrap è fondamentale per **sincronizzare** il nostro nodo con la blockchain di Avalanche, assicurando che disponga dell'intera catena di blocchi **aggiornata**. Questa fase rappresenta il processo più prolungato e la sua durata varia in base alla rete specifica che si intende bootstrap. Per il nostro obiettivo, il bootstrap sulla rete Fuji di Avalanche richiede approssimativamente da **20 a 50 ore e necessita di almeno 100 GB** di spazio su disco. Per iniziare questo processo, è sufficiente eseguire lo script chiamato `build` nella cartella che rappresenta il vostro OS ( Linux oppure MacOS), nel nostro caso sarà la cartella `macOS` che si trova nella cartella `avalanche-node` nel modo seguente:

```
1 ./node.sh
```

Dopo alcuni secondi, l'output sul terminale dovrebbe indicare che il nodo è stato avviato e che il processo di bootstrap è in corso. A questo punto, è necessario attendere il completamento del processo, che può variare tra le 20 e le 50 ore.

```
1 [02-12|13:32:22.391] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 50000, "n
2 [02-12|13:32:28.387] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 55000, "n
3 [02-12|13:32:34.593] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 60000, "n
4 [02-12|13:32:34.754] INFO health/worker.go:256 check started passing {"namespace": "health", "name": "network",
5 [02-12|13:32:42.263] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 65000, "n
6 [02-12|13:32:57.817] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 70000, "n
7 [02-12|13:33:16.603] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 75000, "n
8 [02-12|13:33:24.230] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 80000, "n
9 [02-12|13:33:33.063] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 85000, "n
10 [02-12|13:33:45.044] INFO <P Chain> bootstrap/bootstrapper.go:651 fetching blocks {"numFetchedBlocks": 90000, "n
```

Per verificare che il nodo si stia **bootstrappando** alla rete desiderata, si può aprire un nuovo terminale ed eseguire il seguente comando sempre nella cartella `macos` :

```
1 ./check-network.sh
```

e dovreste ottenere quest'uscita:

```
1 {"jsonrpc": "2.0", "result": {"networkName": "fuji"}, "id": 1}
```

**i** In questo caso otteniamo **"fuji"** perché abbiamo lanciato il processo di **bootstrap** su di essa.

Per verificare il completamento del processo di bootstrap, si può aprire un nuovo terminale ed eseguire il seguente comando sempre nella cartella `macos` :

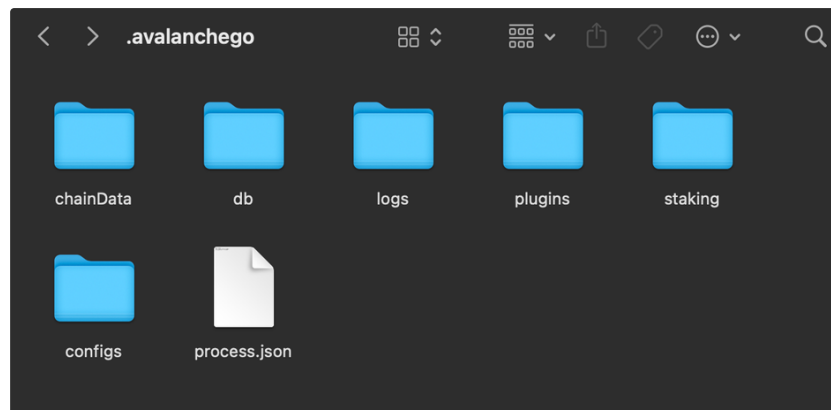
```
1 ./check-bootstrap.sh
```

e dovreste ottenere quest'uscita :

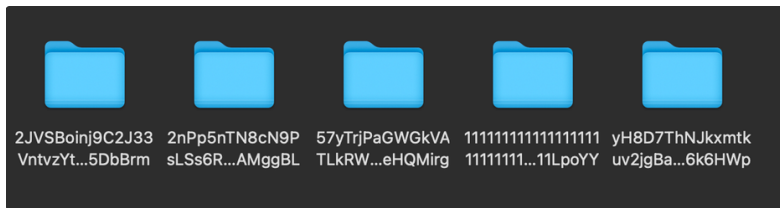
```
1 {"jsonrpc": "2.0", "result": {"isBootstrapped": true}, "id": 1}
```

L'output atteso confermerà che il nodo è stato completamente sincronizzato con la rete Fuji di Avalanche. Tuttavia, è importante notare che, a questo stadio, il nodo **non è ancora configurato come nodo validatore**.

All'avvio del nodo per la prima volta, si creerà una cartella `~/avalanchego` che avrà più o meno queste cartelle :



**chainData**: questa cartella contiene tutte le blockchain dove il nodo è validatore, nel nostro caso la blockchain di ContshipNetV1 è :  
57yTrjPaGWGkVATLkRWUercvMpAe1eH6d6zNvHA9H7eHQMirg



**db:** Questa cartella contiene tutta la database della blockchain dove il nodo si è boosttrappato.

**plugins:** Questa cartella contiene la VM per la blockchain di ContshipNetV1. Questa Virtual Machine è un binario che deve essere chiamata esattamente come la VM del subnet !

Su questo link si può andare a scaricare la VM la più recente di Avalanche perché sia compatibile con la versione del nodo : [Releases](#) .

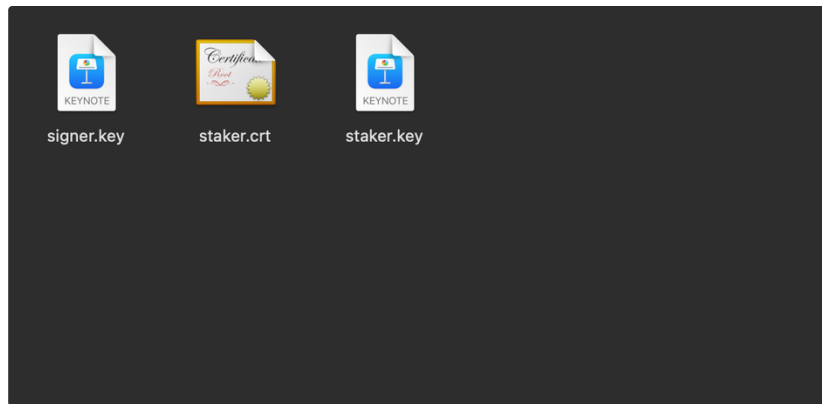
ava-labs/subnet-evm



Qui il file si chiama così perché nel nostro subnet la VM si chiama :  
"WhYY15RH8t2CHtkEa41ZPmML8YQgwUeJhwpmaMjtJR6h94k4pf"

