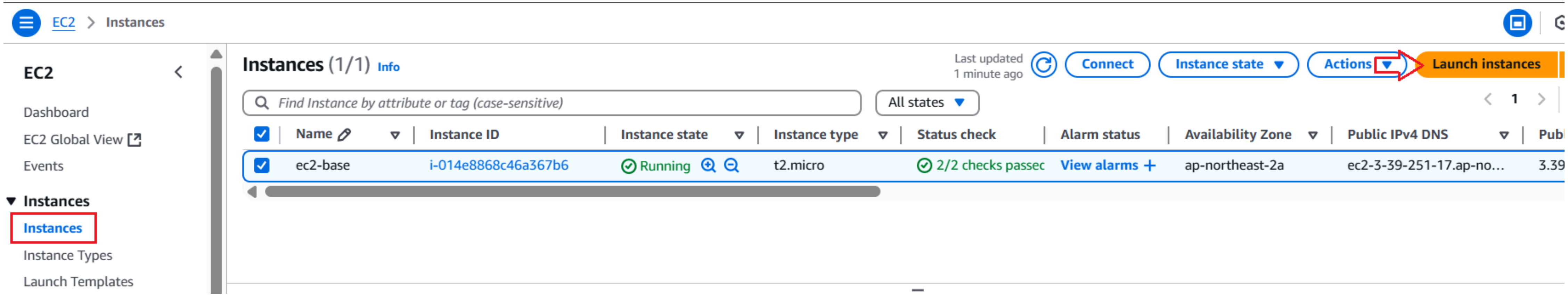
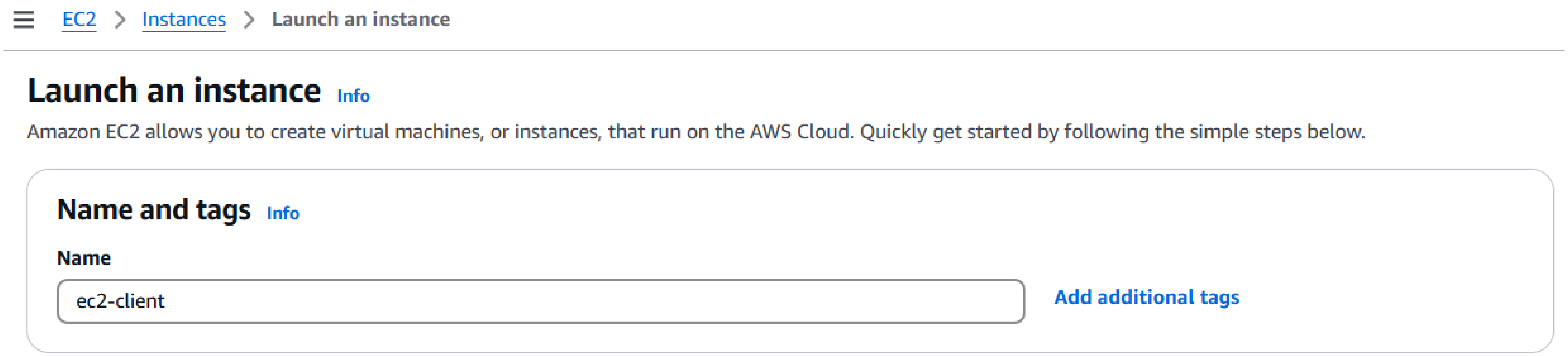
AMI를 이용하여 ec2 생성

단계1: Launch instances



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Pub
☑	ec2-base	i-014e8868c46a367b6	Running	t2.micro	2/2 checks passec	View alarms +	ap-northeast-2a	ec2-3-39-251-17.ap-no...	3.39

단계2: Name and tags



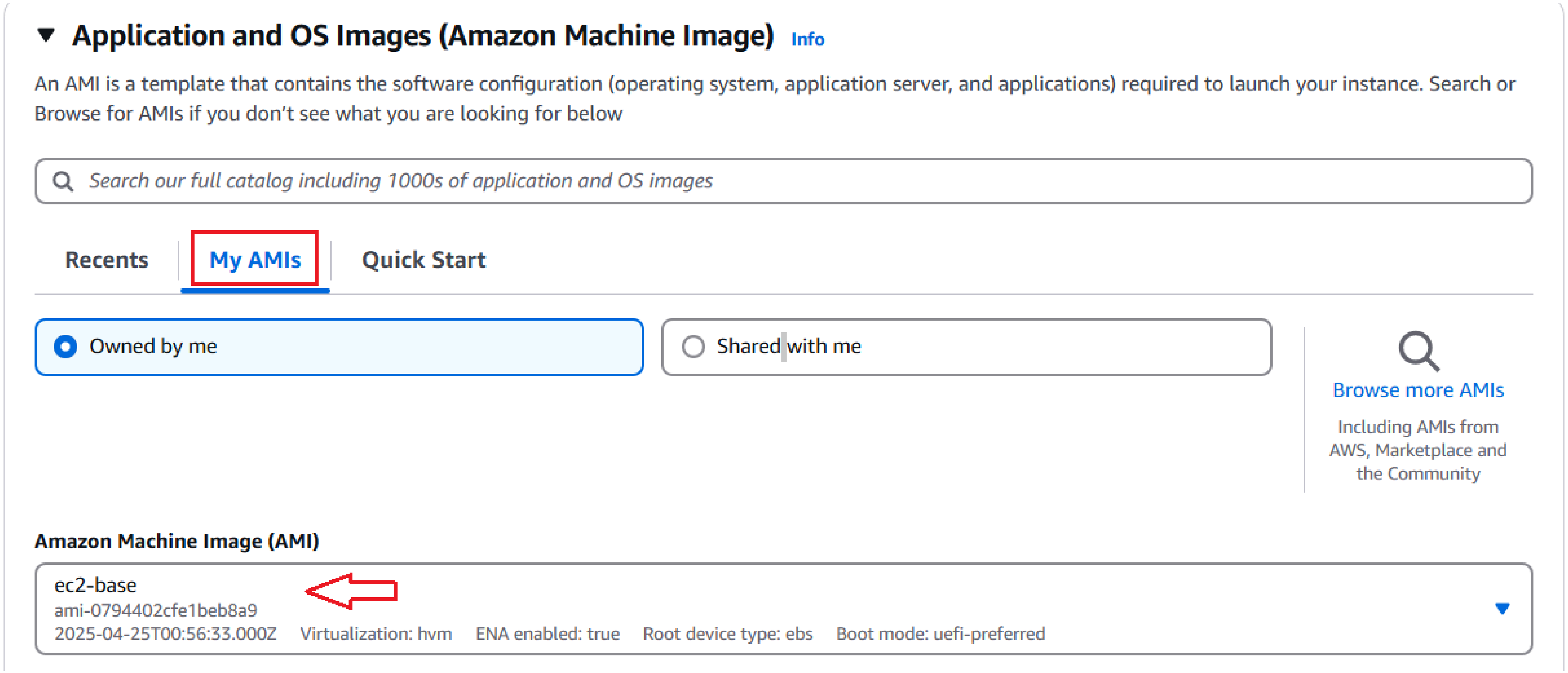
Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: ec2-client

단계3: Application and OS Images (Amazon Machine Image)



▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | **My AMIs** | Quick Start

● Owned by me ○ Shared with me

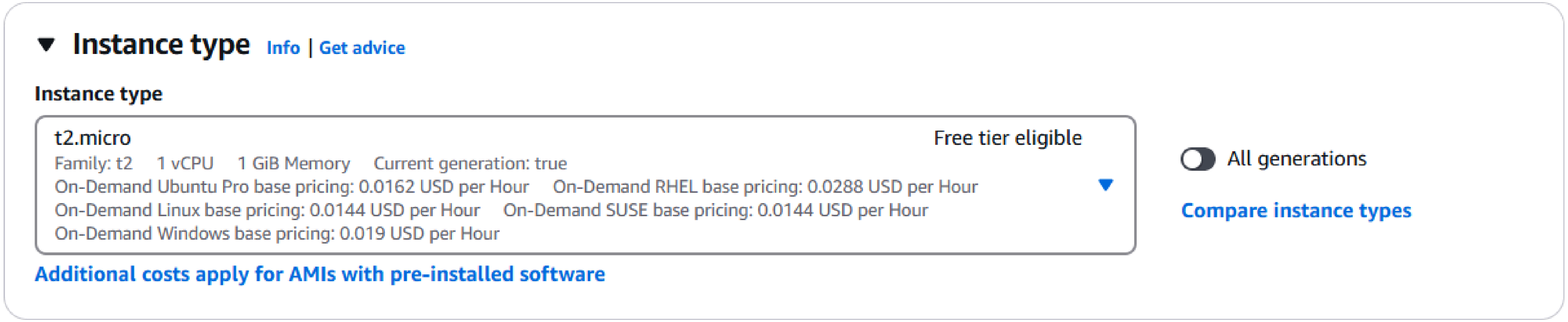
Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

ec2-base
ami-0794402cfe1beb8a9
2025-04-25T00:56:33.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred

