

02. AWS 컴퓨팅 서비스

2.1. AWS 컴퓨팅 서비스

2.1.1. 컴퓨팅 정의

어떤 것에 대해 계산하여 답을 구하고 추정하는 행위를 컴퓨팅(computing)이라고 볼 수 있다. 이런 컴퓨팅을 전문적으로 수행하기 위해 인간이 아닌 컴퓨팅을 목적으로 하는 특화된 장비(device)들이 있는데, 서버(server)가 대표적이다. 참고로 서버는 특수성과 전문성을 위해 일반적인 컴퓨터에 비해 높은 사양이 요구되는 고성능 컴퓨팅 자원이다. 이런 서버 자원은 온프레미스 형태로 구축해서 사용할 수도 있지만, 퍼블릭 클라우드 환경에서 컴퓨팅 자원을 제공받아 가상 서버를 구성하여 손쉽게 워크로드(workload)를 수행할 수 있다.

2.1.2. AWS 컴퓨팅 서비스

AWS 컴퓨팅 서비스는 퍼블릭 클라우드에서 컴퓨팅 자원을 활용하여 다양한 워크로드를 수행할 수 있는 서비스이다. 컴퓨팅 자원을 사용할 수 있는 비용만 지불하면 수분 내로 가상의 서버 자원을 생성하고 관리할 수 있다.

EC2(Elastic Compute Cloud)

클라우드 환경에서 서버 자원을 인스턴스(instance)라는 가상 머신(Virtual Machine, VM) 형태로 제공하는 가장 기본적인 AWS 컴퓨팅 서비스

ECS(Elastic Container Service)

EC2 기반 관리형 클러스터에서 실행되는 컨테이너 형태의 자원에 대해 배포, 스케줄링(scheduling), 스케일링(scaling) 등을 관리하는 서비스

Lambda

서버리스(serverless) 컴퓨팅 형태로, 서버리스라는 말 그대로 별도의 서버 설정이 없는 환경을 제공하여 코드만 실행해 주는 서비스

Lightsail

독립적인 환경을 제공하며, 최소한의 설정만으로도 손쉽게 사용 가능한 컴퓨팅 서비스

2.2. Amazon EC2 소개

Amazon EC2(Amazon Elastic Compute Cloud)는 AWS의 퍼블릭 클라우드 환경에서 확장 가능한 컴퓨팅 자원을 제공하여 가상의 서버를 운영할 수 있는 서비스이다. 여기서 Elastic은 '탄력적인'이라는 뜻이며, 컴퓨팅 자원에 대해 원하는 만큼 확장하거나 축소하여 유연하게 사용할 수 있는 개념으로 이해하면 된다.

Amazon EC2는 인스턴스라는 가상 컴퓨팅 환경을 기반으로 하며, AMI(Amazon Machine Image)를 이용하여 인스턴스에 필요한 소프트웨어 정보를 정의한다. 이렇게 Amazon EC2 인스턴스는 사용자가 요구하는 CPU, 메모리, 디스크, 운영 체제, 소프트웨어 등을 제공하여 워크로드에 맞는 최적화된 가상의 서버를 생성하고 관리할 수 있다.

2.2.1. Amazon EC2 인스턴스

Amazon EC2 인스턴스는 가상의 컴퓨팅 환경으로 CPU, 메모리, 스토리지, 네트워킹 용량을 결정하는 다양한 인스턴스(실행 중인 임의의 프로세스) 유형을 제공한다. 클라우드 환경에서 컴퓨팅 자원을 필요한 만큼 사용하고, 쓰임을 다하면 자원을 반납하는 형태로 임의로 구성된 인스턴스라고 할 수 있다.

인스턴스 유형

Amazon EC2는 500개가 넘는 인스턴스 유형을 제공하여 사용자 목적과 비즈니스 환경에 맞게 최적화된 선택을 할 수

있다. 현재 AWS 프리 티어 정책은 1년간 매달 750시간의 EC2 인스턴스 무료 서비스를 제공한다. 단 모든 인스턴스 유형에 해당하지 않으며 t2.micro 인스턴스 유형만 해당된다.

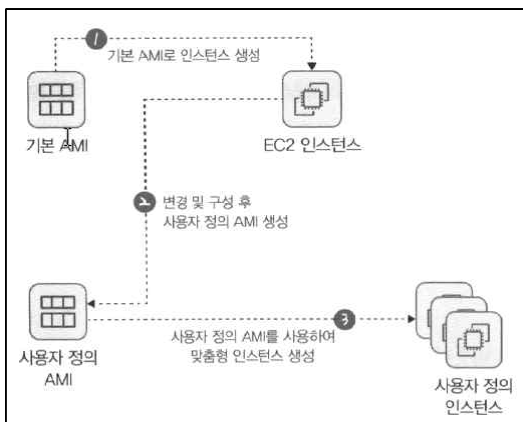
인스턴스 상태

인스턴스 상태는 일반적으로 일곱 가지로 분류할 하며, 어떤 행위에 최종적으로 도달하는 상태와 진행 과정에 따른 상태로 나눌 수 있다. 최종적으로 도달하는 상태는 '실행중(running)', '중지됨(stopped)', '종료됨(terminated)' 상태이며, 진행 과정에 따른 상태는 '대기 중(pending)', '중지 중(stopping)', '재부팅(rebooting)', '종료 중(shutting-down)' 상태이다.

'종료됨' 상태가 되어도 관리 콘솔상에서 바로 삭제되지 않음에 유의한다. 종료되어도 인스턴스가 금방 사라지지 않고 일정 시간 정보를 출력한다.

2.2.2. AMI(Amazon Machine Image)

AMI는 인스턴스를 시작할 때 필요한 정보를 제공하는 것으로 운영 체제와 소프트웨어를 적절히 구성한 상태로 제공 되는 템플릿(template)이다. 인스턴스를 생성할 경우 AMI를 지정해야 하며, 하나의 AMI로 동일한 구성의 여러 인스턴스를 손쉽게 생성할 수 있다.



AWS에서 자체적으로 제공하는 기본 AMI를 활용하여 Amazon EC2 인스턴스를 생성하고, 사용자 요구에 따라 변경하거나 구성하여 사용자 정의 AMI 이미지를 생성한다. 이렇게 만들어진 사용자 정의 AMI를 활용하여 다른 Amazon EC2 인스턴스를 생성할 수도 있다.

2.2.3. Amazon EC2 스토리지

AMI를 통해 Amazon EC2 인스턴스 환경을 설정했다면, 다음으로 어떤 저장소를 사용할지 결정해야 한다. Amazon EC2는 유연하고 효율적이며 사용하기 쉬운 데이터 스토리지 기능을 제공한다. 여기에서 스토리지(storage)는 데이터를 저장하는 공간이라는 의미로, 서버 자원에서 발생하는 다양한 데이터를 보관하려면 필수적으로 존재해야 하는 요소이다. Amazon EC2 인스턴스용 스토리지 유형인 기본적으로 인스턴스 스토어(instance store)와 블록 스토리지인 Amazon EBS(Elastic Block Store)로 나뉜다.

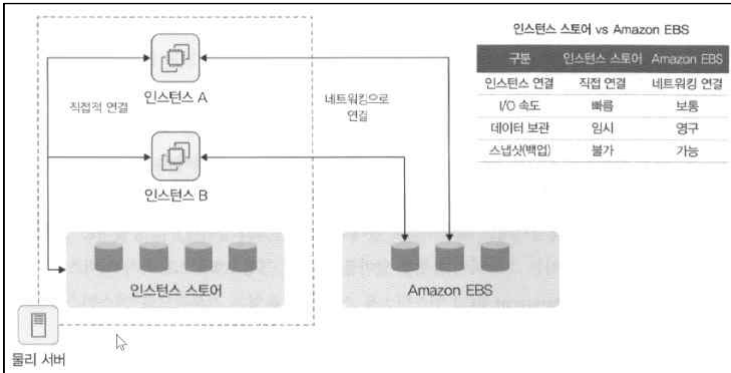
인스턴스 스토어

인스턴스 스토어는 인스턴스에 바로 붙어 있는 저장소로, Amazon EC2 인스턴스를 생성하면 기본적으로 존재하는 스토리지이다. 다만 일부 인스턴스 유형은 인스턴스 스토어를 지원하지 않는다. 직접 붙어 있는 구조 덕분에 매우 빠른 I/O를 보장한다는 장점이 있지만, 인스턴스를 중지하거나 종료하면 인스턴스 스토어에 저장된 데이터가 모두 손실되는 단점이 있다. 이런 측면에서 임시적인 데이터 저장소로 생각할 수 있으며, 장기적으로 보존해야 하는 데이터는 인스턴스 스토어에 저장하지 않는 것이 좋다.

Amazon EBS

외장 하드디스크와 비슷한 개념으로 인스턴스에 연결 및 제거를 하는 형태로 구성되는 블록 스토리지이다. 인스턴스 스토어와 다르게 직접 연결된 구조가 아닌 인스턴스가 네트워킹을 통해 Amazon EBS에 접근하여 연결하는 구조

로, 영구 보존이 가능한 스토리지이다. Amazon EC2 인스턴스가 중지되거나 종료되어도 Amazon EBS에 보존하는 데이터는 그대로 유지할 수 있다.



2.2.4. Amazon EC2 네트워킹

근본적으로 Amazon EC2 인스턴스는 어떤 서비스를 수행하는 특수한 목적으로 만들어진 가상 서버 자원이다. 서비스를 제공하려면 통신이 가능한 환경으로 구성되어야 한다.