PARAMETER	VALUE
Subnet Name	ContshipNetV1
ChainID	2819
Mainnet ChainID	0
Token Name	TEST
VM Version	v0.5.11
VM ID	WhYY15RH8t2CHtK Ea41ZPmML8YQgwUeJhwp aMjtjR6h94k4pf
Fuji SubnetID	2Cv7gS1bxTNVD1rDLFRVNuCsP2WSYaFwfaJVs7RXy4pBUi3MdV
Fuji BlockchainID	57yTrjPaGWGkVATLkRWUercvMpAe1eH6d6zNvHA9H7eHQMirg

**stacking:** Questa cartella contiene l'identità del nodo e va protetta :



⚠ Se si vuole trasferire il nodo su un'altro server, è molto importante salvare tutti i file contenuti nella cartella stacking and anche salvare la DB se non si vuole ricominciare tutto il bootstrap. Per ricreare il nodo su un'altro server basta eseguire `./bootstrap-node.sh` e rimpiazzare i file nella cartella staking del nuovo nodo con i staking file del nodo precedente e la db del nuovo nodo con la db del nodo precedente

⚠ Se non vedete le stesse cartelle potete crearle a mano chiamare i file come sono stati chiamati nei screenshot per avere la configurazione necessaria per collegarsi al subnet

## [-] La conversione di un nodo standard in un nodo validatore

### 1. Installazione della Command Line Interface ( CLI ) di Avalanche.

Per trasformare un nodo normale in un nodo validatore sulla rete Fudji di Avalanche, è fondamentale iniziare installando la Command Line Interface (CLI) di Avalanche. Questo strumento facilita la gestione del nodo e la sua configurazione come validatore.

Per iniziare, bisogna navigare nella cartella `subnet` ed eseguire questo script :

```
1 ./dependency.sh
```

Una volta finito, dovresti trovarvi con quest'uscita :

```
1 ava-labs/avalanche-cli info checking GitHub for latest tag
2 ava-labs/avalanche-cli info found version: 1.3.7 for darwin/arm64
3 ava-labs/avalanche-cli info installed /Users/patrickdev/bin/avalanche
4 warning: brew bash-completion package not found. avalanche command completion for bash not installed
5 Installation complete! Please restart your terminal or run 'source /Users/patrickdev/.bashrc' to apply the change
```

Per finalizzare l'installazione, bisogna eseguire questa linea `source /Users/patrickdev/.bashrc` .

⚠ questa linea sarà diversa nel vostro caso con un path diverso.

Dopo l'installazione, è possibile verificare che la CLI di Avalanche sia stata installata correttamente eseguendo un comando che ne stampa la versione o un comando di aiuto. Ad esempio:

```
1 avalanche --version
```

E questa dovrebbe essere l'uscita :

```
1 avalanche version 1.3.7
```

### 2. Creazione di un wallet Core e delle chiavi Avalanche.

Core rappresenta un portafoglio digitale specificamente progettato per l'ecosistema Avalanche, essenziale per la configurazione di subnet e nodi all'interno di questa rete. Per lo sviluppo di smart contract, invece, il portafoglio Metamask risulta adeguato e sufficiente.

**a. Installazione dell'Estensione Core:**

Per iniziare, è necessario installare l'estensione Core per Google Chrome. Questo può essere fatto accedendo al Chrome Web Store e cercando "Core Crypto Wallet" oppure seguendo il link fornito <https://chromewebstore.google.com/detail/core-crypto-wallet-nft-ex/agoakfejjabomempkjlepdlaleebhb>. Una volta trovata l'estensione, selezionare "Aggiungi a Chrome" per procedere con l'installazione.

**b. Creazione del Wallet:**

Dopo l'installazione dell'estensione Core, il passo successivo è la creazione di un nuovo wallet. Questo processo tipicamente include la generazione di una frase segreta, che deve essere conservata con la massima cura, in quanto rappresenta la chiave di accesso ai fondi e ai token.

**c. Generazione delle Chiavi Avalanche:**

Con il wallet Core ora configurato, è possibile procedere con la generazione delle chiavi specifiche per l'ecosistema Avalanche. Questo è un passaggio fondamentale per interagire con la rete, partecipare alla validazione di transazioni o allo sviluppo di applicazioni decentralizzate. La generazione di una chiave pubblica personalizzata può essere effettuata attraverso l'uso di questo comando :

```
1 avalanche key create ContshipDocumentazione
```

⚠ Al posto di "ContshipDocumentazione" si può mettere un'altro nome, per la documentazione manterremo questo esempio

L'output atteso si presenta così :

```
1 Generating new key...
2 Key created
```

Adesso possiamo vedere la lista delle chiavi che abbiamo creato eseguendo questo comando :

```
1 avalanche key list
```

Vi si chiederà di scegliere una rete, nel nostro caso scegliamo la rete Fuji.

```
1 Use the arrow keys to navigate: ↓ ↑ → ←
2 ? Choose network for which to list addresses:
3   Mainnet
4   ► Fuji
5   Local Network
```

E questa è la lista delle chiavi che avete creato :

1	+-----+-----+-----+-----+							
2		KIND		NAME		CHAIN		ADDRESS
3	+-----+-----+-----+-----+							
4		stored		ContshipDocumentazione		C-Chain (Ethereum hex format)		0xA0A15211cC5613DD51c8746106a74dCCa088525c
5	+-----+-----+-----+-----+							
6						P-Chain (Bech32 format)		P-fuji16e6nad8qm4f62utsje6huk7qx2x139fnstvc4
7	+-----+-----+-----+-----+							
8				PatrickAdmin		C-Chain (Ethereum hex format)		0x7474cE7b605687942543fc9789fbB56BE6849D13
9	+-----+-----+-----+-----+							
10						P-Chain (Bech32 format)		P-fuji1h6gzhmeefq9yw7nudv5a9msmkgfymwuzm5r5r
11	+-----+-----+-----+-----+							
12				Patricktest		C-Chain (Ethereum hex format)		0xb83f5b4A2dCd63eDAF97BF252bAdC122b85Df818
13	+-----+-----+-----+-----+							
14						P-Chain (Bech32 format)		P-fuji1ymep62cuxhzuey9sw30z852g2myme4jd3lr6t
15	+-----+-----+-----+-----+							

Attualmente, si osserva che il saldo dei token sulla C-Chain e sulla P-Chain è pari a zero, condizione che rientra nella norma per una nuova configurazione. Questo stato iniziale conferma l'assenza di fondi associati alla chiave appena creata, ma allo stesso tempo garantisce l'esistenza e la corretta creazione della chiave stessa nel nostro wallet.