단계4: Instance type & Key pair



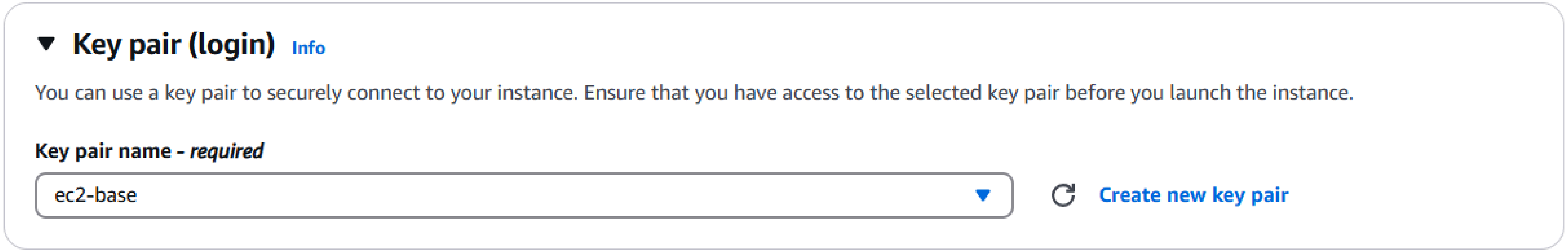
▼ Instance type

Instance type

t2.micro — Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0162 USD per Hour On-Demand RHEL base pricing: 0.0288 USD per Hour
On-Demand Linux base pricing: 0.0144 USD per Hour On-Demand SUSE base pricing: 0.0144 USD per Hour
On-Demand Windows base pricing: 0.019 USD per Hour

Additional costs apply for AMIs with pre-installed software



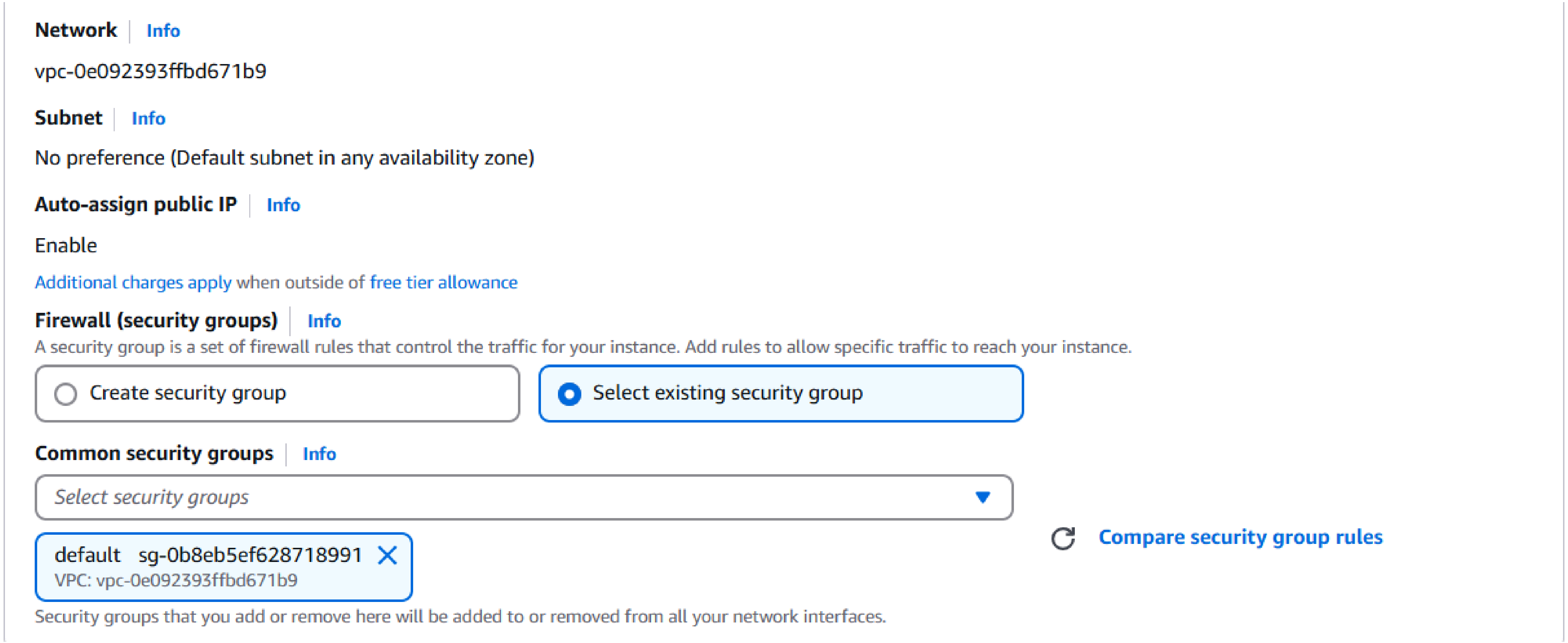
▼ Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

ec2-base

단계5: Network settings



Network | Info

vpc-0e092393ffbd671b9

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group

● Select existing security group

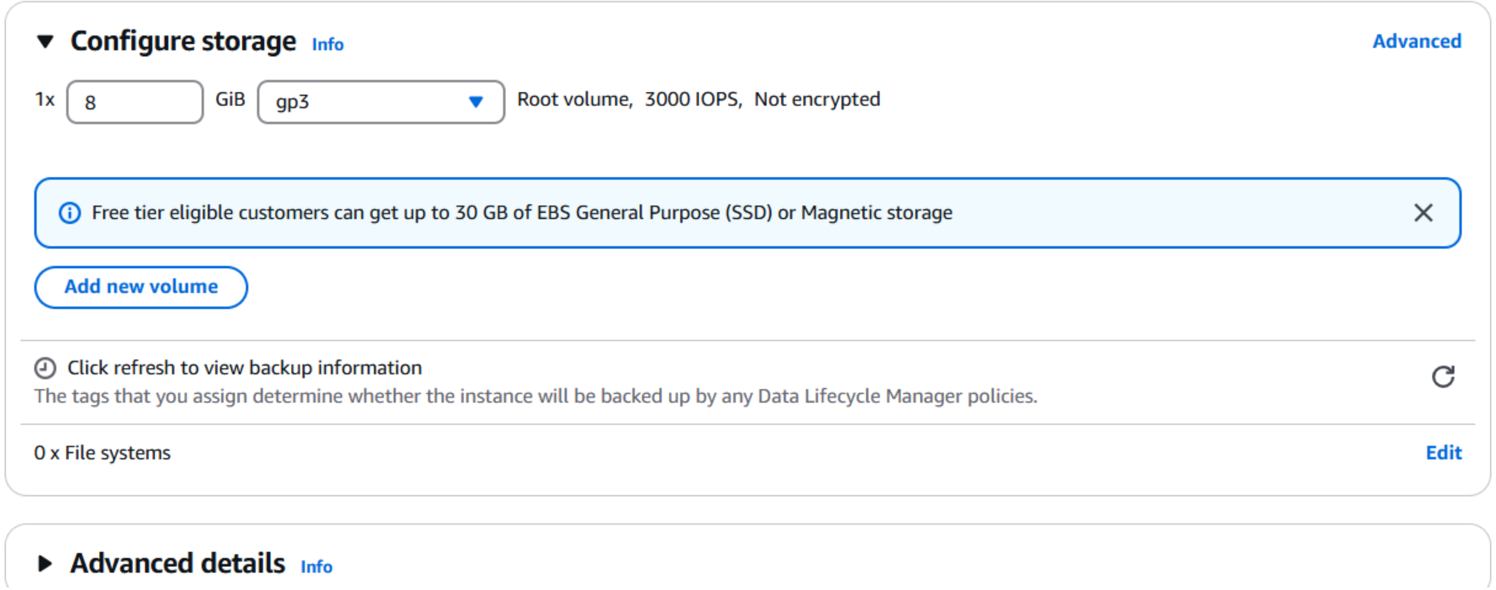
Common security groups | Info

Select security groups ▼

default sg-0b8eb5ef628718991 ✕
VPC: vpc-0e092393ffbd671b9

Security groups that you add or remove here will be added to or removed from all your network interfaces.

단계6: Launch instance



Configure storage Info

Advanced

1x 8 GiB gp3 ▼ Root volume, 3000 IOPS, Not encrypted

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

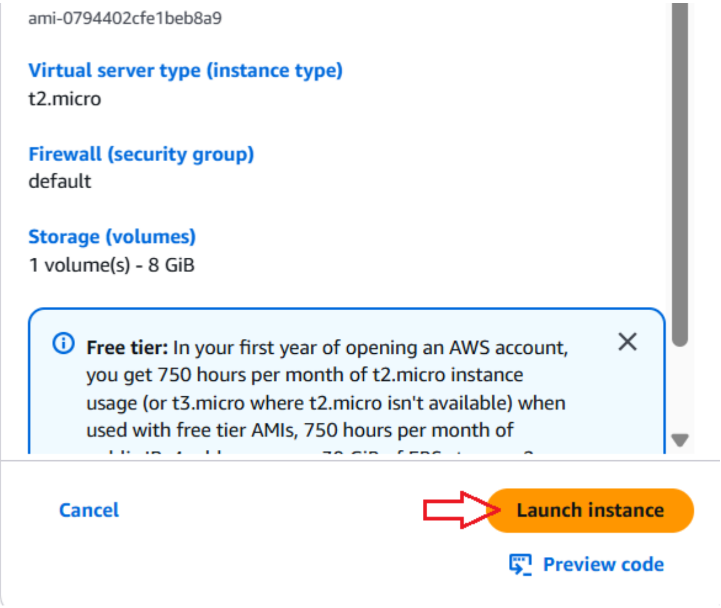
Add new volume

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

▶ **Advanced details** Info



ami-0794402cfe1beb8a9

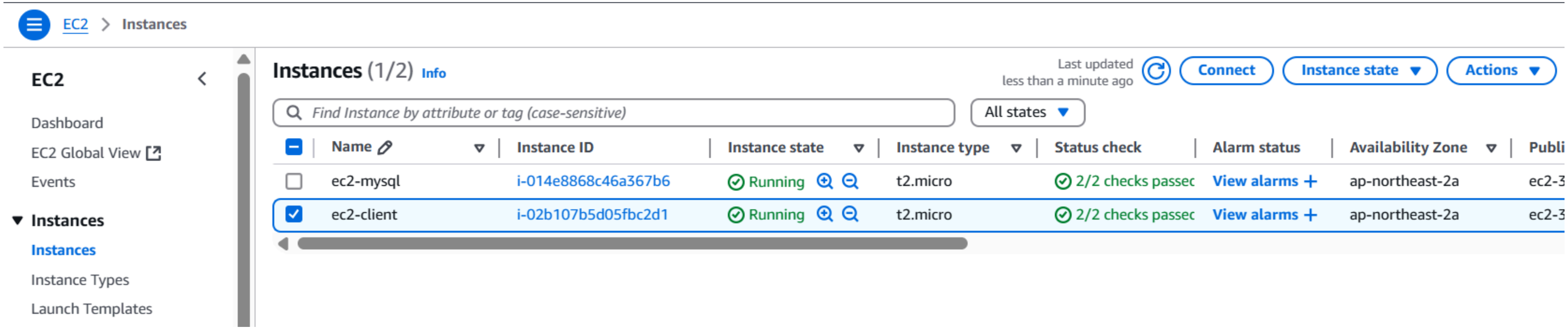
Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