Amazon VPC(Virtual Private Cloud)

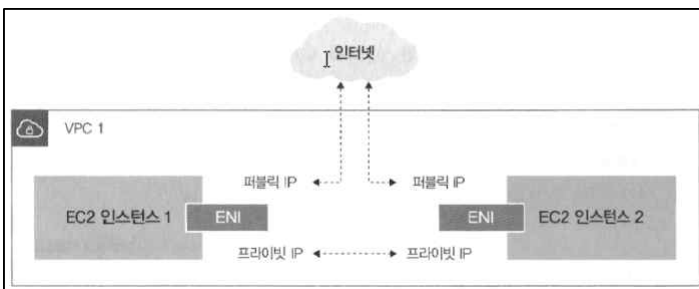
Amazon VPC는 AWS 퍼블릭 클라우드 안에서 논리적으로 격리된 가상의 클라우드 네트워크이다. 생성된 Amazon EC2 인스턴스는 별도로 구성된 하나의 Amazon VPC 안에 생성되어 네트워킹된다.

네트워크 인터페이스(network interface)

네트워킹을 수행하려면 네트워크 인터페이스가 필요하다. Amazon EC2 인스턴스도 마찬가지로 네트워킹을 수행할 네트워크 인터페이스가 필요하다. AWS에서는 ENI(Elastic Network Interface)라는 논리적 네트워크 인터페이스가 VPC 내 생성되며, ENI를 EC2 인스턴스에 연결하여 네트워킹을 수행한다.

IP(Internet Protocol) 주소

네트워크 인터페이스에는 IP 주소가 있으며, IP 주소로 대상을 구분하고 네트워킹을 수행한다. 이런 IP 주소는 크게 프라이빗 IP와 퍼블릭 IP로 나뉜다. 프라이빗 IP는 내부 구간의 통신을 위한 사설 IP고, 퍼블릭 IP는 통신을 위한 공인 IP이다.



2.2.5. Amazon EC2 보안

AWS에서는 클라우드 보안을 중요하게 인식하고 있으며, Amazon EC2도 안정적으로 서비스를 제공하고 관리하기 위해 EC2 보안 기능을 제공한다.

보안 그룹(security group)

보안 그룹은 Amazon EC2 인스턴스의 송수신 트래픽을 제어하는 가상의 방화벽 역할을 한다. 수신 트래픽에 대한 인바운드(inbound) 규칙과 송신 트래픽에 대한 아웃바운드(outbound) 규칙으로 되어 있다.

키 페어(key pair)

키 페어는 Amazon EC2 인스턴스에 연결할 때 자격을 증명하는 보안 키이다. 퍼블릭 키는 Amazon EC2 인스턴스에 저

장되고, 프라이빗 키는 사용자 컴퓨터에 별도로 저장된다.

2.2.6. Amazon EC2 모니터링

Amazon EC2 인스턴스의 모니터링(monitoring)은 서비스 안정성과 가용성, 성능 유지에 필요한 중요한 영역이다. 특히나 불특정 다수에게 제공되는 서비스는 언제 어느 시점에 트래픽이 몰려와 부하가 발생할지 할 수 없다. 인스턴스의 자원은 유한하며, 자원 이상의 부하가 있을 때는 분명 장애가 발생할 수 있다.

수동 모니터링 도구

관리자가 직접 관리 콘솔을 이용하여 모니터링을 수행한다.

자동 모니터링 도구

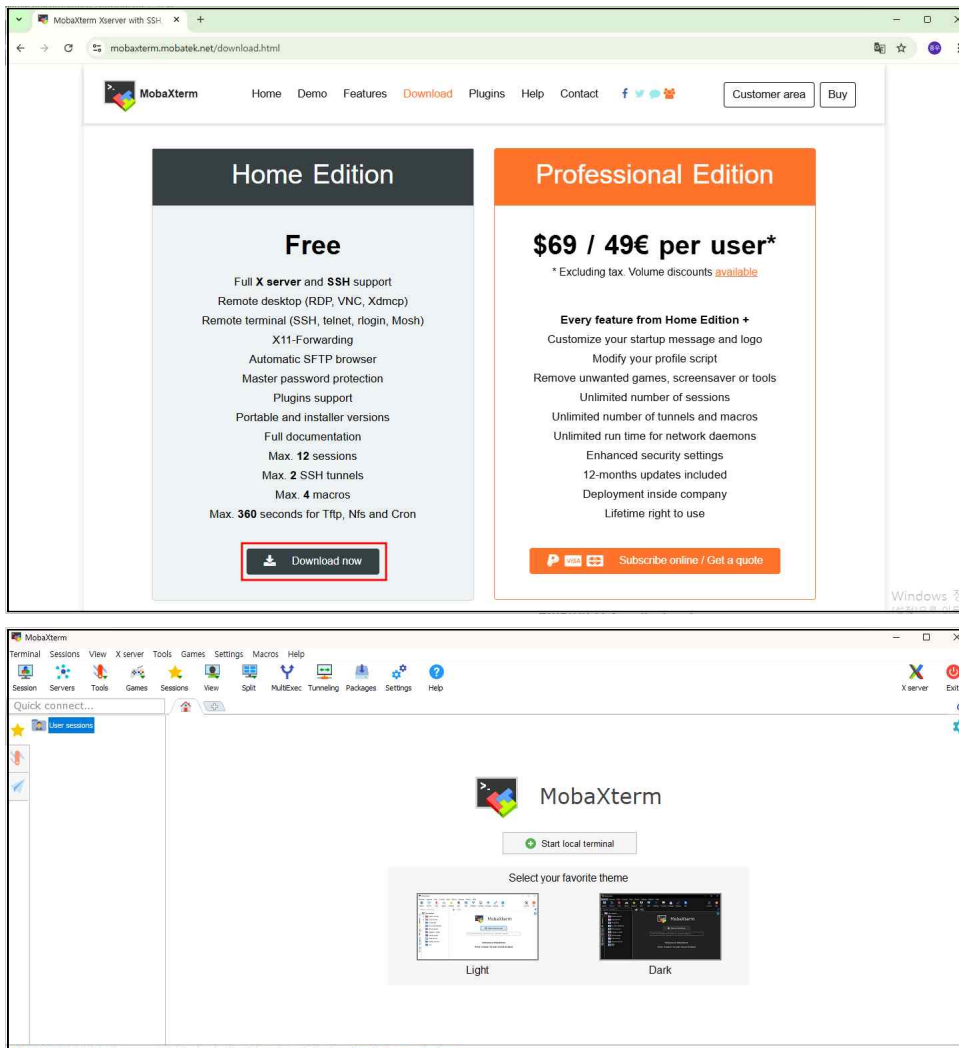
대상 자원의 지표에 대해 임계값을 정하고, 임계값을 초과하면 경보(alarm)를 내리는 형태의 동적 모니터링 방법을 의미한다.

2.3. Amazon EC2 인스턴스 배포 및 접근하기

SSH 클라이언트 설치하기

Amazon EC2 인스턴스를 생성하고 해당 가상 서버에 접속하여 설정하는 단계가 있다. 실습에서 주로 사용하는 EC2 인스턴스의 운영 체제는 리눅스이며, 해당 인스턴스 접근 SSH(Secure Shell)를 통해 셸 명령어를 기반으로 설정한다. 따라서 실습에 사용할 SSH 클라이언트 프로그램을 설치해야 한다.

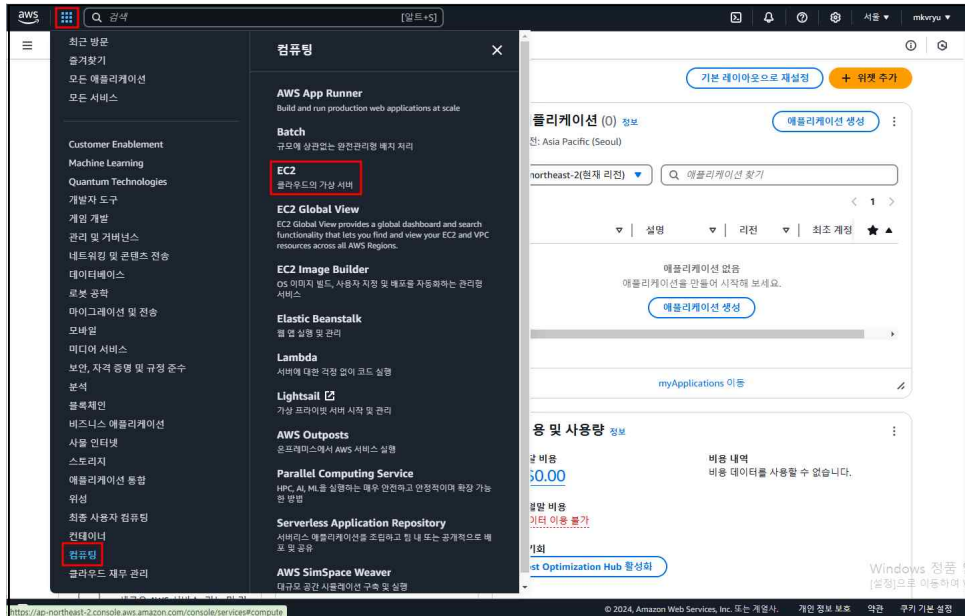
<https://mobaxterm.mobatek.net/download.html>에 접속해서 MobaXterm Home Edition 프로그램을 설치한다.



EC2 키 페어 내려받기

EC2 인스턴스에 접근할 때 자격을 증명하는 데 보안 키 정보가 필요하고 보안 키를 발급받으려면 AWS에서 키 페어를 생성해야 하며, 프라이빗 키 파일을 본인의 실습용 데스크톱에 저장해 두어야 한다.

