

PERSONAL WEB-SERVER CON VPC

23/09/2024

Lettura: 25min

Dawei Zhou 890643

Benvenuto in questo laboratorio, dove avrai l'opportunità di utilizzare Amazon Virtual Private Cloud (VPC) per creare la tua rete personalizzata. Aggiungendo vari componenti e creando un gruppo di sicurezza, acquisirai esperienza pratica sulla personalizzazione del tuo VPC per soddisfare i tuoi requisiti specifici. Inoltre, configurerai e personalizzerai un'istanza EC2 per fungere da server Web e avviarla all'interno di una sottorete VPC. Oltre che aggiungere forme di sicurezza per sito sul server.

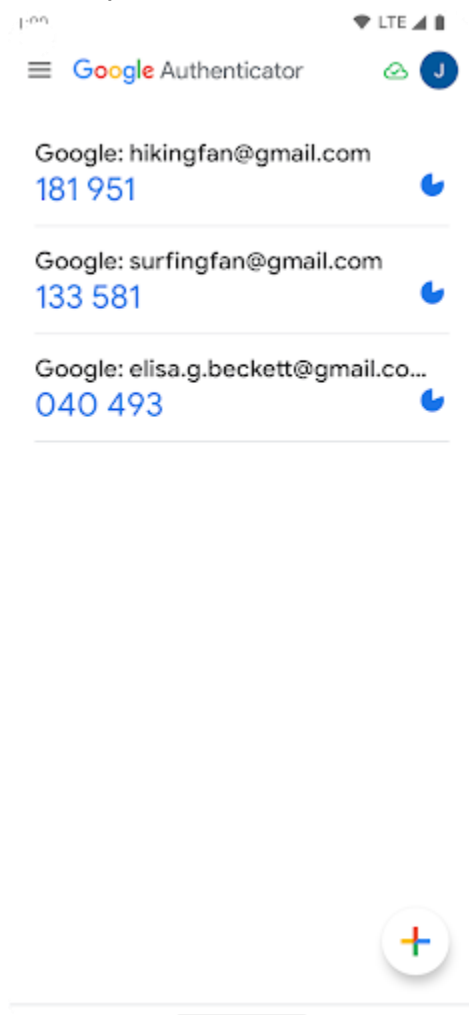
Amazon Virtual Private Cloud (Amazon VPC) ti consente di distribuire risorse Amazon Web Services (AWS) all'interno di una rete virtuale di tua progettazione. Questa rete virtuale somiglia molto a una rete convenzionale che verrebbe generalmente gestita all'interno del tuo data center fisico, sfruttando al tempo stesso l'infrastruttura flessibile e scalabile di AWS. È anche possibile creare un VPC che si estende su più zone di disponibilità.

Prerequisiti

- 1) Google Authenticator (autenticazione MFA)
- 2) Account Aws (free per 12 mesi)
- 3) Account Docker

Google Authenticator(circa 1m)

- 1) Scaricare app da play store.
- 2) Entra sul profilo [Amazon](#) classico.
- 3) Vai su mio account-> Accesso e Sicurezza ->Verifica in 2 passaggi (OFF->ON). Segui i passaggi e arriverai a “due opzioni”, scegli App di Autenticazione.
- 4) Intanto apri google authenticator e in basso a destra clicca sul “+”. Decidi se usare QR code o inserire chiave entrambi forniti al passaggio 3).
- 5) Completato tutto avrai account collegato con codice temporaneo che si aggiorna.

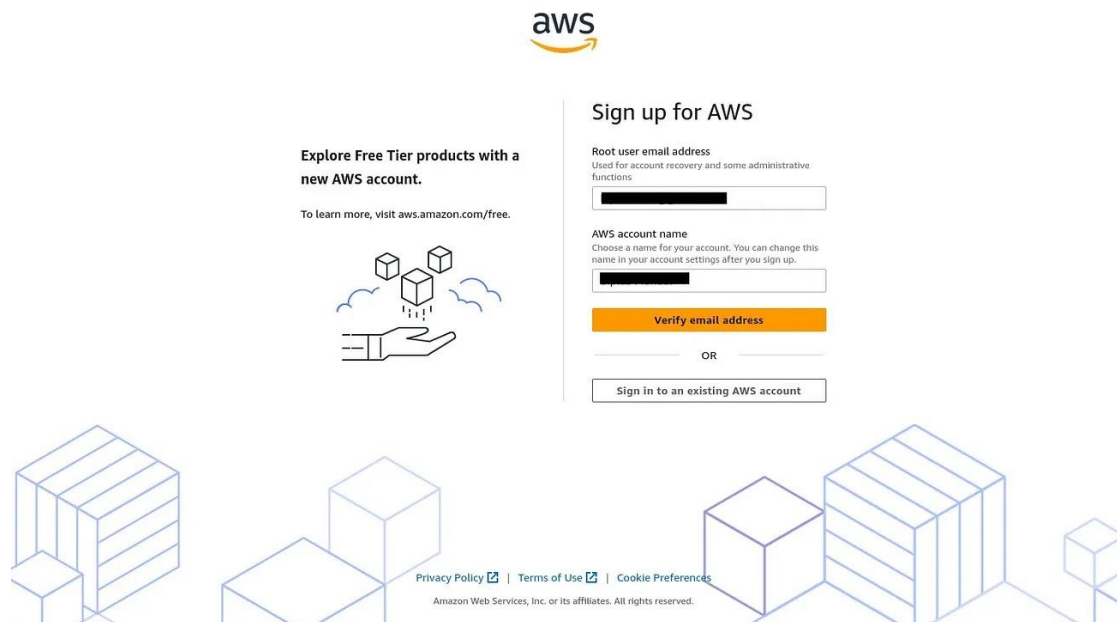


Ogni volta che accederai al profilo Amazon AWS, dovrai fornire il codice MFA fornito in App.

