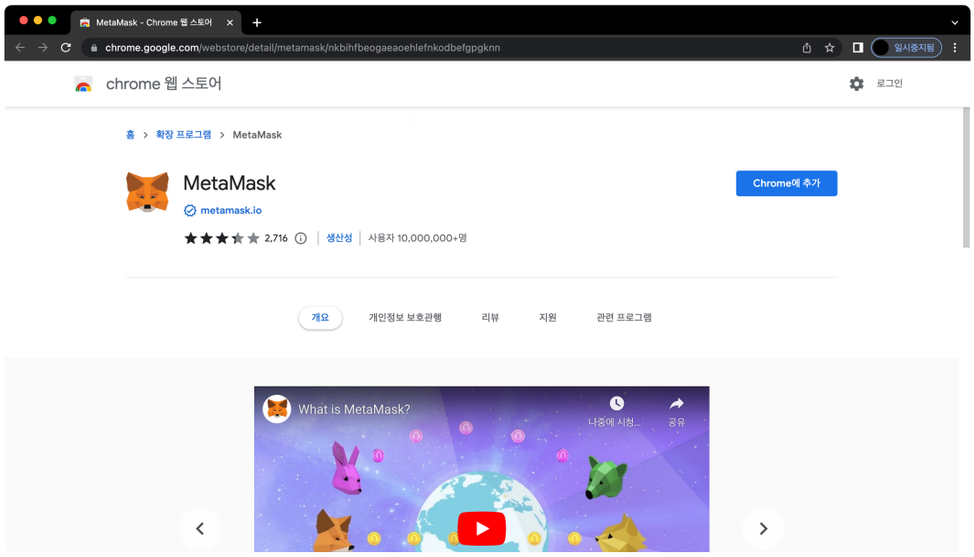
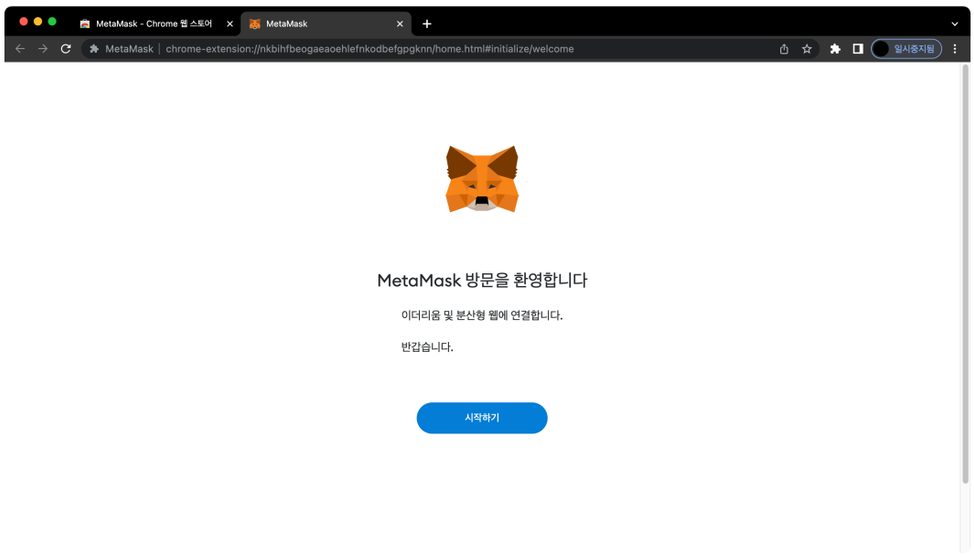
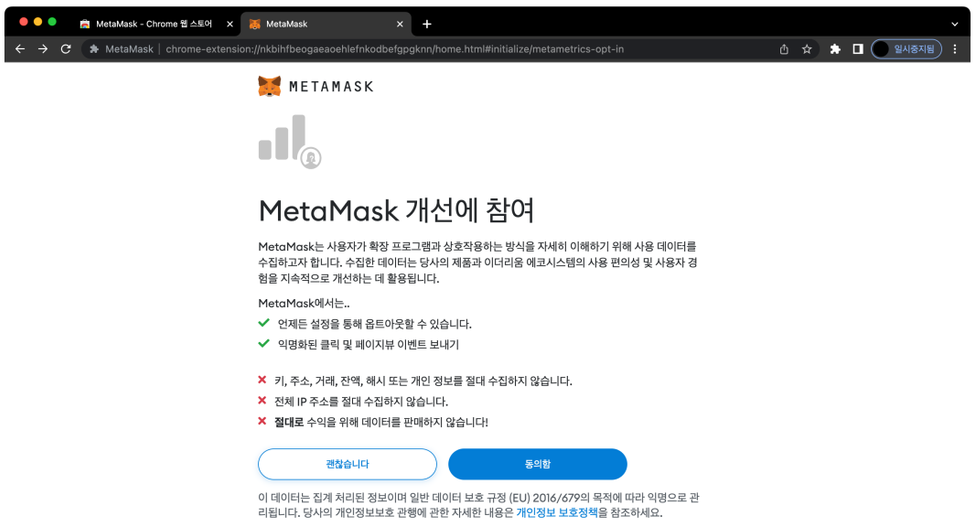
포팅 매뉴얼