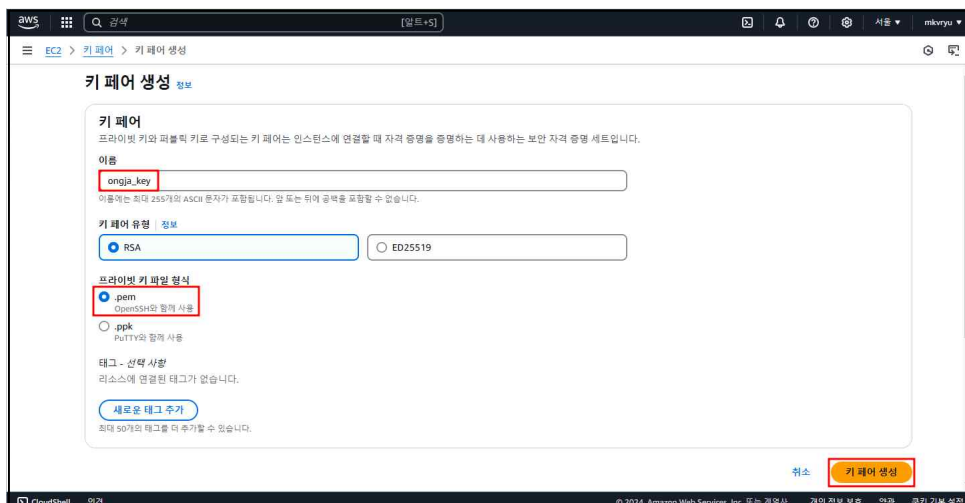
AWS 웹 페이지에 접근해서 관리 콘솔에 로그인한다. 왼쪽 위에 있는 서비스를 클릭하고 컴퓨팅 => EC2를 선택해서 EC2 설정 페이지로 진입한다.



왼쪽 EC2 설정 메뉴에서 네트워크 및 보안 => 키 페어를 선택하고 키 페어 생성을 누른다.



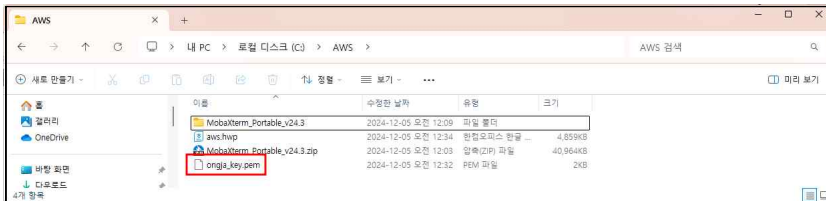
키 페어 설정 페이지에서 이름을 입력하고, 프라이빗 키 파일 형식에서 .pem을 선택한 후 키 페어 생성을 누른다.



키 페어가 생성된 것을 확인할 수 있다.



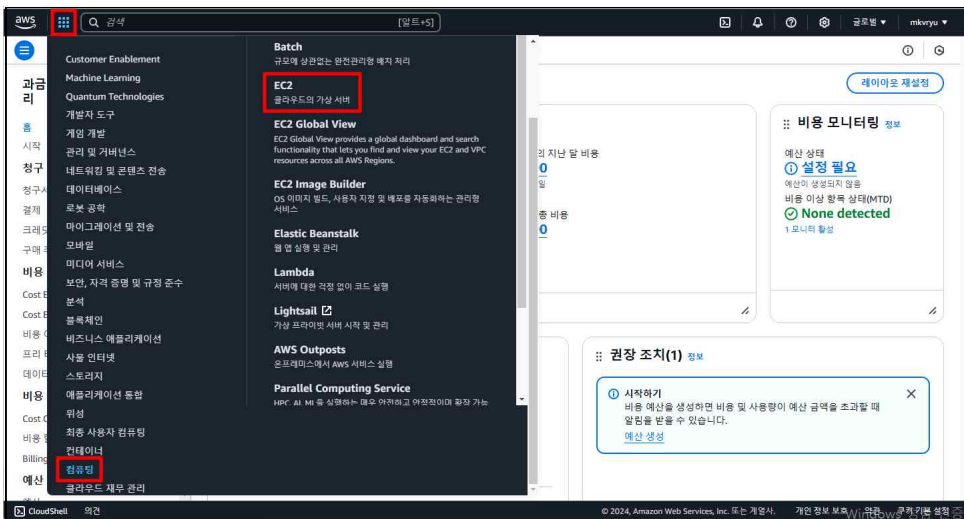
키 페어가 생성되면 이름.pem 파일을 내려받아 데스크톱의 적당한 폴더에 해당 파일을 보관한다.



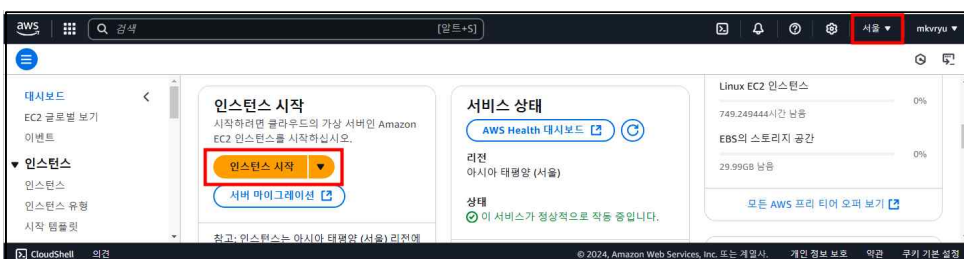
2.3.1. AMI를 이용한 EC2 인스턴스 배포하기

AWS 관리 콘솔에서 EC2 서비스에 대한 인스턴스를 손쉽게 배포할 수 있다. 사용자 요구에 따라 기본적으로 다음 사항을 정의해야 한다.

AWS에 로그인하고 서비스 => 컴퓨팅 => EC2를 선택한다.



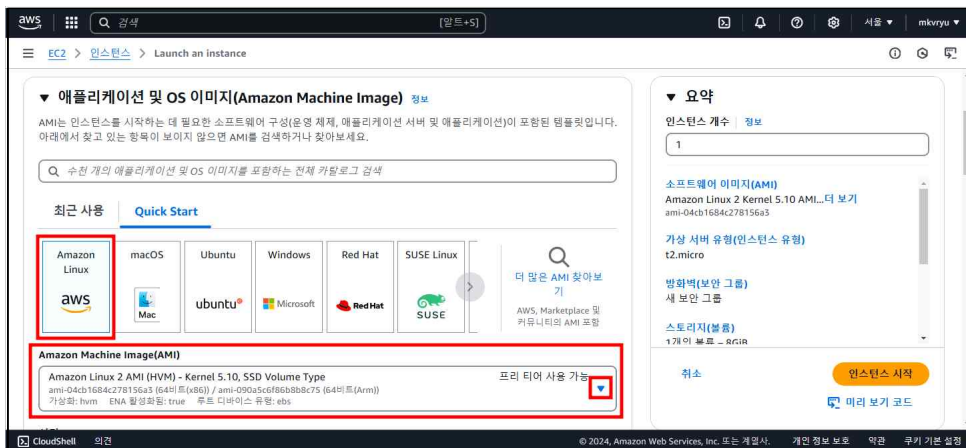
EC 대시보드가 나타나면 인스턴스 시작을 눌러 EC2 인스턴스를 생성하는 설정 페이지로 넘어간다. 이 실습은 서울 리전에서 수행하니 혹시 다른 리전에 있다면 서울 리전으로 변경한 후 진행한다.



이름 및 태그에서 EC2 인스턴스 생성을 위해 이름 및 태그를 정의한다. 이름은 EC2 인스턴스별로 독립적이므로 다른 EC2 인스턴스와 중복되어도 상관없다.



애플리케이션 및 OS 이미지에서 인스턴스의 운영 체제 및 소프트웨어 영역을 정의하는 AMI를 선택해야 한다. 프리 티어를 사용한다면 선택한 AMI는 1년간 매달 750시간 무료로 사용 가능하다.



인스턴스 유형에서는 다양한 인스턴스 유형을 이용하여 CPU와 메모리를 지정하고 있는데, 이 중에서 t2.micro 유형을 선택한다. 프리 티어 용도로 t2.micro를 제공한다.

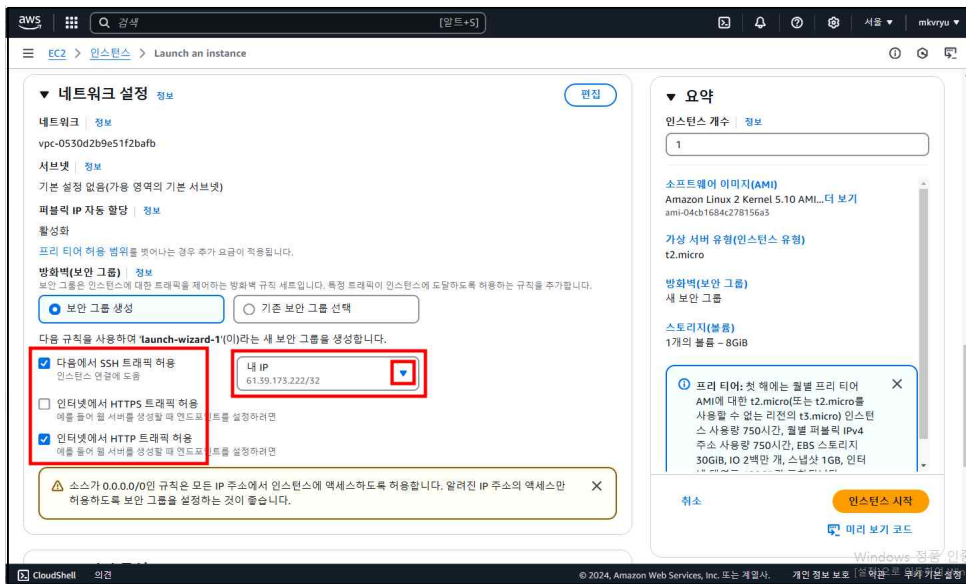


키 페어에서 EC2 인스턴스에서 사용할 키 페어를 선택한다. 각자 생성해서 저장한 키 페어 이름을 선택한다.