ⓘ **Free tier:** In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of ✕

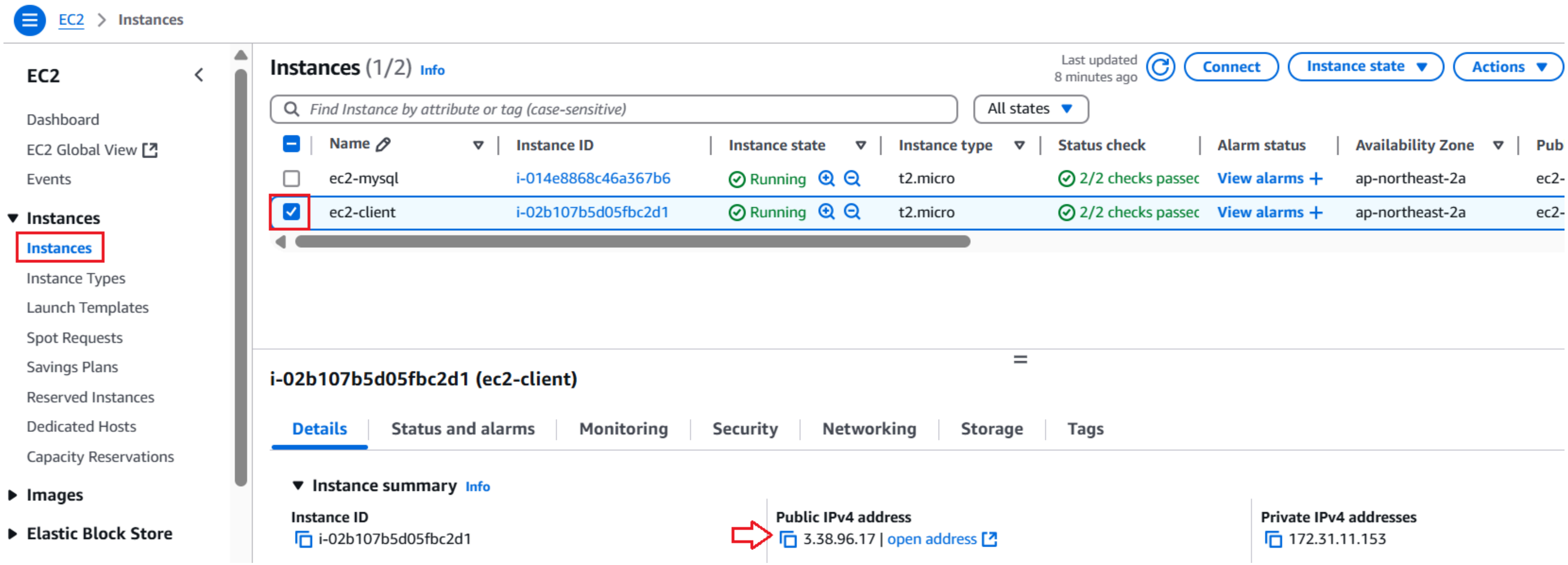
단계7: 결과 확인



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publi
☐	ec2-mysql	i-014e8868c46a367b6	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-northeast-2a	ec2-3
☑	ec2-client	i-02b107b5d05fbc2d1	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-northeast-2a	ec2-3

SSH

단계1: Public IP 복사



EC2 > Instances

EC2

- Dashboard
- EC2 Global View
- Events
- ▼ Instances
  - Instances
  - Instance Types
  - Launch Templates
  - Spot Requests
  - Savings Plans
  - Reserved Instances
  - Dedicated Hosts
  - Capacity Reservations
- ▶ Images
- ▶ Elastic Block Store

Instances (1/2) Info

Last updated 8 minutes ago

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
□	ec2-mysql	i-014e8868c46a367b6	Running	t2.micro	2/2 checks passec	View alarms +	ap-northeast-2a	ec2-
☑	ec2-client	i-02b107b5d05fbc2d1	Running	t2.micro	2/2 checks passec	View alarms +	ap-northeast-2a	ec2-

i-02b107b5d05fbc2d1 (ec2-client)

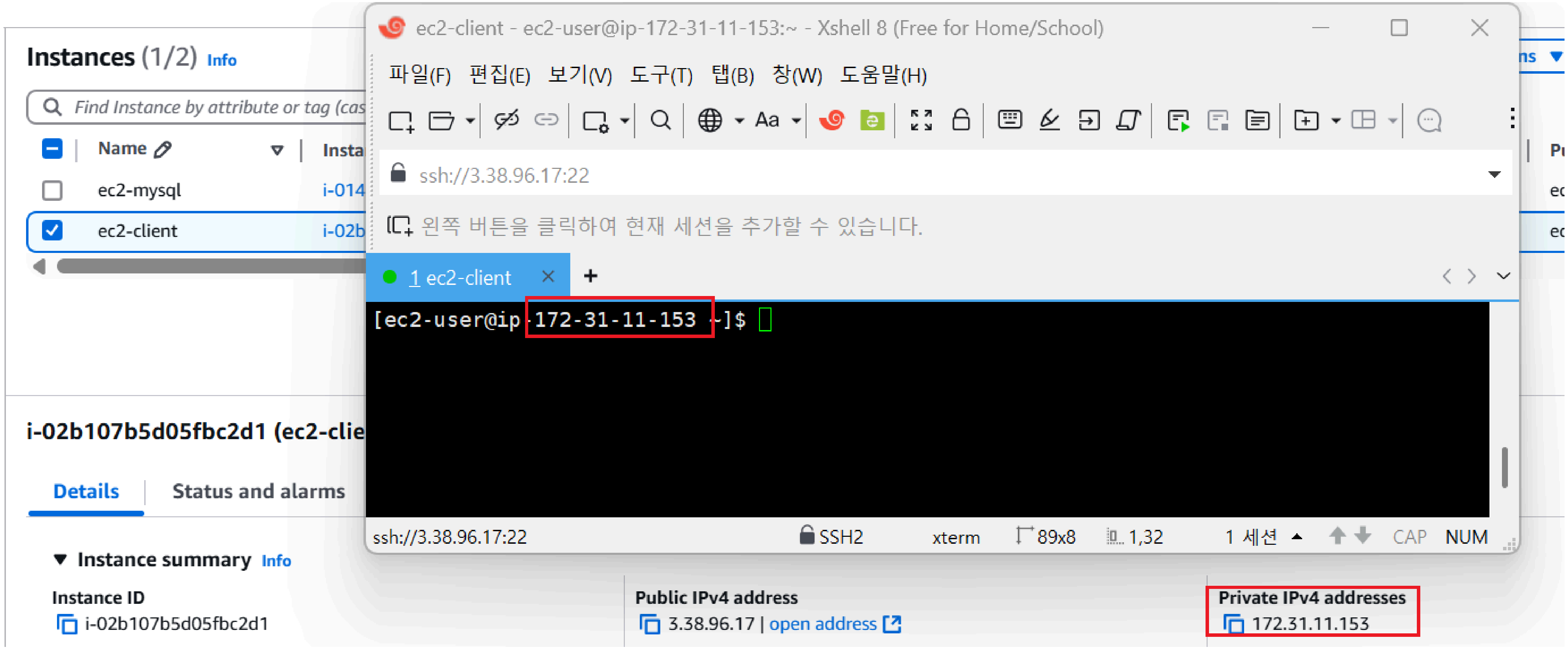
▼ Instance summary Info

Instance ID: i-02b107b5d05fbc2d1

Public IPv4 address: 3.38.96.17 | open address

Private IPv4 addresses: 172.31.11.153

단계2: XSHELL을 이용해서 ec2 접속



Instances (1/2) Info
Find Instance by attribute or tag (cas
Name
Insta
ec2-mysql
i-014
ec2-client
i-02b
i-02b107b5d05fbc2d1 (ec2-clie
Details
Status and alarms
Instance summary Info
ec2-client - ec2-user@ip-172-31-11-153:~ - Xshell 8 (Free for Home/School)
파일(F) 편집(E) 보기(V) 도구(T) 탭(B) 창(W) 도움말(H)
ssh://3.38.96.17:22
왼쪽 버튼을 클릭하여 현재 세션을 추가할 수 있습니다.
1 ec2-client
ssh://3.38.96.17:22
SSH2
xterm
89x8
1,32
1 세션
CAP
NUM