Ⅰ. 외부 서비스 Metamask

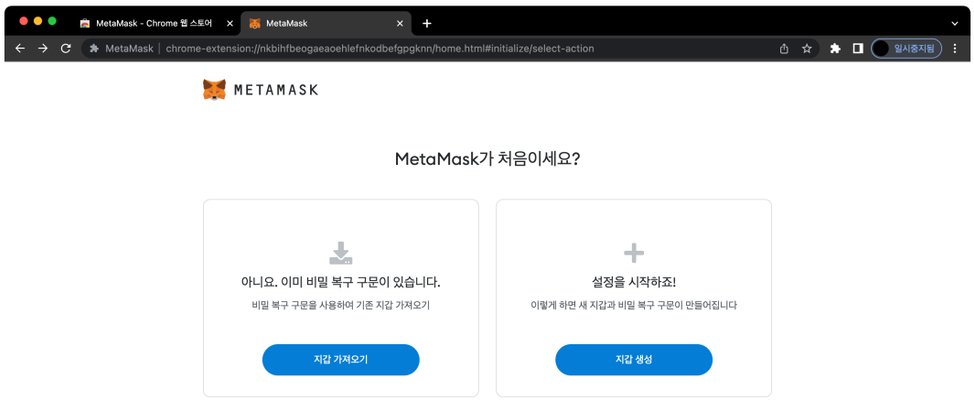
1. 메타마스크 크롬 확장프로그램 설치

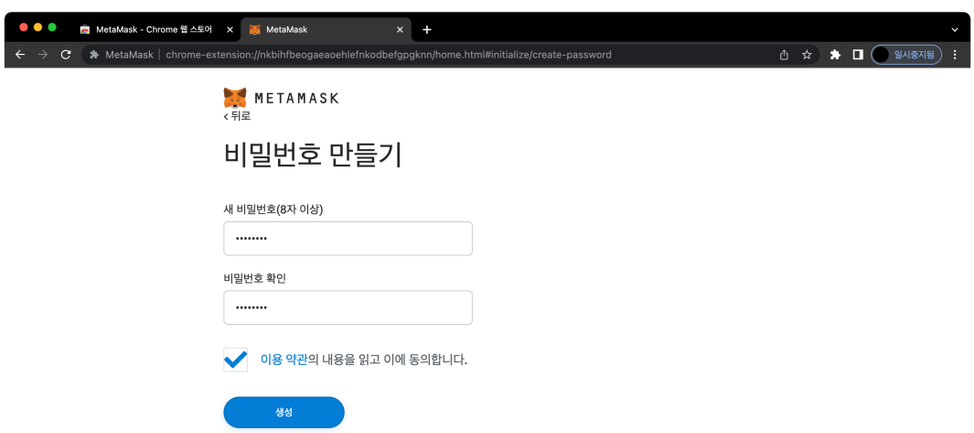


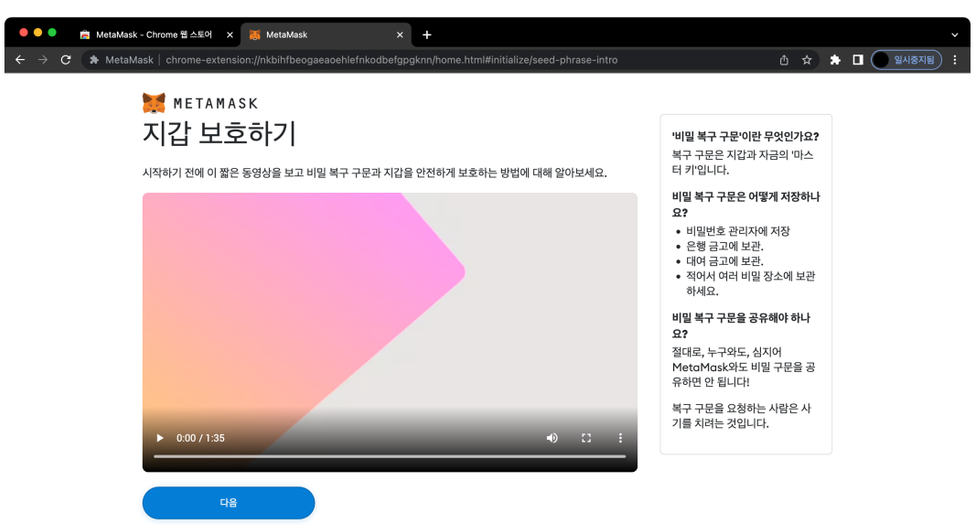


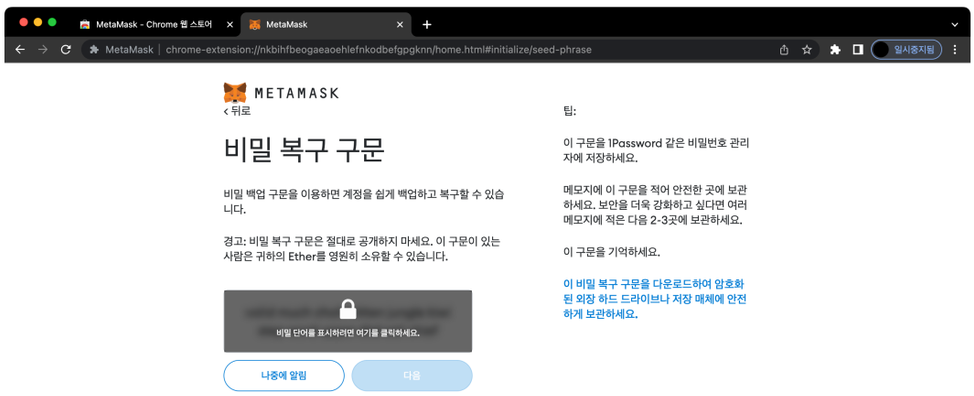


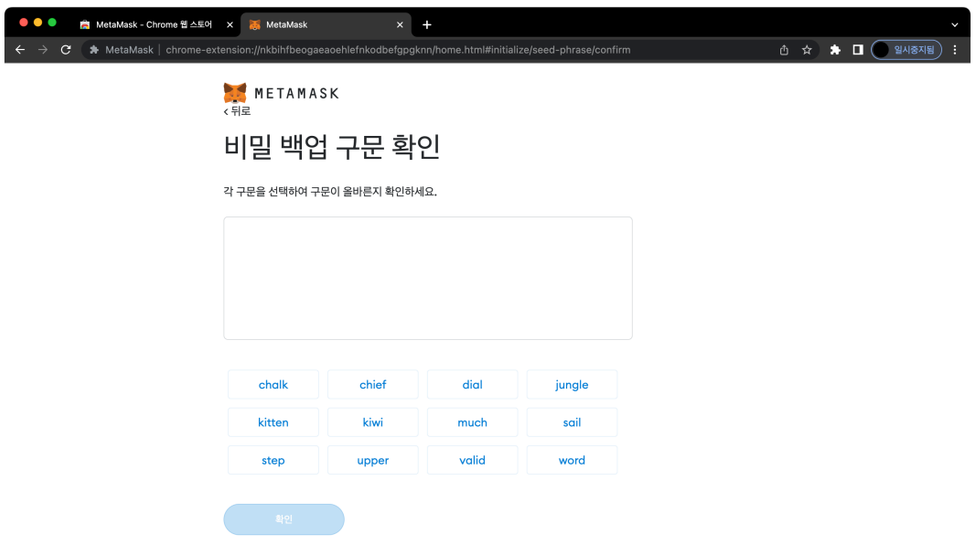
2. 지갑 생성

