

비트코인 사용 가이드

개인 지갑 · 결제 · 풀 노드 · 라이트닝 노드 · 노스터 · 홈 채굴 가이드



2판
2025. 9. 1. 기준

필레몬 지음
HYPE 감수
익스투스 출판

비트코인 사용 가이드

개인 지갑 · 결제 · 풀 노드 · 라이트닝 노드 · 노스터 · 홈 채굴 가이드

비트코인 사용 가이드: 개인 지갑, 결제, 폴 노트, 라이트닝 노트, 노스터, 홈 채굴 가이드, 제2판
저작권 없음 © ① 필레몬, 2025

필레몬은 2025년 『비트코인 사용 가이드』 제2판을 CC0 1.0 Universal에 따라 퍼블릭 도메인에 헌정합니다.

| 필레몬의 퍼블릭 도메인 선언 |

지식과 문화는 인류 모두의 자산입니다. 정보는 희소하지 않으며, 따라서 희소한 재화에 적용하는 재산권이 정보에는 적용될 수 없습니다. 정보에 대한 독점적 재산권 부여는 오히려 정보를 정당하게 취득한 소비자의 물리적 재산권을 침해합니다.

소비자 각각의 재산권 보호가 훨씬 중요하므로 저자는 본 저작물(비트코인 사용 가이드 제2판)에 대한 모든 저작재산권을 최대한도로 포기합니다. 이에 따라 소비자는 일반적으로 저작재산권에 따라 제한되는 복제, 전시, 배포, 전송, 수정, 상업적 이용을 자유롭게 할 수 있습니다. 본 선언은 크리에이티브 커먼즈 CC0 1.0 Universal에 따라 이루어집니다.

선언문 해시값: 45046C4A0858AD664122B30974353D46580D7F107A68CF761724A4E30170BA0D

비트코인 메시지 서명: H9gTJU0T1JIYQ6VxsDd89A0TgiE1by7bdK4EDXf7arjFDg3gqP/wdqoRcL
SiwXwp/rNNSen3t/pxK2AFDDxmKjE=

서명 검증을 위한 저자의 공개된 비트코인 주소는 keybase.io/philemon21에서 확인할 수 있습니다.

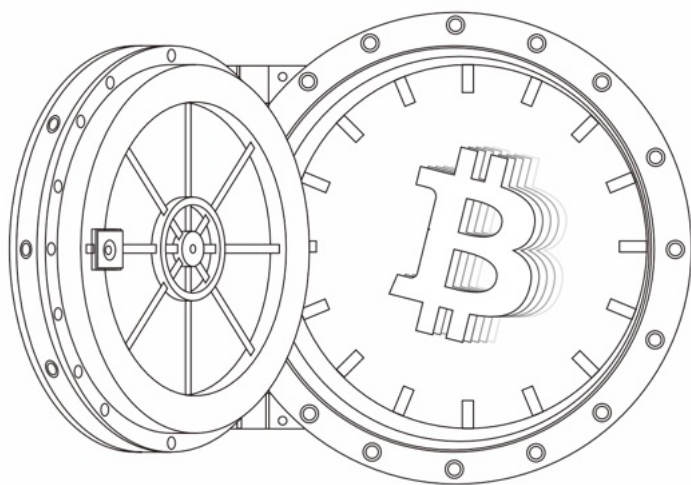
저자의 저작재산권 포기가 상표권, 저작권격권, 퍼블리시티권, 프라이버시권, 저자에게 귀속되지 않는 제3자의 권리 등의 포기를 의미하지 않습니다. 이를 무시하여 발생하는 문제에 대한 책임은 전적으로 사용자에게 있습니다.

본 전자 파일은 배포용으로 제작되었습니다. 본 전자 파일에 표시된 출판사명이나 로고는 원전 출처 확인을 돕기 위한 것으로, 전자 파일의 배포는 해당 출판사와 무관하게 진행되었습니다.

파일의 무결성을 확인하기 위해서는 해시값, 저자 서명이 올바른지 검증해 보십시오.

비트코인 사용 가이드

개인 지갑 · 결제 · 풀 노드 · 라이트닝 노드 · 노스터 · 홈 채굴 가이드



2판

2025. 9. 1. 기준

필레몬 지음

HYPE 감수

익스투스 출판

| 목차 |

서문. 당신의 돈을 통제하라	7
감수의 글	12

1부. 셀프 커스터디 가이드

■ 비트코인 지갑 사용을 위한 지식	26
셀프 커스터디 · 26 비트코인의 소유권과 셀프 커스터디의 필요성, 책임 · 27 BTC와 sats 단위 · 28 잔고 모델과 UTXO 모델 · 28 에어-갭 지갑과 워치-온리 지갑 · 31 PSBT · 33 개인키와 주소 · 34 니모닉과 개인키, 주소 · 35 확장 공개키 · 37 주사위를 굴릴 때 주의할 점 · 38 거래 데이터(트랜잭션) · 39 UTXO에 대한 비유 · 40 거래 데이터와 블록 · 41 수수료 · 43 멤폴 웹사이트 · 43 UTXO 정리 · 45 주소 재사용 주의 · 46 파생 경로 · 46 갭 리밋과 주소 순차 사용 · 47 패스프레이즈 · 48 니모닉 체크섬과 MFP · 49 5달러 렌치 공격과 수량 발설 주의 · 52 KYC (고객 확인) 제도와 트래블 룰 · 54 라이트닝 네트워크와 인보이스, 라이트닝 주소 · 55	
■ 키스톤 지갑	57
필수 준비물 · 57 권장 준비물 · 59 업데이트를 위한 마이크로SD카드 준비 · 60 기기 검증 · 63 펌웨어 2.0.4 검증 및 업그레이드 · 66 최신 펌웨어 업데이트 · 70 지갑 생성 · 77 키스톤 사전 설정 · 85 블루월렛에 확장 공개키 내보내 워치-온리 지갑 만들기 · 88 년척에 확장 공개키 내보내 워치-온리 지갑 만들기 · 94 코코넛 월렛에 확장 공개키 내보내 워치-온리 지갑 만들기 · 101 블루월렛으로 서명 연습 · 105 년척으로 서명 연습 · 111 코코넛 월렛으로 서명 연습 · 116 복구 연습 · 119	
■ 시드사이너 지갑	125
필수 준비물 · 125 권장 준비물 · 128 이미지 파일 다운로드 · 129 소프트웨어 번조 여부 확인(윈도우OS) · 130 소프트웨어 번조 여부 확인(맥OS) · 141 부팅 마이크로SD카드 만들기 · 150 발레나에 처로 시드사이너 이미지 파일 플래싱이 안 될 경우 해결 방법 · 156 무선 통신 모듈 제거(라즈베리파이 제로 W 보드만 해당) · 164 시드사이너 조립 · 166 시드사이너 케이스까지 조립 · 174 지갑 생성 · 182 시드 QR 제작 · 188 니모닉 입력하기 or 시드 QR 스캔하기 · 194 블루월렛에 확장 공개키 내보내 워치-온리 지갑 만들기 · 196 년척에 확장 공개키 내보내 워치-온리 지갑 만들기 · 204 코코넛 월렛에 확장 공개키 내보내 워치-온리 지갑 만들기 · 212 블루월렛으로 서명 연습 · 218 년척으로 서명 연습 · 224 코코넛 월렛으로 서명 연습 · 231 시드사이너를 게임기로 만들기 · 236	

<ul style="list-style-type: none"> ■ 공기계 지갑 247 <ul style="list-style-type: none"> 스마트폰 공기계를 콜드월렛으로 사용해 지갑 생성하기 · 247 블루월렛에 확장 공개키 내보내 워치-온리 지갑 만들기 · 256 년척에 확장 공개키 내보내 워치-온리 지갑 만들기 · 262 블루월렛으로 서명 연습 · 269 년척으로 서명 연습 · 274 공기계 블루월렛에서 간접 복구 테스트 · 279 ■ 거래소에서 지갑으로 비트코인 옮기기 282 <ul style="list-style-type: none"> 거래소에서 비트코인으로 환전하는 방법 · 282 빗썸 가입 및 KYC 인증 · 285 바이낸스 가입 및 KYC 인증 · 298 빗썸에서 원화 입금하고 테더 구매하기 · 304 빗썸에서 바이낸스로 테더 보내기 · 307 바이낸스에서 테더로 비트코인 구매하기 1: Convert 사용 · 310 바이낸스에서 테더로 비트코인 구매하기 2: 시장가 매수 · 312 바이낸스에서 온-체인을 통해 바로 개인 지갑으로 전송하기 · 316 바이낸스에서 라이트닝 네트워크와 볼츠 스와프 서비스를 통해 개인 지갑으로 전송하기 · 317 ■ 지갑에서 거래소로 비트코인 옮겨 원화 출금하기 322 <ul style="list-style-type: none"> 전송 경로 · 322 개인 지갑에서 해외 거래소로 전송 · 323 해외 거래소에서 국내 거래소로 전송 · 324 국내 거래소에서 원화 환전 후 은행 계좌로 출금 · 330 ■ 스패로우 지갑 사용 방법 및 UTXO 정리하기 334 <ul style="list-style-type: none"> 준비물 · 334 스패로우 설치 · 335 풀 노드 서버 설정 · 337 워치-온리 연동하기 · 340 UTXO 정리 · 345 앨리스의 UTXO 정리 · 346 스패로우에서 UTXO 정리하기 · 348 년척에서 UTXO 정리하기 · 368 ■ 수수료율 설정, RBF와 CFPF 376 <ul style="list-style-type: none"> 온-체인 수수료 · 376 멤풀 웹사이트 보는 방법 · 379 적정 수수료율 설정하기 · 382 RBF · 386 CFPF · 401 ■ 패스프레이즈 418 <ul style="list-style-type: none"> 패스프레이즈와 주의 사항 · 418 키스톤에서 패스프레이즈 설정하기 · 420 시드사이너에서 패스프레이즈 설정하기 · 424 공기계 콜드월렛에서 패스프레이즈 설정하기 · 425 서명 기기에서 서명이 안 될 때 · 430 ■ 멀티시그 431 <ul style="list-style-type: none"> 멀티시그(다중서명) · 431 블루월렛에서 멀티시그 지갑 생성 · 434 년척에서 멀티시그 지갑 생성 · 446 스패로우 지갑에서 멀티시그 지갑 생성 · 461 블루월렛 멀티시그 지갑에서 서명하기 · 478 년척 멀티시그 지갑에서 서명하기 · 487 스패로우 멀티시그 지갑에서 서명하기 · 495 멀티시그 워치-온리 지갑 삭제 후 복구하기 · 505

2부. 비트코인 스탠다드 가이드

■ 비트코인은 돈이다	524
비트코인은 돈이다 · 524 교환 매개 · 525 구매력 보존 · 529 회계 단위 · 534 비트코인의 레이어 구조 · 536 라이트닝 네트워크를 사용하는 방법 · 539 비트코인 결제 체험이 중요한 이유 · 539	
■ 라이트닝 수탁 지갑 이용 방법	541
라이트닝 수탁 지갑 설치 · 541 커스텀 라이트닝 주소 발급 · 544 온-체인으로 라이트닝 수탁 지갑에 비트코인 입금하기 · 549 라이트닝 수탁 지갑에서 온-체인으로 비트코인 출금하기 · 553 원화 환전을 위해 라이트닝 수탁 지갑에서 해외 거래소로 비트코인 송금하기 · 556	
■ 오프라인 매장에서 라이트닝 결제하기	561
비트코인으로 커피 사 마시기 · 561	
■ 온라인 매장에서 라이트닝 결제하기	565
비트코인으로 물건 구매하고 택배 받기 · 565	
■ 1분 만에 비트코인 결제 매장 되기	571
매장에서 라이트닝 결제받는 방법 · 571	

3부. 풀 노드 운영 가이드

■ 풀 노드 운영을 위한 지식	574
풀 노드와 풀 노드 운영의 중요성 · 574 풀 노드가 수행하는 검증 작업 · 575 풀 노드가 보관하는 데이터 · 576 가지치기 풀 노드 · 579 비트코인 클라이언트: 비트코인 코어와 노츠 · 579 초기 블록 다운로드(IBD) · 580 아웃바운드 연결과 인바운드 연결, 인바운드 허용 노드 · 581 일렉트럼 서버 · 584 RPC 인터페이스 · 585 진정한 금융 주권의 실천 · 586	
■ 엠브렐 홈 구매 및 세팅	587
풀 노드 구축 방법 · 587 엠브렐 홈 구매 방법 · 588 엠브렐 홈 세팅 · 596	
■ 미니 PC 조립하고 엠브렐OS 설치하기	598
미니 PC 준비물 · 598 미니 PC 조립하기 · 603 바이오스에서 램 설정하기 · 609 OS 설치용 USB 만들기 · 611 엠브렐OS 설치하기 · 618	
■ 라즈베리파이5 조립하고 엠브렐OS 설치하기	623
라즈베리파이5 준비물 · 623 SSD에 엠브렐OS 설치하기 · 630 라즈베리파이5 조립 · 639 부팅이 안 될 경우 · 650 케이스 조립 · 660	

<ul style="list-style-type: none"> <ul style="list-style-type: none"> 남는 노트북에 엠브렐OS 설치하기 · 664 OS 설치용 USB 만들기 · 665 노트북에 엠브렐OS 설치하기 · 673 노트북에서 엠브렐 화면 띄우기 · 677 	664
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 엠브렐 설정 및 업데이트 · 685 비트코인 노드(코어) 또는 노츠 설치 · 690 가지치기(프루닝) 설정 · 694 노츠의 사용자 정책 설정 · 696 	685
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 테일스케일 설치 및 연결 · 704 	704
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 일렉트스(Electrs) 설치 · 710 블루월렛과 자신의 풀 노드 연결하기 · 712 년척과 자신의 풀 노드 연결하기 · 714 코코넛 월렛과 자신의 풀 노드 연결하기 · 716 스페로우와 자신의 풀 노드 연결하기 · 718 토르를 이용해 자신의 풀 노드와 워치-온리 지갑 연결하기 · 722 블루월렛에서 토르를 이용해 워치-온리 지갑 연결하기 · 728 년척에서 토르를 이용해 워치-온리 지갑 연결하기 · 730 	710
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 멤폴 앱 연결하기 · 733 RPC 익스플로러 사용하기 · 734 터미널에서 RPC 명령어 사용하기 · 745 	733
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 자기 노드가 도달 가능한 노드인지 확인해 보기 · 753 익명 네트워크에서 도달 가능한 노드 되기 · 755 클리어넷에서 도달 가능한 노드 되기 · 757 TP링크 공유기: DHCP 서버 설정, 포트 포워딩 · 758 IP타임 공유기: DHCP 서버 설정, 포트 포워딩 · 763 인터넷 서비스 업체의 공유기를 사용하는 경우 · 768 엠브렐 인바운드 연결 허용 및 방화벽 해제 · 770 도달 가능한 노드가 되었는지 확인하기 · 774 	752
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 윈도우OS에 비트코인 코어 설치하고 동기화하기 · 775 윈도우OS에 비트코인 노츠 설치하고 동기화하기 · 784 같은 기기에서 스페로우 지갑 연결하기 · 793 	775
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 맥OS에 비트코인 코어 설치하고 동기화하기 · 800 맥OS에 비트코인 노츠 설치하고 동기화하기 · 810 같은 기기에서 스페로우 지갑 연결하기 · 822 	800
<ul style="list-style-type: none"> <ul style="list-style-type: none"> 코어, 노츠가 설치된 기기의 로컬 IP 주소 알아내기 · 828 bitcoin.conf 파일 설정하기 · 830 윈도우OS에 코어, 노츠가 설치되어 있는 경우 방화벽 해제 · 833 맥OS에 코어, 노츠가 설치되어 있는 경우 방화벽 해제 · 836 로컬 네트워크에서 스페로우 지갑 연결하기 · 837 	828