Account Aws(circa 10m)

Basta iscriversi dal sito [AWS](#), andare su crea un account gratuito e compilare i campi. Non preoccupatevi della iniziale somma da pagare con carta di credito visto che vi restituiranno la somma dopo aver confermato conto bancario.

Step 1)



Inserisci e-mail e username da usare come root. Clicca “verifica e-mail”.



Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.



Sign up for AWS

Confirm you are you

Making sure you are secure -- it's what we do.

We sent an email with a verification code to [redacted] (not you?)

Enter it below to confirm your email.

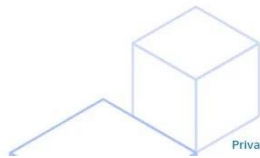
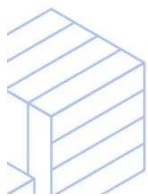
Verification code

Verify

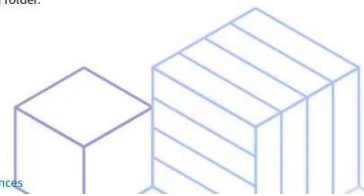
Resend code

Didn't get the code?

- Codes can take up to 5 minutes to arrive.
- Check your spam folder.



[Privacy Policy](#) | [Terms of Use](#) | [Cookie Preferences](#)



Verifica e-mail usando il codice ricevuto per posta. Clicca “Verifica”.



Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.



Sign up for AWS

Create your password

✔ It's you! Your email address has been successfully verified. ✕

Your password provides you with sign in access to AWS, so it's important we get it right.

Root user password

Confirm root user password

Security check



Type the characters as shown above




Continue (step 1 of 5)

OR

Inserisci password forte, usa simboli (__, *, /, etc....), Maiuscole, Cifre. Completa Captcha e clicca su “Continua (Step 1 su 5)”.

Step 2)

All AWS accounts can explore 3 different types of free offers, depending on the product used.

	Always free Never expires
	12 months free Start from initial sign-up date
	Trials Start from service activation date

How do you plan to use AWS?

☐ Business - for your work, school, or organization

☒ Personal - for your own projects

Who should we contact about this account?

Full Name

Phone Number

Country or Region

Address

City

State, Province, or Region


Postal Code

Customers with an Indian contact address are served by Amazon Web Services India Private Limited, the local seller for AWS services in

Scegli Business se è account aziendale altrimenti Personale. Compila il resto dei campi e vai su “Continua.”

Step 3)

Secure verification

 We will not charge you for usage below AWS Free Tier limits. We may temporarily hold up to \$1 USD (or an equivalent amount in local currency) as a pending transaction for 3-5 days to verify your identity.



Billing Information

Credit or Debit card number



AWS accepts all major credit and debit cards. To learn more about payment options, review our [FAQ](#)

Expiration date

Month Year

Cardholder's name

CVV

Billing address

☒ Use my contact address



☐ Use a new address

Do you have a PAN?

Permanent Account Number (PAN) is a ten-digit alphanumeric number issued by the Indian Income Tax Department. This 10-digit number is printed on the front of your PAN card.

☐ Yes

☒ No

You can go on the Tax Settings Page on Billing and Cost Management Console to update your PAN

Inserire le coordinate bancarie + contatti.



Please enter your secure code in order to complete your transaction. This information will not be shared with the merchant.

Merchant Name: AMAZON

Amount: ₹2.00

Date: Aug 6, 2023

Card Number: XXXX XXXX XXXX 

Personal Greeting: Transaction is protected by 3D Secure service.

One Time Password :

Haven't receive OTP? please click to [Re-Generate OTP](#)

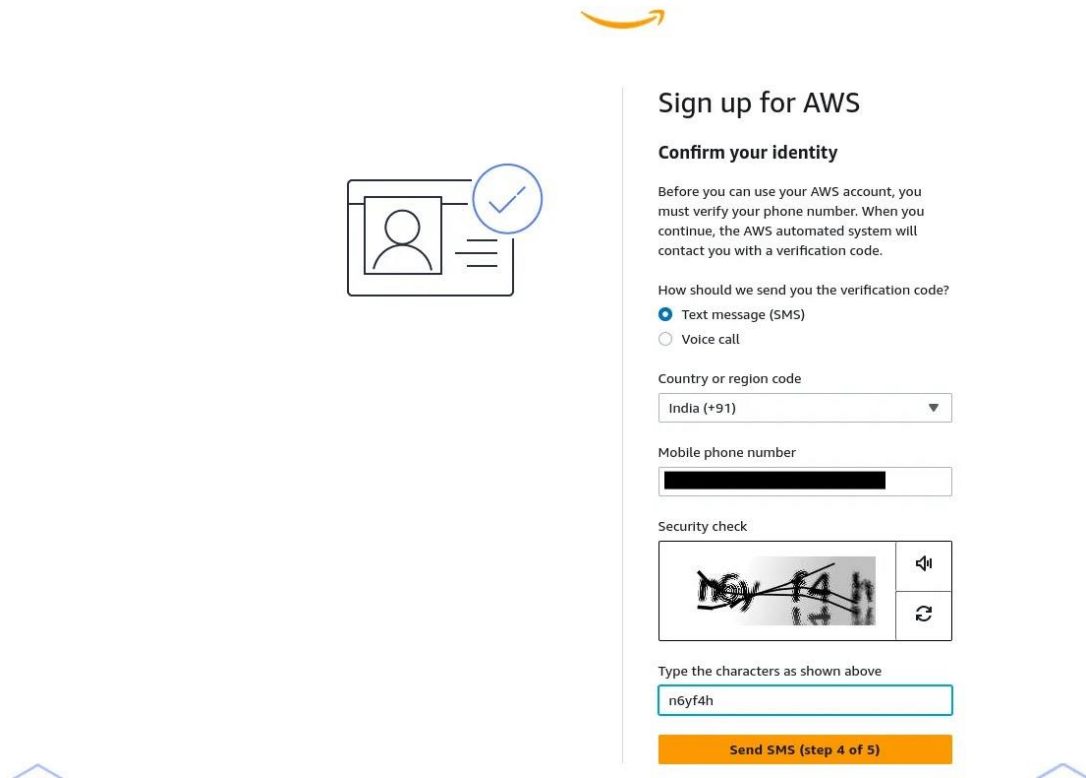
CANCEL

SUBMIT

Powered by


Ti verrà richiesto di fare una piccola transazione (0.02 centesimi esempio) per confermare conto, poi ti verrà restituito l'importo.