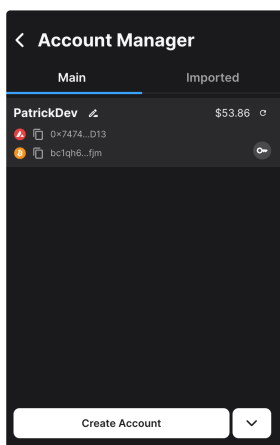
Per procedere ulteriormente, è ora necessario estrarre la chiave privata associata al nostro set di chiavi denominato "ContshipDocumentazione". Questo passaggio è fondamentale per garantire l'accesso e la gestione sicura delle risorse digitali. Per effettuare l'estrazione della chiave privata, è richiesto l'esecuzione del seguente comando:

```
1 avalanche key export ContshipDocumentazione
```

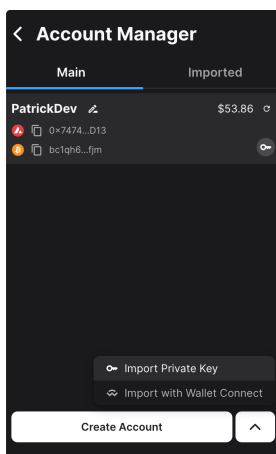
Ed otterrete la chiave privata che assomiglia a questo :

```
1 93183b62bb06dc34a74a4f7b2ae794bf6d702267fc63170b7a1e5f57b8f9ddb0
```

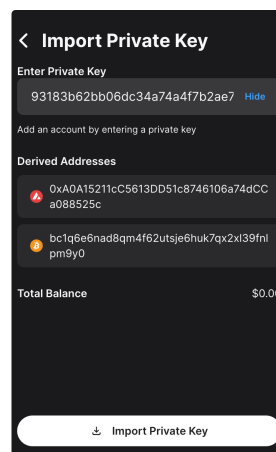
Dopo aver estratto con successo la chiave privata, il passaggio successivo consiste nell'importare questo account nel wallet Core precedentemente configurato.



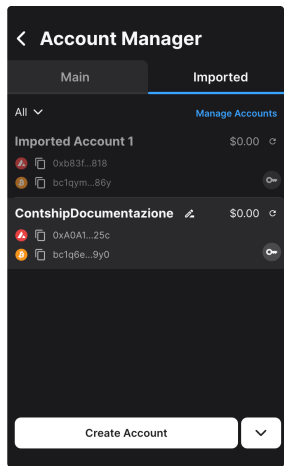
1. Qui si deve cliccare sulla freccia girata verso il basso vicino al "Create Account"



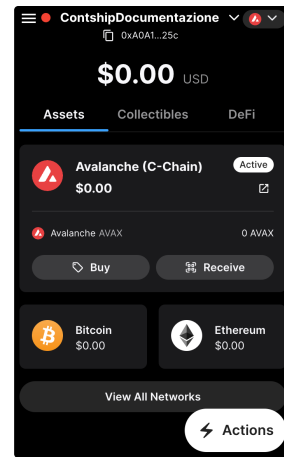
2. Poi si seleziona "Import Private Key"



3. Qui si può incollare la chiave privata ottenuta prima e cliccare su "Import Private Key"



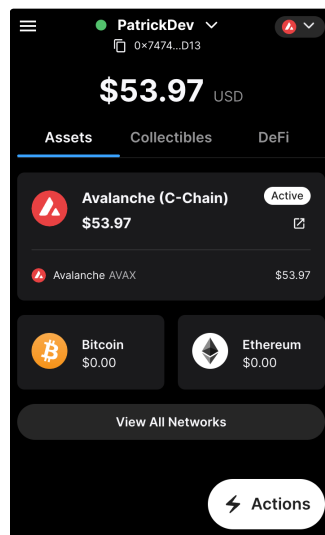
4. Qui potrete visualizzare tutte le chiavi che avete importato. Adesso basta cliccare 2 volte sull'account che si desidera accedere.



5. Questo è l'account che è collegato alla rete Avalanche.

Abbiamo ora stabilito con successo un account e un wallet connessi alla rete Avalanche. Il passo successivo nel processo di configurazione e test consiste nell'acquisire "Test Token" utilizzabili all'interno dell'ambiente di test di Avalanche. Questi token possono essere richiesti direttamente tramite il faucet ufficiale di Avalanche, accessibile al seguente indirizzo: [Avalanche Testnet Faucet | Core](#)

Esistono varie modalità per ottenere questi test token. Se disponete già di AVAX sulla rete principale, potete utilizzare direttamente tale saldo per ricevere i test token. In alternativa, per coloro che non possiedono AVAX, è possibile richiedere un coupon attraverso il canale Discord ufficiale di Avalanche.

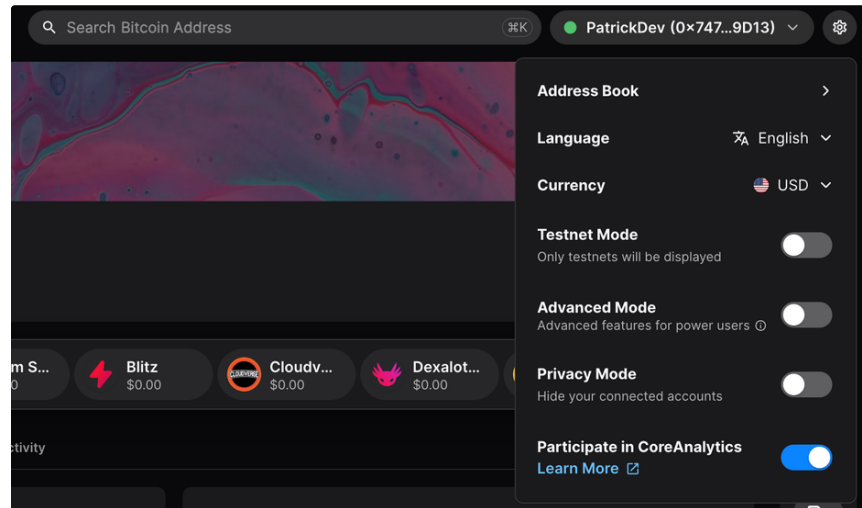


Per la documentazione userò il mio vero account con dei AVAX sulla rete principale.