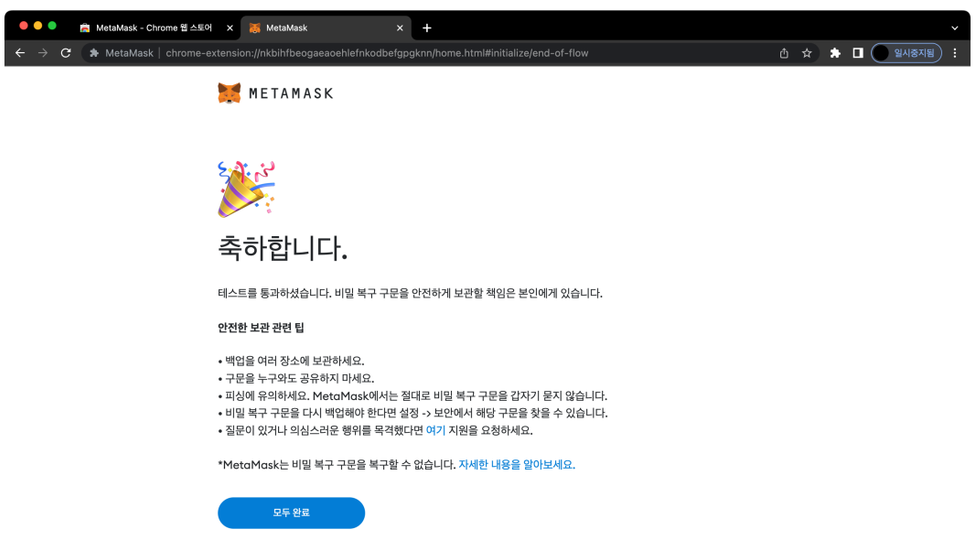

MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#initialize/select-action
일시중지됨
METAMASK
MetaMask가 처음이세요?
아니요. 이미 비밀 복구 구문이 있습니다.
비밀 복구 구문을 사용하여 기존 지갑 가져오기
지갑 가져오기
설정을 시작하죠!
이렇게 하면 새 지갑과 비밀 복구 구문이 만들어집니다
지갑 생성