4부. 라이트닝 노드 운영 가이드

- **라이트닝 노드 운영을 위한 지식** 842
라이트닝 네트워크 · 842 | 라이트닝 채널의 원리 · 843 | 인바운드 유동성과 아웃바운드 유동성 · 847 | 다중 경로 결제 · 852 | HTLC · 854 | 협력적 종료와 비협력적 종료, CSV, 페널티 · 855 | 라이트닝 노드의 유형 · 857
- **라이트닝 노드 설치, 복구, RTL 설치** 859
라이트닝 노드(LND) 설치 · 859 | 라이트닝 노드 제거 후 복구 · 863 | CLN을 설치하는 경우 · 866 | RTL 설치 · 873
- **일상적인 지갑 목적으로 라이트닝 노드 운영하기** 876
라이트닝 노드 온-체인 지갑에 자금 전송 · 878 | 라이트닝 노드 검색 및 피어 추가, 채널 개설 · 880 | 두 번째 채널 개설 · 889 | 세 번째 채널 개설 · 892 | 네 번째 채널 개설 · 894 | 다섯 번째 채널 개설 · 896 | 인바운드 유동성 확보 · 898 | 여섯 번째 채널 개설 · 908 | 일곱 번째 채널 개설 · 910 | 여덟 번째 채널 개설 · 912 | 채널 추천 목록 · 914
- **외부에서 라이트닝 노드 사용하기** 917
토르를 통해 라이트닝 노드와 제우스 앱 연동하기 · 917 | 테일스케일을 통해 라이트닝 노드와 제우스 앱 연동하기 · 920 | 제우스 앱 사용 방법 · 925
- **라이트닝 노드 설정하기** 931
라이트닝 노드 네트워크 설정과 개인 맞춤 설정 · 931 | 채널 설정 · 935 | 라우팅 설정 · 938 | 위치타워 설정 · 946
- **채널 관리 가이드** 953
라우팅 수수료 부과 원리 · 953 | 특정 노드가 유동성을 다 흡수할 때 · 955 | 채널별 라우팅 수수료, 최대/최소 HTLC 금액 조정하기 · 957 | 수수료 조정보다는 적절한 노드 찾고 채널 맺기 · 959 | 채널 닫기 · 961 | 라이트닝 노드 SCB 파일 백업과 복구 · 963
- **라이트닝 주소 설정, 자신의 노드 알리기** 969
알비 허브로 라이트닝 주소 연결하기 · 969 | 알비 유료 결제하고 커스텀 라이트닝 주소 만들기 · 978 | 앱 보스에서 노드 정보 입력하기 · 984
- **라이트닝 노드로 온라인 비트코인 결제 매장 구축하기** 990
워드프레스에 BTC Pay Server, 우커머스 플러그인 설치 · 990 | 우커머스 기본 설정 및 테마 선택 · 992 | 우커머스 상품 올리기 · 996 | 엠브렐에서 BTC Pay Server 다운로드하고 설정하기 · 1001 | 클라우드플레이어 회원가입 · 1009 | 클라우드플레이어 터널 연결 · 1016 | 도메인 연결 · 1020 | SSL 적용 · 1025 | 워드프레스 우커머스와 자신의 BTC Pay Server 연결 · 1029 | 기타 설정 · 1033 | 법률 문제, 세금 문제 · 1034

5부. 노스터 가이드

■ 노스터 사용을 위한 지식	1038
기존 소셜 미디어의 문제점과 노스터 · 1038 노스터 클라이언트 · 1041 노스터 릴레이와 이벤트, 작동 원리 · 1042 노스터 구현 제안(NIP) · 1044 개인키(nsec)와 공개키(npub), 노스터 주소 · 1045 노스터의 DM과 중단간 암호화 · 1046 잼(Zaps)과 NWC (노스터 지갑 연결) · 1047 노스터의 단점과 광고 필터, 리스트 구독 · 1049	
■ 프라이멀 사용 방법	1051
프라이멀 앱 설치 및 개인키-공개키 쌍 생성 · 1052 다른 사람들에게 npub 알려주기 · 1056 팔로우 추가 · 1057 잼을 위한 지갑 추가 · 1058	
■ 다무스 사용 방법	1062
다무스 앱 설치 및 개인키-공개키 쌍 생성 · 1062 다른 사람들에게 npub 알려주기 · 1065 팔로우 추가 · 1066 잼을 받기 위한 라이트닝 주소 연결 · 1068 다른 사람에게 잼 보내기 · 1070	
■ 피닉스 사용 방법	1073
피닉스에서 개인키-공개키 쌍 생성 · 1073 다른 사람들에게 npub 알려주기 · 1079 팔로우 추가 · 1081 잼을 받기 위한 라이트닝 주소 연결 · 1083	
■ 노스터 서명 확장 프로그램	1084
서명 확장 프로그램을 쓰는 이유 · 1084 크롬에서 알비 익스텐션 사용 방법 · 1085 알비에서 노스터 주소 사용하기 · 1092 웹 클라이언트에서 알비 익스텐션으로 로그인하기 · 1094	
■ 노스터에서 기사, 칼럼 등의 긴 글 쓰기	1098
하블라에서 긴 글 쓰기 · 1098 마크다운 문법 간략히 알아보기 · 1103	
■ 엠프렐에서 노스터 릴레이 서버 운영하고 연결하기	1112
프라이빗 릴레이 서버 운영하기 · 1112 로컬 네트워크에서 자신의 릴레이 서버에 연결하기 · 1114 테일스케일을 이용해 원격으로 자신의 릴레이 서버에 연결하기 · 1117 도메인을 연결해 퍼블릭 릴레이 서버로 만들기 · 1119	
■ NWC를 이용해 자신의 라이트닝 노드에서 잼 보내기	1124
알비 허브를 통해 NWC 지갑 생성하기 · 1124 다무스에서 NWC 지갑 연결하기 · 1126 피닉스에서 NWC 지갑 연결하기 · 1129	
■ NWC를 이용해 제우스에서 라이트닝 주소 발급하기	1132
제우스에서 라이트닝 주소 발급하기 · 1132	

6부. 홈 채굴 가이드

■ 홈 채굴을 위한 지식	1136
비트코인 채굴 · 1136 채굴 방식의 분류 · 1139 채산성 계산하기 · 1141 채굴 풀 보상 방식 · 1144 스트라텀 프로토콜 · 1151 채굴 풀의 한계 · 1153 다팀과 채굴 주권 · 1155 홈 채굴의 의미 · 1157	
■ 비트엑스 감마 601로 솔로 채굴하기, 채굴 풀 참여하기	1159
준비물 · 1159 비트엑스 스탠드 조립 및 전원 연결 · 1161 비트엑스 네트워크 연결 · 1162 비트엑스 펌웨어 업데이트 · 1164 솔로 채굴 설정하기(ckpool) · 1167 채굴 풀 참여하기(브레인스 풀) · 1170 라이트닝 네트워크로 보상 받기(브레인스 풀) · 1176	
■ 아발론 나노 3로 솔로 채굴하기, 채굴 풀 참여하기	1179
준비물 · 1179 아발론 나노 3 전원 연결 · 1181 아발론 나노 3 설정하기 · 1182 솔로 채굴 설정하기 (ckpool) · 1186 채굴 풀 참여하기(브레인스 풀) · 1190 라이트닝 네트워크로 보상 받기(브레인스 풀) · 1197	
■ 다팀으로 풀 노드와 채굴기 연결하기	1200
비트코인 노츠 설치 · 1200 다팀 설치 · 1202 다팀에서 솔로 채굴 설정하기 · 1204 비트엑스를 다팀 에 연결하기 · 1208 아발론 나노 3를 다팀에 연결하기 · 1210 채굴이 잘 되는지 확인하기 · 1211 다 팀을 이용하여 채굴 풀(오션 풀) 참여하기 · 1213 블루월렛에서 생성된 지갑 주소 사용 · 1214 라이트 닝 노드에서 생성된 온-체인 주소 사용 · 1216 다팀 설정하기 · 1216 비트엑스를 다팀에 연결하기 · 1220 아발론 나노 3를 다팀에 연결하기 · 1222 채굴이 잘 되는지 확인하기 · 1223 라이트닝 지갑으 로 채굴 보상 받기 · 1228 코어 라이트닝(CLN)으로 Offer 생성하기 · 1231 메시지 서명하기 · 1236 마무리하며 · 1246	

부록

■ 부록 1. 기기별 니모닉 생성 알고리즘	1248
니모닉 생성 알고리즘 검증 · 1248 키스톤 3 프로 기기의 니모닉 생성 알고리즘 · 1249 시드사이너 기 기의 니모닉 생성 알고리즘 · 1251 블루월렛에서의 니모닉 생성 알고리즘 · 1253	
■ 부록 2. 니모닉 복구 방법 및 니모닉 목록	1256
BIP-39 목록 설명 · 1256 니모닉 복구 전 주의 사항 · 1256 ① 영단어 4자리로 백업되어 있는 경우 · 1258 ② 이진법(비트)으로 백업되어 있는 경우 · 1260 ③ 영단어 4자리 순서로 백업되어 있는 경우 · 1263 BIP-39 니모닉 목록 · 1267	

비트코인 사용 가이드

3. 풀 노드 운영 가이드

3. 풀 노드 운영 가이드

| 풀 노드 운영을 위한 지식

풀 노드와 풀 노드 운영의 중요성

비트코인 네트워크에서 풀 노드는 네트워크의 블록과 거래가 합의 규칙에 맞는지 독립적으로 검증하고 전체 블록 데이터를 스스로 저장하는 노드를 뜻한다. 따라서 풀 노드 운영자는 중앙 서버나 신뢰받는 제3자에 의존하지 않고, 스스로 진실을 판단하는 존재다.

‘블록체인 트릴레마’는 어떤 기술이 분산화, 보안, 확장성 3가지 모두를 챙길 수는 없다는 것을 뜻한다. 그런데 비트코인은 3가지 속성 모두를 챙겼다. 이 중에 건전한 돈이 되기 위해서 가장 중요한 것은 분산화 속성이다.

풀 노드는 비트코인 네트워크의 분산화 속성을 강화한다. 모든 풀 노드들은 동일한 권력을 갖고 독립적으로 거래와 블록이 합의 규칙에 맞는지 검증한다. 풀 노드들 덕분에 비트코인 네트워크는 분산된 네트워크 구조를 갖게 되고, 비트코인에 불변성이 부여된다. 비트코인의 공급량이 약 2,100만 개로 고정되어 있다는 규칙이나 난이도 조정 메커니즘, 21만 개 블록마다 수수료 인센티브를 제외한 채굴 보상이 절반으로 줄어든다는 규칙 등은 수많은 풀 노드들이 있기 때문에 영원히 변할

수 없다. 규칙에 맞지 않는 거래나 블록은 풀 노드가 전파를 하지 않고 버리기 때문에 네트워크에서 바로 거부된다. 즉, 풀 노드들은 비트코인 합의 규칙의 수호자들이다.



풀 노드를 운영하는 이유에는 비트코인의 분산화 속성을 지키기 위해 네트워크에 기여하는 것도 있지만 개인적인 이유도 있다(그리고 개인적인 이유가 더 중요하다). 풀 노드를 운영한다는 것은 외부 기관이나 정부, 은행에 전혀 의존하지 않고 스스로 자산을 검증하고 자신의 거래를 네트워크에 전파하는 주체가 되는 일이기 때문이다. 따라서 신뢰 지점을 최소화하기 위해서는 풀 노드 운영이 필수적이다.

풀 노드가 수행하는 검증 작업

풀 노드는 거래와 블록을 몇 가지 기준에 따라 철저히 검증한다. 거래를 검증할 때는 다음 기준에 의해 검증한다. 거래 데이터가 문법에 맞는지, 입력값이나 출력값이 비어 있지는 않은지, 거래 데이터 크기(무게)가 블록 크기(무게) 제한인 1 MvB (4 MwU)를 초과하진 않는지, 출력값이 2,100만 BTC 이하인지, 절대 잠금 시간이나 상대 잠금 시간 규칙이 있다면 그 규칙을 충족하는지, 입력으로 코인베이스 보상을 쓴다면 채굴된 시점으로부터 100블록이 지났는지(COINBASE_MATURITY 규칙), 거래에 담긴 서명 작업 수(SIGOPS, OP 명령어 개수)가 서명 작업

한도(거래 가상 크기/50)보다 작은지, 거래의 입력에서 사용하려는 이전 거래 출력이 블록이나 댐풀에 존재하는지(이중지불 방지), 출력 금액의 총합이 입력 금액의 총합 이하인지, 입력의 해제 스크립트(서명이나 세그윗)에 이전 거래의 잠금 스크립트를 연결했을 때 제대로 검증이 되는지 등을 검증한다.

블록을 검증할 때는 다음 기준에 의해 검증한다. 블록 데이터가 문법에 맞는지, 블록 헤더의 해시값이 목뿔값보다 작은지(작업증명이 됐는지), 블록 헤더의 타임스탬프 값이 중간 과거 시각 규칙과 미래 블록 시각 규칙을 만족하는지, 블록 무게가 4 Mwu를 초과하진 않는지, 첫 번째 거래가 코인베이스 거래인지, 그리고 블록 내에 있는 모든 거래가 거래 검증 규칙에 맞는지 등을 검증한다.

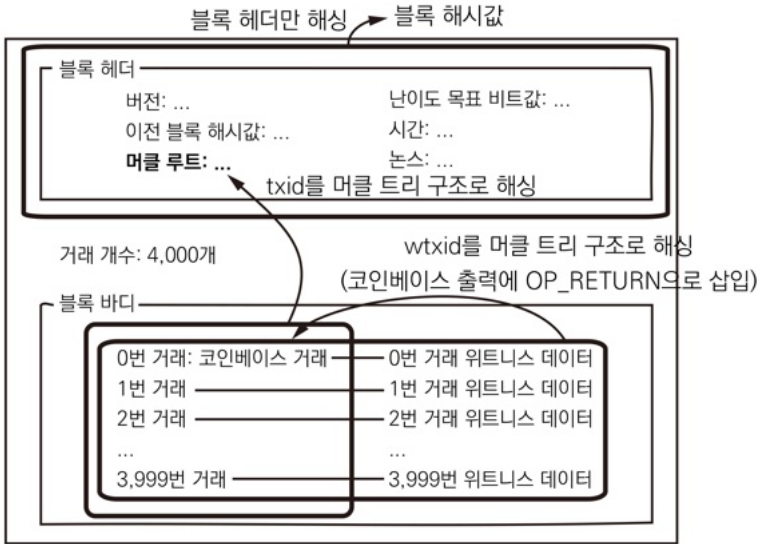
물론 이러한 검증은 컴퓨터가 하는 것이므로 매우 빠르게 일어난다. 만약 거래나 블록 데이터에서 검증 과정에서 단 하나라도 맞지 않는 것이 발견된다면 그 거래나 블록은 네트워크에서 거부된다. 이 철저함 덕분에 비트코인 네트워크는 견고하게 유지된다. 실제로 2017년 1월 30일, 한 채굴자가 블록 크기 제한인 1MB 크기를 넘는 블록을 만들자 풀 노드들은 당연히 이를 유효하지 않은 블록으로 판정했고, 블록은 순식간에 네트워크에서 거부되었다.

풀 노드가 보관하는 데이터

풀 노드가 보관하는 데이터는 여러 가지가 있지만 주요 데이터베이스는 블록 데이터, UTXO 세트, 댐풀 3가지다.