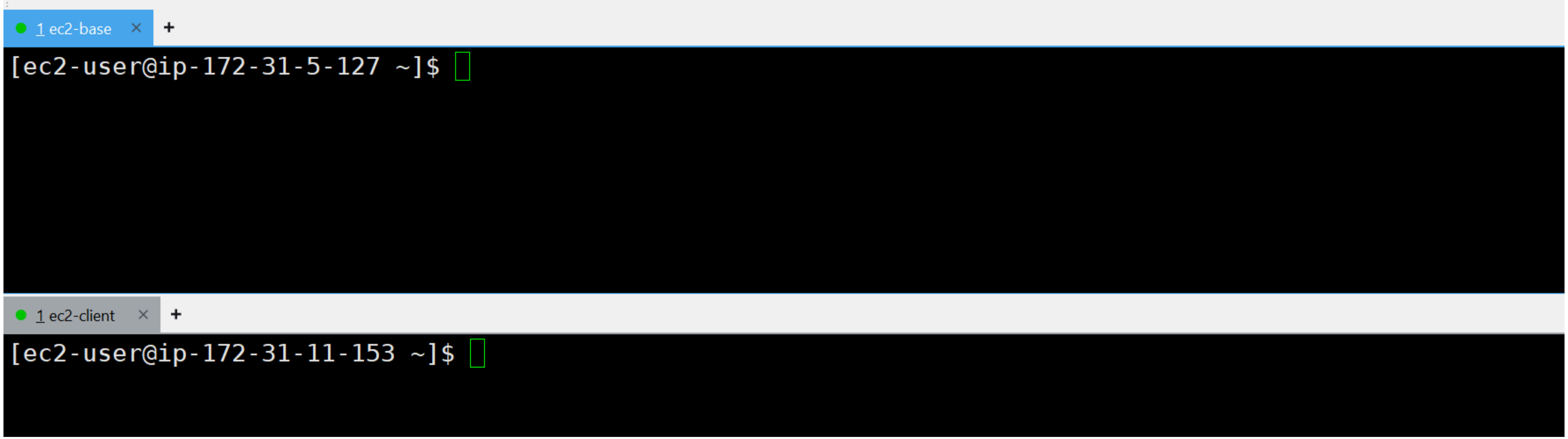
```
[ec2-user@ip-172-31-11-153 ~]$
```

Instance ID: i-02b107b5d05fbc2d1

Public IPv4 address: 3.38.96.17 | open address

Private IPv4 addresses: 172.31.11.153

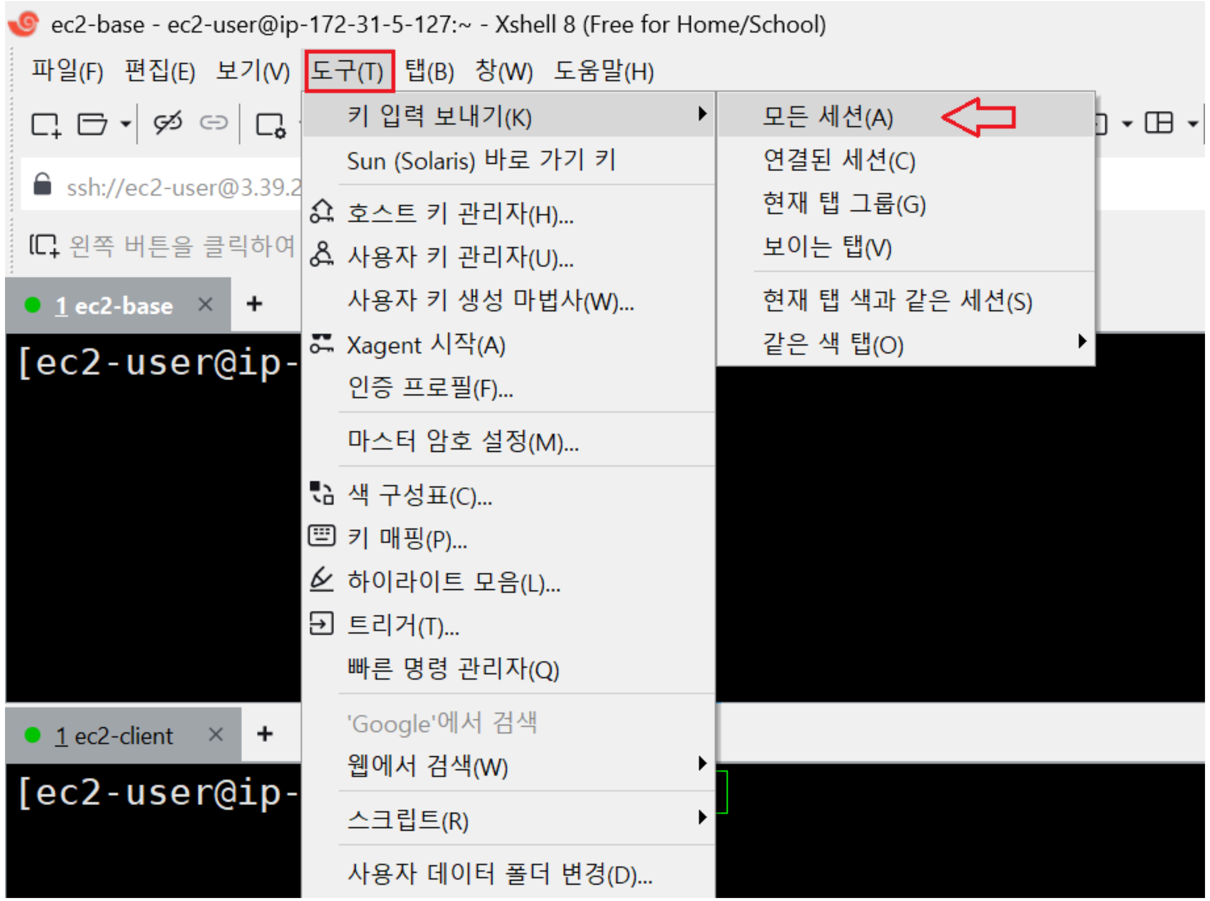
단계3: 2개 ec2 접속



```
[ec2-user@ip-172-31-5-127 ~]$
```

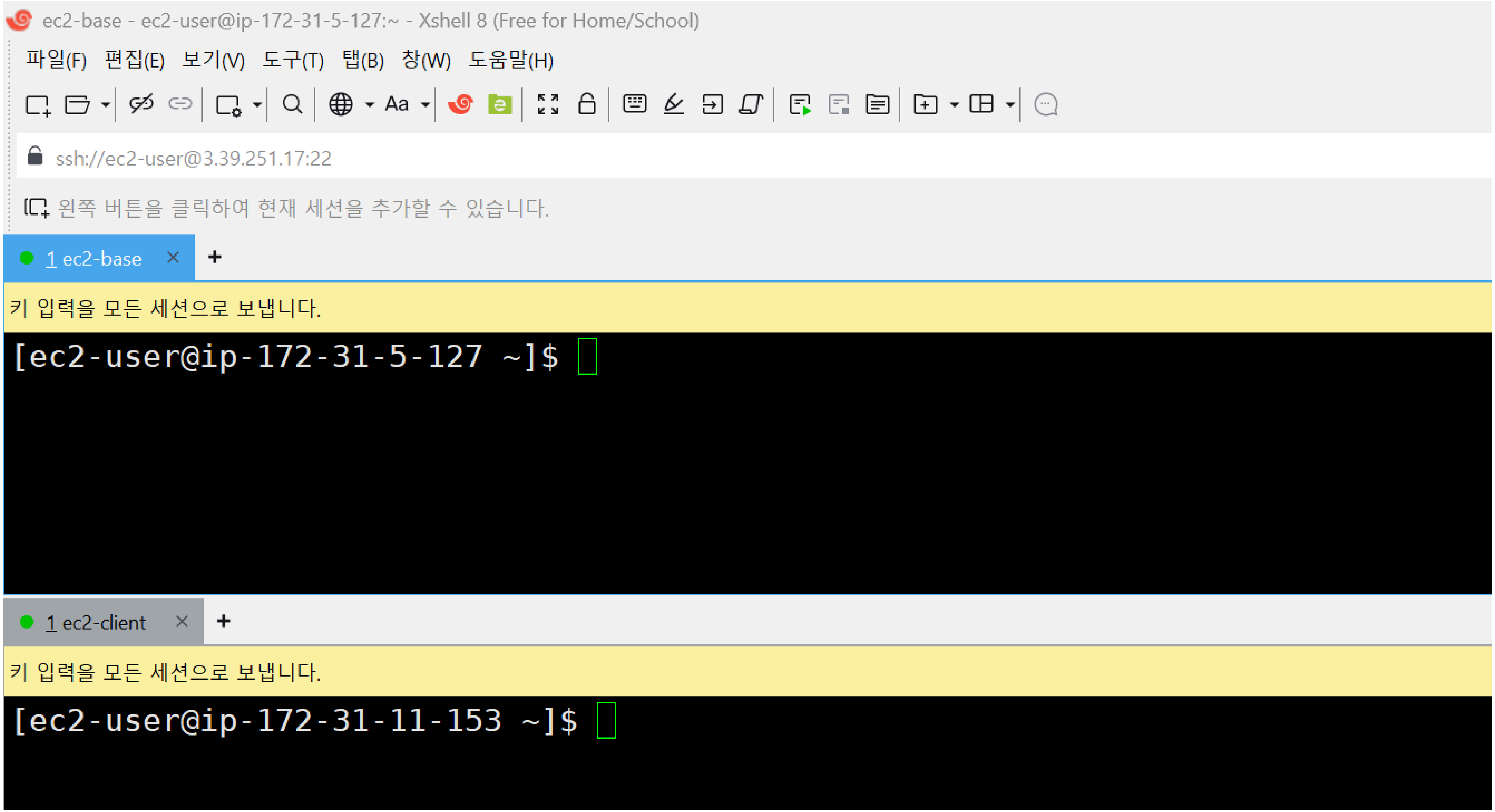
```
[ec2-user@ip-172-31-11-153 ~]$
```

단계4: 모든 세션 적용



```
[ec2-user@ip-
```

```
[ec2-user@ip-
```



ec2-base - ec2-user@ip-172-31-5-127:~ - Xshell 8 (Free for Home/School)
파일(F)
편집(E)
보기(V)
도구(T)
탭(B)
창(W)
도움말(H)
Aa
ssh://ec2-user@3.39.251.17:22
왼쪽 버튼을 클릭하여 현재 세션을 추가할 수 있습니다.
1 ec2-base
1 ec2-client

키 입력을 모든 세션으로 보냅니다.