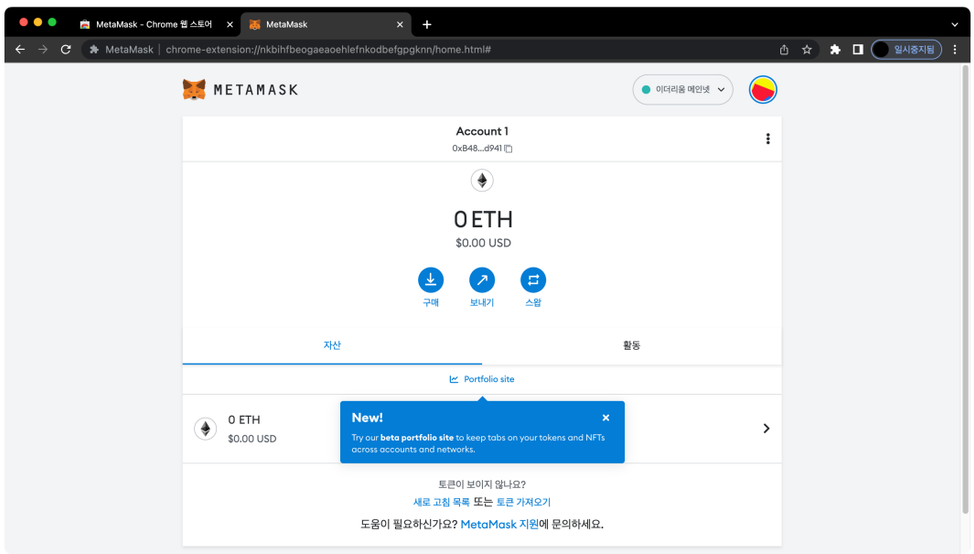

MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#initialize/create-password
일시중지됨
METAMASK
뒤로
비밀번호 만들기
새 비밀번호(8자 이상)
비밀번호 확인
이용 약관의 내용을 읽고 이에 동의합니다.
생성


MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#initialize/seed-phrase-intro
일시중지됨
METAMASK
지갑 보호하기
시작하기 전에 이 짧은 동영상을 보고 비밀 복구 구문과 지갑을 안전하게 보호하는 방법에 대해 알아보세요.
0:00 / 1:35
다음
'비밀 복구 구문'이란 무엇인가요?
복구 구문은 지갑과 자금의 '마스터 키'입니다.
비밀 복구 구문은 어떻게 저장하나요?
비밀번호 관리자에 저장
은행 금고에 보관.
대여 금고에 보관.
적어서 여러 비밀 장소에 보관하세요.
비밀 복구 구문을 공유해야 하나요?
절대로, 누구와도, 심지어 MetaMask와도 비밀 구문을 공유하면 안 됩니다!
복구 구문을 요청하는 사람은 사기를 치려는 것입니다.

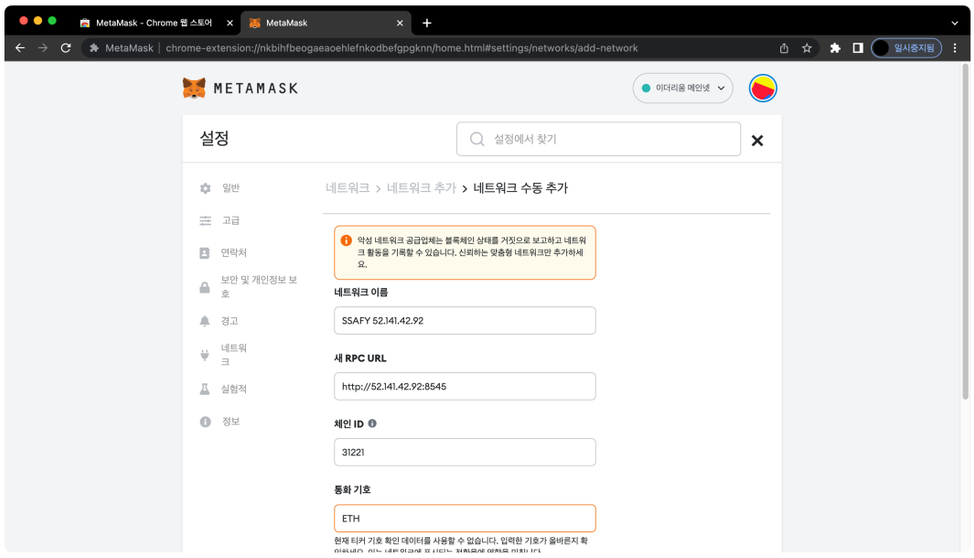

MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#initialize/seed-phrase
일시중지됨
METAMASK
뒤로
비밀 복구 구문
비밀 백업 구문을 이용하면 계정을 쉽게 백업하고 복구할 수 있습니다.
경고: 비밀 복구 구문은 절대로 공개하지 마세요. 이 구문이 있는 사람은 귀하의 Ether를 영원히 소유할 수 있습니다.
비밀 단어를 표시하려면 여기를 클릭하세요.
나중에 알림
다음
팁:
이 구문을 1Password 같은 비밀번호 관리자에 저장하세요.
메모지에 이 구문을 적어 안전한 곳에 보관하세요. 보안을 더욱 강화하고 싶다면 여러 메모지에 적은 다음 2-3곳에 보관하세요.
이 구문을 기억하세요.
이 비밀 복구 구문을 다운로드하여 암호화된 외장 하드 드라이브나 저장 매체에 안전하게 보관하세요.