Step 4)



The image shows the AWS 'Sign up for AWS' page, specifically the 'Confirm your identity' step. On the left, there is an icon of a person's profile with a checkmark. The main content area is titled 'Sign up for AWS' and 'Confirm your identity'. It explains that before using the AWS account, the user must verify their phone number. Below this, there are two radio buttons for 'How should we send you the verification code?': 'Text message (SMS)' (selected) and 'Voice call'. A dropdown menu for 'Country or region code' is set to 'India (+91)'. A text field for 'Mobile phone number' is shown with a blacked-out number. A 'Security check' section displays a distorted image of the characters 'n6yf4h' with a refresh button. Below the image, a text field contains the characters 'n6yf4h'. At the bottom, there is an orange button labeled 'Send SMS (step 4 of 5)'.

Sign up for AWS

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

☒ Text message (SMS)

☐ Voice call

Country or region code

India (+91)

Mobile phone number

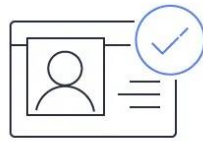
Security check

Type the characters as shown above

n6yf4h

Send SMS (step 4 of 5)

Conferma la tua identità usando un recapito telefonico valido.



Sign up for AWS

Confirm your identity

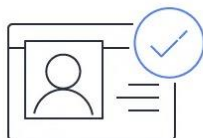
Verify code

Continue (step 4 of 5)

Having trouble? Sometimes it takes up to 10 minutes to retrieve a verification code. If it's been longer than that, [return to the previous page](#) and try again.



Inserisci codice ricevuto.



Sign up for AWS

Confirm your identity [Info](#)

Primary purpose of account registration

Choose one that best applies to you. If your account is tied to a business, select the one that applies to your business.

Personal use ▼

Ownership type


Individual ▼

Continue (step 4 of 5)



Scegli modalità d'uso del profilo. Clicca su "Continua".




Step 5)




Sign up for AWS



Select a support plan

Choose a support plan for your business or personal account. [Compare plans and pricing examples](#)
[You can change your plan anytime in the AWS Management Console.](#)

<p><input checked="" type="radio"/> Basic support - Free</p> <ul style="list-style-type: none">Recommended for new users just getting started with AWS24x7 self-service access to AWS resourcesFor account and billing issues onlyAccess to Personal Health Dashboard & Trusted Advisor 	<p><input type="radio"/> Developer support - From \$29/month</p> <ul style="list-style-type: none">Recommended for developers experimenting with AWSEmail access to AWS Support during business hours12 (business)-hour response times 	<p><input type="radio"/> Business support - From \$100/month</p> <ul style="list-style-type: none">Recommended for running production workloads on AWS24x7 tech support via email, phone, and chat1-hour response timesFull set of Trusted Advisor best-practice recommendations 
---	---	---

 **Need Enterprise level support?**
From \$15,000 a month you will receive 15-minute response times and concierge-style experience with an assigned Technical Account Manager. [Learn more](#)

Scegli piano Amazon AWS per il tuo profilo (nel nostro caso Free).




Congratulations

Thank you for signing up for AWS.

We are activating your account, which should only take a few minutes. You will receive an email when this is complete.

[Go to the AWS Management Console](#)

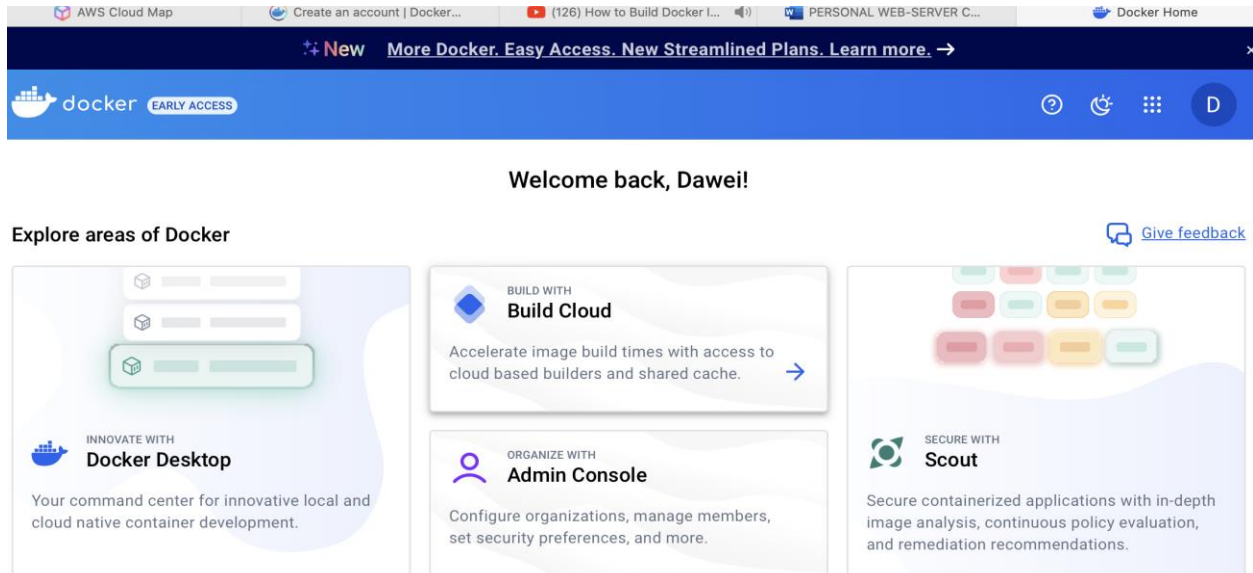
[Sign up for another account or contact sales.](#)



Finito , entro qualche minuto riceverai un'e-mail.