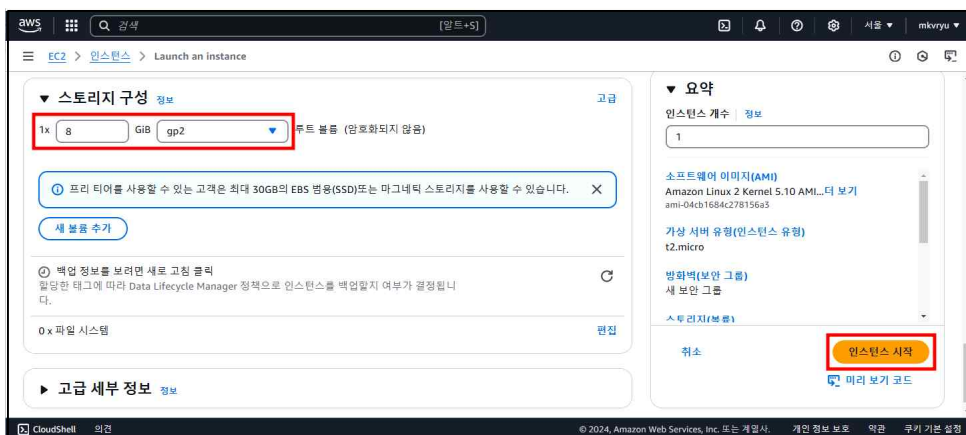


네트워크 설정은 EC2 인스턴스에 대한 네트워크 설정을 수행하는 영역으로, 먼저 인스턴스가 속할 VPC를 지정해야 한다. 사용자가 직접 생성해도 되지만 기본적으로 생성되어 있는 VPC를 지정한다. 웹 서비스를 제공하려면 퍼블릭 IP 자동 할당은 활성화로 지정해야 하며, 기본값으로 활성화된 상태이다.

보안 그룹 설정으로 EC2 인스턴스가 제공할 서비스 접근 제어를 설정할 수 있다. '다음에서 SSH 트래픽 허용'과 '인터넷에서 HTTP 트래픽 허용'에 체크한다. SSH 접근은 관리자 자신만 접근하면 되는 서비스이기에 보안 측면을 고려하여 내 IP로 선택한다.



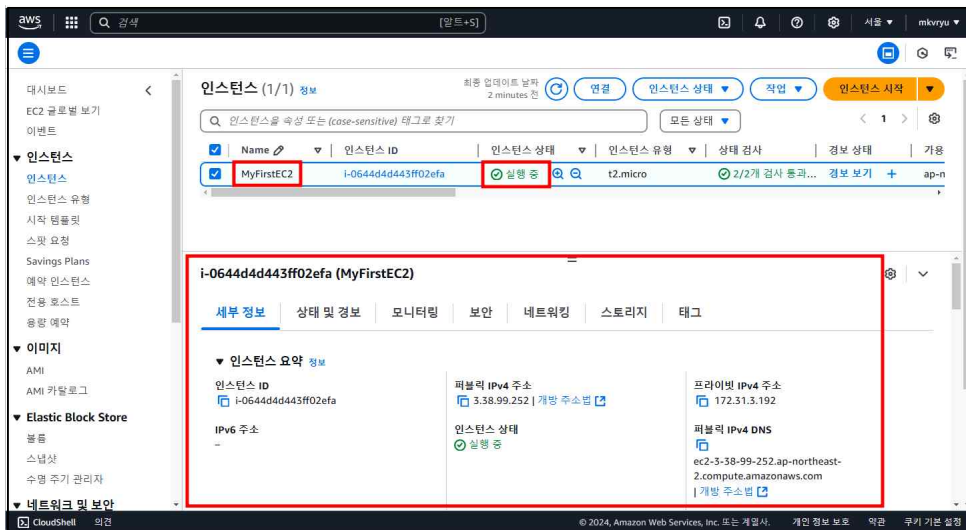
마지막으로 스토리지 구성에서 EC2 인스턴스에 구성될 스토리지를 설정할 수 있다. AWS에서 제공하는 문구를 보면 ‘프리 티어를 사용할 수 있는 고객은 최대 30GB의 EBS 범용(SSD)또는 마그네틱 스토리지를 사용할 수 있습니다.’고 한다. 기본값으로 8GiB 용량과 gp2로 지정되어 있는데, 설정으로 충분한 용량이므로 그대로 유지한다.



모든 작업이 완료되면 EC2 인스턴스 설정 페이지 맨 아래쪽에 있는 인스턴스 시작을 누른다. 모든 설정이 정상적으로 입력되면 다음 그림과 같이 표시된다. 아래쪽에 있는 모든 인스턴스 보기를 누른다.



현재 리전에 존재하는 EC2 인스턴스 정보를 확인할 수 있다. 인스턴스 상태는 약간의 시간이 지난 후 ‘실행 중’인 상태로 전환된다. 생성한 EC2 인스턴스를 선택하면 아래쪽에 지금까지 설정한 인스턴스 정보들을 볼 수 있다.



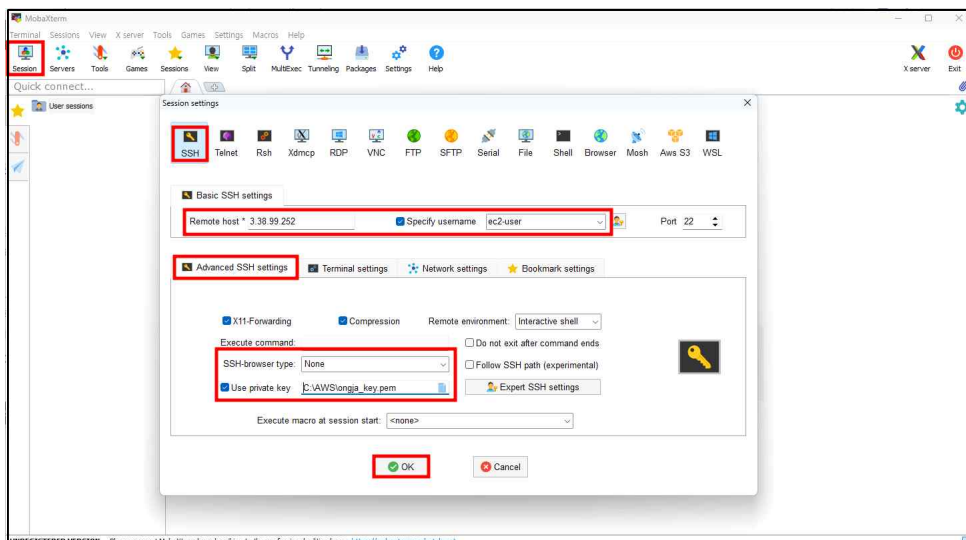
2.3.2. SSH로 EC2 인스턴스에 접속하여 웹 서비스 설정하기

앞서 생성한 EC2 인스턴스에 웹 서비스를 설정한다.

EC2 인스턴스 접속을 위해서는 퍼블릭 IP 주소를 확인해야 한다. 생성된 EC2 인스턴스에 체크하면 아래쪽 인스턴스 세부 정보에서 '퍼블릭 IP 주소'를 확인할 수 있다. IP 주소 왼쪽에 있는 아이콘을 클릭하면 주소가 복사되니 이 퍼블릭 IP 주소를 따로 메모해 둔다.



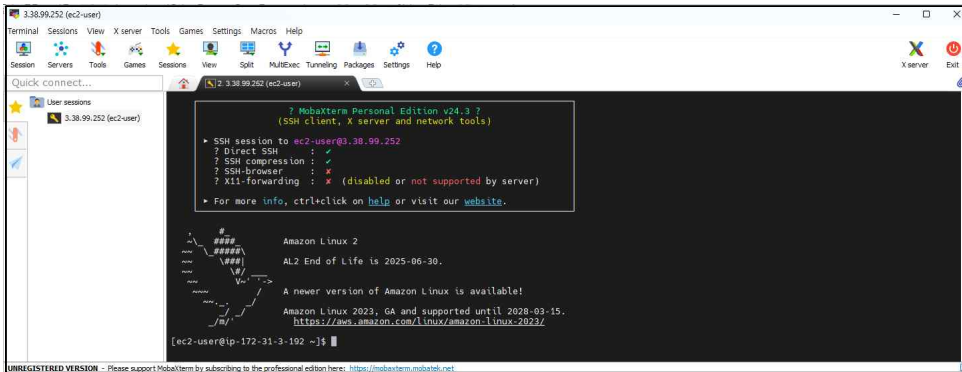
SSH 프로그램(MobaXterm)을 실행하여 다음의 순서대로 설정한다.



메인 메뉴에서 Session을 클릭하여 Session settings 창 열기

settings 창에서 SSH 탭을 클릭하여 SSH 설정 정보 표시
 Remote host에 앞서 복사한 EC2 인스턴스 퍼블릭 IP 주소 붙여 넣기
 'Specify username'에 체크한 후 'ec2-user' 입력
 아래 'Advanced SSH settings'에 체크하여 상세 설정 정보 표시
 SSH-browser type은 None으로 선택
 'Use private key'에 체크한 후 내려받은 키 페어 파일 지정
 모든 설정이 끝나면 OK 누르기

정상적으로 설정되면 다음 그림과 같이 SSH 접속을 하며, 리눅스 셸 명령어를 수행할 수 있는 터미널이 보인다.