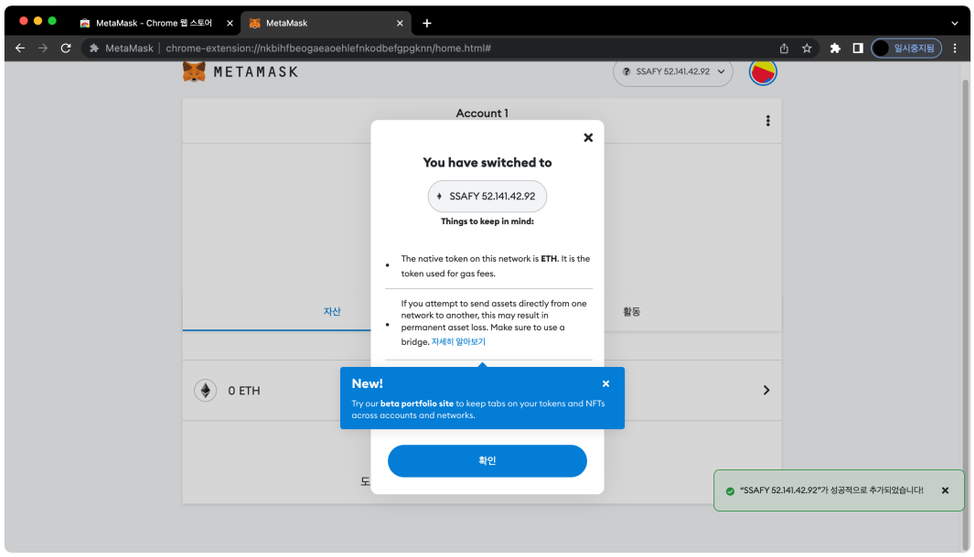

MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#initialize/seed-phrase/confirm
일시중지됨
METAMASK
뒤로
비밀 백업 구문 확인
각 구문을 선택하여 구문이 올바른지 확인하세요.
chalk
chief
dial
jungle
kitten
kiwi
much
sail
step
upper
valid
word
확인


MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#initialize/end-of-flow
일시중지됨
METAMASK
축하합니다.
테스트를 통과하셨습니다. 비밀 복구 구문을 안전하게 보관할 책임은 본인에게 있습니다.
안전한 보관 관련 팁
• 백업을 여러 장소에 보관하세요.
• 구문을 누구와도 공유하지 마세요.
• 피싱에 유의하세요. MetaMask에서는 절대로 비밀 복구 구문을 갑자기 묻지 않습니다.
• 비밀 복구 구문을 다시 백업해야 한다면 설정 -> 보안에서 해당 구문을 찾을 수 있습니다.
• 질문이 있거나 의심스러운 행위를 목격했다면 여기 지원을 요청하세요.
*MetaMask는 비밀 복구 구문을 복구할 수 없습니다. 자세한 내용을 알아보세요.
모두 완료