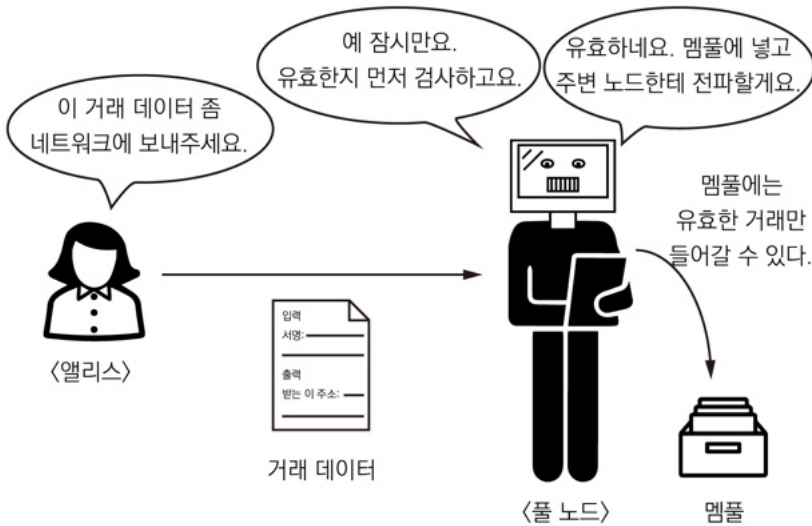
블록 데이터는 말 그대로 블록 데이터를 보관하는 곳이다. 블록은 블록 헤더와 거래 개수, 블록 바디로 구성된다. 블록 헤더에는 ① 버전, ②

이전 블록 해시값, ③ 머클 루트, ④ 타임스탬프, ⑤ 난이도 목표 비트 값, ⑥ 논스값 데이터가 들어있다. 거래 개수는 블록 헤더와 블록 바디 어디에도 속하지 않는 데이터다. 블록 바디는 거래 데이터와 위트니스 (증인) 데이터로 이루어져 있다. 거래 데이터의 해시값인 txid를 머클 트리 구조로 해싱하면 블록 헤더의 머클 루트가 된다.



UTXO 세트는 UTXO를 따로 보관하는 곳이다. 거래를 검증할 때는 거래의 입력값에 쓰인 UTXO가 정말로 이전에 쓰인 적이 없는 UTXO인지 검사를 해야 한다. 하지만 이것을 블록 데이터 전체에서 찾는 것은 컴퓨터에게도 너무 힘든 일이다. 따라서 풀 노드는 UTXO들을 모아 또 다른 공간에도 저장해놓는데, 이것이 UTXO 세트다. 풀 노드들은 새 블록을 받아서 검증할 때마다 그 블록에서 쓰인 출력들을 UTXO 세트에서 삭제하고, 새로운 UTXO들을 추가한다. 참고로 말하자면 비트코인의 첫 데이터가 담겨 있는 '제네시스 블록'은 블록 검증 대상이 아

니기 때문에 제네시스 블록에 있는 코인베이스 보상은 영원히 쓰일 수가 없다. UTXO 세트에 추가되지 않기 때문이다.



뱀풀은 아직 블록에 포함되지 않은 미확정 거래들을 풀 노드가 임시로 저장하는 공간이다. 어떤 노드가 새로운 거래를 비트코인 네트워크에 전파하면 풀 노드들은 그 거래를 검증한다. 만약 그 거래가 검증을 통과했다면 풀 노드들은 이 거래를 임시 저장소인 뱀풀에 보관한다. 채굴자들은 다음 블록을 만들 때 뱀풀에 있는 거래를 모아(가상 바이트 당 수수료가 높은 거래부터 모아) 블록을 구성한다. 만약 거래가 유효하지 않으면 풀 노드들은 거래를 뱀풀에 넣지 않고 그냥 버린다. 만약 거래 수수료를 너무 낮게 설정한 거래가 있다면 그 거래는 뱀풀에 오랫동안 남겨져 있거나, 너무 오래되면 뱀풀에서 튕겨 나오기도 한다. 뱀풀은 모든 풀 노드가 똑같이 동기화하는 데이터가 아니고, 모든 풀 노드가 각각 따로 관리한다. 풀 노드마다 뱀풀 저장 공간 크기를 다르게 설정할 수

있으므로 메모리의 구성은 다양할 것이다. 새로운 블록이 전파됐을 때, 풀 노드들은 메모리에 있던 거래들 중 새 블록에 실린 거래들은 메모리에서 삭제한다.

가지치기 풀 노드

노드에는 가지치기 `pruning` 기능이 있다. 이를 이용하면 블록 데이터 용량을 낮게 유지할 수도 있다. 가지치기 기능을 사용하면 블록의 모든 데이터를 보관하지 않고 최근 블록 데이터 일부와 블록 헤더만 보관한다. 설정한 용량이 될 때까지 이전 거래 데이터들을 삭제하는 것이다.

하지만 이렇게 가지치기 기능을 켜고 바로 SPV 노드(라이트 노드)가 되는 것이 아니다. 가지치기 기능을 켜도 검증 방식은 풀 노드와 동일하게 한다. 이런 가지치기 풀 노드는 UTXO 세트는 그대로 보관하고 있기 때문이다. 새 블록 데이터를 받을 때마다 계속 UTXO 세트를 업데이트하기 때문에 가지치기 풀 노드도 풀 노드와 동일한 방식으로 검증할 수 있는 것이다.

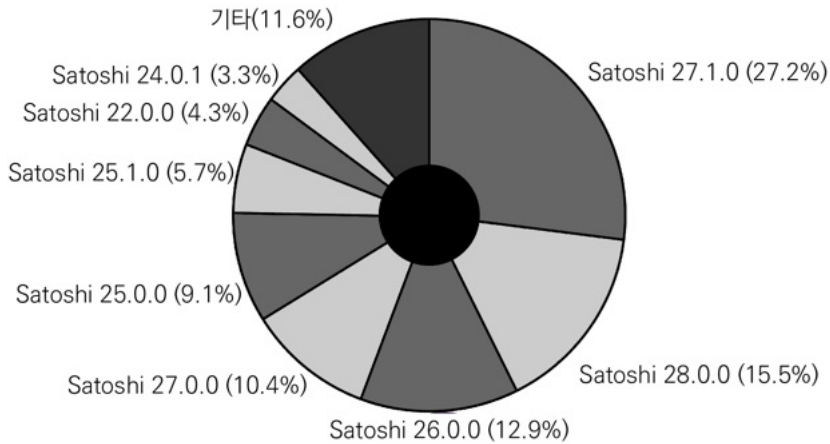
비트코인 클라이언트: 비트코인 코어와 노츠

풀 노드를 실행하는 대표적인 소프트웨어는 비트코인 코어 `Bitcoin Core`다. 비트코인 코어의 첫 번째 버전은 사토시 나카모토가 만든 것으로, 비트코인 코어는 사토시가 만든 오리지널 클라이언트를 이어받은 프로젝트다. 현재까지 가장 널리 쓰이고 있는 클라이언트다.

또 다른 클라이언트가 비트코인 노츠 `Bitcoin Knots`다. 노츠는 코어를 기반으로 더 다양한 기능과 선택지를 추가했다. 예를 들어 노츠에는 `OP_RETURN` 거래나 오디널스나 룬 토큰과 관련된 거래를 메모리에 추

가하지 않고 전파를 거부하는 옵션 등이 있다. 만약 노드를 사용자의 채굴기와 연결하면 사용자가 원하는 방식대로 블록 템플릿을 구성해 채굴할 수가 있다.

비트코인 클라이언트의 사용 비율은 여전히 비트코인 코어가 압도적이다. 그러나 사용자들은 비트코인 코어에 새로운 업데이트가 나오면 업데이트를 할지 말지 자유롭게 결정할 수 있다. 즉, 업데이트는 강제가 아니다. 비트코인은 지금까지 하드포크보다는 항상 이전 버전과 호환이 되는 쪽으로 소프트포크를 지향해왔다. 언젠가 거의 만장일치의 찬성을 필요로 하는 하드포크를 피할 수 없을지도 모르는 일이지만, 지금까지는 소프트포크를 지향해왔다. 따라서 비트코인 네트워크에는 다양한 버전의 비트코인 코어가 공존하고 있다.



초기 블록 다운로드(IBD)

풀 노드를 처음 가동하게 되면 초기 블록 다운로드(Initial Block Download, 이하 IBD) 과정을 거쳐야 한다. IBD는 제네시스 블록을 시작

점으로 하여 그다음 블록부터 현재 블록까지를 이웃 노드로부터 다운로드하고, 각 거래와 블록을 직접 검증하여 네트워크 상태를 맞추는 절차다. 쉽게 말해 블록 데이터들을 다운로드하는 과정이다. 참고로 0번 블록인 제네시스 블록은 클라이언트 자체에 하드코딩되어 있으므로 이웃 노드에게서 다운로드하는 대상이 아니다. 이렇게 블록 데이터를 다운로드할 때 모든 거래와 블록을 검증하고, 동시에 UTXO 세트도 업데이트하기 때문에 시간이 오래 걸린다. 이 과정을 거쳐 100% 동기화가 되어야 진정한 비트코인 풀 노드가 된다.

가지치기 풀 노드도 IBD는 동일하게 수행한다. 다만 IBD를 수행하며 디스크 공간을 절약하기 위해 오래된 블록의 가지치기를 병행한다.

아웃바운드 연결과 인바운드 연결, 인바운드 허용 노드

비트코인 네트워크에서는 수많은 풀 노드들이 서로 연결되어 거래와 블록 데이터를 주고받는다. 노드끼리 처음 연결을 시도할 때 먼저 연결을 시도하는 것을 아웃바운드 연결이라 하고, 상대방이 시도한 연결을 받아주는 것을 인바운드 연결이라 한다. 쉽게 말하자면 전화를 거는 사람은 아웃바운드, 전화를 받는 사람은 인바운드인 것이다.

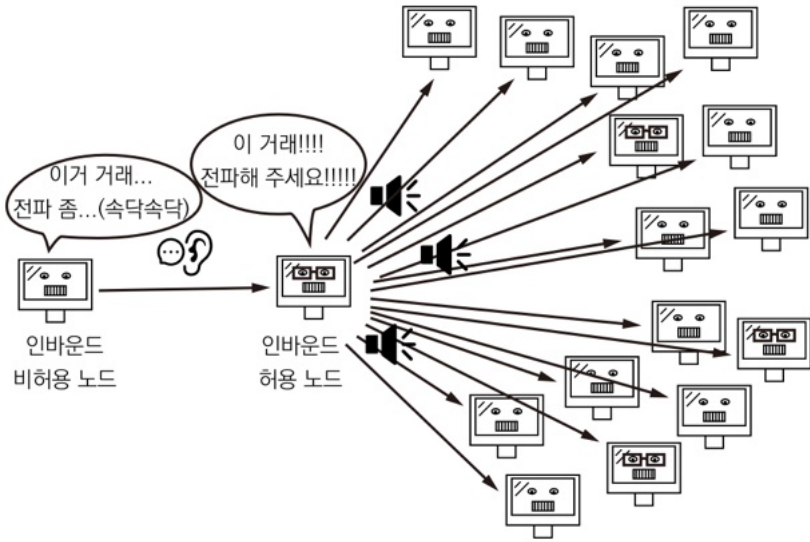


비트코인 풀 노드 운영을 시작하면 기본적으로 인바운드 연결이 비허용되어 있고 아웃바운드 연결만 가능하다. 무슨 뜻이냐면 다른 풀 노드

가 나의 노드에 연결을 시도할 수는 없고, 내 노드가 다른 노드에게 연결을 시도하는 것만 가능하다는 뜻이다. 보통 공유기나 방화벽 설정 때문에 외부에서 내 노드로 인바운드 연결이 들어올 수 없기 때문이다. 인바운드 연결을 허용하지 않은 노드는 아웃바운드 연결만 가능하기 때문에 당연히 인바운드 연결을 허용한 노드와 연결될 것이다. 상대 노드가 내 노드가 신청한 연결을 받으려면, 상대 노드는 인바운드 연결이 허용되어 있어야 하기 때문이다. 쉽게 설명하자면 내 노드는 수신(연결 요청 받기) 금지가 되어있고, 발신(연결 요청 보내기)만 가능한 상태인 것이다.

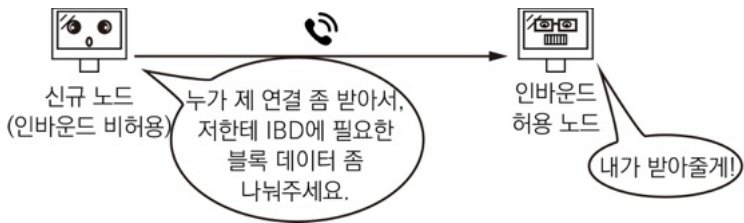
아웃바운드 연결만 가능할 때 연결할 수 있는 이웃 노드(Peer)의 수는 최대 8개다. 클리어넷뿐만 아니라 토르, I2P 연결까지 합하면 최대 11-12개까지 피어 연결이 가능하다. 반면 인바운드를 허용한 노드는 아웃바운드 연결을 신청하는 노드들과 연결되는데 이때는 최대 125개의 노드와 연결이 가능하다.

인바운드를 비허용한 노드나 허용한 노드 모두 새 거래나 새 블록을 받으면 똑같은 방식으로 검증하고 전파한다. 내 노드가 인바운드 연결이 비허용이라서 1번부터 8번까지 8개의 노드와 연결되어 있다고 해보자. 1번 이웃 노드로부터 새로운 거래나 블록을 전파받으면 내 노드는 해당 거래 혹은 블록을 검증한 뒤 다시 2번부터 7번 노드에게 전파한다. 인바운드 허용 노드는 그 전파를 125개 노드에게 한다. 인바운드 연결 비허용 노드가 인바운드 연결 허용 노드에게 귓속말을 하는 것이라면, 인바운드를 허용한 노드는 확성기를 들고 큰 소리로 네트워크에 전파하는 것과 같다.



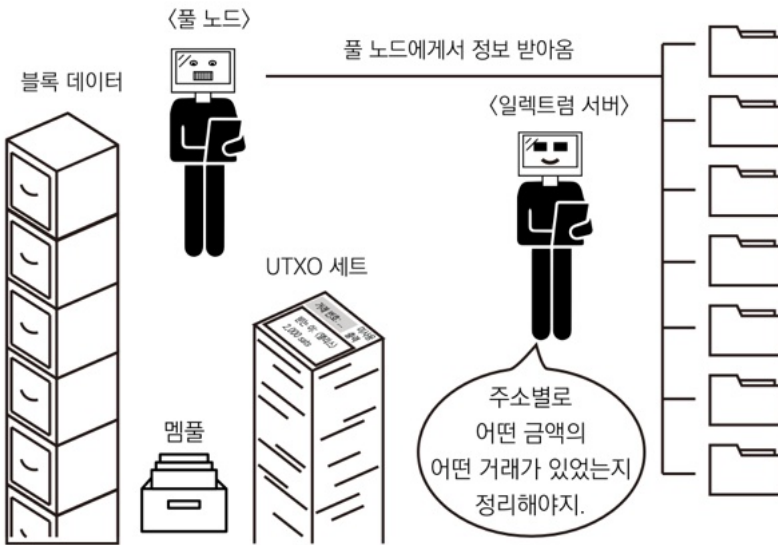
인바운드 허용 노드는 특히 IBD를 갖 시작한 노드에게 더 많은 기여를 한다. 새롭게 네트워크에 참여한 노드는 당연히 인바운드가 비허용되어 있으므로 인바운드를 허용한 노드와 아웃바운드 연결을 시도해야 한다. 어딘가에 전화를 걸어야 한다는 뜻이다. 그러면 전화를 받아주고 (비유임을 잊지 말자) 블록 데이터를 나눠줄 노드가 필요한데, 이게 바로 인바운드를 허용한 노드다.