Adesso, bisogna andare sul sito Core : [The Best Way to Connect to Web3 | Core](#) e connettere il vostro wallet qui :



Una volta il wallet connesso, dobbiamo metterci in modalità "Test". Per farsi dobbiamo cliccare sul logo delle impostazioni vicino al nome del wallet et attivare il modo "Testnet Mode" :

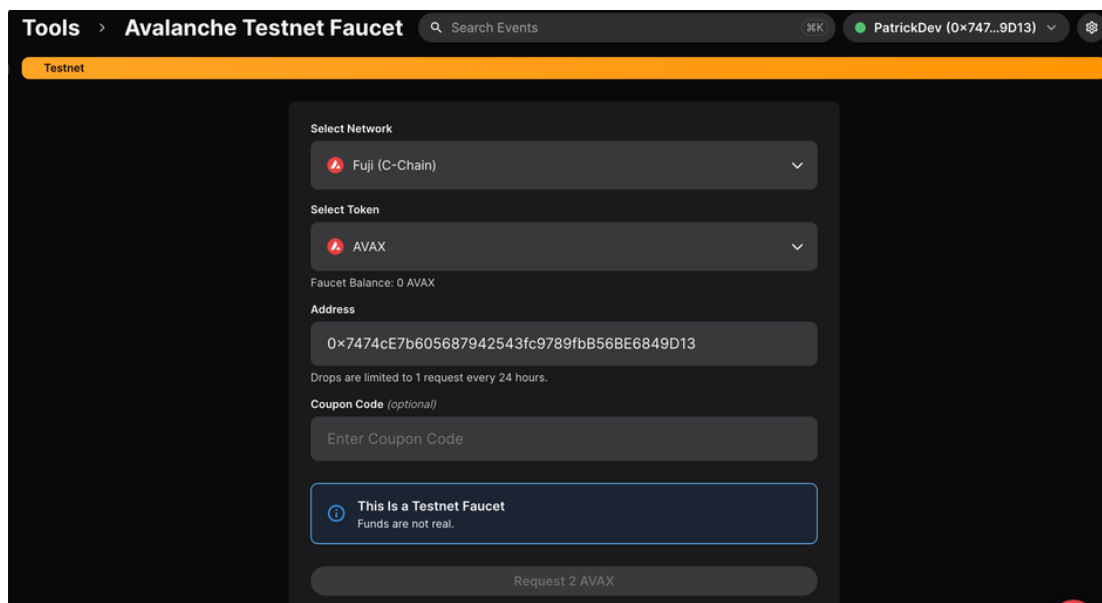


Questo fara apparire una transazione che cambierà anche la modalità del vostro wallet. Si deve accettare tutto.

Una volta fatto questo, si deve andare alla sinistra della pagina e cliccare su "Tools" e poi "Testnet Faucet". Se tutto e andato bene, senza fare nulla dovreste vedere una schermata cosi :

⚠ Come evidenziato dalla seguente schermata, si nota la dicitura "Faucet Balance: 0 AVAX", indicando l'assenza di test token disponibili al momento. In questa situazione, una soluzione praticabile è quella di rivolgersi alla comunità di Avalanche tramite il canale Discord ufficiale : [Discord - A New Way to Chat with Friends & Communities](#) . Qui, è possibile richiedere cortesemente agli amministratori di inviare test token al vostro wallet, fornendo loro l'indirizzo del vostro wallet per facilitare il trasferimento.

Questo approccio è stato adottato per la preparazione della presente documentazione. Alternativamente, qualora il faucet disponesse di test token, l'ottenimento di questi ultimi sarebbe semplice e diretto: sarebbe sufficiente cliccare su "Request 2 AVAX" per ricevere i token sull'account specificato.



Una volta ricevuti i token, possiamo eseguire di nuovo questo comando per verificare se tutto e impostato correttamente :



```

1 avalanche key list
2
3 ✓ Fuji
4 +-----+-----+-----+-----+
5 |  KIND  |           NAME           |           CHAIN           |           ADDRESS           |
6 +-----+-----+-----+-----+
7 | stored | ContshipDocumentazione | C-Chain (Ethereum hex format) | 0xA0A15211cC5613DD51c8746106a74dCCa088525c
8 +      +      +-----+-----+
9 |      |      | P-Chain (Bech32 format)      | P-fuji16e6nad8qm4f62utsje6huk7qx2x139fnstvc4
10 +     +-----+-----+-----+-----+
11 |      | PatrickAdmin           | C-Chain (Ethereum hex format) | 0x7474cE7b605687942543fc9789fbB56BE6849D13
12 +     +-----+-----+-----+-----+
13 |      |      | P-Chain (Bech32 format)      | P-fuji1h6gzhmeefq9yw7nudv5a9msmkgfymwuzm5r5r
14 +     +-----+-----+-----+-----+
15 |      | Patricktest           | C-Chain (Ethereum hex format) | 0xb83f5b4A2dCd63eDAF97BF252bAdC122b85Df818
16 +     +-----+-----+-----+-----+
17 |      |      | P-Chain (Bech32 format)      | P-fuji1ymep62cuxhzuey9sw30z852g2myme4jd3lr6t
18 +-----+-----+-----+-----+

```

Qui possiamo vedere che la bilancia dei token nella C-Chain è passato da 0 a 0.9. Questo ci informa che l'account ha ricevuto i test token.

d. Trasferta dei token dalla C-Chain alla P-Chain

Adesso, per potere interagire con il nostro nodo, dobbiamo trasferire i nostri token dalla C-Chain alla P-Chain. Per farsi, dobbiamo andare sul sito web di Core : [Core | The Best Way to Connect to Web3](#)

⚠ Non si deve dimenticare di mettere il sito in modalità "Test net mode"

Una volta sul sito, si deve andare a sinistra su Stake e poi scegliere Cross-chain transfer e dovrete avere questa schermata :

CORE

BETA

Portfolio

Stake

Delegate

Validate

Cross-Chain Trans...