다음과 같이 명령어를 하나씩 입력하여 웹 서비스(HTTP 데몬)를 설치하고, 미리 만들어 놓은 최초 웹 페이지를 내려받는다.

```
# MyFirstEC2 인스턴스에 SSH 접속하기
# 슈퍼 유저로 변경
[ec2-user@ip-172-31-3-192 ~]$ sudo su -

# http 데몬 설치
[root@ip-172-31-3-192 ~]# yum install httpd -y
...
Complete!

# http 데몬 실행
[root@ip-172-31-3-192 ~]# systemctl start httpd

# 웹 서비스 최초 페이지 내려받기
[root@ip-172-31-3-192 ~]# curl -L https://bit.ly/afbtest01 > /var/www/html/index.html
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100	113	100	113	0	0	506	0 --:--:-- --:--:-- --:--:-- 506
100	180	100	180	0	0	613	0 --:--:-- --:--:-- --:--:-- 613

sudo su -
 특정 명령어는 권한에 따라 사용할 수 있는 명령어들이 있다. 이에 따라 슈퍼 유저로 변환하는 명령어를 미리 수행한다.
 yum
 특정 패키지를 설치 및 삭제할 수 있는 명령어이다.
 systemctl
 특정 데몬을 실행, 중지, 재시작하는 명령어이다.
 curl
 HTML URL을 실행하는 명령어이다.

2.3.3. EC2 인스턴스에 생성된 웹 서비스 접속하기

현재 동작 중인 EC2 인스턴스의 웹 서비스에 접속하기 위해 앞서 확인한 퍼블릭 IP 주소를 다시 확인하고 인터넷 웹 브라우저(browser)에서 입력하여 접속한다.

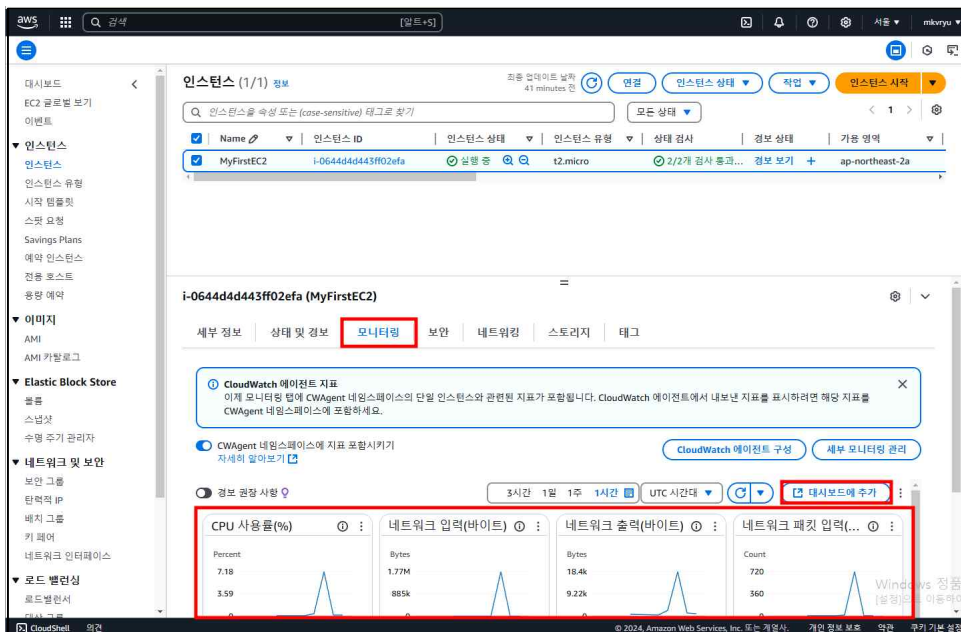


2.3.4. EC2 인스턴스의 모니터링 설정하기

Amazon EC2 인스턴스에 대해 수동 모니터링과 자동 모니터링을 할 수 있다.

수동 모니터링

다음 그림과 같이 앞서 진행했던 Amazon EC2 인스턴스 생성 화면에서 아래쪽에 있는 모니터링 탭을 클릭하면 다양한 지표에서 통계 정보를 확인할 수 있고, 필요에 따라 특정 지표 화면을 최대화하여 출력하거나 통계 기간 및 자동 갱신 기간도 설정할 수 있다.

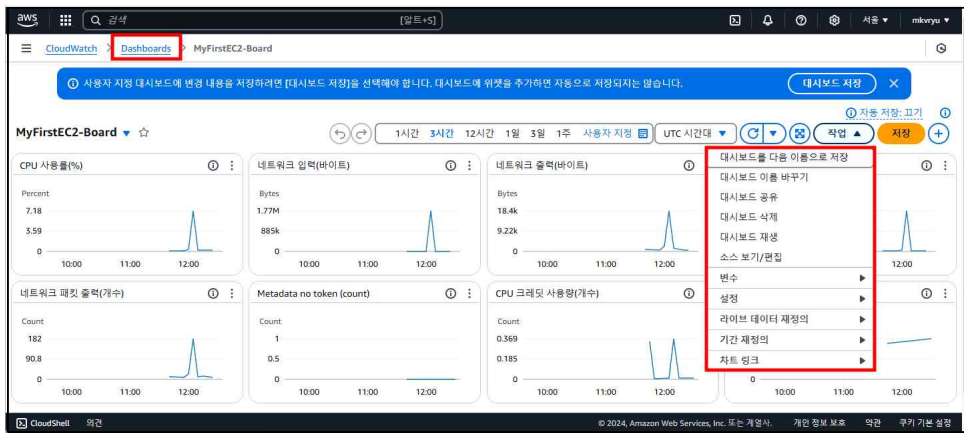


Amazon EC2 인스턴스의 기본적인 모니터링에서 좀 더 상세한 지표나 규칙을 부여하고 싶다면 대시보드에 추가를 눌러 Amazon CloudWatch 대시보드에서 모니터링을 수행할 수도 있다.

새로 생성 => 새 대시보드 이름 입력 => 생성 => 대시보드에 추가를 차례대로 진행하여 CloudWatch 대시보드를 생성한다.

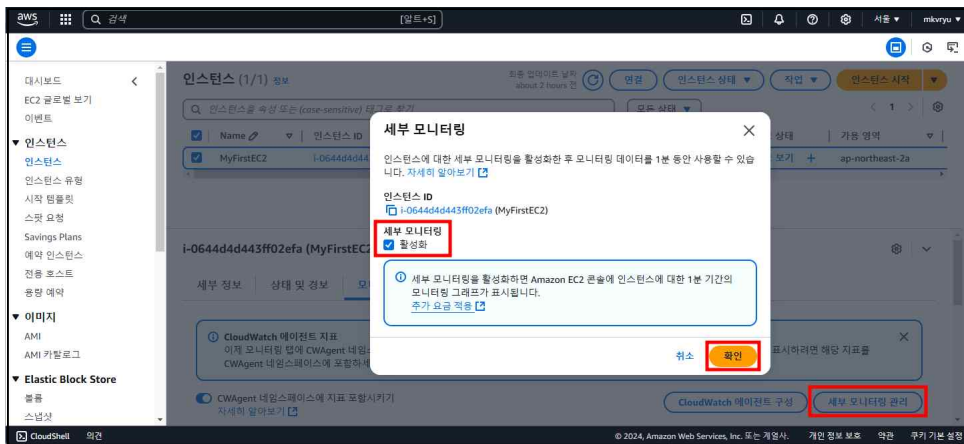


Amazon EC2 인스턴스 모니터링에 있던 지표들이 CloudWatch 대시보드에 출력되며, 좀 더 상세한 작업이나 추가적인 위젯도 생성할 수 있다. 간단히 살펴보고 왼쪽 맨 위에 Dashboards를 클릭하여 대시보드 초기 화면으로 이동한다.

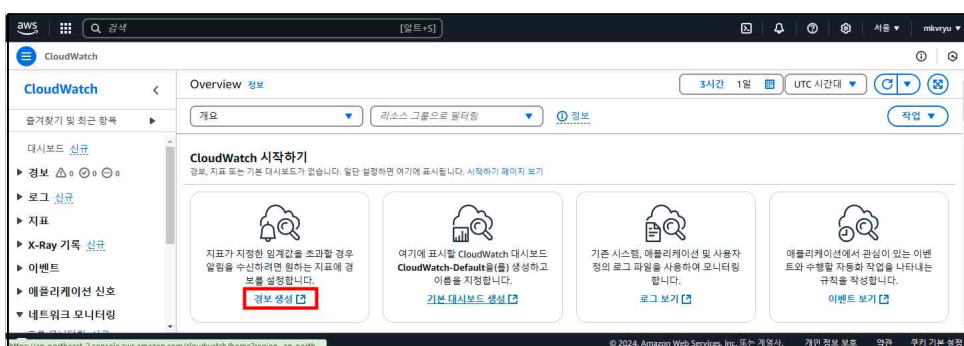
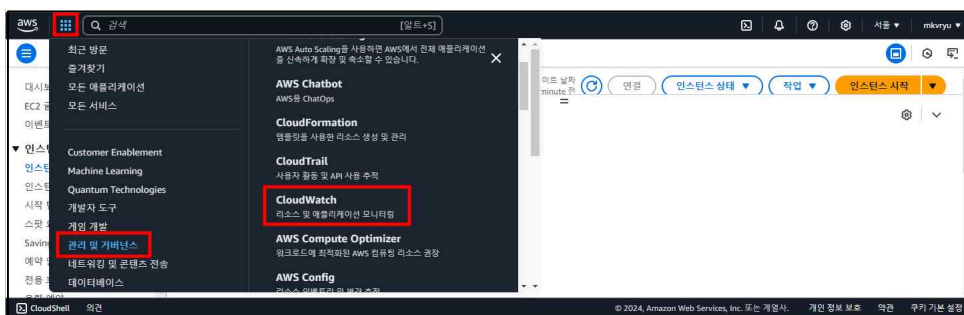


자동 모니터링