인바운드 허용 노드나 인바운드 비허용 노드나 네트워크에 기여하고 있는 것은 똑같지만 그 중요도는 다르다. 인바운드 연결을 허용한 노드가 더 많은 기여를 한다.



일렉트럼 서버

내 위치-온리 지갑과 풀 노드를 연결하기 위해서는 일렉트럼 서버가 필요하다. 풀 노드(코어나 노츠)는 블록 데이터와 UTXO 세트는 보관하고 있지만 이것은 주소별 거래 목록은 아니다. 위치-온리 지갑이 지갑에 있는 금액을 빠르게 조회하기 위해서는 주소별로 어떤 거래(txid)를 통해 얼마의 금액이 들어왔고, 얼마의 금액이 나갔는지 정리되어 있는 ‘색인’이 필요하다. 일렉트럼 서버는 바로 이 역할을 한다. 일렉트럼 서버는 풀 노드가 갖고 있는 데이터를 읽으면서 주소별 거래 색인(인덱스)을 만든다. 풀 노드 자체에는 주소별 인덱스가 없기 때문에 일렉트럼 서버가 별도로 정리하는 것이다.



따라서 일렉트럼 서버를 통해 위치-온리 지갑과 풀 노드를 연결할 수 있으며, 위치-온리 지갑은 매우 빠르게 주소에 얼마가 들어있는지 스캔할 수가 있다. 이러한 일렉트럼 서버에는 Electrs, Fulcrum,

ElectrumX 등이 있다. 일렉트럼 서버를 직접 운영하면 사용자는 제3자에게 지갑 주소나 거래 기록을 노출하지 않고도 안전하게 자기 지갑의 비트코인을 조회할 수 있다.

RPC 인터페이스

비트코인 풀 노드를 운영하면 블록 데이터를 직접 보관하고 검증하게 된다. 우리가 풀 노드에게 ‘최신 블록의 높이는 얼마인지’, 아니면 ‘이 주소로 얼마가 들어왔는지’와 같은 질문을 할 때 RPC 인터페이스를 사용하게 된다.

RPC는 Remote Procedure Call의 약자로, 원격 프로시저 호출이라는 뜻이다. 별도의 코딩 없이 다른 공간에서 함수나 프로시저(데이터를 불러오는 절차)를 실행할 수 있게 하는 통신 기술이다. 원격으로 풀 노드에 명령을 내릴 수 있게 해주는 통신 방식이라고 생각하면 된다.

RPC 인터페이스를 사용하기 위해서는 먼저 RPC 요청을 받도록 사용자 이름과 비밀번호가 설정되어 있어야 한다. 누구나 원격으로 내 풀 노드에 접속해서 명령을 내릴 수 있다면 위험하지 않겠는가? 그래서 먼저 `rpcuser` (사용자 이름)와 `rpcpassword` (비밀번호)가 설정되어 있어야 하는 것이다.

비트코인은 RPC 통신을 할 때 JSON-RPC라는 형식을 사용한다. JSON은 데이터를 주고받을 때 널리 쓰이는 포맷으로, 비유하자면 문서 양식이라고 생각하면 된다. JSON-RPC는 이를 기반으로 한 호출 방식이다.

자주 쓰는 RPC 명령어 예시를 몇 개 들어보겠다.

`getblockchaininfo`는 현재 내 블록 데이터의 정보를 알려주는 명령어다. `getblockhash` <블록 높이>는 블록 해시값을 알아내는 명령어다. `getblock` <블록 해시값>은 해당 블록의 정보를 출력하게 하는 명령어다. `getrawtransaction` <txid>는 해당 거래(txid)의 거래 정보를 출력하게 하는 명령어다. `sendrawtransaction` <직렬화된 거래 데이터 HEX>는 거래 데이터를 네트워크에 전파하는 명령어다.

진정한 금융 주권의 실천

풀 노드는 단순한 홈 서버가 아니다. 풀 노드를 운영하는 것은 비트코인의 철학을 구현하는 행위이다. 풀 노드를 운영함으로써 제3자를 신뢰하지 않고 거래와 블록을 스스로 검증할 수 있다. 동시에 네트워크의 분산화 속성을 지키고 강화한다. 비트코인 네트워크에 직접 참여하고 싶다면 풀 노드를 운영하면 된다.

| 엠브렐 홈 구매 및 세팅

풀 노드 구축 방법

풀 노드를 구축하는 방법은 여러 가지가 있다. 윈도우OS나 맥OS에 비트코인 코어나 노드를 설치할 수도 있다. 하지만 입문자의 경우 앱 설치만 한 번 하면 모든 설정이 되는, 풀 노드 운영에 특화된 운영체제를 설치할 수도 있다. 이러한 운영체제에는 엠브렐OS나 스타트9의 스타트OS가 있다. 이 책에서는 좀 더 많은 확장성과 편리성을 가진 엠브렐OS 위주로 설명할 것이다. 뒤에서는 윈도우OS와 맥OS에서 풀 노드 프로그램을 설치하고, 위치-온리 지갑인 스페로우 지갑과 연결하는 방법도 살펴볼 것이다.

엠브렐OS를 설치하려면 홈 서버 기기가 필요하다. 가장 간편한 방법은 엠브렐OS를 만든 회사에서 판매하는 ‘엠브렐 홈’이라는 기기를 구매하는 것이다. 이 외에 직접 미니컴퓨터를 조립해 엠브렐OS를 설치하는 방법도 있는데, 미니 PC를 조립하거나 라즈베리파이를 조립할 수 있다. 혹은 남는 노트북이나 컴퓨터가 있다면 여기에 설치하는 방법도 있다. 다만 풀 노드는 보통 24시간 가동하므로 이러한 방법은 저전력 미니 PC를 이용하는 것보다 전기료가 많이 나올 수 있다.

이렇게 엠브렐OS를 직접 설치하는 방법들은 엠브렐 홈을 그냥 사는 것보다 조금 더 저렴하게 노드를 구축할 수 있지만, 거꾸로 입문자들에게 다소 어렵게 느껴질 수 있다. 반면에 엠브렐 홈은 구매하고 나서 공유기에서 나오는 인터넷 선과 전원선만 연결하면 세팅이 끝날 정도로 매우 간단하다. 이번 장을 서술하기 전에 필자는 엠브렐로부터 어떠한 대가도 받지 않고 직접 엠브렐 홈을 구매했으며 특정 제품을 광고하려

는 목적이 아님을 밝힌다. 만약 미니 PC 조립에 도전해 보고 싶다면 미니 PC나 라즈베리파이를 이용해 풀 노드를 구축하고, 그게 두렵다면 엄브렐 홈을 구매하는 것을 고려해 볼 수 있다. 이제 엄브렐 홈을 구매하는 방법에 대해 알아보자.

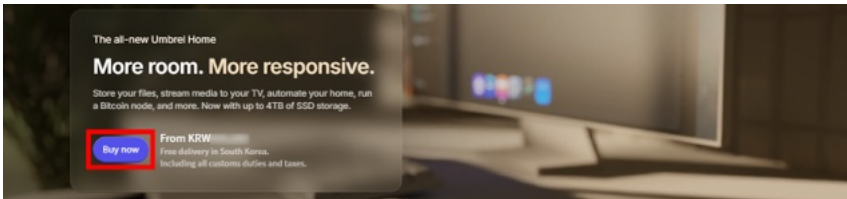
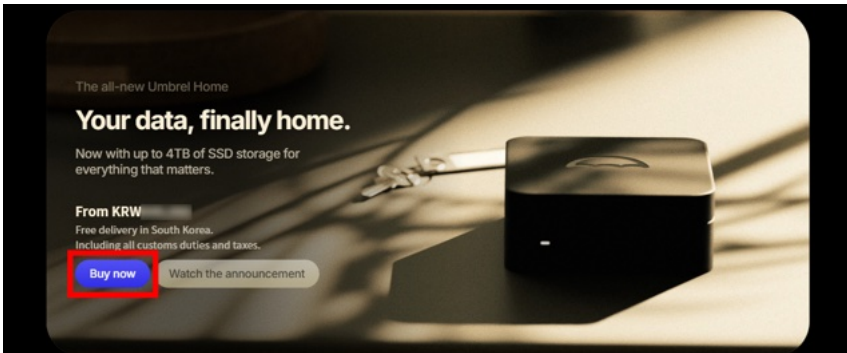
엄브렐 홈 구매 방법

엄브렐 홈을 구매해서 풀 노드를 구축하려는 경우 가장 어려운 것은 구매 과정이다. 구매만 하면 그 이후의 세팅은 매우 간단하다. 엄브렐 웹사이트에 접속한다.

<https://umbrel.com/?country=KR>

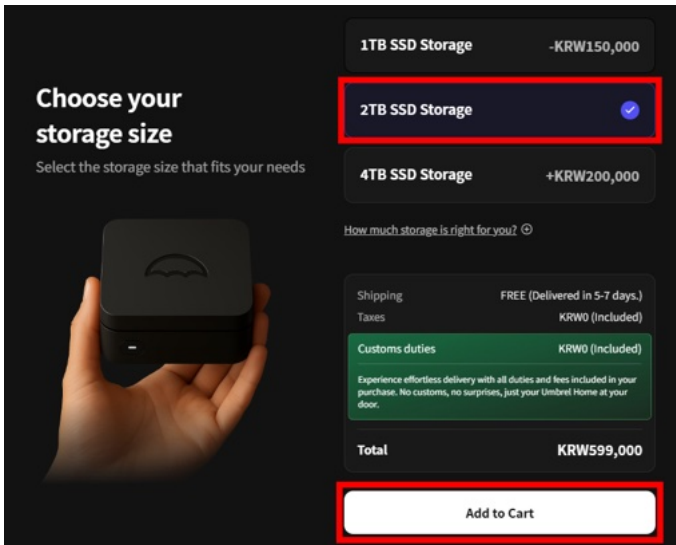


웹사이트에서 [Buy now] → [Buy now]를 누른다.

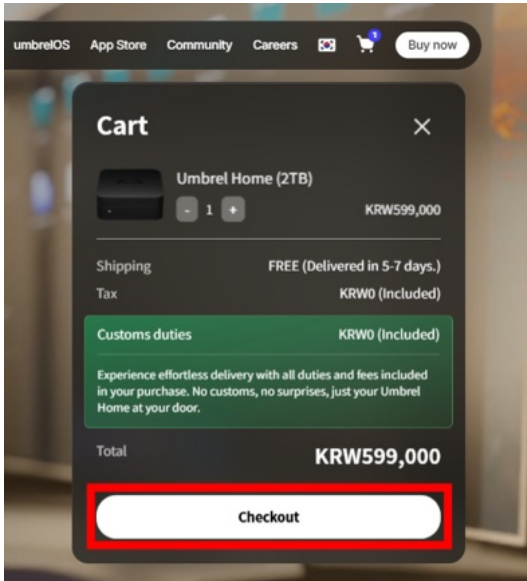


엄브렐 홈 기기의 가격은 계속 변한다. 2025년 8월 기준, 저장 용량이 2TB인 엄브렐 홈 기기의 가격은 599,000원이다. 저장 용량에 따라 가격이 달라지는데 비트코인의 블록 데이터 크기가 약 800GB 정도 되는 2025년 8월 기준으로는 멤폴이나 일렉터스, 라이트닝 노드와 같은 다른 앱도 사용하기 위해서는 적어도 2TB는 필요하다. 1TB는 풀 노드 앱만 설치해도 용량이 거의 찰 것이다.

필자는 [2TB SSD Storage]를 선택하고 진행했다. 이후 [Add to Cart]를 누른다.



가격을 확인하고 [Checkout]을 누른다.



이제 배송 주소를 입력하는 칸이 나온다. 'Email' 칸에는 이메일을 입력하면 된다. 엠브렐 홈의 구매 내역이나 배송 추적 내역 등은 모두 여기에 입력한 이메일로 수신되므로 정확히 입력한다. 'Last name'에는 성, 'First name'에는 이름을 입력한다. 'Postal code'에는 우편번호를 입력하고, 'Province'에서 도/시를 선택한다. 'City'에는 시/구를 입력하고, 'Address'에 도로명 주소, 'Apartment, suite, etc.'에 상세 주소를 입력한다. 이때 모두 영어로 입력해야 한다. 'Phone'에는 전화번호를 입력한다.

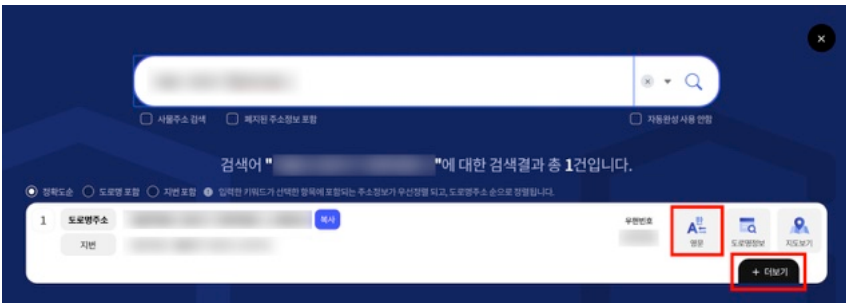
The image shows a checkout page for Umbrel Home. The page is dark-themed. At the top left is the Umbrel Home logo. Below it, there are two payment options: 'shop 24' and 'G Pay'. A red box highlights the 'Email' field under the 'Contact' section. Under the 'Delivery' section, several fields are highlighted with red boxes: 'Last name', 'First name', 'Postal code', 'Province' (set to Seoul), 'City', 'Address', and 'Phone'. On the right side, the product 'Umbrel Home (2TB)' is listed for ₩599,000. Below that is a 'Discount code' field and an 'Apply' button. The subtotal is ₩599,000, and the total is ₩599,000. The shipping address field is currently empty and labeled 'Enter shipping address'.

만약 영어 주소를 작성하는 방법을 모른다면 다음 웹사이트에 접속해서 배송 받을 주소를 검색한다.

<https://juso.go.kr>



그리고 [영문]을 누르면 영문 주소가 나온다. [더보기]를 눌러 [우편주소 표기방법]을 누르면 더 자세한 영문 주소 작성 방법을 알 수 있다. 이를 참고하여 배송 정보를 채우면 된다.



이제 결제할 신용카드 정보를 입력해야 한다. 결제할 카드가 비자나 마스터카드 등 해외 결제를 지원하는 카드여야 한다. 'Card number'에 카드번호를 입력하고, 'Expiration date'에 카드 유효기간(달/년), 'Security code'에 CVC 번호, 'Name on card'에 카드에 입력된 자기 이름을 입력하면 된다.

이 모든 과정이 싫다면 비트코인으로 결제할 수도 있다! 아래 [Bitcoin]의 라디오 버튼을 선택하면 된다. 비트코인은 돈이니 라이트닝 네트워크 혹은 온-체인으로 지구 반대편에 있는 회사에 결제할 수도 있다.

배송 주소 정보와 결제하는 사람의 정보가 같다면 'Use shipping address as billing address' 체크박스에 체크하면 된다.

아래에 빠른 배송 조회를 위해 자신의 정보를 저장할 것인지 묻는 체크박스가 있다. 여기에 연락처를 입력하는 것은 선택사항이다.

다 입력했다면 [Pay now]를 누른다.