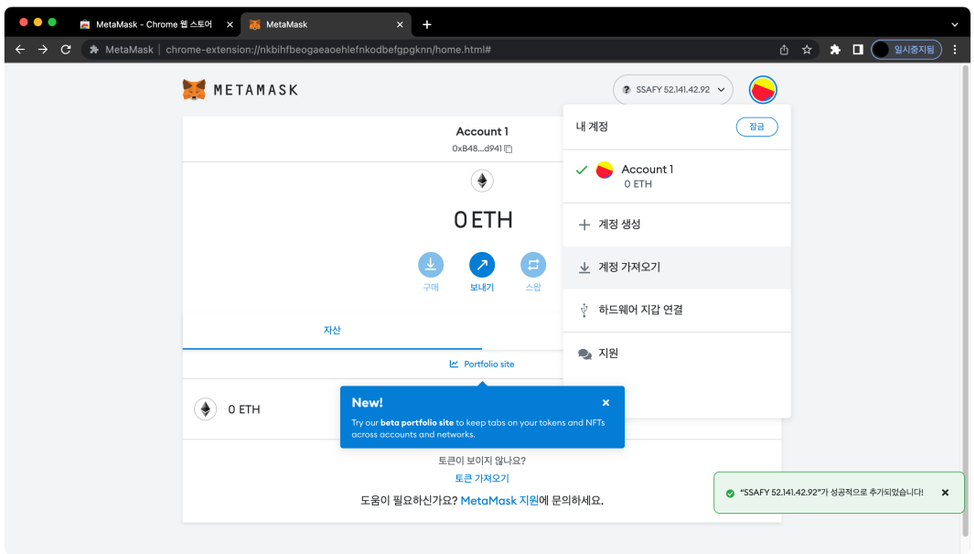


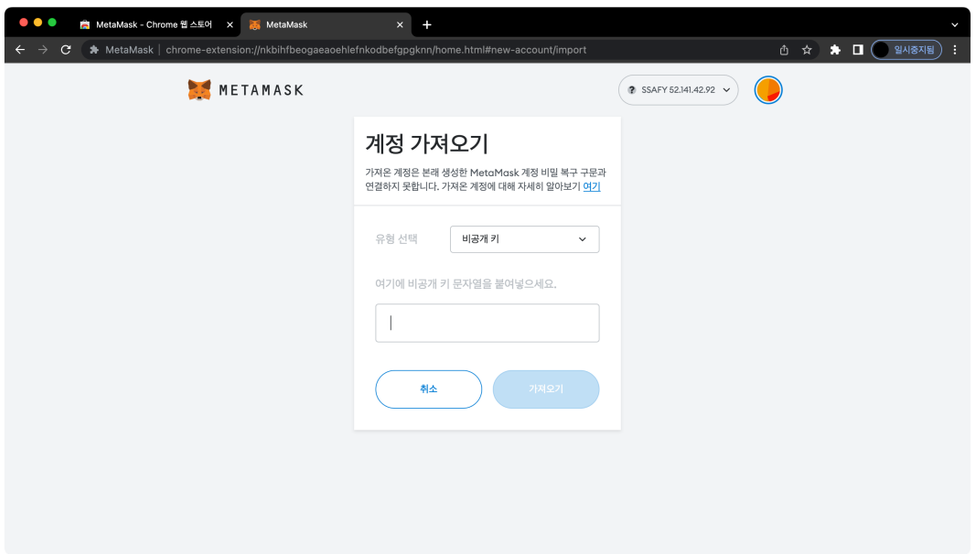
3. SSAFY 블록체인 네트워크 설정

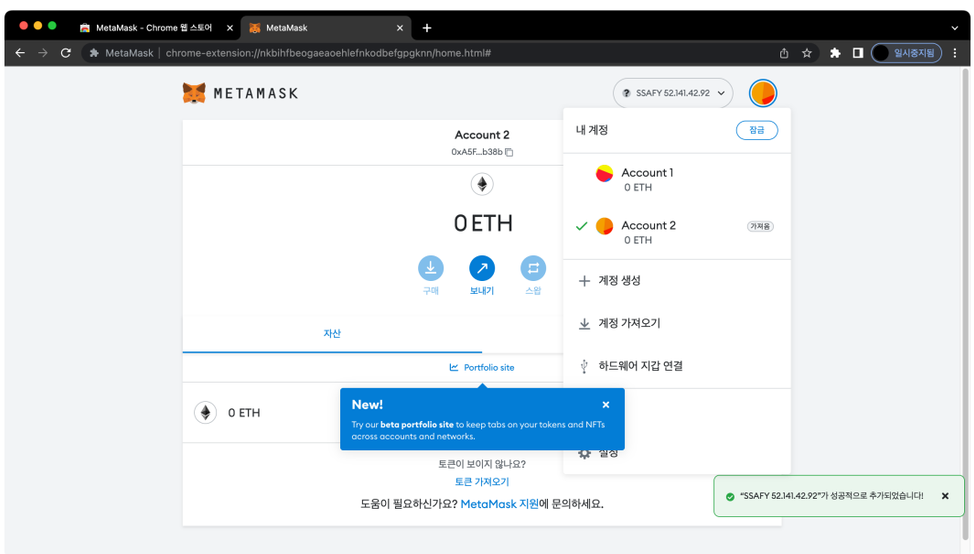




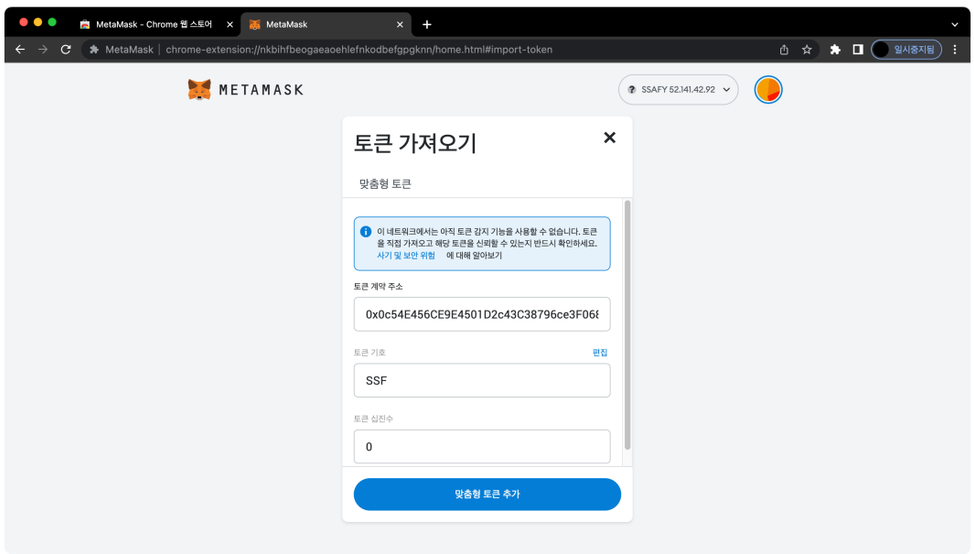
4. SSAFY wallet 계정 가져오기

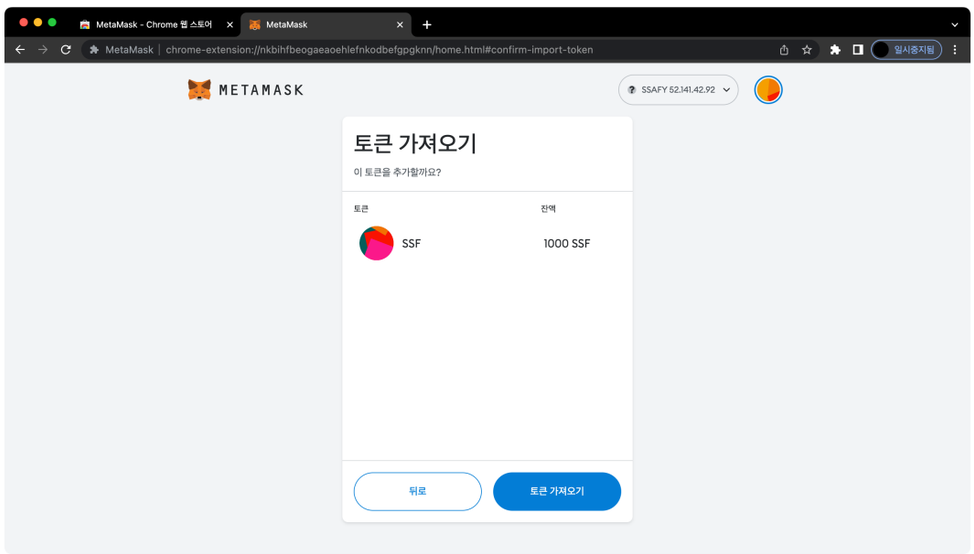

MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#
일시중지됨
METAMASK
SSAFY 52.141.42.92
Account 1
0xB48...d941
0 ETH
구매
보내기
스왑
자산
Portfolio site
0 ETH
New!
Try our beta portfolio site to keep tabs on your tokens and NFTs across accounts and networks.
토큰이 보이지 않나요?
토큰 가져오기
도움이 필요하신가요? MetaMask 지원에 문의하세요.
내 계정
잠금
Account 1
0 ETH
계정 생성
계정 가져오기
하드웨어 지갑 연결
지원
"SSAFY 52.141.42.92"가 성공적으로 추가되었습니다!

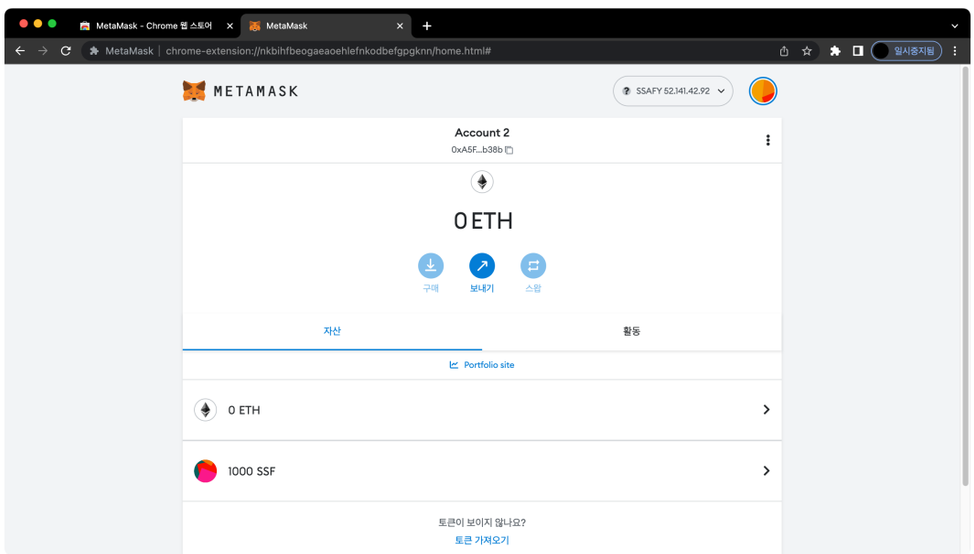

MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#new-account/import
일시중지됨
METAMASK
SSAFY 52.141.42.92
계정 가져오기
가져온 계정은 본래 생성한 MetaMask 계정 비밀 복구 구문과 연결하지 못합니다. 가져온 계정에 대해 자세히 알아보기 여기
유형 선택
비공개 키
여기에 비공개 키 문자열을 붙여넣으세요.
취소
가져오기


MetaMask - Chrome 웹 스토어
MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#
일시중지됨
METAMASK
SSAFY 52.141.42.92
Account 2
0xA5F...b38b
0 ETH
구매
보내기
스왑
자산
Portfolio site
0 ETH
New!
Try our beta portfolio site to keep tabs on your tokens and NFTs across accounts and networks.
토큰이 보이지 않나요?
토큰 가져오기