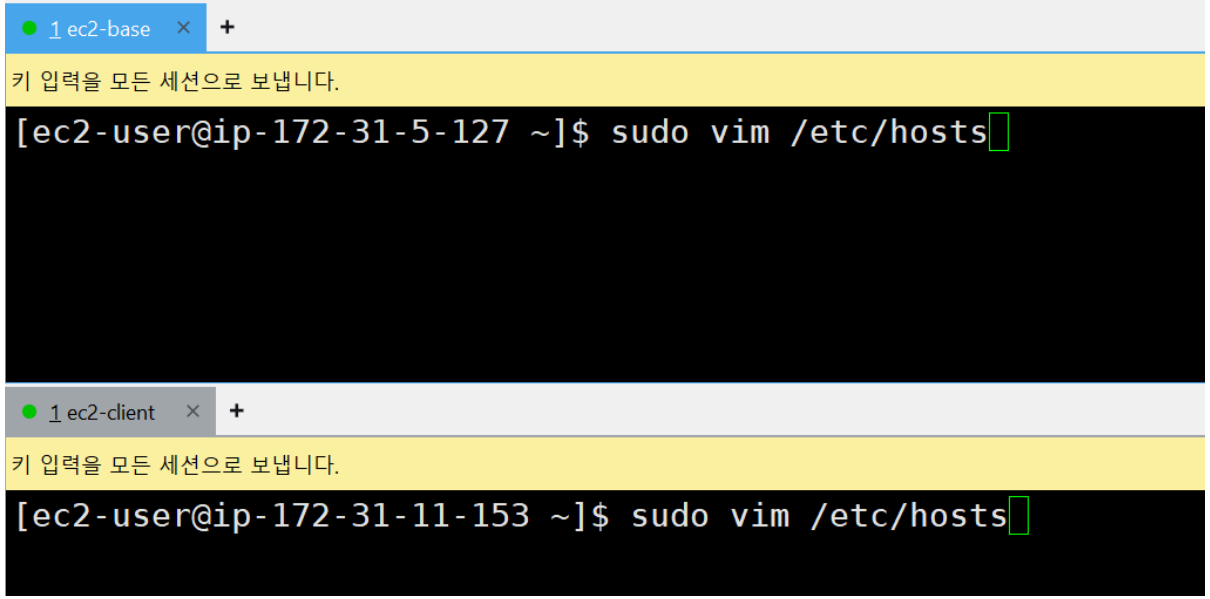
```
[ec2-user@ip-172-31-5-127 ~]$ 
```

키 입력을 모든 세션으로 보냅니다.

```
[ec2-user@ip-172-31-11-153 ~]$ 
```

단계5: hosts 파일 편집

```
# hosts 파일 편집
sudo vim /etc/hosts
```

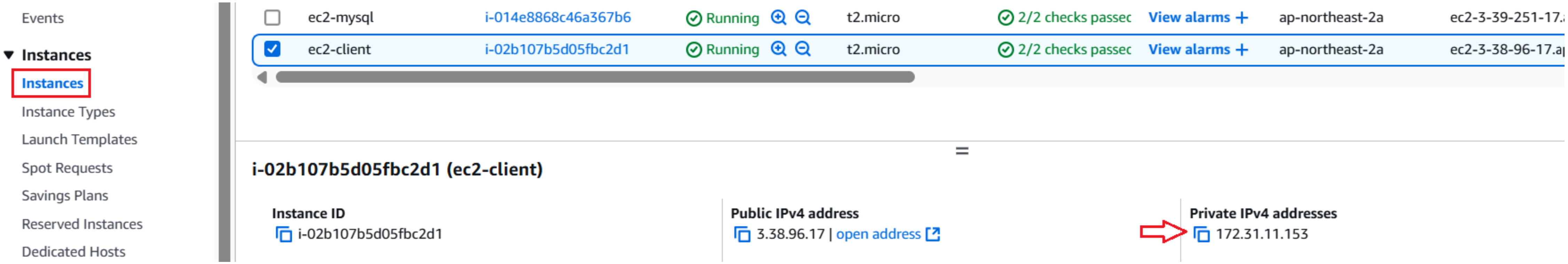

1 ec2-base
키 입력을 모든 세션으로 보냅니다.

```
[ec2-user@ip-172-31-5-127 ~]$ sudo vim /etc/hosts
```

1 ec2-client
키 입력을 모든 세션으로 보냅니다.

```
[ec2-user@ip-172-31-11-153 ~]$ sudo vim /etc/hosts
```

단계6: Private IP 확인 후 hosts 파일에 작성



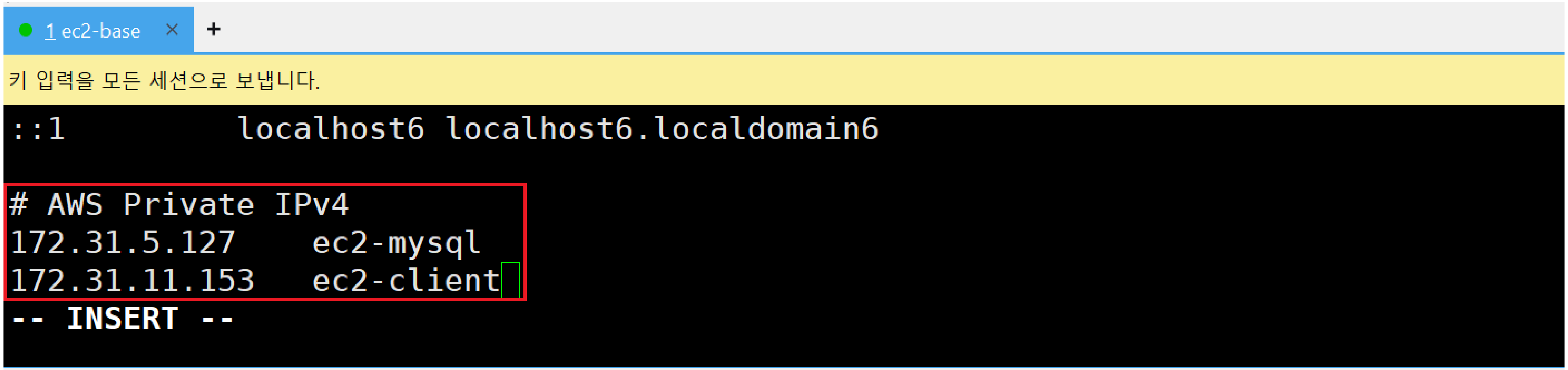
☐	ec2-mysql	i-014e8868c46a367b6	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-northeast-2a	ec2-3-39-251-17.:
☑	ec2-client	i-02b107b5d05fbc2d1	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-northeast-2a	ec2-3-38-96-17.a

i-02b107b5d05fbc2d1 (ec2-client)

Instance ID
i-02b107b5d05fbc2d1

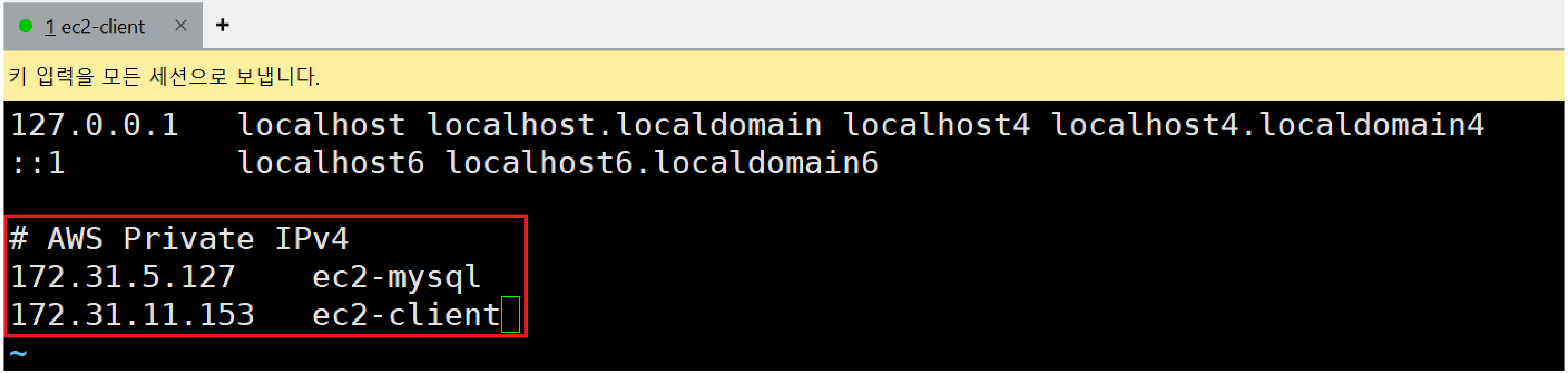
Public IPv4 address
3.38.96.17 | open address

Private IPv4 addresses
172.31.11.153



키 입력을 모든 세션으로 보냅니다.

```
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
-- INSERT --
```