Account Docker(2m-5m)

Prima di tutto entra sul sito ufficiale e inserisci credenziali account [Docker](#) o registrati con e-mail Google.



Dopo Login potrete vedere questa schermata.

Adesso installiamo Docker da terminale (Ubuntu - MacOS):

- 1) Elimino tutte le dispense inutili:

```
$ for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

- 2) Installo certificati, chiavi e repository:

```
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
```

3) Per essere sicuri che installiamo la versione ufficiale e non di Ubuntu

```
$ apt-cache policy docker-ce
```

4) Installiamo Docker:

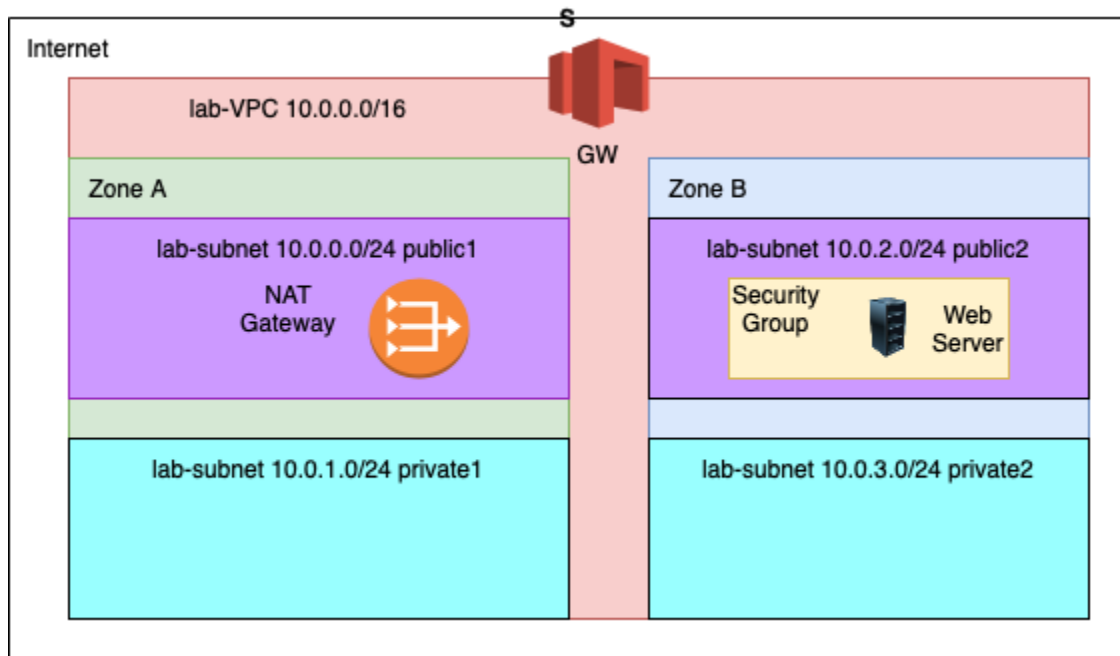
```
$ sudo apt install docker-ce
```