Discover

Explorer

Tools

Stake > Cross-Chain Transfer

Search .avax Address

ContshipDocum... (fuji1\_vc4g)

Testnet

Transfer tokens between Exchange (X), Platform (P), and Contract (C) chains. To move tokens between Avalanche, Ethereum, and Bitcoin go to the [Bridge tool](#).

Source Chain

C-Chain

Show Details

↓

Destination Chain

P-Chain

Show Details

You will be prompted to sign 2 separate transactions: one export, followed by one import.

Transfer Amount

0,9983136398

Max

Available Balance: 1 AVAX \$40

Estimated Export Fee 0.00066 AVAX

Import Fee 0.001 AVAX

Estimated Total 0.00166 AVAX

Confirm Transfer

L'ultimo passo da compiere consiste nella selezione delle catene di origine e destinazione per il trasferimento dei token: specificatamente, scegliere la C-Chain come "Source Chain" e la P-Chain come "Destination Chain". Successivamente, è necessario indicare la quantità di token da trasferire. Nel nostro caso, opteremo per il trasferimento dell'intero saldo disponibile. Una volta definite queste opzioni, occorre procedere cliccando su "Confirm Transfer" per iniziare il processo di trasferimento.

```

1 avalanche key list
2 ✓ Fuji
3 +-----+-----+-----+-----+
4 |  KIND  |          NAME          |          CHAIN          |          ADDRESS          |
5 +-----+-----+-----+-----+
6 | stored | ContshipDocumentazione | C-Chain (Ethereum hex format) | 0xA0A15211cC5613DD51c8746106a74dCCa088525c

```

7	+	+	+-----+-----
8			P-Chain (Bech32 format)   P-fuji16e6nad8qm4f62utsje6huk7qx2x139fnstvc4
9	+	+	+-----+-----
10		PatrickAdmin	C-Chain (Ethereum hex format)   0x7474cE7b605687942543fc9789fbB56BE6849D13
11	+	+	+-----+-----
12			P-Chain (Bech32 format)   P-fuji1h6gzhmeefq9yw7nudv5a9msmkgyfmwuzm5r5r
13	+	+	+-----+-----
14		Patricktest	C-Chain (Ethereum hex format)   0xb83f5b4A2dCd63eDAF97BF252bAdC122b85Df818
15	+	+	+-----+-----
16			P-Chain (Bech32 format)   P-fuji1ymep62cuxhzuey9sw30z852g2myme4jd3lr6t
17	+	+	+-----+-----

Una volta completate queste operazioni, è possibile verificare lo stato del trasferimento e il saldo aggiornato delle catene interessate eseguendo nuovamente il comando sopra.

### 3. Conversione nodo standard a nodo validatore

Per questa parte, si deve avviare il nodo boostrapato con lo stesso comando nella cartella `avalanche-node` :

```
1 ./node.sh
```

 Si devono aspettare circa 5 minuti per lasciare al nodo il tempo di avviarsi

Una volta scorsi i 5 minuti, possiamo eseguire questo comando per verificare che il nodo è boostrapato e funzionale :

```
1 ./check-bootstrap.sh
```

Se tutto va bene, dovremo vedere apparire questo :

```
1 {"jsonrpc":"2.0","result":{"isBootstrapped":true},"id":1}
```

Bene ! Adesso dobbiamo recuperare l'ID del nostro nodo, per fare questo si deve eseguire questo comando :

```
1 ./get-id.sh
```

Una volta eseguito questo comando, dovremo trovarci con quest'uscita :

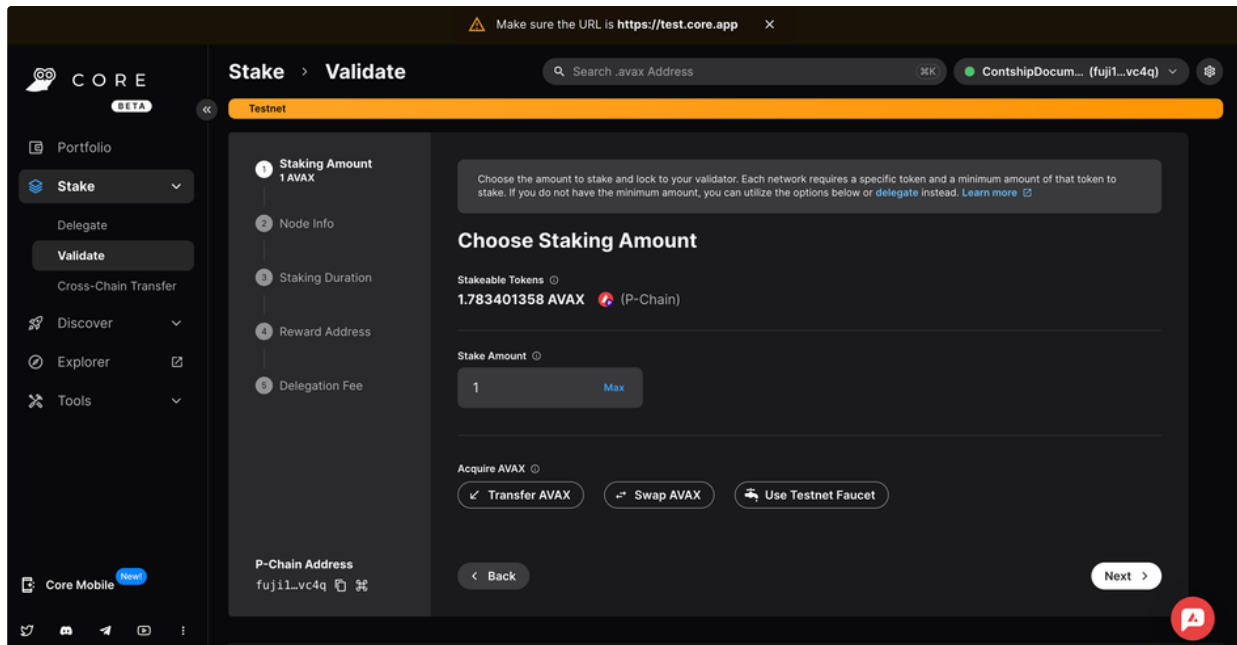
```
1 {"jsonrpc":"2.0","result":{"nodeID":"NodeID-4JwxRwRY8LgeSdDLNargL9NBEBUuvTk9v","nodePOP":{"publicKey":"0xb38828ac
2
```

L'unica informazione che ci importa qui è il node ID, in questo caso : `NodeID-4JwxRwRY8LgeSdDLNargL9NBEBUuvTk9v` .