가장 먼저 수행할 작업은 Amazon EC2 인스턴스의 세부 모니터링 관리 기능을 활성화시키는 것이다. 기능을 활성화하면 미미한 비용이 발생할 수 있다. 세부 모니터링 관리를 눌러 세부 모니터링 활성화에 체크하고 확인을 누른다.



세부 모니터링 기능을 활성화했다면 Amazon CloudWatch 서비스로 진입한다. Amazon CloudWatch 서비스는 서비스 => 관리 및 거버넌스 => Amazon CloudWatch를 눌러 접근한다. 그 다음 경보 생성을 위해 경보 생성 링크를 클릭한다. 경보 페이지로 진입하면 가운데 위치한 경보 생성을 눌러 설정한다.





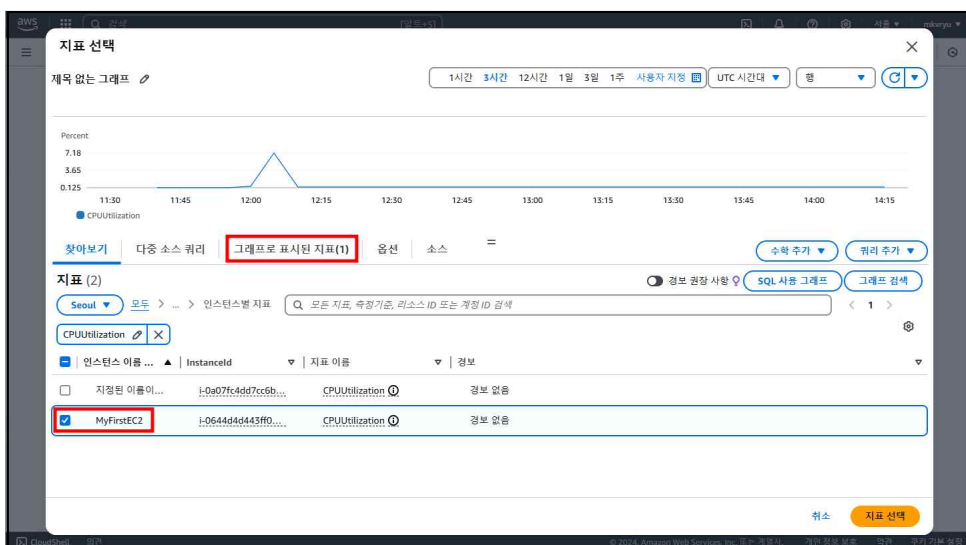
Amazon CloudWatch 경보를 생성하는 첫 번째 단계로 지표 및 조건을 설정하는 페이지로 진입한다. 설정을 위해 지표 선택을 누른다.



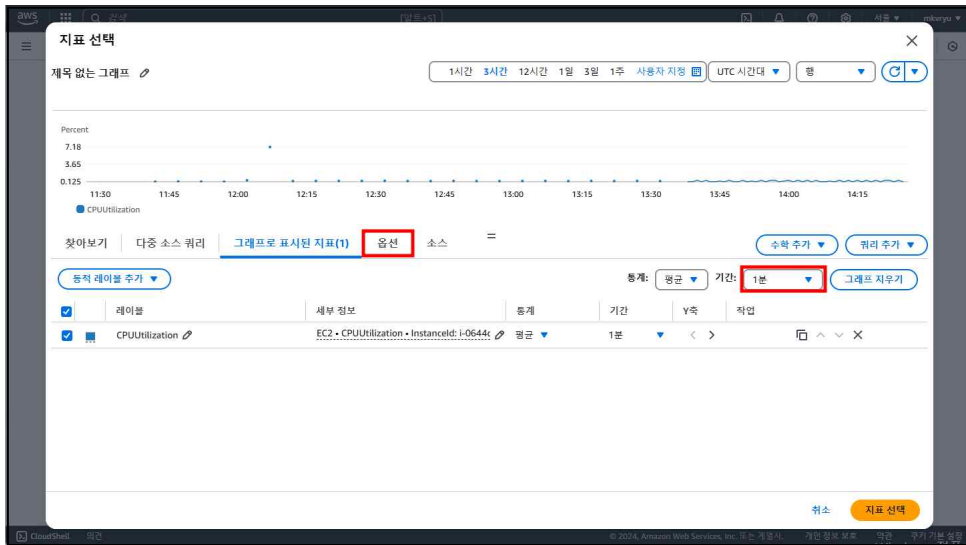
지표 선택 페이지가 표시되면 찾아보기 탭에서 'CPUUtilization'을 검색한 후 EC2 > 인스턴스별 지표를 선택한다.



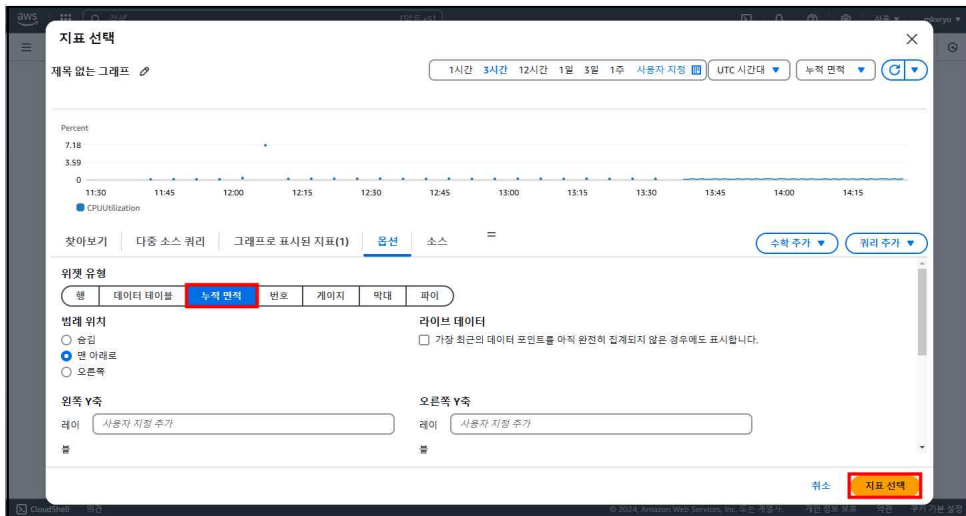
필터링 대상 중 자신이 생성한 EC2 인스턴스에 체크한 후 그래프로 표시된 지표 탭을 클릭한다.



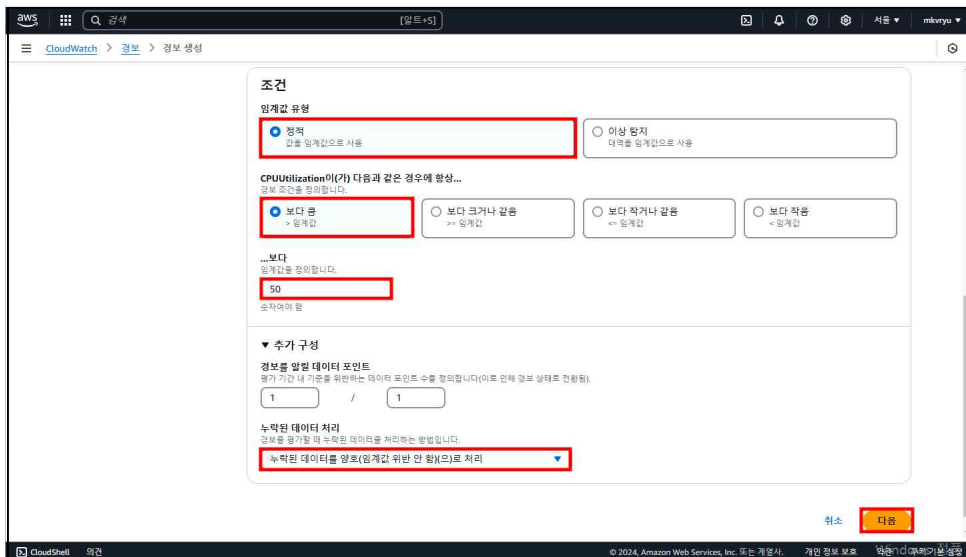
그래프로 표시된 지표 페이지에 진입하면 기간을 1분으로 변경하고 옵션 탭을 클릭하여 다음 단계로 넘어간다.



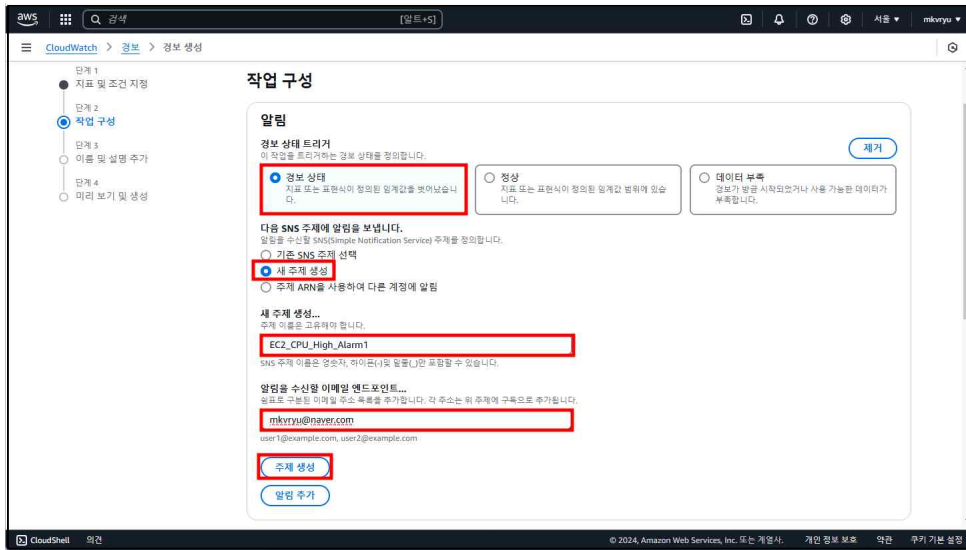
옵선 페이지에서 위젯 유형을 누적 면적으로 선택하고 지표 선택을 누른다.



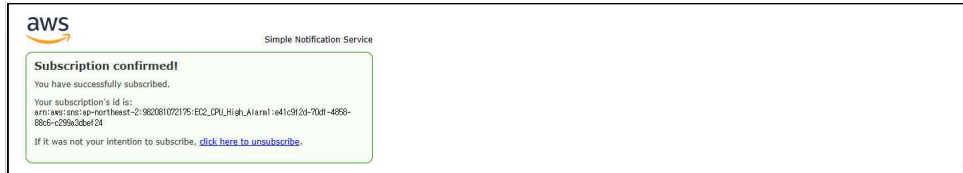
쪽 아래로 내려 지표 선택 아래쪽에 위치한 조건 설정 영역으로 넘어간다. 조건은 다음과 같이 설정하고 다음을 누른다.



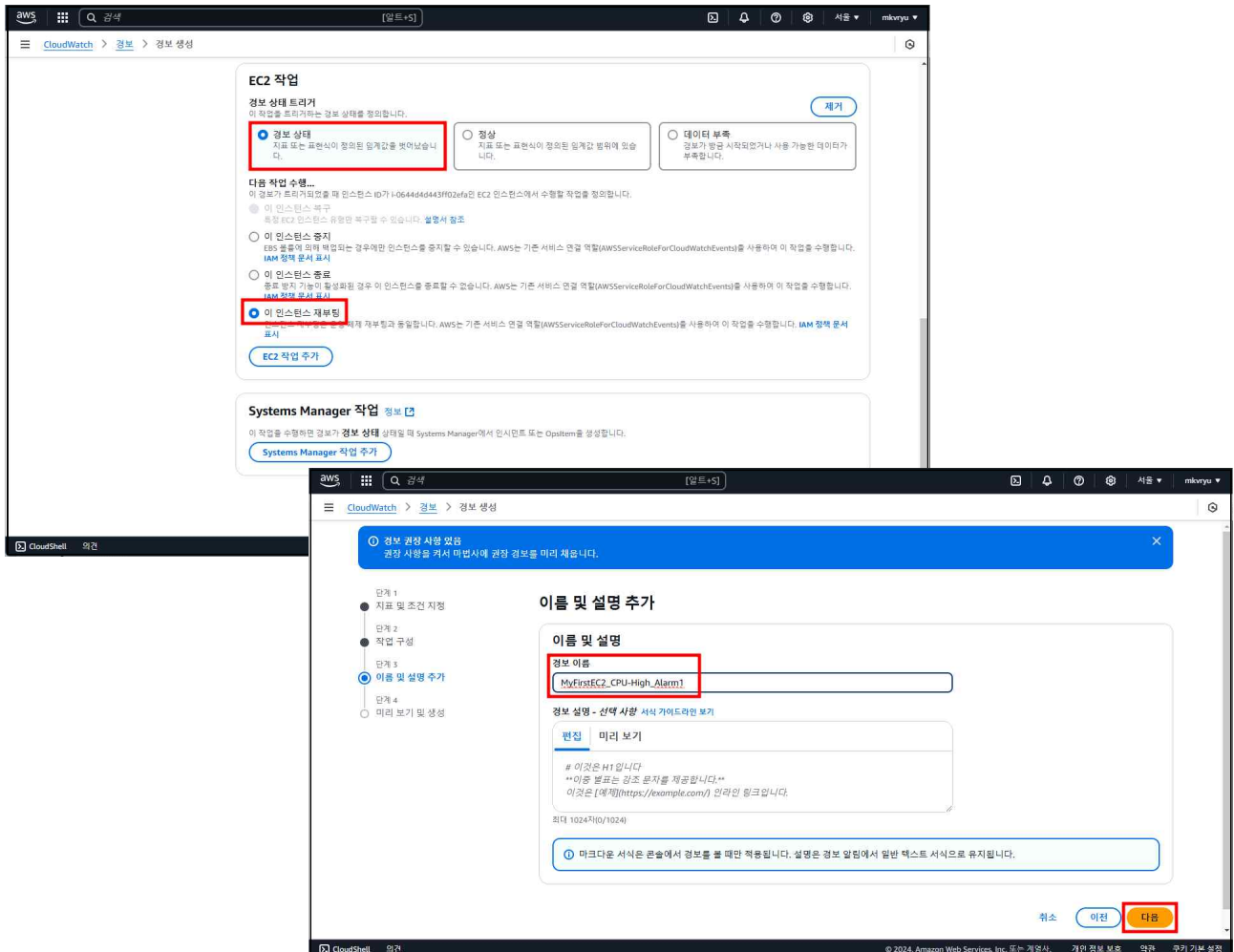
작업 구성 페이지로 넘어가면 경보 설정에 따른 알림을 설정하고 주제 생성을 누른다.



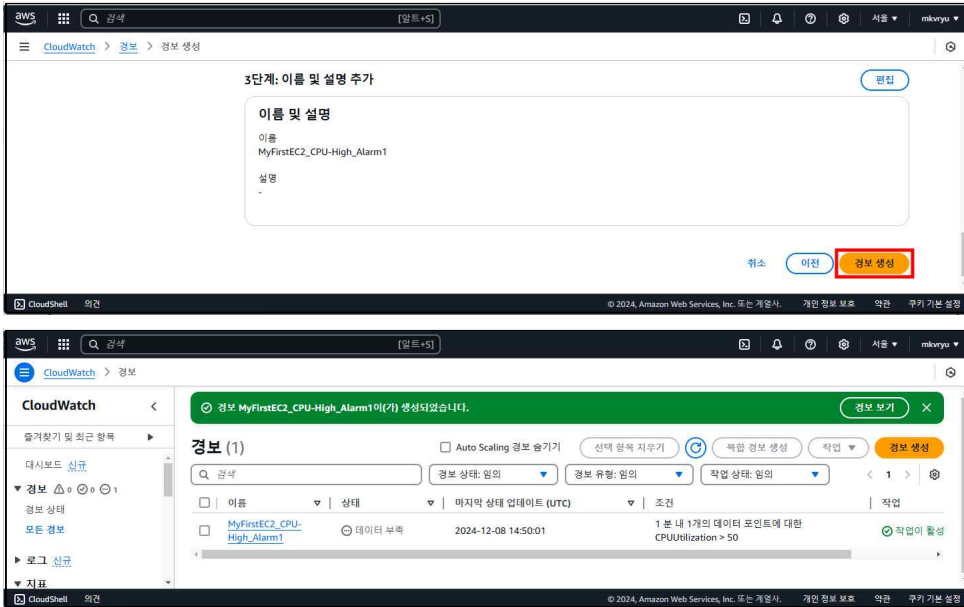
주제 생성을 누르면 입력한 이메일 주소로 Subscription Confirmation 이메일이 전송된다. Confirm subscription 링크를 클릭하면 다음과 같이 승인 메시지가 출력된다.



그다음 알람 설정 아래쪽에 EC2 작업을 설정한다. EC2 작업 설정 영역에서 다음과 같이 설정을 진행하고 다음을 누른다. 생성할 경보 이름과 설명을 입력하는 단계로 넘어간다. 자신만의 경보 이름을 입력하고 다음을 누른다.



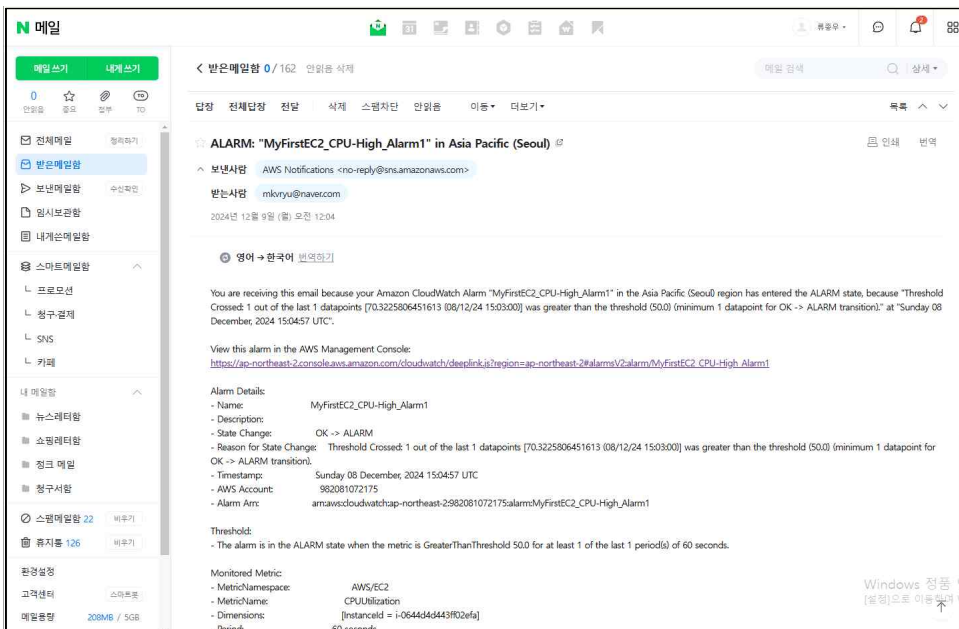
마지막으로 지금까지 설정한 정보를 미리보기하는 단계로 설정된 내용을 훑어보고 맨 아랫쪽 경보 생성을 눌러 마무리한다.



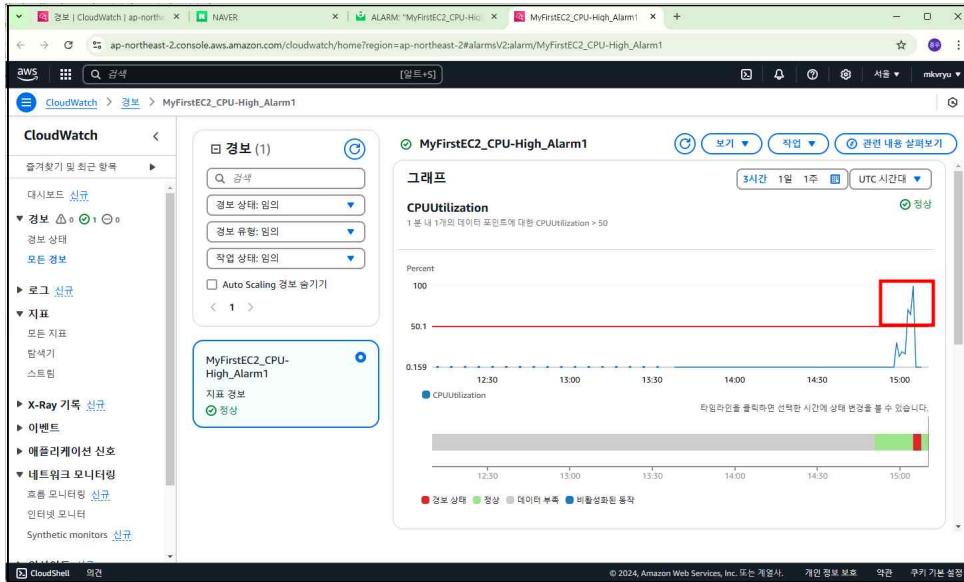
지금까지 설정한 Amazon CloudWatch 경보 설정은 CPU 사용률이 임계값(50%) 이상일 경우 이메일로 알림을 수행하는 것이다. 하지만 현재는 CPU 부하가 거의 없는 상태이기 때문에 자동 경보 실습을 진행하기가 어렵다. 따라서 CPU 부하를 강제로 발생하는 툴을 설치하여 실습을 한다.