```
$ sudo systemctl status docker
```

Copy

```
Output
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enable
   Active: active (running) since Tue 2020-05-19 17:00:41 UTC; 17s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 24321 (dockerd)
      Tasks: 8
     Memory: 46.4M
    CGroup: /system.slice/docker.service
            └─24321 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.
```

Struttura VPC

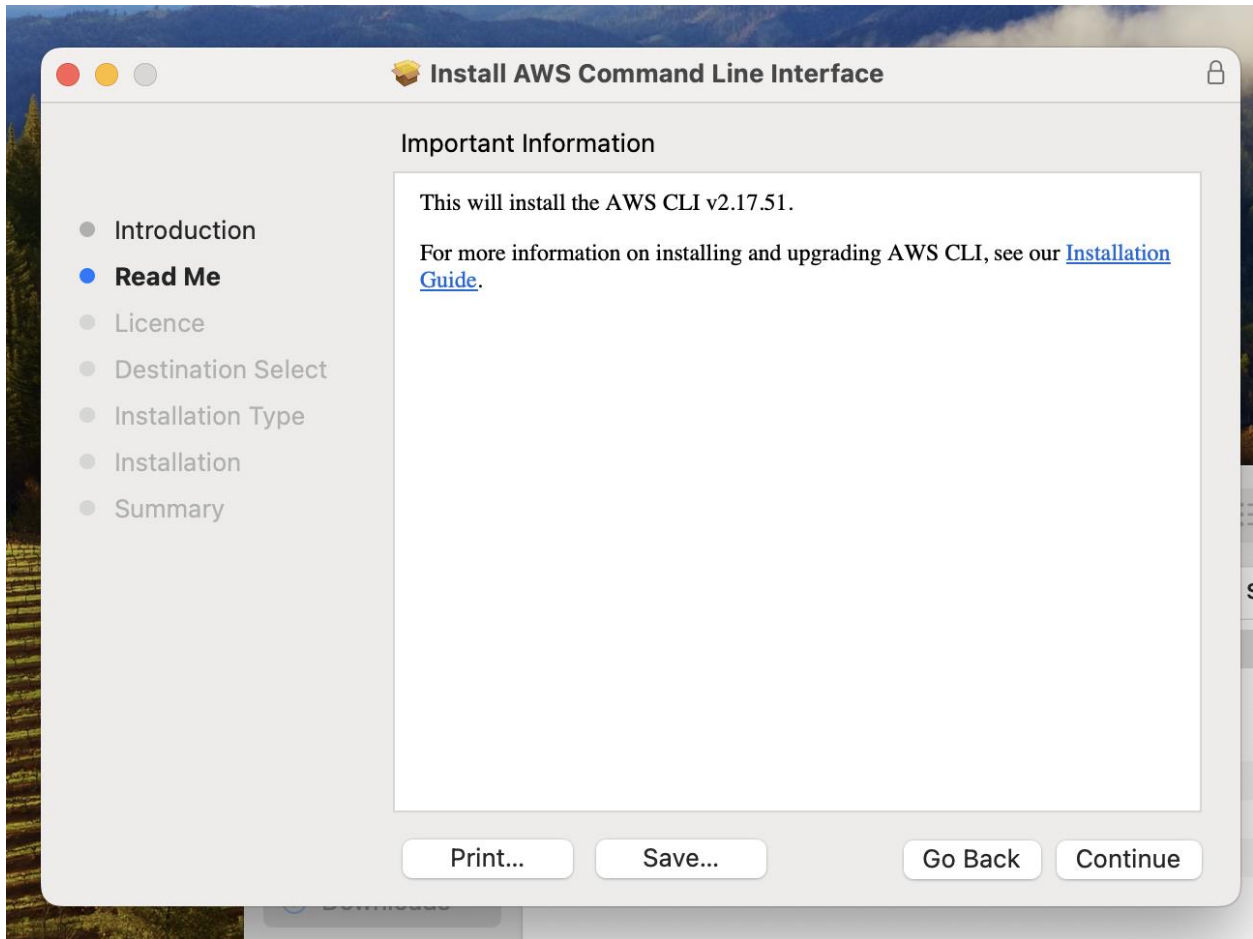


Architettura della rete privata.

Passi del Progetto (MacOs circa 45min)

Installazione AWS CLI

- 1) Scarica questo [package](#).
- 2) Avvia il file scaricato e segui i passaggi.



- 3) Al passaggio di selezione degli utenti dove installare AWS puoi andare su: tutti, solo attuale, disco specifico.
- 4) Testa che ci sia AWS. Con il comando:

```
$ which aws
```

```
$ aws --version
```

```
daweizhou@daweis-Air-1688 ~ % aws --version
```

```
aws-cli/2.17.51 Python/3.11.9 Darwin/23.5.0 exe/x86_64
```

Se non trovi comando aws, restarta terminale o segui [troubleshoot](#).

Configurazione AWS

Inserisci i comandi:

\$ aws configure sso

\$ SSO session name (Recommended): **my-sso**

\$ SSO start URL [None]: **<https://my-sso-portal.awsapps.com/start>** <- se non lo conosci
segui il punto successivo

\$ SSO region [None]: **us-east-1** <- dipende da che regione volete che il profilo sia
settato/server hostato. Le regioni le trovi [qua](#).

\$ SSO registration scopes [None]: **sso:account:access**

Fatto questo si aprirà una schermata sul browser dove dovrete inserire username+
password+MFA(se prima volta che loggi ti chiederà di registrarlo su google authenticator).

Set up the authenticator app

Username:

ADMIN ([not you?](#))

[Back to MFA device options](#)

1



Install either the Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer. [See a list of compatible apps](#)

2

Show QR code

Use your virtual MFA app or your device's camera to scan the QR code ([show secret key](#))

3

Please enter the six digit code from your authenticator app

Authenticator code

Assign MFA

Inserito codice e vi farà connettere al vostro utente.

```
jqen zvrp
The only AWS account available to you is: 398666360603
Using the account ID 398666360603
The only role available to you is: AdministratorAccess
Using the role name "AdministratorAccess"
[CLI default client Region [None]: eu-north-1
[CLI default output format [None]: json
[CLI profile name [AdministratorAccess-398666360603]: Admin
```

Connesso Utente, adesso inserite regione, formato output: JSON e nome profilo così da quel momento in poi potrete usarlo per collegarmi immediatamente da terminale.

Per Collegare profilo da ora in poi basta digitare il comando:

```
$ aws sso login --profile my-dev-profile *
```

```
[daweizhou@daweis-Air-1688 ~ % aws sso login --profile Admin
Attempting to automatically open the SSO authorization page in your default browser.
If the browser does not open or you wish to use a different device to authorize this request, open the following URL:

https://device.sso.eu-north-1.amazonaws.com/

Then enter the code:

XMXQ-SCNZ
Successfully logged into Start_URL: https://daweizhou.awsapps.com/start
```

Ecco qua, profilo impostato e collegato correttamente! Adesso però dobbiamo inserire le chiavi di sicurezza altrimenti molte operazioni non saranno permesse:

```
$ aws configure
```

```
AWS Access Key ID [*****admin]: AS
AWS Secret Access Key [*****u12.]:
v26voZ
Default region name [eu-north-1]:
```

Per le credenziali basta che tu acceda ad [IAM console](#), in alto a destra sul tuo username->Credenziali di sicurezza (Security credentials) e crea una chiave d'accesso, ricordati di salvarti i dati o scaricare file .csv .

Dopo di questo potrete mandare i primi comandi con AWS CLI