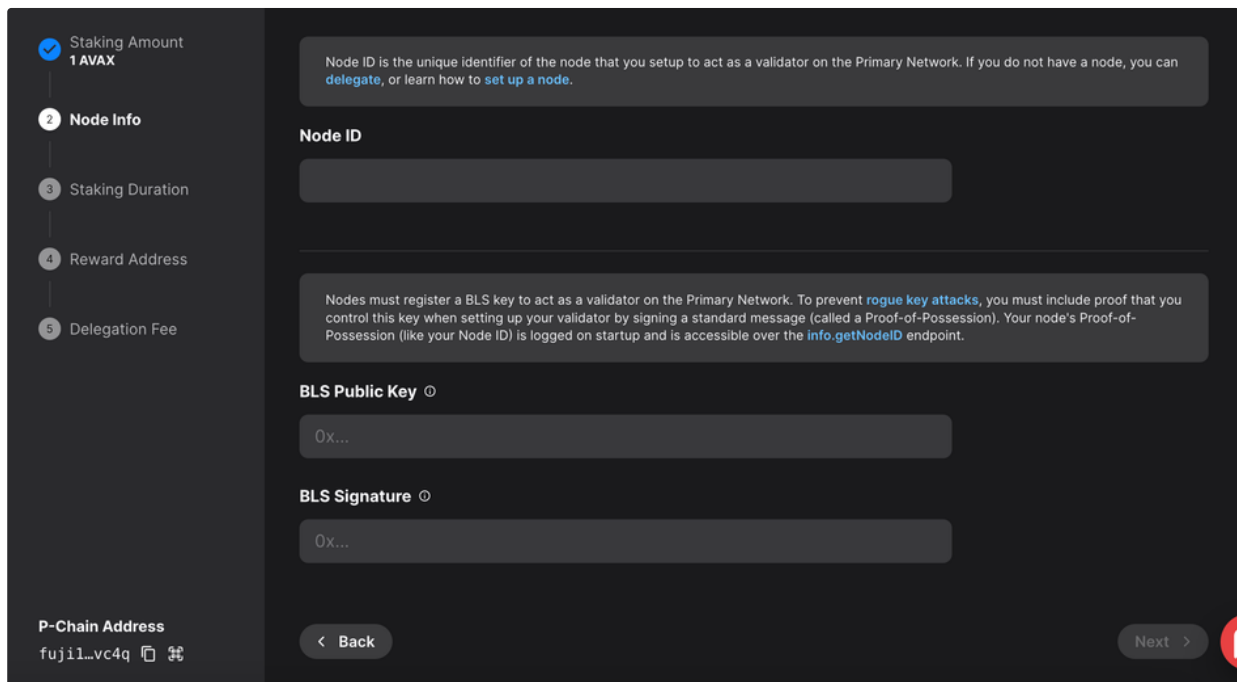
Adesso, dobbiamo andare di nuovo sul sito Core : [🗨️ Core | The Best Way to Connect to Web3](#)

 Non si deve dimenticare di mettere il sito in modalità "Test net mode"

Questa volta andremo su `Stake` e poi su `Validate` . Dovremo essere su questa schermata :



⚠️ **IMPORTANTE** : Per avere un nodo validatore su Fuji, dobbiamo stake al minimo 1 AVAX. Per avere un nodo validatore sulla rete principale, dobbiamo stake al minimo 2000 AVAX. Nel nostro caso andiamo a stake solo 1 AVAX.



Qui possiamo entrare i valori seguenti ottenuti qui :

```
1 {"jsonrpc": "2.0", "result": {"nodeID": "NodeID-4JwxRwRY8LgeSdDLNargL9NBEBUuvTk9v", "nodePOP": {"publicKey": "0xb38828ac0b7599bc73c25e55e4b30739172d08029d8751e38a982190a19ca1f94ae3efab3091b0c15dc1ee161554f64e", "signature": "0x97462e430bc51823aae230157f490ac8b1943ba055cbe03c4da9523eb817d677c46ecdffdafc360b233b689caf7f29900afe2425a953b31668918f14e12d0b07ad86e0110913750517145327bbb46074ae2e8cd3aeb5c3950db3894619c94aef"}}
```

Node ID : NodeID-4JwxRwRY8LgeSdDLNargL9NBEBUuvTk9v ,

BLS Public Key : 0xb38828ac0b7599bc73c25e55e4b30739172d08029d8751e38a982190a19ca1f94ae3efab3091b0c15dc1ee161554f64e ( si trova nella "publicKey" )

BLS Signature :

0x97462e430bc51823aae230157f490ac8b1943ba055cbe03c4da9523eb817d677c46ecdffdafc360b233b689caf7f29900afe2425a953b31668918f14e12d0b07ad86e0110913750517145327bbb46074ae2e8cd3aeb5c3950db3894619c94aef ( si trova nella "proofOfPossession" )



Staking Amount  
1 AVAX

Node Info  
NodeL...Tk9v

Staking Duration  
End Date: 02/11/2025

Reward Address  
fujil...vc4q

Delegation Fee

Other individuals can stake to your validator and receive rewards too, known as "delegating." You can benefit from the rewards they receive by charging a fee for using your validator. [Learn more](#)

### Enter Delegation Fee

Delegation Fee (%)

0.0 %

This is the percentage of delegators' rewards you will charge for use of your node

Choose a wallet address to which delegation rewards will be sent

☒ Connected Wallet
 

fujil6e6nad8qm4f62utsje6huk7qx2xl39fnstvc4q

☐ Custom Address
 

Enter P-Chain address

< Back

Next >

P-Chain Address  
fujil...vc4q

In fine, possiamo entrare una “ Delegation Fee “ che rappresenta una tassa per la gente che desidera usare il nostro nodo per i loro Subnet. In questo caso ho messo 2% di tassa.