The image shows a checkout page with a dark theme. On the left, there are input fields for 'Apartment, suite, etc. (optional)' and 'Phone'. Below these is a 'Shipping method' section with a placeholder text: 'Enter your shipping address to view available shipping methods.' The 'Payment' section is highlighted with a red border and contains the following elements:

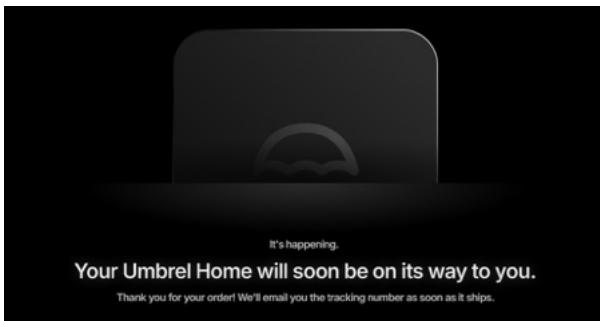
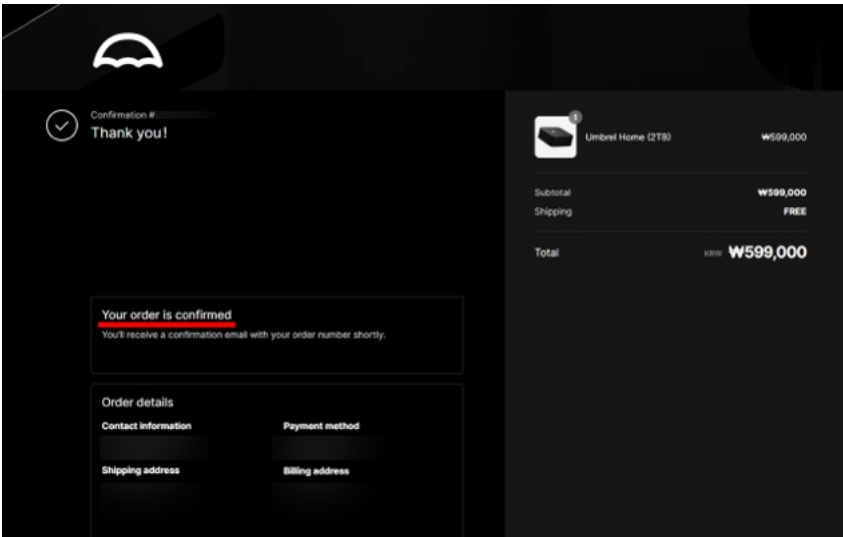
- A red box around the 'Credit card' option, which includes icons for Visa, Mastercard, and others.
- Input fields for 'Card number', 'Expiration date (MM / YY)', and 'Security code', each with a red box around it.
- An input field for 'Name on card' with a red box around it.
- A checked checkbox for 'Use shipping address as billing address' with a red box around it.
- Radio buttons for 'shop Pay in full or in installments' and 'Bitcoin' with a red box around the entire section.

Below the payment section is a 'Remember me' section with a checked checkbox for 'Save my information for a faster checkout with a Shop account' and an input field for 'Mobile phone number' with a red box around it. At the bottom of the form is a large blue 'Pay now' button with a red border. To the right of the form, the order summary shows 'Umbrel Home (2TB)' for ₩599,000, a 'Subtotal' of ₩599,000, and a 'Total' of ₩599,000. A 'Discount code' field and an 'Apply' button are also visible.

그러면 입력한 정보들이 맞는지 한 번 더 확인하는 창이 나오고, 'Personal Customs Code'를 입력하는 창이 나온다. 여기에 개인통관 부호를 입력하면 된다.



결제를 마치면 결제가 잘 되었다는 창이 나온다.



이메일로도 구매가 잘 되었다는 내용과 주문 번호(ORDER)가 발송된다.




ORDER

Thank you for your purchase!

We're getting your order ready to be shipped. We will notify you when it has been sent.

Order summary

 Umbrel Home (2TB) × 1	₩599,000
Subtotal	₩599,000
Shipping	₩0
Taxes	₩0
Total	₩599,000 KRW

참고로 엠브렐 홈을 구매하면 관세가 가격에 포함되어 있다. 즉, 엠브렐 측에서 관세를 내준다. 한국에 제품이 통관될 때 관세를 따로 낼 필요가 없다. 혹여나 관세를 납부하라는 연락이 온다면 정말 관세청이나 택배사의 전화가 맞는지, 보이스피싱은 아닌지 의심해 보자. 또한, 이미 통관부호와 주소를 전부 입력했음에도 빠른 통관을 위해 어떤 웹사이트로 접속해서 다시 배송 정보를 입력하라는 문자가 온다면 그것도 스미싱일 가능성이 높다. 확인되지 않은 링크는 누르지 말고, 정말 택배사에서 온 문자가 맞는지 의심하자. 만약 통관부호와 개인정보를 입력하면 다른 사기 등에 자신의 정보가 쓰일 수 있으므로 그 즉시 통관부호를 재발급받아야 한다.

엠브렐 홈 세팅

엠브렐이 도착하기 전 구비해야 할 것이 있다. 엠브렐에서는 어댑터로 EU 규격을 보내준다. 한국에서 EU 규격의 플러그는 꽂을 수 있지만 험령하다. 그러면 기기에 전력이 안정적으로 공급되지 못할 수 있다. 따라서 KC 인증이 있는 멀티플러그 어댑터를 꼭 미리 구매하자. 참고로 엠브렐 기기의 충전 전력은 12V, 2.5A 고정형이다. 이는 일반적으로 C타입으로 충전하는 기기들에 비해 전압이 매우 높다. 따라서 엠브렐 어댑터에 스마트폰 등의 다른 기기를 충전하면 고장 날 가능성이 크니 삼가도록 하자.

엠브렐 기기 포장을 열어보면 엠브렐 기기, 랜선, 그리고 어댑터가 있을 것이다(사진에서 맨 오른쪽은 따로 구매한 KC 인증 멀티플러그 어댑터이다).



엠브렐 홈의 세팅은 정말 간단하다. 만약 컴퓨터에서 엠브렐 기기에 접속할 것이라면 엠브렐을 연결할 컴퓨터가 엠브렐과 같은 공유기에 연결되어야 한다. 또한, 스마트폰에서 엠브렐 기기에 접속할 것이라면, 스마트폰이 엠브렐이 연결된 공유기에서 나오는 와이파이에 연결되어 있

어야 한다. 이렇게 같은 공유기를 통해 연결되어 있는 것을 '로컬 네트워크'에 있다고 표현한다.

엄브렐 기기에 랜선을 꽂아 공유기와 연결한다. 그다음에 전원선도 꽂는다.



전원이 들어오면 전원 버튼에 불빛이 들어온다. 그러면 세팅이 끝난 것이다. 여기까지 잘 따라왔다면 이제 '엄브렐 설정 및 풀 노드 동기화' 장으로 넘어가면 된다.



| 미니 PC 조립하고 엠브렐OS 설치하기

이 장에서는 미니 PC를 조립하고 엠브렐OS를 설치하는 방법을 알아볼 것이다. 풀 노드는 기본적으로 홈 서버다. 24시간 365일 켜져 있는 것을 전제로 하기 때문에 전력 소모가 적은 미니 PC를 장만하여 구축하는 것이 좋은 선택이다. 남은 PC나 노트북에 엠브렐OS를 설치하는 방법도 있으나, 24시간 매일 켜놓는 경우 일반적인 노트북과 저전력 미니 PC는 전기요금이 한 달에 약 3,000원 정도까지도 차이가 나기 때문이다. 미니 PC는 직접 조립하고 OS를 설치해야 한다는 번거로움이 있지만 엠브렐 홈을 구매하는 것보다 적은 비용으로 풀 노드를 구축할 수 있고, 직접 PC를 조립하는 것은 재밌는 경험이 될 수도 있다.

미니 PC 준비물

미니 PC의 경우 라즈베리파이5를 이용하여 구축할지, 다른 CPU를 사용하여 구축할지 선택할 수 있다. 풀 노드 목적이거나 라즈베리파이를 이용해도 좋은 스펙이다. 라즈베리파이5를 이용해 풀 노드를 구축하는 방법은 다음 장을 참고하라.

필자의 주관적인 의견으로는 2025년 8월 기준, CPU로 N100을 추천한다. 저전력 중 가장 좋은 스펙이기 때문이다. 좀 더 좋은 N200이나 새로 나온 N150 등도 있지만 가격 대비 성능이 그렇게 좋다고 생각하지는 않는다.

미니 PC를 구축할 때는 베어본을 구매하는 것이 편리하다. 베어본은 일반적으로 메인보드에 CPU가 조립되어 나오고 케이스까지 한 번에 제공되는 경우가 많기 때문이다.

필자는 T8 PLUS 베어본 모델에 램 DDR5 16GB가 탑재된 미니 PC를 구매했다. N100이 탑재된 다른 베어본 모델을 사용해도 좋다. 또한 저장 장치는 없는 버전(램 16G, 하드디스크 미포함)으로 준비했다. (다음 사진 참고)



○ CNS타입 >

FIREBAT T8PLUS N100 미니PC EU버전 데스크탑 4K 게임 미니컴퓨터 16G

175,690원 186,900원

스> 해외배송 상품 | 배송정보 ⓘ

배송비 주문시 결제 (20,000원) | CJ택배

카드무이자 | 카드추가혜택

원산지 : 상세설명 참조

옵션선택

타입

16G-하드 디스크 미포함

175,690원 ×

총 금액 **175,690원**

장바구니 **구매하기**

저장 장치는 2TB는 필요하다. 풀 노드만 돌릴 용도라면 1TB여도 되지만, 현재 블록이 차는 속도로 봤을 때 1TB는 앞으로 3-5년 내로 꺾인다. 따라서 여유롭게 2TB는 마련하는 것을 추천한다. 저장 장치를 구매할 때는 자신이 구매한 베어본에 맞는 SSD를 구매해야 한다. T8 PLUS의 경우 M.2 2242 소켓을 지원하므로 SSD도 'M.2 SSD 2TB 2242'라고 검색해야 한다. 다른 베어본을 구매하는 경우 이러한 부분을 고려하여 저장 장치를 준비하자.

The image shows a product listing for a Transcend MTE410S M.2 2TB NVMe SSD. On the left is a product card with the Transcend logo, a '정품' (Genuine) seal, and the model name 'MTE410S'. The card also specifies '2242 / GEN4' and '5년 보증' (5-year warranty). On the right is a shopping interface showing the product title '트랜센드 MTE410S M.2 2TB NVMe SSD (GEN4 / 2242 / 5년)', the price '268,420원', a quantity selector set to '1', and three buttons: '선물하기' (Gift), '장바구니 담기' (Add to cart), and '바로 구매하기' (Buy now).

마지막으로 어댑터가 필요하다. T8 PLUS 베어본의 경우 EU 규격 어댑터가 온다. 한국 플러그 규격과 EU 규격은 다르다. EU 규격이 한국 규격보다 좀 더 얇기 때문에 한국형 콘센트에 꽂으면 약간 헐렁하다. 따라서 안정적인 전력 공급을 위해 한국형 규격의 플러그를 따로 사는 것이 좋다. T8 PLUS의 정격 전압은 12V에 전류 2.5A이다. 전압[V]은 꼭 지켜야 하고, 전류[A]는 더 높아도 상관없다. 따라서 한국형 플러그

12V 5A를 준비했다. 다른 N100 베어본을 구매하는 경우 이러한 부분을 고려하여 어댑터를 준비하자.



(상) 한국형 플러그, (하) EU형 플러그. EU형 플러그가 좀 더 얇은 것이 보인다. EU형을 한국 콘센트에 꽂으면 헐렁하고 전기 공급이 안정적이지 않을 수 있다.



하늘컴플렉스 어댑터 DC 12V 5A HP1250CC 오늘출발

11,000원
배송비 3,000원

씨씨티비이를 고객을 위한 혜택

최대 적립 포인트	1,970원 ?
- 기본적립	110원
- 네이버 현대카드 E&S로 결제 시 >	770원
- 내인바탕이 아니 결제 시 최대 적립 >	220원
회원 가입 추가 적립	440원
최대 5% 적립 시작하기 >	

추가로 미니 PC의 세부 설정을 위해 연결할 모니터가 필요하고(처음에만 잠깐 필요), 엠브렐OS 설치를 위한 USB 저장 장치가 필요하다.

종합하면 준비물은 다음과 같다.

1. T8 PLUS N100 베어본(RAM 16GB, 저장장치 미포함)
2. M.2 SSD 2TB 2242
3. 12V 5A 어댑터
4. 엠브렐OS 설치를 위한 USB 저장 장치
5. 엠브렐OS 세팅을 위해 잠깐 사용할 컴퓨터 모니터, 키보드, 모니터에 연결할 HDMI선(T8 PLUS 구매 시 선은 같이 왔음)
6. 십자 드라이버, 일자 드라이버, 컴퓨터 조립용 미세 십자 드라이버
7. 인터넷 연결할 랜선

미니 PC 조립하기

모든 물품이 배송되었다면 조립을 시작하자. T8 PLUS 베어본 구성품은 아래와 같다. 다음 사진 중 EU형 플러그(왼쪽 아래)는 사용하지 않을 것이고, 베사홀에 조립할 브라켓(오른쪽 아래)도 사용하지 않을 것이다.



한국형 플러그와 2TB SSD, 엠브렐OS를 설치하기 위한 USB도 준비가 되었다.



먼저 SSD를 베어본에 조립해야 한다. 베어본을 분해하자. 십자 드라이버를 이용하여 네 군데 나사를 풀어준다.



십자 드라이버를 이용하여 다음 사진과 같은 부분에 틈을 만들면 덮개를 쉽게 분리할 수 있다.





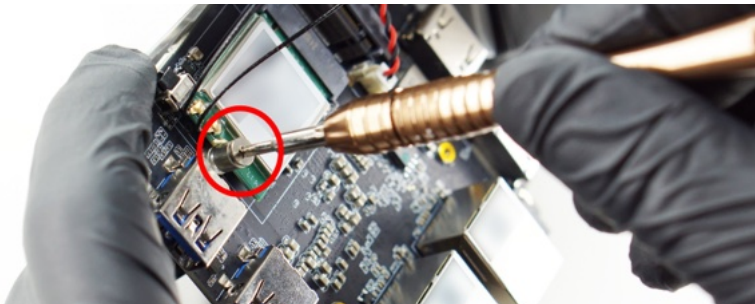
뒷개를 분리하면 안쪽에 다시 4개의 나사를 풀어줘야 한다. 십자 드라이버를 이용하여 4개의 나사를 풀어준다.