도움이 필요하신가요? MetaMask 지원에 문의하세요.
내 계정
잠금
Account 1
0 ETH
Account 2
0 ETH
가져옴
계정 생성
계정 가져오기
하드웨어 지갑 연결
"SSAFY 52.141.42.92"가 성공적으로 추가되었습니다!

5. SSF 토큰 가져오기







II. 서버

1. ubuntu 서버에 docker 설치

사전 패키지 설치

```
sudo apt-get update
sudo apt-get install -y ca-certificates \
    curl \
    software-properties-common \
    apt-transport-https \
    gnupg \
    lsb-release
```

gpg 키 다운로드

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/k
eyrings/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] http
s://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

docker 설치

```
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose
```

2. Jenkins 설치 및 설정

docker-compose.yml

```
version: '3'

services:
    jenkins:
        image: jenkins/jenkins:lts
        container_name: jenkins
        volumes:
            - /var/run/docker.sock:/var/run/docker.sock
            - /jenkins:/var/jenkins_home
        ports:
            - "9090:8080"
        privileged: true
        user: root
```

Jenkins 설치

```
sudo docker-compose up -d
```

플러그인 설치 목록

```
gitlab
  GitLab
  Generic Webhook Trigger
  Gitlab API
  GitLab Authentication
docker
```

```
  Docker
  Docker Commons
  Docker Pipeline
  Docker API
ssh
  Publish Over SSH
```