```
# MyFirstEC2 인스턴스 SSH 터미널 접속
# CPU 부하 설정 툴 설치
[root@ip-172-31-3-192 ~]# amazon-linux-extras install -y epel
[root@ip-172-31-3-192 ~]# yum install -y stress-ng
# CPU 부하 발생(70%)
[root@ip-172-31-3-192 ~]# stress-ng --cpu 1 --cpu-load 70% --timeout 10m --metrics --times --verify
stress-ng: info: [440] dispatching hogs: 1 cpu
```

stress-ng라는 툴의 명령어로 강제로 CPU 사용률을 70%로 발생시킨다. 현재 경보 설정 임계값 기준이 CPU 50% 이상에서 발생하므로 경보 요건에 충족된다. 이것으로 Amazon EC2 인스턴스는 재부팅할 것이고 경보에 따른 이메일을 송신할 것이다. 잠시 기다려 보면 다음 그림과 같이 AWS에서 보낸 경보 메일이 수신된다.



알림 설정에 따라 60초 동안 1회 CPU 임계값인 50%를 넘어서 Amazon SNS를 통해 이메일을 전송한 것이며, EC2 작업 설정에 따라 Amazon EC2 인스턴스는 재부팅된다. 자세한 확인을 위해 이메일에 있는 링크를 클릭해 접속해 본다.



2.3.5. 리눅스 명령어를 사용하여 EC2 인스턴스의 정보 확인하기

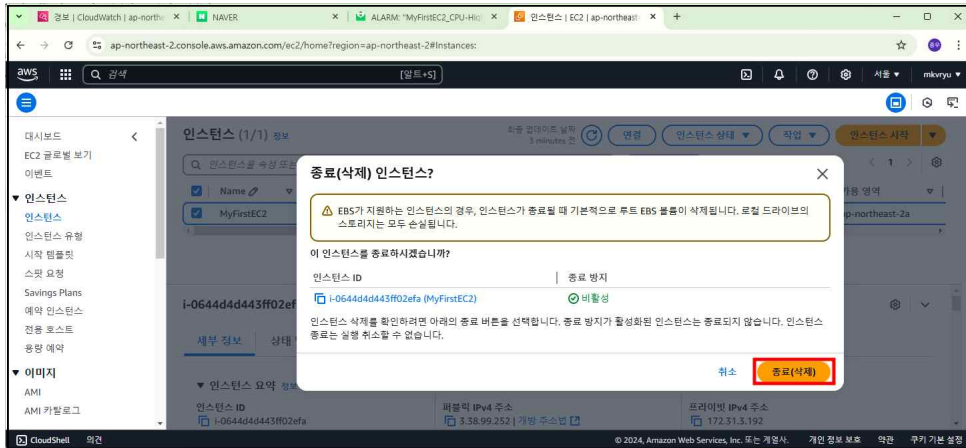
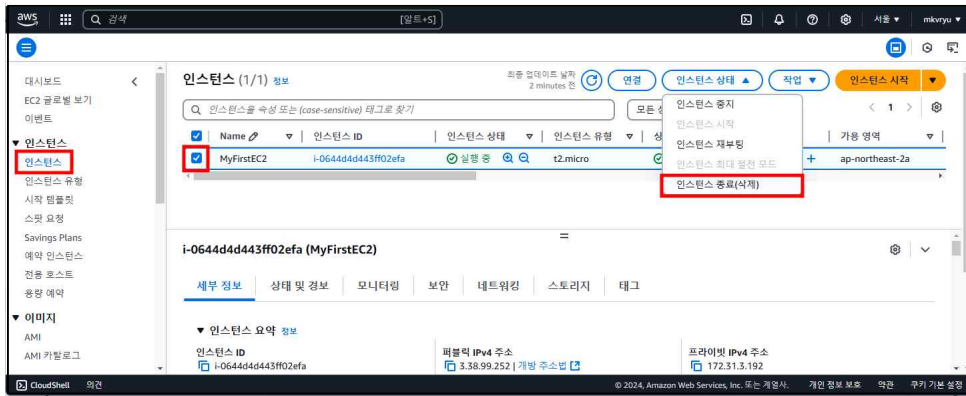
추가로 SSH 터미널에서 리눅스 명령어를 사용하여 EC2 인스턴스의 다양한 구성 요소를 확인해 볼 수 있다.