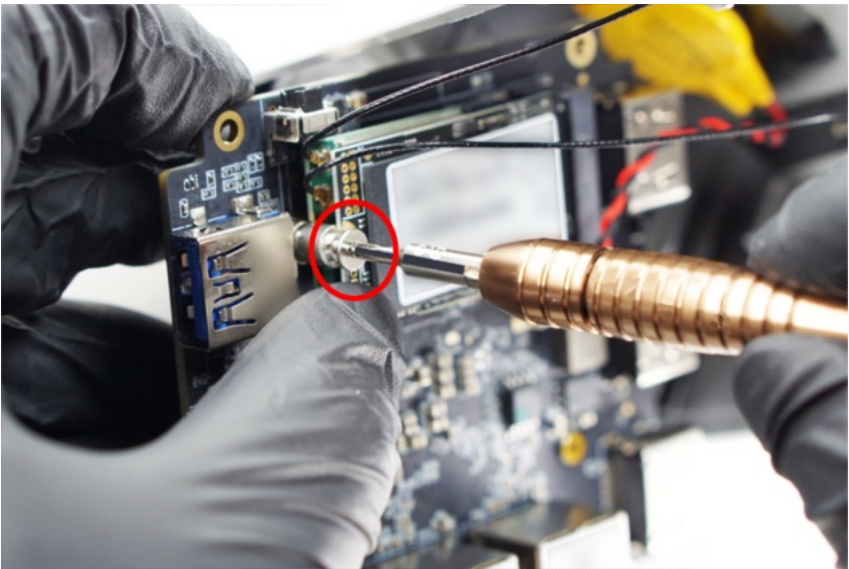
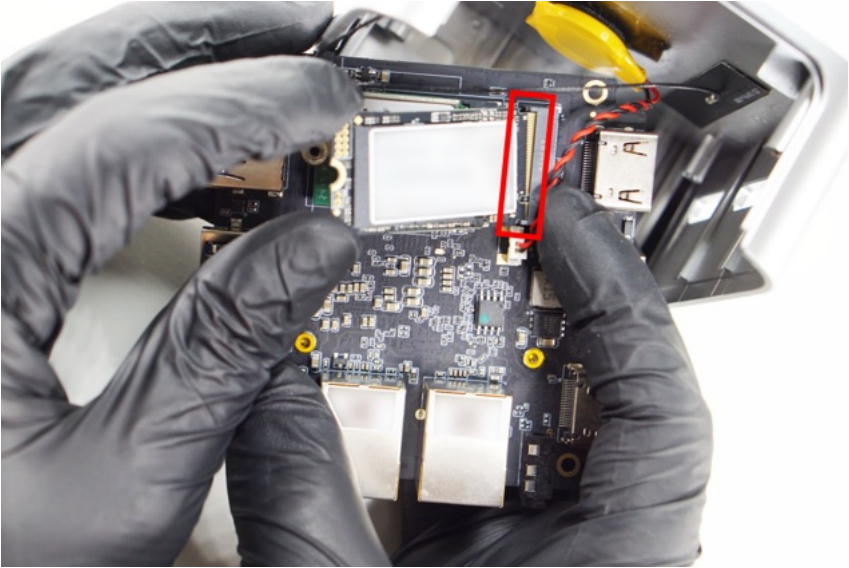
분리가 되면 아래와 같은 모습이다. 아래에서 빨간색 네모 박스 부분에 SSD를 끼울 것이다. 이때 나사를 풀기 위해 컴퓨터 조립용 미세 십자 드라이버가 필요하다. 주방용 칼 등으로 해도 될지는 모르겠지만, 미세 십자 드라이버가 하나 있으면 편하다.



미세 드라이버로 얇은 나사 하나만 풀어줘야 한다. 아래에 있는 두꺼운 나사까지 풀지 않도록 주의하여 풀어준다. 두꺼운 나사까지 풀리면 일단 분리한 뒤 펜치로 잡고 얇은 나사를 다시 풀어주면 된다.



이제 SSD를 끼우고, 다시 나사를 조여준다.



조립은 분해의 역순이다. T8 PLUS는 분해되어 있을 때 전원 버튼 부분이 잘 빠지니 이 부분을 잘 고정해 놓고 조립하자.



이제 조립이 끝났다. 엠브렐OS를 설치하기 전에 먼저 모니터를 연결하고 램 설정을 해야 한다.

바이오스에서 램 설정하기

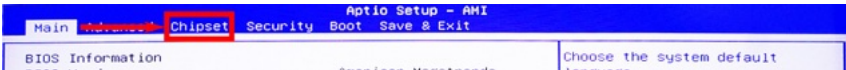
T8 PLUS는 램 16GB 베어본을 구매하면 DDR5 4,800 MHz 램이 장착되어 나온다. 하지만 기본적으로 램 성능이 3,200 MHz으로 설정되어 있다. 따라서 이 설정을 바꿔줄 필요가 있다. (DDR4 램의 경우에는 이 설정을 수정할 필요가 없다.)

먼저 모니터 선(HDMI)과 전원선을 연결한다. 키보드도 연결한다.

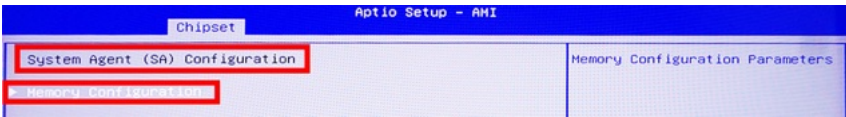


컴퓨터 전원을 켜다. 그러면 바로 바이오스 설정 창으로 진입할 것이다. 만약 윈도우 등의 OS 설치가 이미 되어있는 베어본을 구매했다면 부팅 중 F7이나 F2, F10 혹은 Delete 키를 연타하여 바이오스 메뉴로 진입하면 된다.

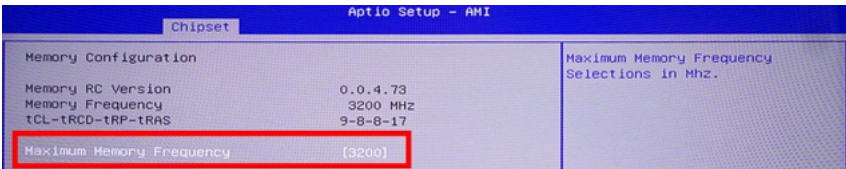
바이오스 메뉴에서 오른쪽 방향키를 두 번 눌러 [Chipset] 설정으로 이동한다.



[System Agent Configuration]에서 엔터를 누른다. 그러면 [Memory Configuration]이 뜬다. 여기서 또 엔터를 누른다.



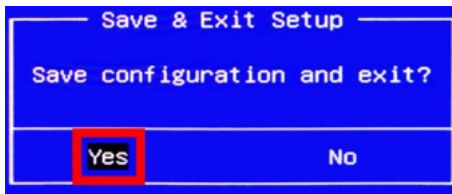
[Maximum Memory Frequency]가 뜨면 엔터를 누른다.



방향키를 내려 [4800]으로 설정을 바꾼 뒤 엔터를 누른다.



F4 키를 눌러 저장하고 종료한다.



이제 바이오스 설정이 끝났다. 엠브렐OS를 설치해 보자.

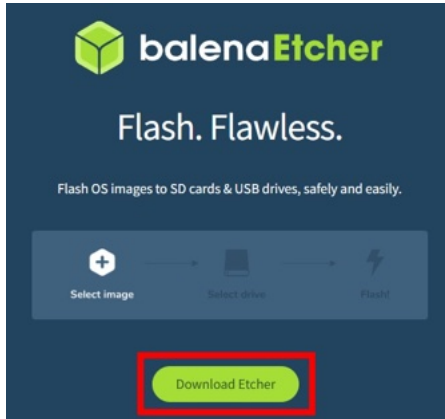
OS 설치용 USB 만들기

먼저 USB 저장 장치를 OS 설치용으로 만들기 위해 발레나에처로 구워야 한다. 아래 링크에서 발레나에처 프로그램을 다운로드한다.

<https://etcher.balena.io/>



[Download Etcher]를 누른다.



자신의 컴퓨터 운영체제에 맞는 버전을 다운로드한다.

DOWNLOAD

Download Etcher

ASSET	OS	ARCH	
ETCHER FOR WINDOWS (X86 X64) (INSTALLER)	WINDOWS	X86 X64	Download
ETCHER FOR MACOS	MACOS	X64	Download
ETCHER FOR MACOS (ARM64)	MACOS	ARM64	Download
ETCHER FOR LINUX X64 (64-BIT) (ZIP)	LINUX	X64	Download
ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)	LINUX	X86	Download

설치가 완료되었으면 발레나에처를 실행한다.



이제 깃허브에서 움브렐OS iso 파일을 다운로드할 것이다. 아래 링크에 접속한다.

<https://github.com/getumbrel/umbrel/wiki/install-umbrelOS-on-x86-Systems>



[latest umbrelOS USB installer]를 누르면 iso 파일이 다운로드된다.

Install umbrelOS on x86 Systems

Luke Childs edited this page on Aug 1, 2024 - [3 revisions](#)

Note

umbrelOS is specifically built for [Umbrel Home](#). Support for other devices is best-effort and not guaranteed. Compatibility issues may arise due to the differences in hardware, drivers, and such.

Minimum System Requirements

- **CPU:** Dual-core 64-bit Intel or AMD processor. (Quad-core or higher recommended)
- **RAM:** 4GB. (8GB+ recommended)
- **Storage:** 32GB. (Storage needs depend on your use case. For instance, running a Bitcoin node typically requires 1TB or more, while file/media storage needs can be estimated based on your cloud services usage.)

How to install

Step 1: Prepare a bootable USB stick

- Download the [latest umbrelOS USB installer](#) on your computer.
- Download and install [Balena Etcher](#) on your computer.
- Plug a USB stick (at least 4GB) into your computer. Ensure that it doesn't contain any important data as it will be formatted.
- Use Balena Etcher to flash the `umbrelOS-amd64-usb-installer.iso` file to the USB stick.

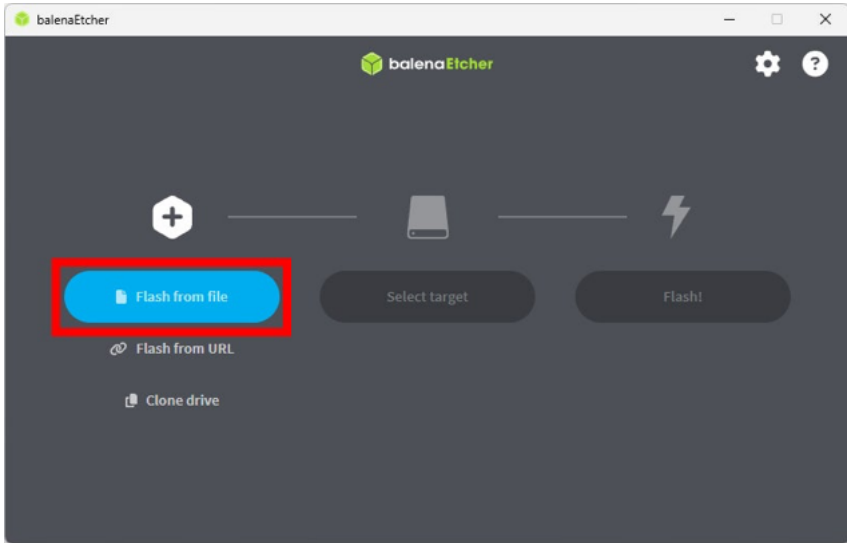


umbrelOS-amd64-usb-installer.iso

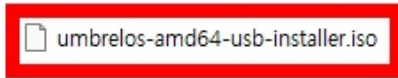
컴퓨터에 USB 저장 장치를 꽂는다(USB 포맷이 안 되어 있다면 FAT32 형식으로 포맷한다).



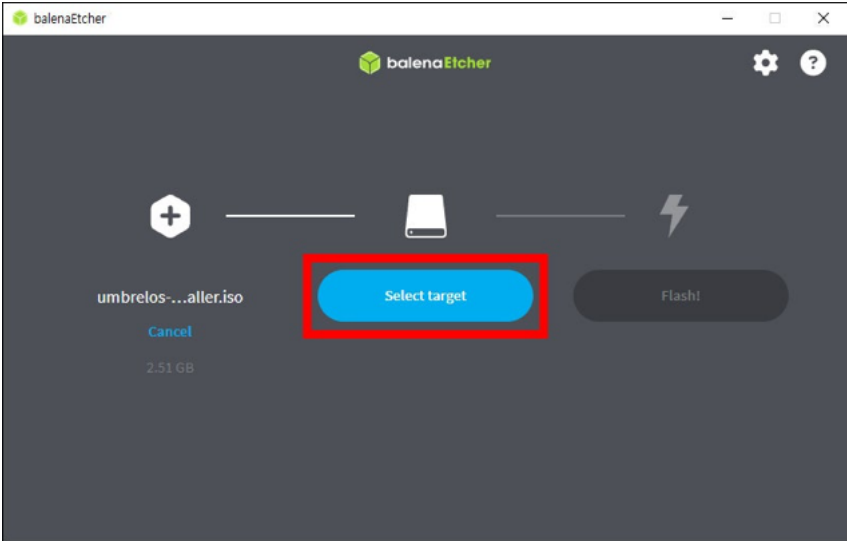
발레나에처를 실행하면 다음과 같은 화면이 나온다. [Flash from file]을 누른다.



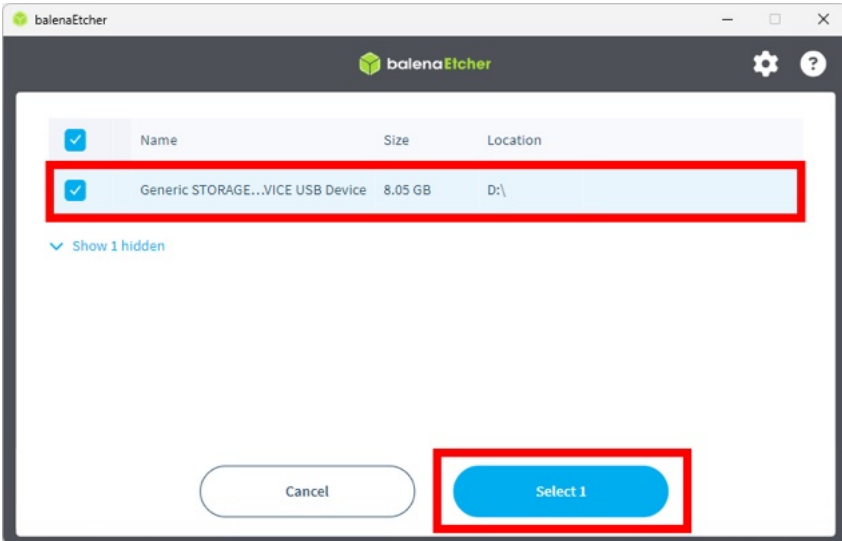
iso 파일을 선택하는 창이 뜰 것이다. 아까 다운로드했던 엮브렐OS의 iso 파일을 찾아서 더블 클릭하자.



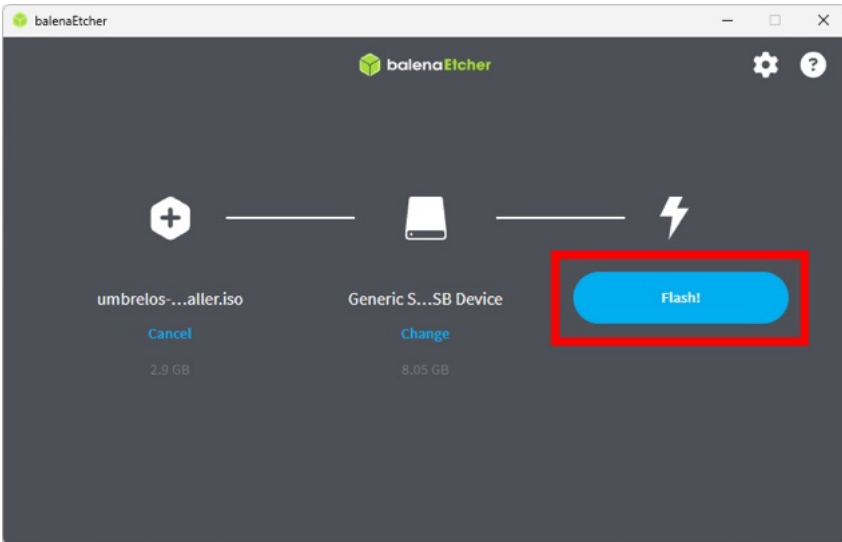
[Select target]을 누른다.



우리가 쫓은 USB 저장 장치를 선택한다.



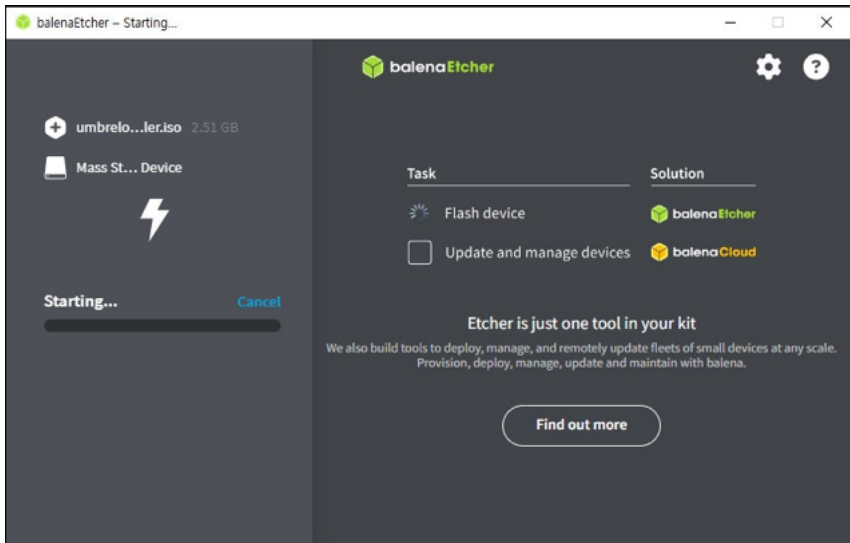
이제 [Flash!]를 누른다.