Jenkins 프로젝트 생성

```
[소스 코드 관리]
Git
  Repository URL
    https://[gitlab사용자계정]:[gitlab사용자AccessToken]@lab.ssafy.com/s07-blockchain-nft-sub
2/S07P22E206.git
  Credentials Add
    Kind: Uername with password
    Username: gitlab id
    Password: gitlab pw
    ID: Credential 구별을 위한 텍스트

[빌드 유발]
Build when a change is pushed to GitLab. GitLab webhook URL: -
  checkbox check
고급
  Secret token Generate 및 기록
```

Jenkins 내 docker 사전 패키지 설치

```
apt-get update
apt-get install -y ca-certificates \
    curl \
    software-properties-common \
    apt-transport-https \
    gnupg \
    lsb-release
```

Jenkins 내 gpg 키 다운로드

```
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrin
gs/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] http
s://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Jenkins 내 docker 설치

```
apt update
apt install docker-ce docker-ce-cli containerd.io docker-compose
```

frontend/Dockerfile

```
FROM node:16.13.2-alpine
RUN npm install -g serve
RUN mkdir /app
WORKDIR /app
```

```
RUN mkdir ./build
COPY ./build ./build
ENTRYPOINT ["serve", "-s", "build"]
```

backend/NG/Dockerfile

```
FROM openjdk:11
ARG JAR_FILE=/build/libs/*.jar
COPY ${JAR_FILE} ng_springboot.jar
ENTRYPOINT ["java", "-jar", "/ng_springboot.jar"]
```

Jenkins 프로젝트 설정(1)

```
[빌드]
ExecuteShell
  docker image prune -a --force
  mkdir -p /var/jenkins_home/images_tar

  cd /var/jenkins_home/workspace/deploy-test/backend/NG/
  docker build -t ng_springboot .
  docker save ng_springboot > /var/jenkins_home/images_tar/ng_springboot.tar

  ls -al /var/jenkins_home/images_tar
```

Jenkins SSH 연결 설정

```
Jenkins home | Jenkins 관리 | 시스템 설정
[Publish over SSH]
SSH Server 추가
  Name
    deploy-test
  Hostname
    j7e206.p.ssafy.io
  Username
    ubuntu
  고급
    Use password authentication, or use a different key
      checkbox check
    Key
      pem 파일 내용 삽입
  Test Configuration
```

certbot container 생성 및 인증서 발급

```
cd
sudo mkdir certbot
cd certbot
sudo mkdir conf www logs
sudo docker run -it --rm --name certbot -p 80:80 -v "/home/ubuntu/certbot/conf:/etc/letsen
crypt" -v "/home/ubuntu/certbot/log:/var/log/letsencrypt" -v "/home/ubuntu/certbot/www:/va
r/www/certbot" certbot/certbot certonly

* standalone, agree, no, <domain_name> 으로 작성
```

Jenkins 프로젝트 설정(2)

```
[빌드 후 조치]
SSH Server Name
```

```
  deploy-test 선택
Transfers
  Source files
    /README.md
  Exec command
    ls -al /jenkins/images_tar

    sudo docker load < /jenkins/images_tar/ng_springboot.tar

    if (sudo docker ps | grep "ng_springboot"); then sudo docker stop ng_springboot; sudo
 docker rm ng_springboot; fi

    sudo docker run -it -d --rm -p 8080:8080 --name ng_springboot ng_springboot
```

frontend/deploy_conf/nginx.conf

```
server {
        listen 80;
        listen [::]:80;
        server_name j7e206.p.ssafy.io;

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        location / {
                return 301 https://$server_name$request_uri;
        }
}

server {
        listen 443 ssl;
        listen [::]:443 ssl;
        server_name j7e206.p.ssafy.io;

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ssl_certificate /etc/letsencrypt/live/j7e206.p.ssafy.io/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/j7e206.p.ssafy.io/privkey.pem;

        root /usr/share/nginx/html;
        index index.html;

        location / {
                proxy_pass http://localhost:3000;
        }

        location /api/v1 {
                proxy_pass http://localhost:8080/api/v1;
        }
}
```

3. Webhooks

URL & Secret token 작성

Trigger: Push events, Merge request rvents

4. MySQL 설치

```
sudo docker pull mysql
sudo docker run --name mysql -e MYSQL_ROOT_PASSWORD=[PASSWORD] -d -p 13306:3306 mysql:late
st
sudo docker exec -it mysql bash
  mysql -u root -p
```

```
    show databases;
    create user [USER]@'%' identified by [PASSWORD];
    grant all privileges on *.* to 'root'@'%';
    grant all privileges on *.* to [USER]@'%';
    flush privileges;
```

5. IPFS 설치

```
docker pull ipfs/kubo
export ipfs_staging=/home/ubuntu/ipfs/ipfs-staging
export ipfs_data=/home/ubuntu/ipfs/ipfs-data
sudo docker run -d --name ipfs_host -v $ipfs_staging:/export -v $ipfs_data:/data/ipfs -p 4
001:4001 -p 4001:4001/udp -p 8080:8080 -p 5001:5001 ipfs/kubo:latest
```

6. 서버 시간 변경

```
rm /etc/localtime
ln -s /usr/share/zoninfo/Asia/Seoul /etc/localtime
date
```