```
# SSH 터미널 접속
# CPU 정보 확인
[root@ip-172-31-3-192 ~]# cat /proc/cpuinfo | grep name
model name      : Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
# 메모리 용량 확인
[root@ip-172-31-3-192 ~]# cat /proc/meminfo | grep MemTotal
MemTotal:       975524 kB
# 프라이빗 IP 주소 확인
[root@ip-172-31-3-192 ~]# ip -br -c addr show eth0
eth0            UP                172.31.3.192/20 fe80::64:d7ff:fe84:80a1/64
# 퍼블릭 IP 주소 확인
[root@ip-172-31-3-192 ~]# curl ipinfo.io/ip
3.38.99.252
# 스토리지 확인(EBS 볼륨 확인)
[root@ip-172-31-3-192 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda  202:0   0   8G  0 disk
└─xvda1 202:1   0   8G  0 part /

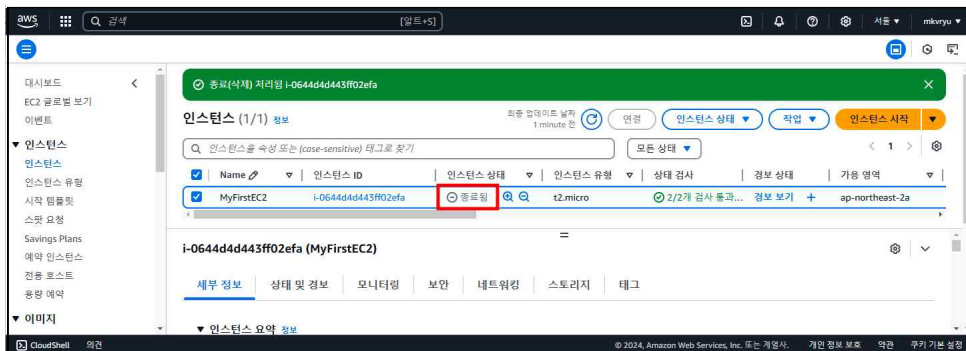
[root@ip-172-31-3-192 ~]# df -hT -t xfs
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/xvda1      xfs   8.0G  1.9G  6.2G  24% /
```

이번 실습으로 EC2 인스턴스를 만들고 웹 서비스를 생성해서 접속하고 정보도 확인해 보았다. 모든 실습이 끝나면 항상 AWS 클라우드 자원을 삭제해야 한다. 이렇게 해야 불필요한 과금을 막을 수 있다.

Amazon EC2 인스턴스를 삭제한다. EC2 => 인스턴스에서 생성한 EC2 인스턴스를 선택한 후 인스턴스 상태 => 인스턴스 종료(삭제)를 선택한다. 이후 새로운 창이 열리면 맨 아래쪽에 있는 종료를 누른다.



그러면 인스턴스 상태는 수분 내로 ‘종료 중 => 종료됨’ 상태로 전환된다. ‘종료됨’ 상태가 되면 정상적으로 EC2 인스턴스가 삭제된 것이니 안심하고 다른 작업을 해도 된다.



CloudWatch => 모든 경보에서 생성한 경보에 체크하고 작업 => 삭제를 선택해서 삭제한다.