```
daweizhou@daweis-Air-1688 ~ % aws ec2 describe-vpcs
{
  "Vpcs": [
    {
      "CidrBlock": "172.31.0.0/16",
      "DhcpOptionsId": "dopt-08fd7cd3beb60fcdd",
      "State": "available",
      "VpcId": "vpc-06005da4907f5f9a4",
      "OwnerId": "398666360603",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-05d93c08d117b8899",
          "CidrBlock": "172.31.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": true
    }
  ]
}
```

Esempio

ATTENZIONE!

Utente deve fare parte della lista di “users”, lo trovi su: IAM Identity Center nella colonna a sinistra, uno di quegli utenti può accedere usando SSO da terminale fornendo i dati di autenticazioni richiesti.

Inoltre, devi collegare Utente con account AWS, per farlo sempre nella stessa colonna, su “AWS Account” -> Assegna utenti e Group -> Nella sezione Utenti collegare utente con account.

* ATTENZIONE! Solo se non conosci start URL altrimenti salta

- 1) Entra e logga su [aws](#).
- 2) Inserisci e-mail e password e codice MFA usando google Authenticator.
- 3) Entrato nel profilo andate su “Cerca” e inserite “IAM Identity Center” (in inglese).

AWS access portal URL

 [https://daweizhou.awsapps.com/st](https://daweizhou.awsapps.com/start)
[art](#) 

Nella colonna a destra troverete URL access. Se c'è scritto "EDIT" aggiungete il vostro dominio e poi potrete usarlo per SSO.



Sign in

☒ **Root user**

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**

User within an account that performs daily tasks. [Learn more](#)

Root user email address

qwe@|

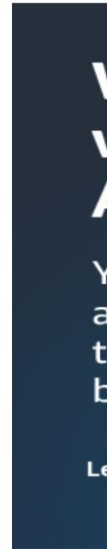


Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

____ New to AWS? ____

Create a new AWS account



See our new improved Amazon Web Services sign in experience



Root user sign in ⓘ

Email: dawei Zhou2002@gmail.com

Password

[Forgot password?](#)

Sign in

[Sign in to a different account](#)

[Create a new AWS account](#)

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Momento di creare VPC

Gli output dei comandi saranno formato Json come impostato durante la creazione del profilo.

1) Inizializziamo vpc:

```
$ aws ec2 create-vpc --cidr-block 10.0.0.0/16 --query Vpc.VpcId --output text
```

Ci darà in output l'ID della VPC-> Esempio : vpc-06c61a7a6af30324e.

Nel nostro progetto per facilità creiamo 2 zone eu-north-1a/eu-north-1b con ognuno di loro che possiede una rete pubblica +privata.

2) Creiamo le reti private/pubbliche:

```
$ aws ec2 create-subnet --vpc-id vpc-06c61a7a6af30324e --cidr-block 10.0.0.0/24 --availability-zone eu-north-1a --query Subnet.SubnetId --output text
```

Output: subnet-08c0ca06a4aa35d56

```
$ aws ec2 create-subnet --vpc-id vpc-06c61a7a6af30324e --cidr-block 10.0.1.0/24 --availability-zone eu-north-1a --query Subnet.SubnetId --output text
```

```
$ aws ec2 create-subnet --vpc-id vpc-06c61a7a6af30324e --cidr-block 10.0.2.0/24 --availability-zone eu-north-1b --query Subnet.SubnetId --output text
```

```
$ aws ec2 create-subnet --vpc-id vpc-06c61a7a6af30324e --cidr-block 10.0.3.0/24 --availability-zone eu-north-1b --query Subnet.SubnetId --output text
```

Come configurare NAT e Internet Gateway

Nei comandi qui sottoelencati, le parti rosse devono essere sostituite con i vostri id, sono diversi da persona a persona.

- Creiamo nostro Internet Gateway

```
$ aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --output text
```

Con output ID-gateway;

- Associamo Internet gateway alla vpc così che la rete “possa navigare su Internet”

```
$ aws ec2 attach-internet-gateway --vpc-id vpc-1a2b3c4d5e6f1a2b3 --internet-gateway-id igw-id
```

- È il momento di creare la nostra route table pubblica

```
$ aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query RouteTable.RouteTableId --output text
```

Con output ID-routetable;

- Aggiungi route nella route-table pubblica che punti all’Internet Gateway

```
$ aws ec2 create-route --route-table-id rtb-id-public --destination-cidr-block 0.0.0.0/0 --gateway-id igw-id
```

Con output “true” se tutto va bene;

- Colleghiamo la route table alle istanze subnets pubbliche

```
$aws ec2 associate-route-table --route-table-id rtb-id-public --subnet-id subnet-id-public-subnet
```

Rifai per l'altra subnet pubblica.

Con output: id-associazione + messaggio “associated”.

- Impostiamo nostro NAT Gateway così che le reti private possano comunicare con l'esterno senza dover essere scoperte
- Prima di tutti ci serve un elastic IP per il NAT Gateway

```
$ aws ec2 allocate-address --domain vpc --query AllocationId --output text
```

Output: id-elastic-ip;

- Crea NAT Gateway in una subnet pubblica con IP pubblica fornito al comando di prima

```
$aws ec2 create-nat-gateway --subnet-id subnet-id-public-subnet --allocation-id eipalloc-id
```

Output: Json di conferma contenente ID-NAT-Gateway;

- Creiamo una route-table come nel caso dell'Internet Gateway ma in questo caso per il NAT.