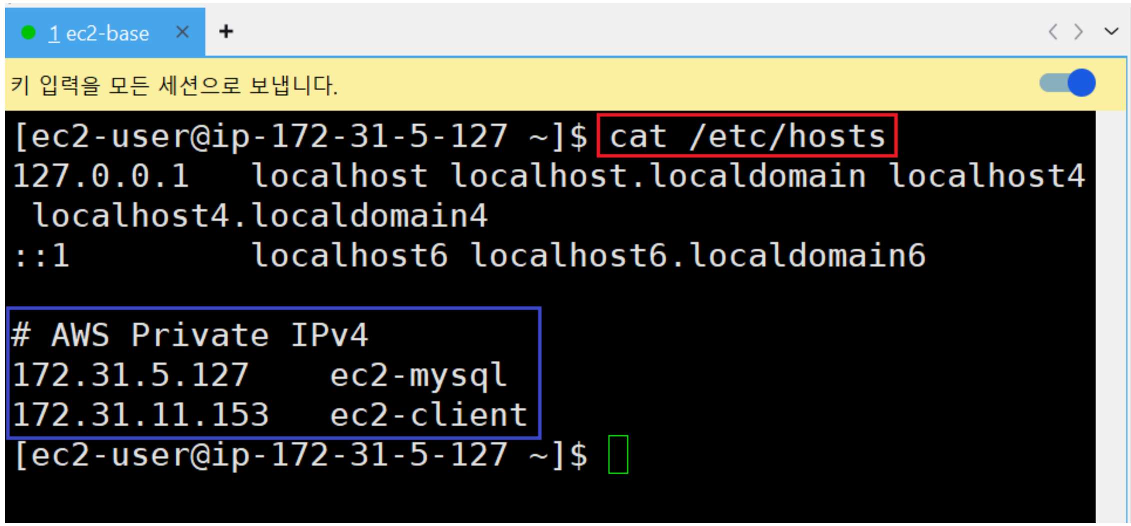


키 입력을 모든 세션으로 보냅니다.

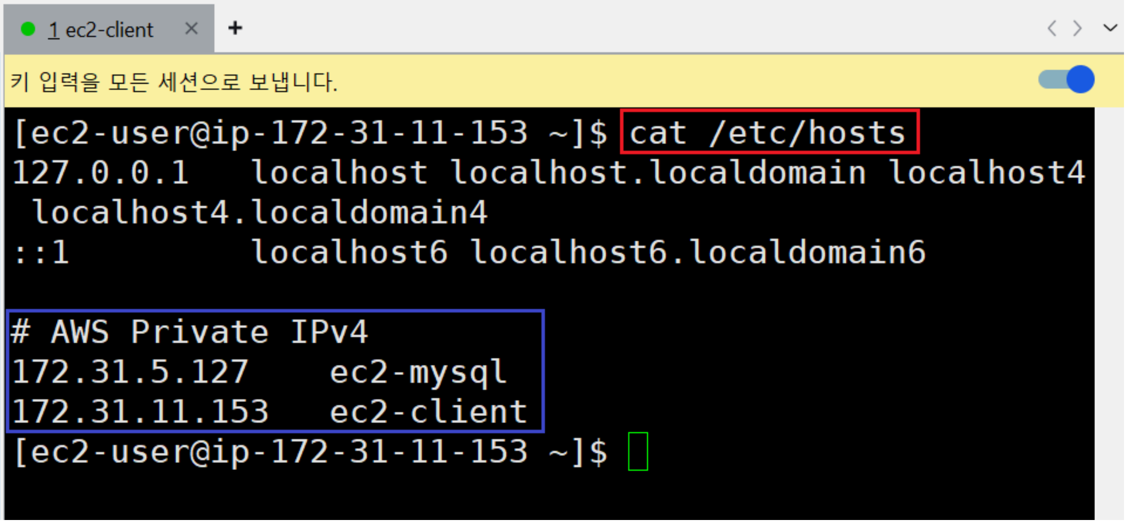
```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
~
```

단계7: 수정된 내용 확인

```
cat /etc/hosts
```

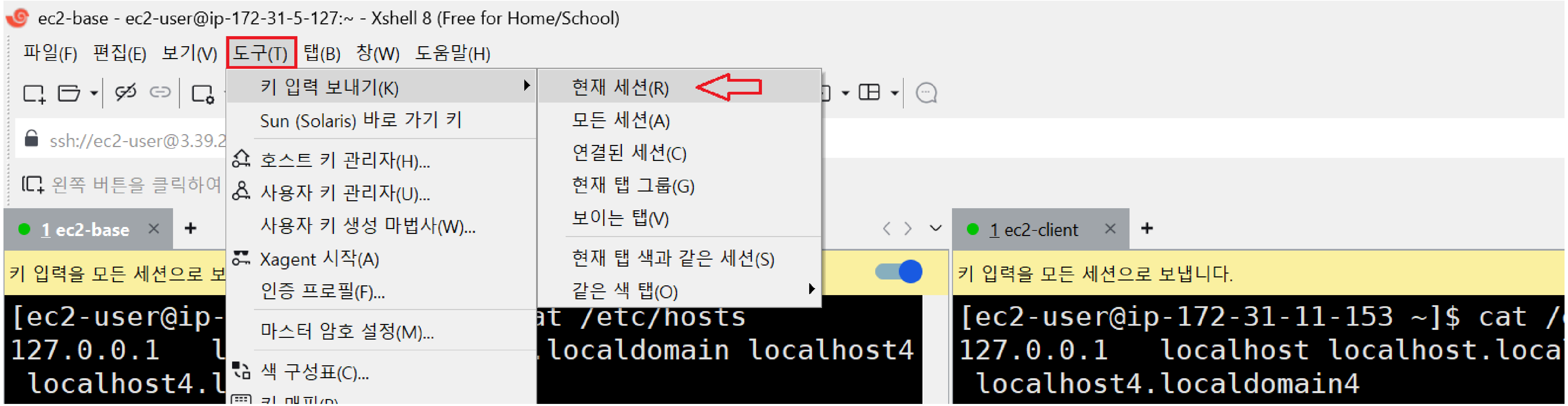


```
[ec2-user@ip-172-31-5-127 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
[ec2-user@ip-172-31-5-127 ~]$ 
```



```
[ec2-user@ip-172-31-11-153 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
[ec2-user@ip-172-31-11-153 ~]$ 
```

단계8: 현재 세션



ec2-base - ec2-user@ip-172-31-5-127:~ - Xshell 8 (Free for Home/School)

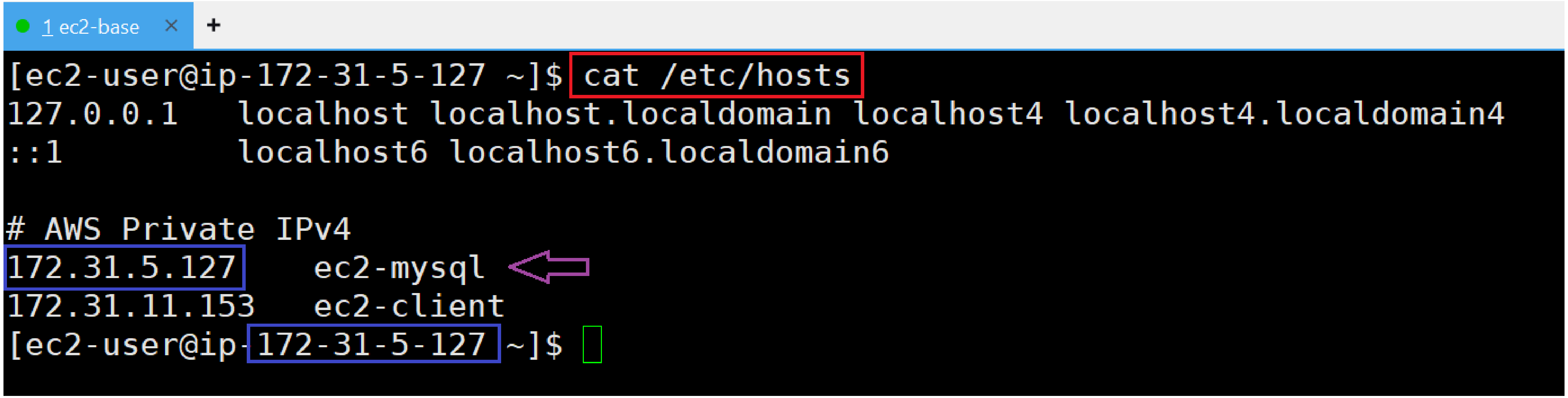
키 입력을 모든 세션으로 보

```
[ec2-user@ip-                          at /etc/hosts
127.0.0.1     l                        .localdomain localhost4
 localhost4.l
```

키 입력을 모든 세션으로 보냅니다.