Staking Amount  
1 AVAX

Node Info  
NodeL...Tk9v

Staking Duration  
End Date: 02/11/2025

Reward Address  
fujil...vc4q

Delegation Fee  
2%

Rewards Address  
fujil6e6nad8qm4f62utsje6huk7qx2xl39fnstvc4q

Delegation Fee  
2%

Delegation Rewards Address  
fujil6e6nad8qm4f62utsje6huk7qx2xl39fnstvc4q

Delegation Fee Rewards

Max Delegation Capacity

4 AVAX

Max Fee Rewards

0.00880006 AVAX

\$0.00

Validation Rewards

Rewards Percentage

11%

11% annually

Estimated Rewards

0.11000072 AVAX

\$0.00

Max Potential Rewards

0.11880077 AVAX

\$0.00

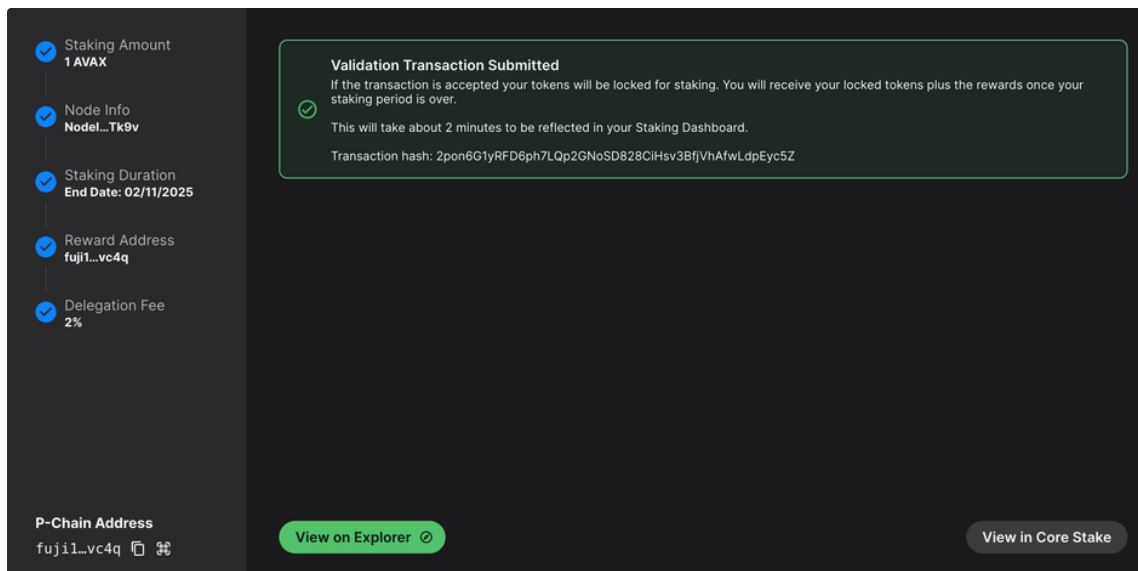
< Back

Start Over

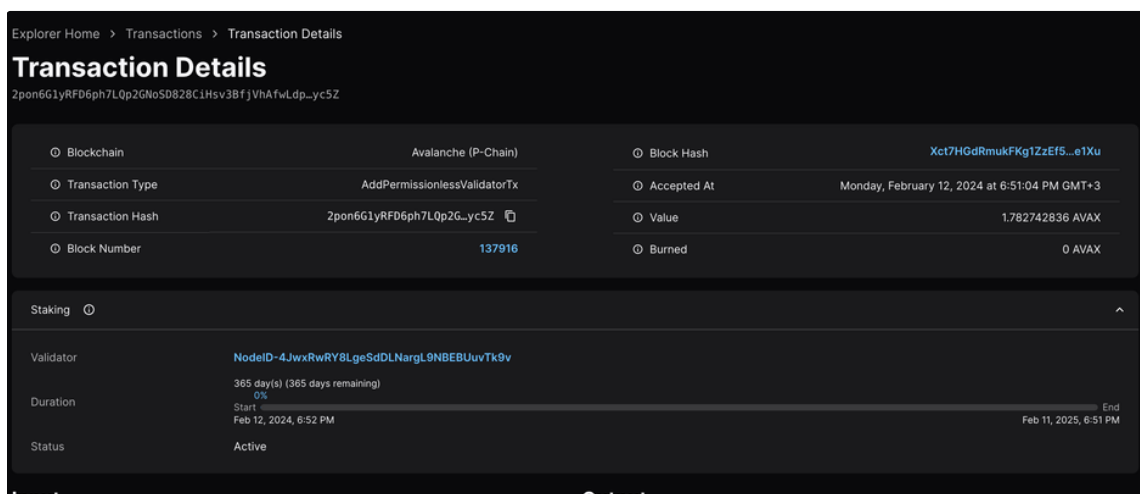
Submit Validation

P-Chain Address  
fujil...vc4q

Alla fine vedremo questa scherma che sarà un recap del nodo che abbiamo configurato. Possiamo cliccare su “Submit Validation” e confermare la transazione.



Una volta confermata la transazione, dovremo vedere apparire questo sulla schermata. Il tempo di attesa per la validazione della transazione è di circa 2 minuti. Se clicchiamo su “ View on Explorer “ e dovremo vedere questa schermata :



Ottimo ! Il nostro nodo è diventato validatore sulla rete Fuji !

## 📄 Aggiunta del nodo al subnet

⚠️ A questo punto, se non avete ancora creato un subnet e lo avete deployato sulla rete Fuji, seguite le istruzioni nella documentazione "Contship - Subnet - Create And Deploy" e tornate qui al termine dell'operazione.

Prima di poter richiedere il permesso di diventare un validatore sul nostro subnet, dobbiamo assicurarci che il nostro nodo sia sincronizzato con la rete Fuji e stia eseguendo l'ultima versione del software Avalanche.

Una volta che il nodo è sincronizzato e in esecuzione, possiamo recarci nella cartella "subnet" e avviare il seguente comando per richiedere il permesso di diventare un validatore sul nostro subnet:

```
1 ./join.sh
```

Successivamente, si dovrà selezionare l'opzione "Fuji" e poi la modalità `automatica`. Il sistema cercherà il file di configurazione, che al momento non esiste, pertanto si potrà indicare una localizzazione a piacere. In seguito, il sistema cercherà la directory `plugins` situata nella cartella `/.avalanhego/plugins`. A questo punto, si potrà cliccare su `yes` per avviare l'installazione della VM. Una volta terminata l'installazione, verrà visualizzata la versione della VM (v0.5.11) e il sistema chiederà se si desidera creare un backup. Anche in questo caso, si consiglia di cliccare su `yes`.

il config file creato dal sistema deve assomigliare a questo :

```
1 {
2   "network-id": "fuji",
3   "track-subnets": "2Cv7gS1bXTNVD1rDLFRVNuCsP2WSYaFwfaJVs7RXY4pBui3MdV"
4 }
```

Prima di tutto, assicurarsi di avere il config file correttamente impostato con la rete Fuji e l'indirizzo del nostro subnet. Questo file è fondamentale per le configurazioni più avanzate del nodo, anche se non verrà utilizzato in questa documentazione. Per ulteriori dettagli, fare riferimento alla documentazione ufficiale di Avalanche: [🔴 Avalanche Dev Docs: Create Without Limits](#) .