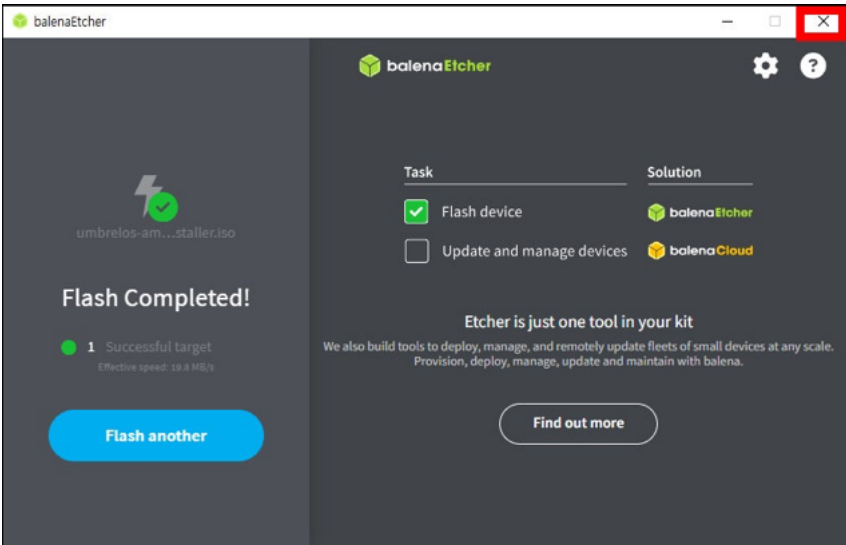


다음과 같은 화면이 뜨다가 'Flash Completed!'가 뜰 것이다. 뜨면 오른쪽 위 [x] 버튼을 누르고 USB를 빼면 된다. 중간에 포맷하라는 알림창이 뜨면 무시하고 취소하면 된다.

만약 Flash에 실패하면 USB를 초기화하고 다시 플래싱해야 한다. 이때 일반적인 포맷 방법으로는 안 되는데, 부팅 USB 초기화 방법으로 검색하면 결과가 많이 나온다.

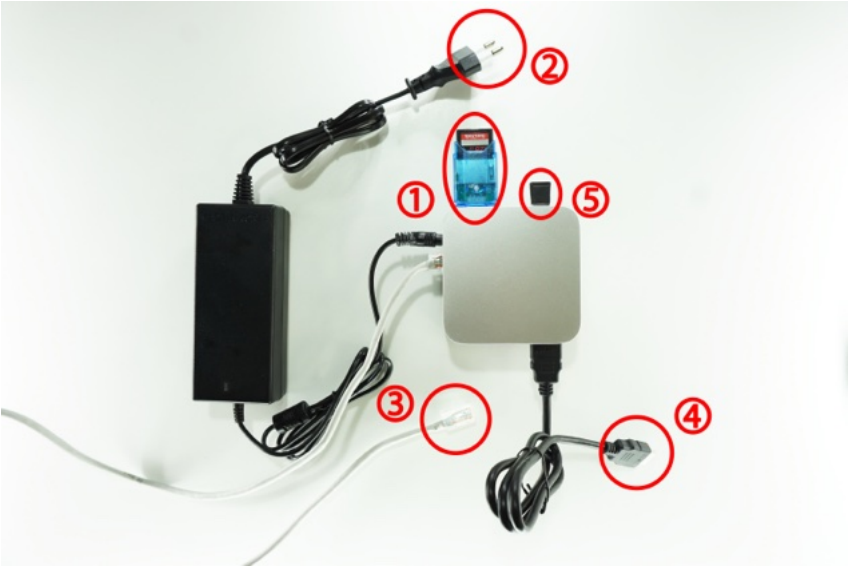
윈도우의 경우 윈도우 키 + R → cmd 입력 후 실행 → 터미널이 나오면 `diskpart` 입력 → `list disk` → 마이크로SD카드 디스크 번호 확인 → `select disk [디스크 번호]` → `clean`을 입력하면 된다. 그리고 다시 포맷하고 처음부터 진행하면 된다.





엠크렐OS 설치하기

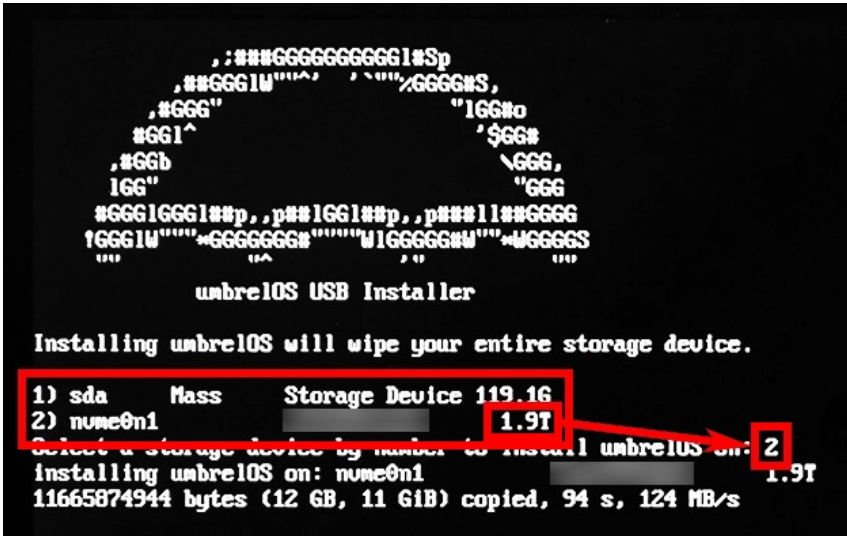
이제 OS 설치용 USB가 만들어졌으니 미니 PC에 엠크렐OS를 설치해 보자. 미니 PC에 ① OS 설치용 USB, ② 전원선, ③ 인터넷 랜선, ④ 모니터에 연결할 HDMI선, ⑤ 마우스/키보드 USB를 연결한다.



전원을 켜면 화면에 무언가 많이 지나갈 것이다.



우산 모양이 뜨면 그 밑에 저장공간을 선택하는 창이 나온다. 여기서 당연히 미니 PC에 설치한 2TB 저장 장치를 선택해야 한다. 다른 하나는 OS 설치용 USB다. 1.9T로 표시되어 있는 숫자를 입력하고 엔터를 누른다.



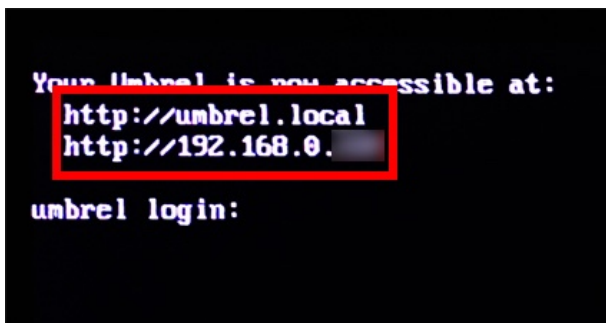
무언가 뜨다가 umbreLOS has been installed! 문구와, Press any key to shutdown 문구가 뜨면 엔터를 누른다.



컴퓨터가 자동으로 켜다 켜질 것이다. 다시 무언가가 많이 지나갈 것이다. 만약 Finished로 시작하는 문구가 나왔는데 시간이 많이 지나도 다음 화면으로 넘어가지 않으면 기기를 재부팅해 보자.



다 지나가고 나면 `Your Umbrel is now accessible at:`이라는 문구가 뜰 것이다. 그 밑에 URL 두 개가 뜨는데 이 URL을 잘 기억하자. `http://192.168.0.???` 이것이 엄브렐 기기의 로컬 IP 주소다.



이제 엮브렐 기기에서 OS 설치용 USB를 제거하고, 모니터 연결선도 뽑아도 된다. 미니 PC는 홈 서버의 역할을 한다. 미니 PC는 24시간 365일 켜져 있으면서 풀 노드의 역할을 할 것이다.

여기까지 잘 따라왔다면 이제 '엮브렐 설정 및 풀 노드 동기화' 장으로 넘어가면 된다.

| 라즈베리파이5 조립하고 엠브렐OS 설치하기

이 장에서는 라즈베리파이5를 조립하고 엠브렐OS를 설치하는 방법을 알아볼 것이다. 라즈베리파이5는 앞 장에서 나온 N100 CPU 기반의 미니 PC보다 CPU 성능이 약 2.5배 안 좋다(CPU 벤치마크 점수 기준). 그러나 전력 소모량도 절반 정도로 전력 효율이 더 좋고, 무엇보다 구축하는 데 드는 비용이 미니 PC보다 적다.

라즈베리파이5 준비물

엠브렐OS는 애초에 라즈베리파이에서도 충분히 돌아가도록 설계되었으므로 라즈베리파이를 사용해도 큰 문제가 없다. CPU보다는 램이 충분해야 한다. 라즈베리파이5는 램 16GB 버전도 출시되었으므로 엠브렐OS를 운영하기에 사양이 매우 괜찮다. 필자는 램 16GB 버전의 라즈베리파이5를 구매했다. (다음 사진 참고)



[정품] 라즈베리파이5 16GB Raspberry Pi 5 Model

오늘출발

196,000원

무료배송

무원전자 고객을 위한 혜택

최대 적립 포인트	23,670원 ?
기본적립	1,960원
내이버 천대카드 Ed으로 결제 시 >	13,720원
내이버캐시 미니결제 시 최대 적립 >	3,920원

멤버십 추가 적립 7,840원

최대 5% 적립 시작하기 >

저장 장치는 2TB는 필요하다. 풀 노드만 돌릴 용도라면 1TB여도 되지만, 현재 블록이 차는 속도로 봤을 때 1TB는 앞으로 3-5년 내로 꼭 찬다. 따라서 여유롭게 2TB는 마련하는 것을 추천한다. 저장 장치로 2TB SSD를 구매한다면 이를 라즈베리파이5에 연결하기 위한 SSD 쉴드가 필요하다. 단, 쉴드가 라즈베리파이5 케이스와 호환되는지 반드시 확인해야 한다. 케이스는 팬이 포함되어 있는 '일렉트로쿠키 컴팩트 알루미늄 방열 케이스'를 구매했다. SSD는 2TB NVMe 2280을 샀는데, 여기서 2280은 SSD의 길이를 말한다. 보통 SSD의 길이가 길수록 같은 용량 대비 가격이 저렴하다. SSD 쉴드는 2280 사이즈 SSD, 케이스 모두와 호환되는 X1001을 구매했다. (다음 사진 참고) 다른 케이스를 구매할 시 쉴드, SSD가 케이스와 호환되는지 꼭 확인하고 구매해야 한다.



라즈베리파이5 컴팩트 알루미늄 방열 케이스 [CP]

11%

25,900원 **22,900원**

배송비 2,800원

모집 Mollip 고객을 위한 혜택

최대 적립 포인트	2,898원 [?]
기본적립	229원
낸이버 현대카드 Edg으로 결제 시 >	1,603원
낸이퍼메이 미니 결제 시 최대 적립 >	457원
+ 멤버십 추가 적립	916원
최대 5% 적립 시작하기 >	

또한, SSD에 엠브렐OS를 설치하기 위해서는 NVMe SSD를 일반 컴퓨터와 연결해 주는 ‘인클로저’가 필요하다. SSD를 USB로 컴퓨터에 꽂아 외장 하드처럼 쓸 수 있게 해주는 어댑터라고 생각하면 편하다.



UNICORN

유니콘 NVMe M.2 SSD외장케이스 Gen2.10Gbps A+C 타입 SM-700P

14% 23,900원 **18,800원**
무료배송

유니콘스토어 고객을 위한 혜택

최대 적립 포인트	3,356원 ?
기본적립	188원
내이버 현대카드 Ed2로 결제 시 >	1,316원
내이버페이 미니 결제 시 최대적립 >	376원
멤버십 추가 적립	752원
최대 6% 적립 시작하기 >	

처음에 라즈베리파이5의 상태를 모니터로 보려면 모니터에 연결할 선이 필요하다. 그런데 라즈베리파이5에는 일반 HDMI선이 아니라 마이크로HDMI선을 꽂아야 한다. 따라서 마이크로HDMI선을 준비해야 한다. 필자는 이미 갖고 있는 걸 썼지만, 구매해야 하는 사람은 다음 사진을 참고하라.



코드웨이
 코드웨이 Micro HDMI to HDMI 2.0 케이블

★★★★★ 1,511 개 상품평

50% 23,500원

11,750원 (1개당 11,750원) 무료배송

- 로켓배송 상품 19,800원 이상 무료배송
- 무료배송 + 무료반품 로켓배송 신청시

길이 * 수량
 5m x 1개

적립 최대 588원 코발트신적립 · 구매이 미니 결제시 혜택보기 >

PC에서도 간편한 결제 위챗페이 카드 계좌이체

1 장바구니 담기 바로구매 >

• 출시 연도: 2021년도
 • 무량당문의: 6484972436 - 1421481719

라즈베리파이5가 초기 버전으로 출고되었다면 부팅을 NVMe SSD로 시도하지 않고 마이크로SD카드로만 시도한다. 우리는 SSD에 엠틀OS를 설치할 것이므로 NVMe SSD로 부팅을 시도해야 한다. 일반적으로 마이크로SD카드 슬롯이 비어 있다면 자동으로 NVMe SSD로 부팅을 시도하지만, 라즈베리파이5 초기 버전은 그렇지가 않다. 그래서 그냥 부팅이 안 되고 멈춰있는 경우가 있다. 따라서 처음에 부팅을 마이크로SD카드가 아닌 NVMe SSD를 통해 시도하게 하려면 부트로더를 업데이트해 주어야 한다. 이때 마이크로SD카드가 필요하다. 만약 한 번에 설치에 성공하면 마이크로SD카드는 필요 없지만, 비트코인과 관련된 하드웨어들을 이용할 때는 8GB의 마이크로SD카드가 많이 쓰이므로 여분의 마이크로SD카드는 장만해 두는 것을 추천한다.



샌디스크 MicroSDHC 8GB Class4 마이크로SD 메모리카드 

★★★★★ (2) | 구매 22 (5만원 이상 99,877개) 

4,930원

📄 이후 4시 종료 예정 📄

배송비 후원서 결제 (2,500원) 

스마일카드 최대 2% 캐시 적립 

신세계포인트 적립 

풀 노드는 안정적인 전원 공급이 매우 중요하다. 라즈베리파이5는 5V, 5A의 전력을 필요로 하므로, 정확한 전압의 어댑터를 준비하는 것이 좋다. 연결부는 C타입이다. (다음 사진 참고) 이 외에 라즈베리파이5 기기와 공유기를 연결할 랜선도 필요하다.