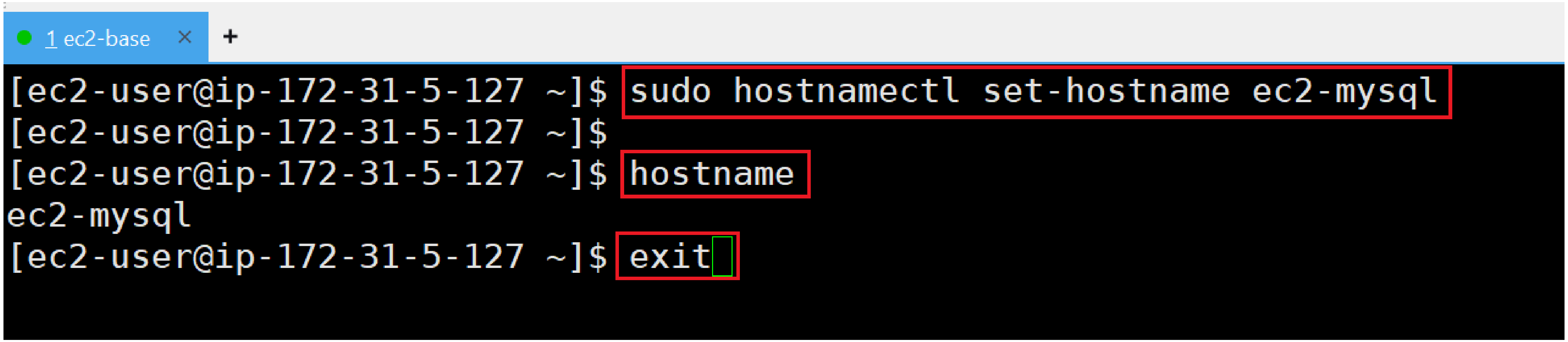
```
[ec2-user@ip-172-31-11-153 ~]$ cat /
127.0.0.1    localhost localhost.loca
 localhost4.localdomain4
```

단계9: Host name 변경 > ec2-mysql



```
[ec2-user@ip-172-31-5-127 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
[ec2-user@ip-172-31-5-127 ~]$ 
```

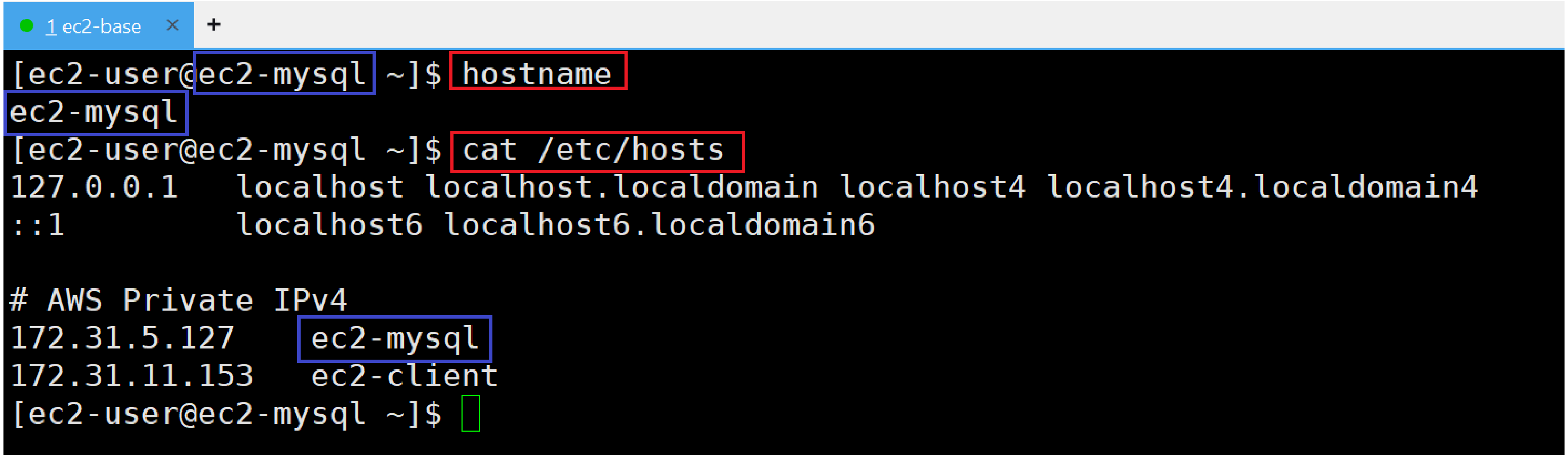
```
# host name 수정
sudo hostnamectl set-hostname ec2-mysql
# host name 확인
hostname
# 종료
exit
```



```
[ec2-user@ip-172-31-5-127 ~]$ sudo hostnamectl set-hostname ec2-mysql
[ec2-user@ip-172-31-5-127 ~]$
[ec2-user@ip-172-31-5-127 ~]$ hostname
ec2-mysql
[ec2-user@ip-172-31-5-127 ~]$ exit
```

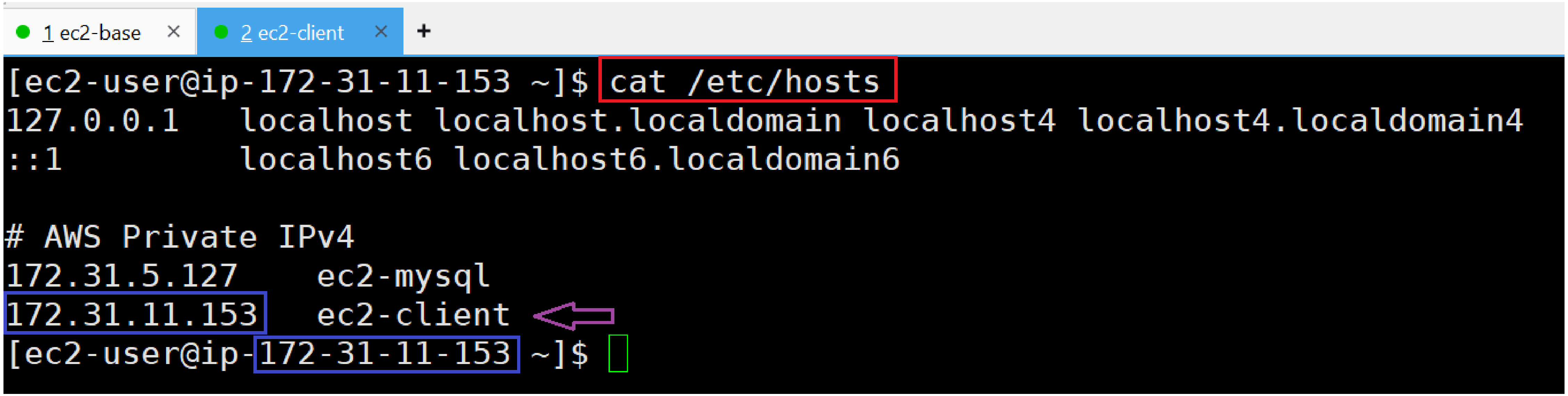
- 재접속 및 결과 확인

```
hostname
cat /etc/hosts
```



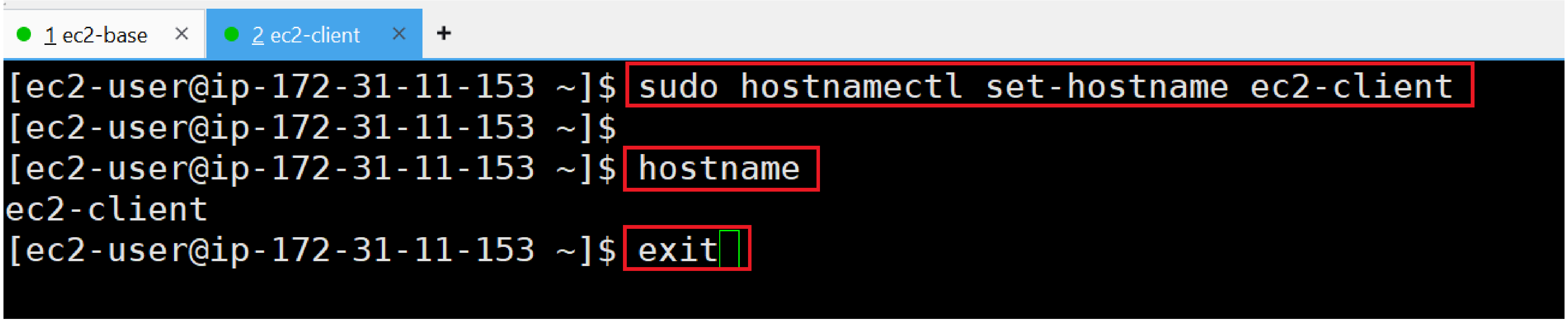
```
[ec2-user@ec2-mysql ~]$ hostname
ec2-mysql
[ec2-user@ec2-mysql ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
[ec2-user@ec2-mysql ~]$ 
```

단계10: Host name 변경 > ec2-client



```
[ec2-user@ip-172-31-11-153 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
[ec2-user@ip-172-31-11-153 ~]$ 
```

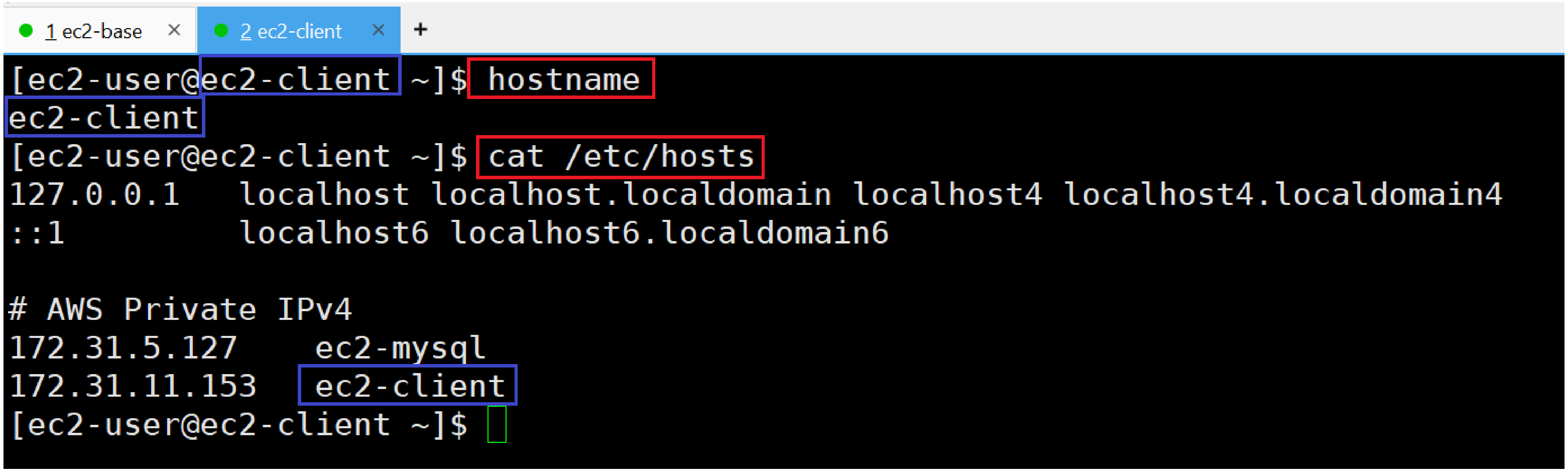
```
# host name 수정
sudo hostnamectl set-hostname ec2-client
# host name 확인
hostname
# 종료
exit
```



```
[ec2-user@ip-172-31-11-153 ~]$ sudo hostnamectl set-hostname ec2-client
[ec2-user@ip-172-31-11-153 ~]$
[ec2-user@ip-172-31-11-153 ~]$ hostname
ec2-client
[ec2-user@ip-172-31-11-153 ~]$ exit
```