```
$ aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query RouteTable.RouteTableId --output text
```

Output:id-route-table;

- Ci serve una route che indirizzi tutto verso la NAT Gateway

```
$ aws ec2 create-route --route-table-id rtb-id-private --destination-cidr-block 0.0.0.0/0 --gateway-id nat-id
```

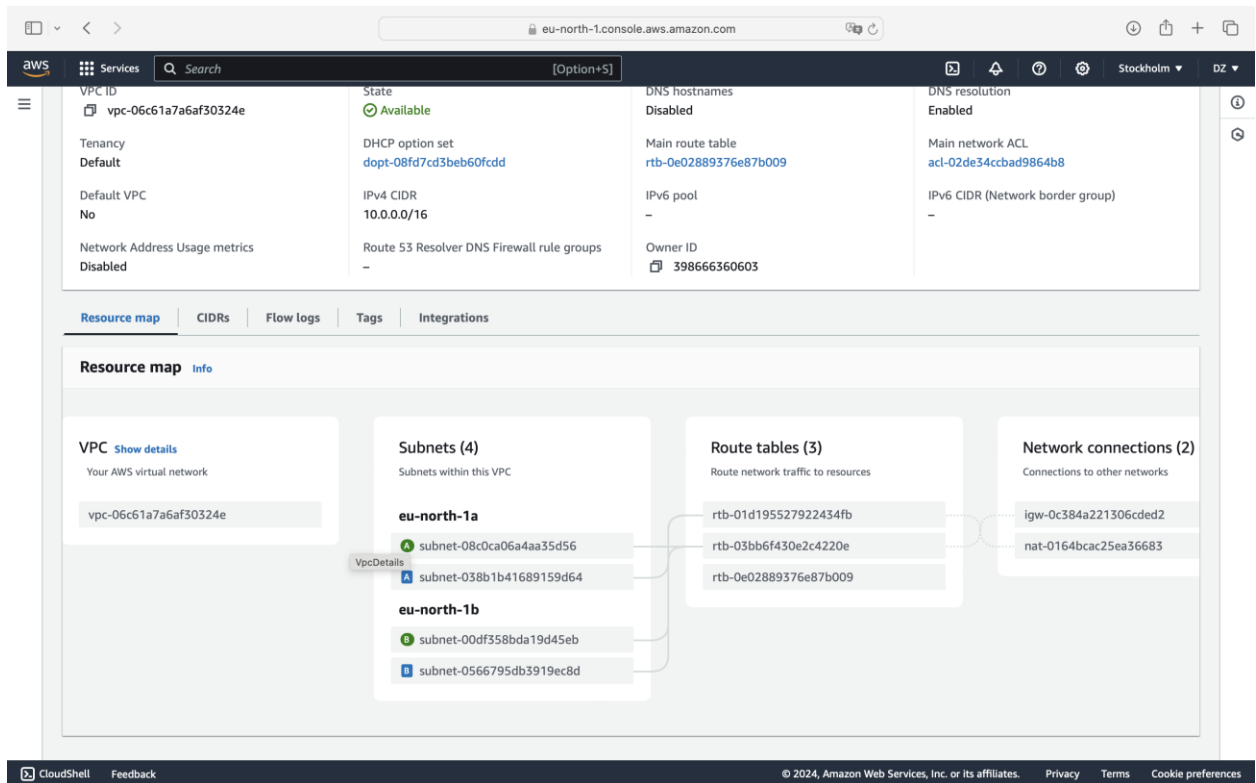
Output: “true” se va tutto bene;

- Associamo questa route-table-privata alle reti private

```
$ aws ec2 associate-route-table --route-table-id rtb-id-private --subnet-id subnet-id-private-subnet
```

Rifai per altra rete privata;

Output: id-associazione + messaggio “associated”.



Risultato post comandi, stato della VPC

Impostiamo web-server con EC2

Ricordo che le parti in rosso sono sostituibili con altri nomi preferibili a voi o con id univici ricevuti come output dai comandi.

- ★ Creiamo la **coppia di chiavi** e li salviamo nel file **.pem** con il comando:

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > MyKeyPair.pem
```

```
$ aws ec2 describe-key-pairs --key-name MyKeyPair
```

Output: per vedere output del file;

- ★ Creiamo una **security-group** per agire come firewall sul traffico di rete:

```
$ aws ec2 create-security-group --group-name MyWebSG --description "Allows SSH and HTTP connections for the Web Server" --vpc-id vpc-06c61a7a6af30324e
```

Output: Id-Security-group;

- ★ Aggiungiamo regole alla security-group:

```
$ aws ec2 authorize-security-group-ingress --group-id sg-000d2c961be941fa0 --protocol tcp --port 22 --cidr 0.0.0.0/0
```

Per autorizzare l'SSH.

Output: Json con “true” se comando andato a buon fine.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-000d2c961be941fa0 --protocol tcp --port 80 --cidr 0.0.0.0/0
```

Per autorizzare richieste HTTP.

```
$ aws ec2 describe-security-groups --group-ids sg-000d2c961be941fa0
```

Verifichiamo status della security-group.


```
[daweizhou@daweis-MacBook-Air-1688 ~ % aws ec2 describe-security-groups --group-ids sg-023c16199299cd6ae
{
  "SecurityGroups": [
    {
      "Description": "Allows SSH and HTTP connections for the Web Server",
      "GroupName": "MyWebSG",
      "IpPermissions": [
        {
          "FromPort": 80,
          "IpProtocol": "tcp",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "ToPort": 80,
          "UserIdGroupPairs": []
        },
        {
          "FromPort": 22,
          "IpProtocol": "tcp",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "ToPort": 22,
          "UserIdGroupPairs": []
        }
      ],
      "OwnerId": "398666360603",
      "GroupId": "sg-023c16199299cd6ae",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "UserIdGroupPairs": []
        }
      ],
      "VpcId": "vpc-06005da4907f5f9a4"
    }
  ]
}
```

★ Ci serve altro elastic-ip così lo colleghiamo alla nostra istanza:

```
$ aws ec2 allocate-address --domain vpc-06c61a7a6af30324e --query  
AllocationId --output text
```

★ Lanciamo server ec2, come immagine uso UBUNTU Arm visto che stiamo tenendo conto di stare usando MacOS:

```
$ aws ec2 run-instances --image-id ami-026b57f3c383c2eec --count 1 --instance-type t3.micro --key-name MyKeyPair --security-group-ids sg-000d2c961be941fa0 --subnet-id subnet-12314
```

In pratichiamo stiamo runnando una istanza specificando che immagine di boot usare(nel mio caso Ubuntu Arm), tipo di istanza (memoria, risorse fisiche), chiave per connessione ssh, security-group da usare, indirizzo ip-pubblico, subnet e vpc da associare.

Output: Json contenente messaggio di conferma + id-istanza.

★ Per vedere lista delle istanza EC2:

```
$ aws ec2 describe-instances
```

★ Associamo un indirizzo ip-pubblico al server:

```
$ aws ec2 associate-address --instance-id i-12345678 --allocation-id eipalloc-12345678
```

Output: ID-associazione.

ATTENZIONE!

Per vedere id immagine su free tier vai su AWS [Console](#) ->CERCA “EC2”, barra laterale sinistra vai su Immagini->Catalogo AMI.

Per vedere tipo di istanza prima scegli immagine.

Passo Finale:

1 - Connetiamoci al nostro server AWS

```
$ chmod 400 MyKeyPair.pem
```

```
$ ssh -i MyKeyPair.pem ubuntu@13.60.88.226
```

Se stai usando istanza **ubuntu**” va bene, ma altre tipi di OS richiedono **“ec2-user”**.

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

2 – Installa Docker CLI(Prerequisiti)

3 – Pullo immagine preparata da me precedentemente (Nginx) per Linux ARM 64

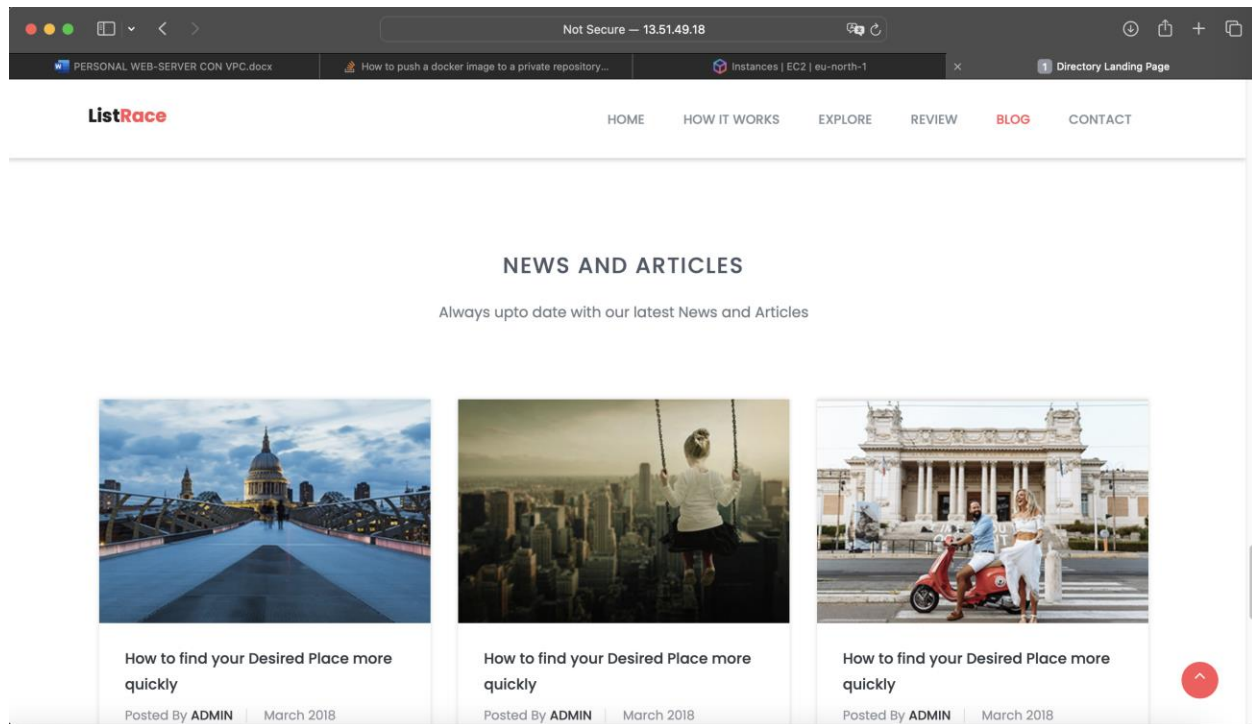
```
$ sudo docker pull daweizhou/my-nginx-image
```

4 – Controllo lista di immagini docker e creo container per poi runnarlo

```
$ sudo docker image ls
```

```
$ sudo docker run -d -p 80:80 daweizhou/my-nginx-image
```

5 -Vai sul browser e inserisci indirizzo ip-pubblico del server EC2 AWS nella barra di ricerca.



COMANDI UTILI PER SUCCESSIVE CONNESSIONI :

\$ aws ec2 terminate-instances --instance-ids INSTANCE_ID1,INSTANCE_ID2,... ->
(per terminare istanza)

\$ aws ec2 stop-instances --instance-ids INSTANCE_ID -> (fermare istanza server
ec2)

\$ aws ec2 start-instances --instance-ids INSTANCE_ID -> (avviare istanza server
ec2)

\$ sudo systemctl start apache2 -> (avviare servizio webserver)

`$ sudo systemctl stop apache2` -> (fermare servizio webserver)

`$ ssh -v -i MyKeyPair.pem ubuntu@13.60.88.226` -> (aprire connessione ssh)

`$ exit -> to close ssh connection` -> (chiudere connessione ssh)