Ora che abbiamo richiesto il permesso di aggiungere il nostro nodo come validatore sul nostro subnet, dobbiamo procedere con la sua configurazione. Per fare ciò, aprire la cartella `subnet` nella riga di comando e digitare il seguente comando:

```
1 ./add-validator.sh
```

Prima di tutto, selezioniamo la rete Fuji come network di riferimento. Successivamente, utilizziamo la chiave precedentemente creata, ovvero `ContshipDocumentazione` , selezionando l'opzione `Use stored key` .

.Poi il sistema ci chiederà l'ID del nostro nodo che possiamo ottenere eseguendo il comando nella cartella `macos` :

```
1 ./get-id.sh
```

e otterremo quest'uscita con il `nodeID` che andiamo a mettere nel sistema :

```
1 "jsonrpc":"2.0","result":{"nodeID":"NodeID-4JwxRwRY8LgeSdDLNargL9NBEBUuvTk9v","nodePOP":{"publicKey":"0xb38828ac0
```

Per impostare il peso del nostro nodo a 30, selezioniamo l'opzione `custom` e inseriamo il valore desiderato. Successivamente, scegliamo l'opzione `Start in 5 minutes` per far partire il processo di validazione del nodo dopo 5 minuti. Infine, dobbiamo indicare per quanto tempo vogliamo che il nodo sia un validatore.

⚠ Per la documentazione, è raccomandabile impostare la durata della validazione del nodo sul nostro subnet a un valore inferiore o uguale alla durata dello staking del nodo sulla rete Fuji. In questo caso, abbiamo configurato la durata dello staking del nostro nodo su Fuji per 4800 ore, pertanto imposteremo la stessa durata per la validazione sul nostro subnet. Ciò garantirà la coerenza e la compatibilità delle impostazioni di validazione tra la rete Fuji e il nostro subnet.

Dopo aver inserito la durata di 4800 ore per il nodo validatore sul nostro subnet, il sistema richiederà una conferma. Clicchiamo su `yes` per procedere. Se tutte le operazioni vengono eseguite correttamente, verrà eseguita una transazione e l'output dovrebbe essere simile al seguente:

```
1 Transaction successful, transaction ID :EhZh8PvQyqA9xggn6EsdemXMnWKyy839NzEJ5DHEXTBiXbjV
```

Bene, il nostro nodo è validatore del subnet !

## 📁 Collegamento al subnet con metamask

Per avviare il nodo sul nostro subnet, apriamo una nuova finestra del terminale nella cartella `macOS` e inseriamo il seguente comando:

```
1 ./run-node-subnet.sh
```

⚠ Quando si esegue il comando per avviare il nodo sul subnet, potrebbe essere restituito un errore di compatibilità tra il nuovo binario del nodo e la versione della VM installata in precedenza. Questo errore può verificarsi a seguito di un aggiornamento del nodo rilasciato dalla rete Avalanche. In questo caso, è necessario aggiornare la VM alla versione compatibile con il nuovo binario del nodo per poter continuare a operare sul subnet.

All'avvio del nodo incontreremo questo problema :

```
1 INFO <57yTrjPaGWGkVATLkRWUercvMpAe1eH6d6zNvHA9H7eHQMirg Chain> subprocess/runtime.go:111 stdout collector shutdown
2 [02-27|22:23:39.921] ERROR chains/manager.go:361 error creating chain {"subnetID": "2Cv7gS1bXTNVD1rDLFRVNuCsP2WSY
```

Durante il processo di configurazione del nodo, potrebbe esserci un errore di compatibilità tra la versione del nodo e quella della VM installata. In questo caso specifico, la VM installata in precedenza (v0.5.11) non è compatibile con la nuova versione del nodo resa disponibile da Avalanche. Per risolvere questo problema, è necessario scaricare manualmente l'ultima versione della VM dal link fornito [Releases · ava-labs/subnet-evm](#) e sostituirla con quella precedentemente installata nella cartella plugins all'interno di `./avalanchego`. Il nome della VM utilizzata nel nostro subnet era `whyY15RH8t2CHtkEa41ZPmML8YQgwUeJhwpamjtjR6h94k4pf`.

Una volta fatto questo possiamo tentare di avviare di nuovo il nodo. Se va tutto bene dovremmo ottenere quest'uscita :

```
1 INFO <57yTrjPaGWGkVATLkRWUercvMpAe1eH6d6zNvHA9H7eHQMirg Chain> snowman/transitive.go:543 consensus starting {"las
```

**⚠ MOLTO IMPORTANTE :** È fondamentale prestare attenzione agli aggiornamenti rilasciati da Avalanche ogni due settimane e assicurarsi che sia il binario del nodo che quello della VM siano sempre aggiornati alla versione più recente. Ciò eviterà errori di compatibilità tra i protocolli e garantirà il corretto funzionamento del nodo sul subnet.

Ecco la configurazione per aggiungere il nuovo network su Metamask:

**i** A malicious network provider can lie about the state of the blockchain and record your network activity. Only add custom networks you trust.

**Network name**

**New RPC URL**

**Chain ID** **i**

**Currency symbol**

Ticker symbol verification data is currently unavailable, make sure that the symbol you have entered is correct. It will impact the conversion rates that you see for this network

**Block explorer URL** (Optional)

Dove l'RPC è questo : `http://127.0.0.1:9650/ext/bc/57yTrjPaGWGkVATLkRWUercvMpAe1eH6d6zNvHA9H7eHQMirg/rpc` dove `57yTrjPaGWGkVATLkRWUercvMpAe1eH6d6zNvHA9H7eHQMirg` è l'ID della nostra blockchain è 2819 la rete.

Perfetto, adesso possiamo sviluppare smart contracts e utilizzare la blockchain che abbiamo creato come qualsiasi blockchain.