- 재접속 및 결과 확인

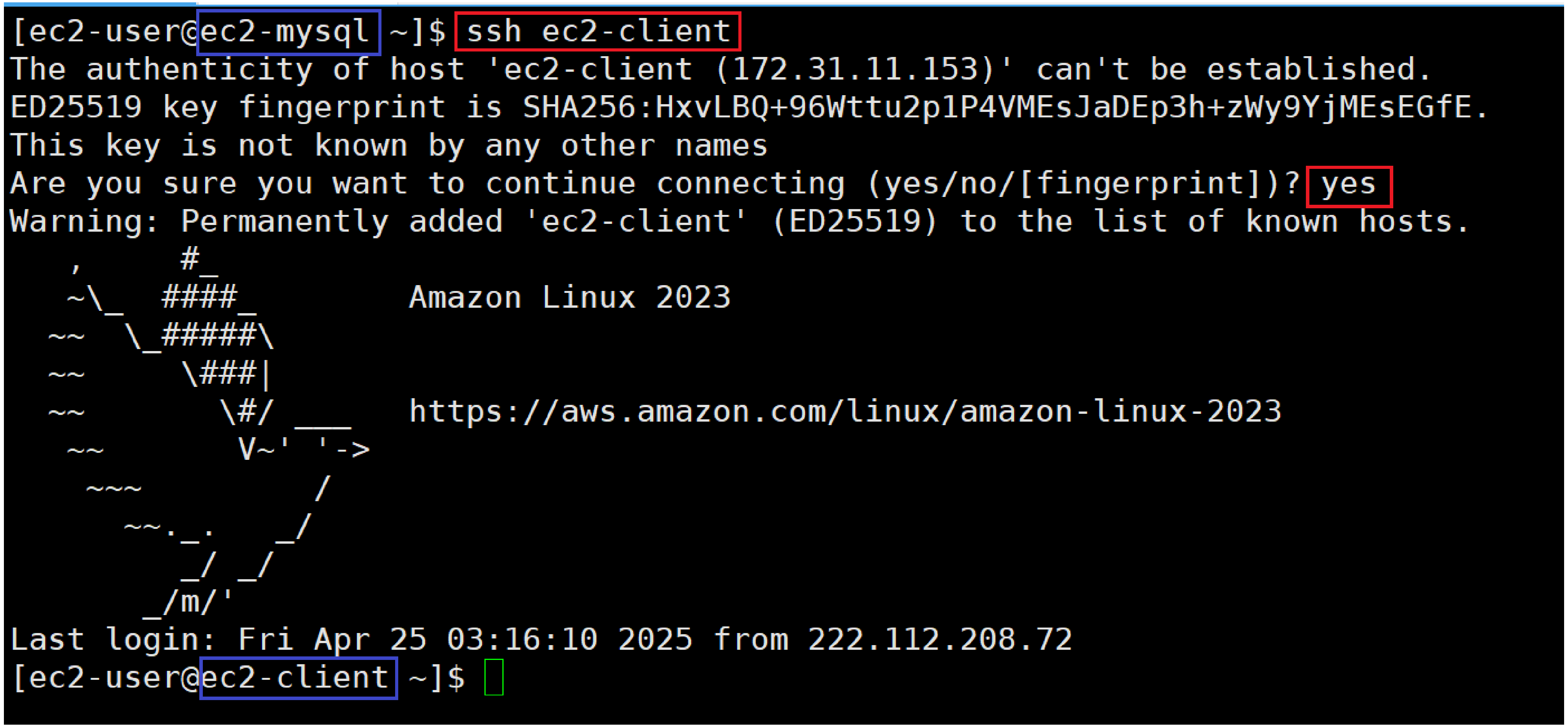
```
hostname
cat /etc/hosts
```



```
[ec2-user@ec2-client ~]$ hostname
ec2-client
[ec2-user@ec2-client ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6

# AWS Private IPv4
172.31.5.127    ec2-mysql
172.31.11.153   ec2-client
[ec2-user@ec2-client ~]$ 
```

단계11: ec2-mysql에서 ec2-client 접속

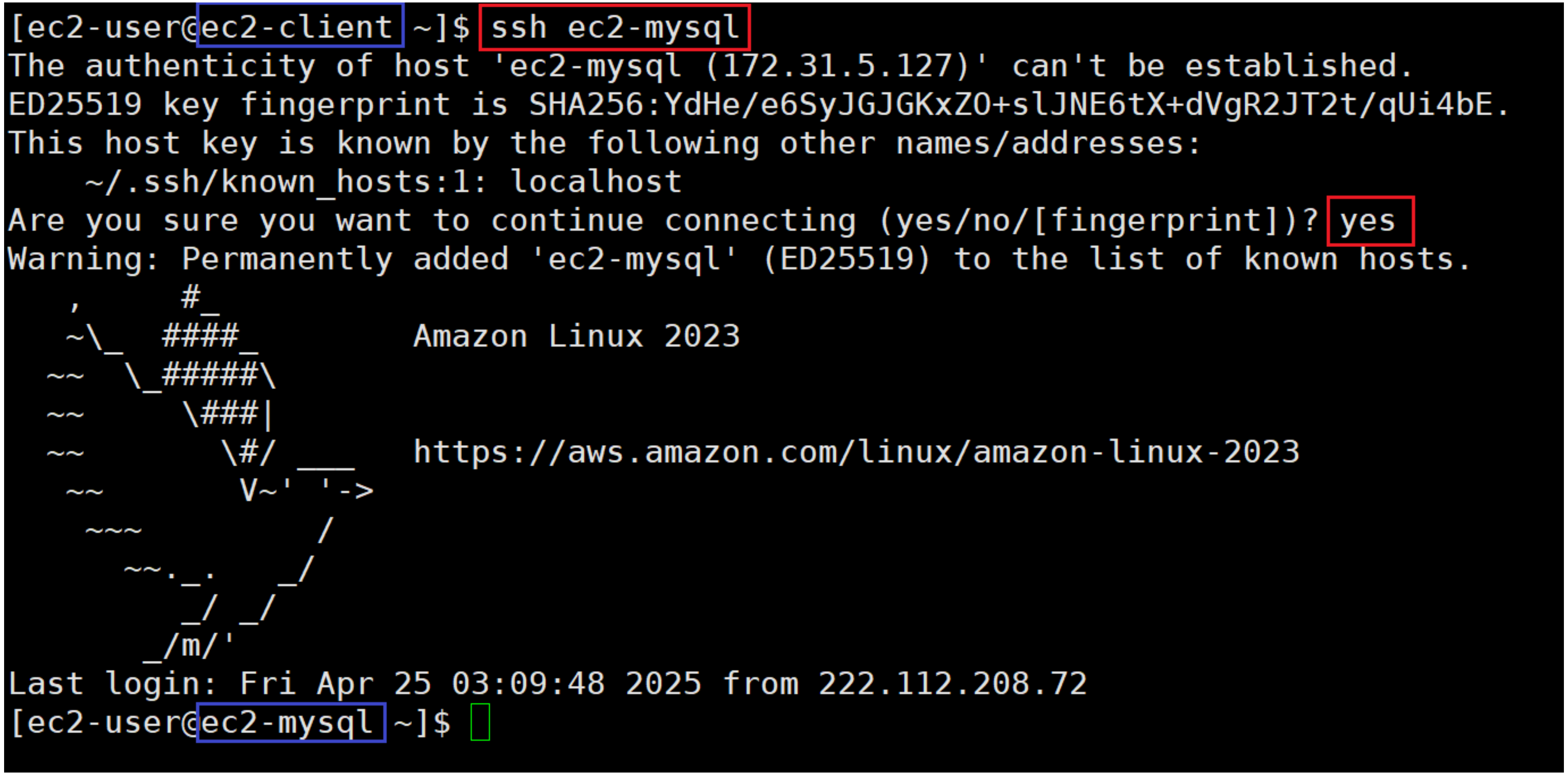
```
ssh ec2-client
# Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```



```
[ec2-user@ec2-mysql ~]$ ssh ec2-client
The authenticity of host 'ec2-client (172.31.11.153)' can't be established.
ED25519 key fingerprint is SHA256:HxvLBQ+96Wttu2p1P4VMEsJaDEp3h+zWy9YjMEsEGfE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-client' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Apr 25 03:16:10 2025 from 222.112.208.72
[ec2-user@ec2-client ~]$
```

단계12: ec2-client에서 ec2-mysql 접속

```
ssh ec2-mysql
# Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```



```
[ec2-user@ec2-client ~]$ ssh ec2-mysql
The authenticity of host 'ec2-mysql (172.31.5.127)' can't be established.
ED25519 key fingerprint is SHA256:YdHe/e6SyJGJGKxZO+slJNE6tX+dVgR2JT2t/qUi4bE.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: localhost
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-mysql' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Apr 25 03:09:48 2025 from 222.112.208.72
[ec2-user@ec2-mysql ~]$ 
```