라즈베리파이5 전용 전원 어댑터 5V 5A 오늘출발
34% 15,000원 **9,900원**
배송비 2,800원

모집 Molip 고객을 위한 혜택

최대 적립 포인트	1,338원 ?
↳ 기본적립	99원
↳ 내인바 확대카드 E02로 결제 시 >	693원
↳ 내인바페이머니 결제 시 최대적립 >	197원
▶ 멤버십 추가 적립	396원
최대 8% 적립 시작하기 >	

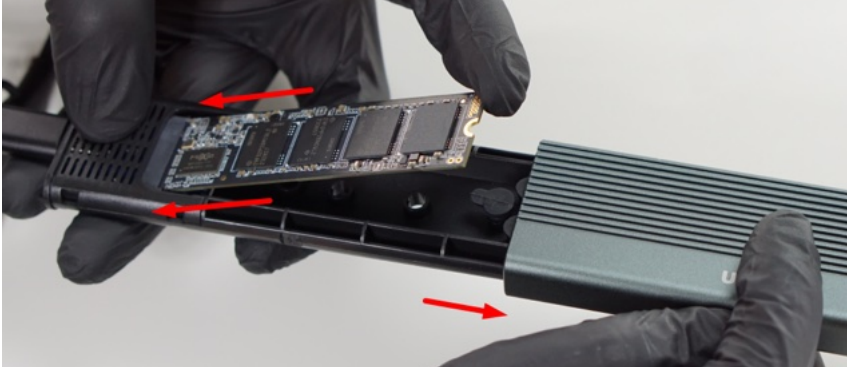
종합하면 준비물은 다음과 같다.

1. 라즈베리파이5 (RAM 16GB)
2. SSD 쉴드를 장착할 수 있는 라즈베리파이5 케이스(팬 포함)
3. NVMe SSD 2TB (케이스와 호환되는 길이 선택)
4. SSD 쉴드(케이스, SSD 길이와 호환되는 제품 선택)
5. NVMe SSD를 컴퓨터에 연결할 수 있게 해줄 인클로저
6. 처음에 엠브렐OS 세팅을 위해 사용할 컴퓨터 모니터와 모니터에 연결할 마이크로HDMI선
7. 라즈베리파이5 전용 5V 5A C타입 어댑터
8. 인터넷 연결할 랜선
9. 십자 드라이버



SSD에 엠브렐OS 설치하기

부품이 모두 준비되었다면 엠브렐OS 설치부터 시작할 것이다. 먼저 NVMe SSD를 SSD 인클로저에 연결해 주고, 이것을 컴퓨터에 꽂는다.

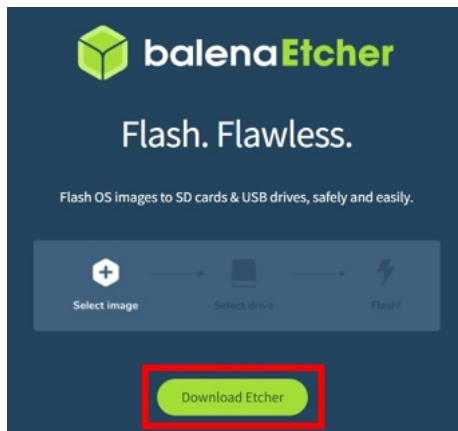


이제 SSD에 OS를 설치하기 위해 발레나에서 프로그램을 다운로드해야 한다. 아래 링크에서 발레나에서 프로그램을 다운로드한다.

<https://etcher.balena.io/>



[Download Etcher]를 누른다.



자신의 컴퓨터 운영체제에 맞는 버전을 다운로드한다.

DOWNLOAD

Download Etcher

ASSET	OS	ARCH	
ETCHER FOR WINDOWS (X86 X64) (INSTALLER)	WINDOWS	X86 X64	Download
ETCHER FOR MACOS	MACOS	X64	Download
ETCHER FOR MACOS (ARM64)	MACOS	ARM64	Download
ETCHER FOR LINUX X64 (64-BIT) (ZIP)	LINUX	X64	Download
ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)	LINUX	X86	Download

설치가 완료되었으면 발레나에처를 실행한다.



이제 깃허브에서 움브렐OS의 이미지 파일을 다운로드할 것이다. 아래 링크에 접속한다.

<https://github.com/getumbrel/umbrel/wiki/Install-umbrelOS-on-a-Raspberry-Pi-5>



스크롤을 내리고 [umbrelOS image]를 누르면 압축된 이미지 파일이 다운로드된다.

Installing umbrelOS on the NVMe or USB drive

In this configuration, you will install umbrelOS directly on an NVMe drive or USB drive, which will be automatically partitioned to store umbrelOS and your data.

Note

Booting from NVMe or USB instead of a microSD card is only supported on Raspberry Pi 5 for umbrelOS 1.2.0 and later. Booting from NVMe or USB is not supported on Raspberry Pi 4. Additionally, certain NVMe drives may not be compatible with specific NVMe HATs. It is crucial to check the compatibility of your NVMe drive with the HAT you are using before proceeding. HAT manufacturers typically provide lists of compatible drives and known quality issues on their websites. Using a drive not on the compatible list may result in instability or failure to boot. Also, make sure to check the quality of your NVMe SSD, as some drives are of poor quality or have reported issues that could affect performance and reliability.

What you will need

- Raspberry Pi 5
- Official Raspberry Pi power supply
- Ethernet cable
- Either an NVMe drive + HAT or USB drive (depending on your preference)

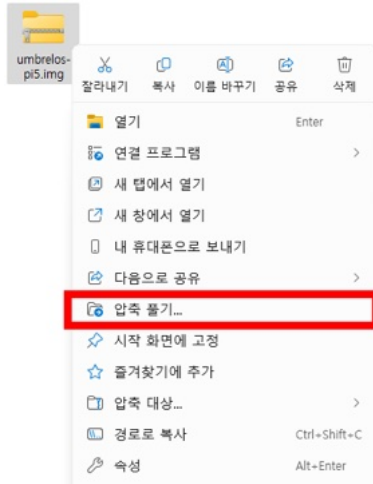
Steps

Tip

If you have an early Raspberry Pi 5 model, you may need to update the bootloader firmware in order to be able to boot from NVMe. If this is the case, please follow the official Raspberry Pi documentation on how to update the bootloader firmware: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#bootloader_update_stable

1. Download the latest **umbrelOS image** for Raspberry Pi 5.
2. Download [Balena Etcher](#) on your computer. It is required to flash the umbrelOS image that you downloaded in the previous step to your NVMe or USB drive.

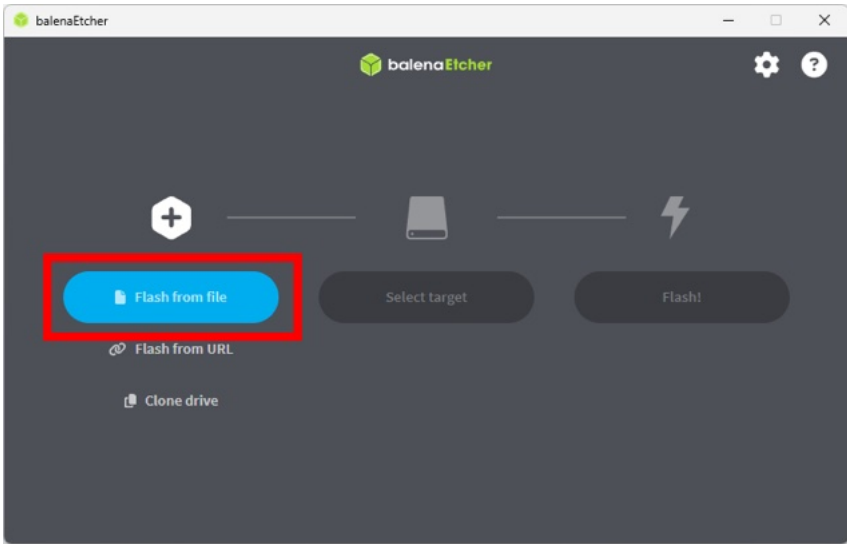
다운로드가 완료되었으면 압축을 푼다.



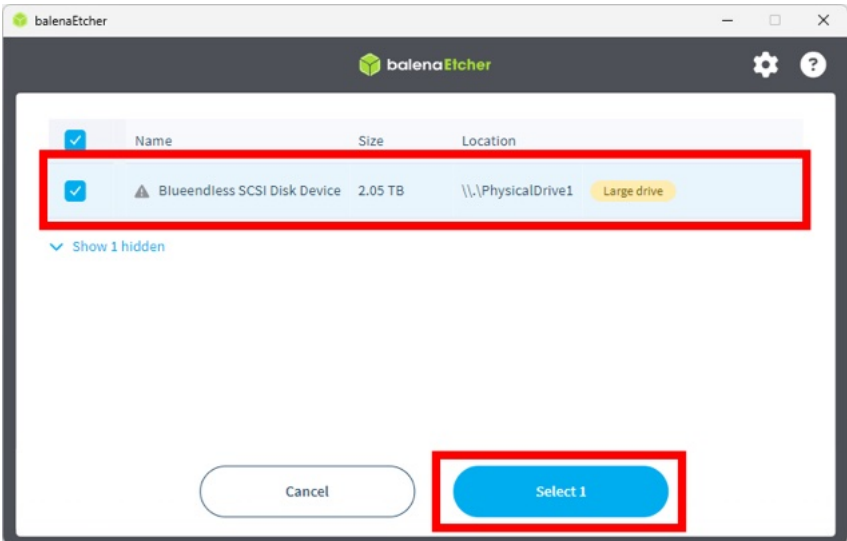
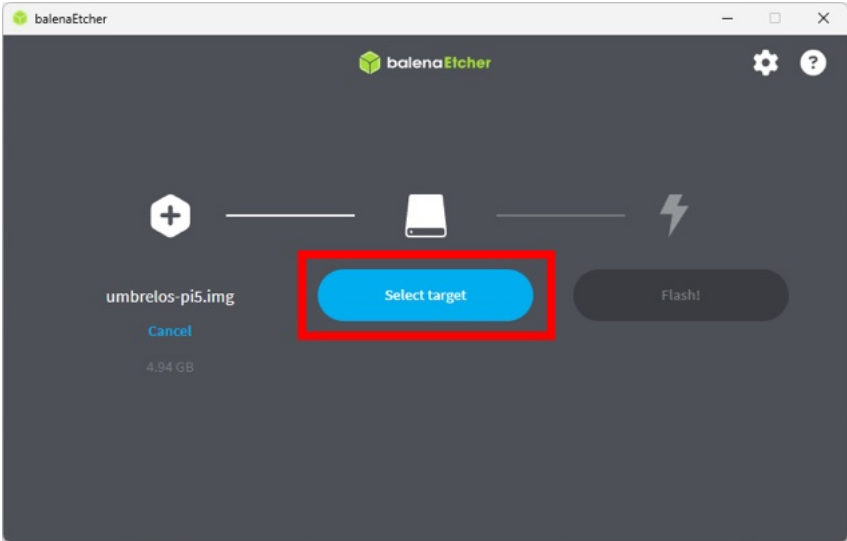
압축을 풀면 다음과 같은 이미지 파일이 나온다.



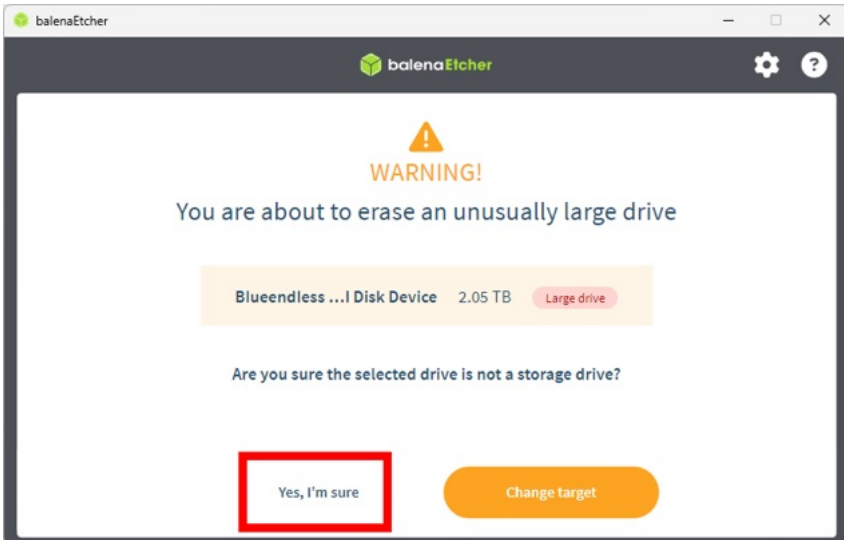
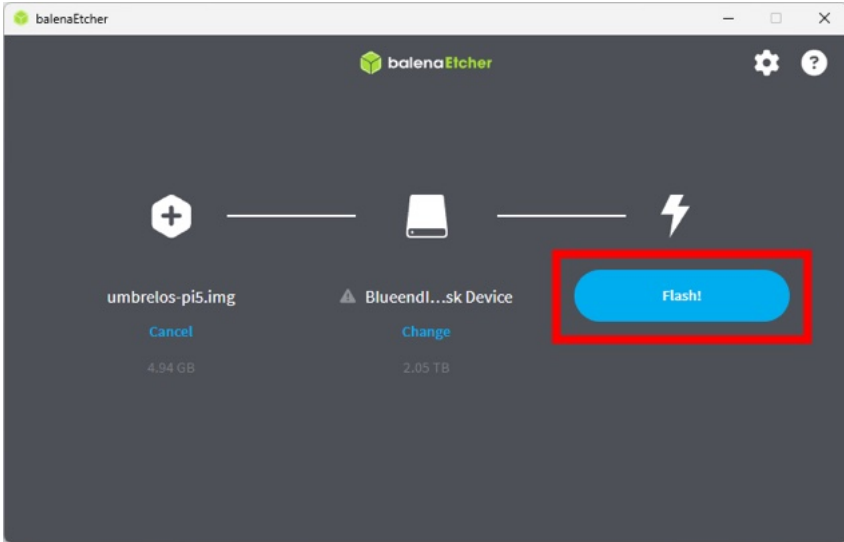
이제 발레나에처 프로그램에서 [Flash from file]을 누르고, 방금 압축을 풀어서 나온 파일을 선택한다.



[Select target]을 누르고, 연결한 NVMe SSD 저장장치를 선택한다.
그다음 [Select ?]을 누른다.



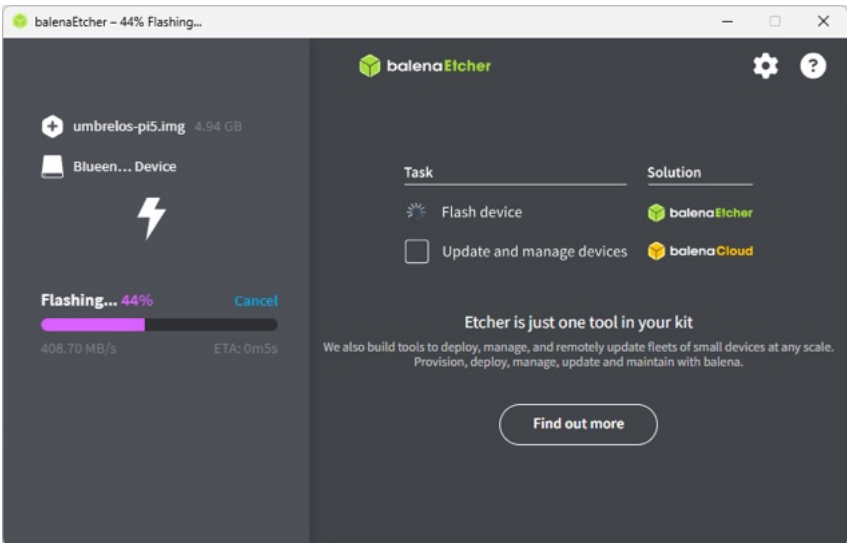
이제 [Flash!]를 누른다. 만약 너무 큰 저장장치를 골라서 정말 계속 진행할 건지 경고하는 창이 나오면 [Yes, I'm sure]를 눌러 설치를 시작한다.

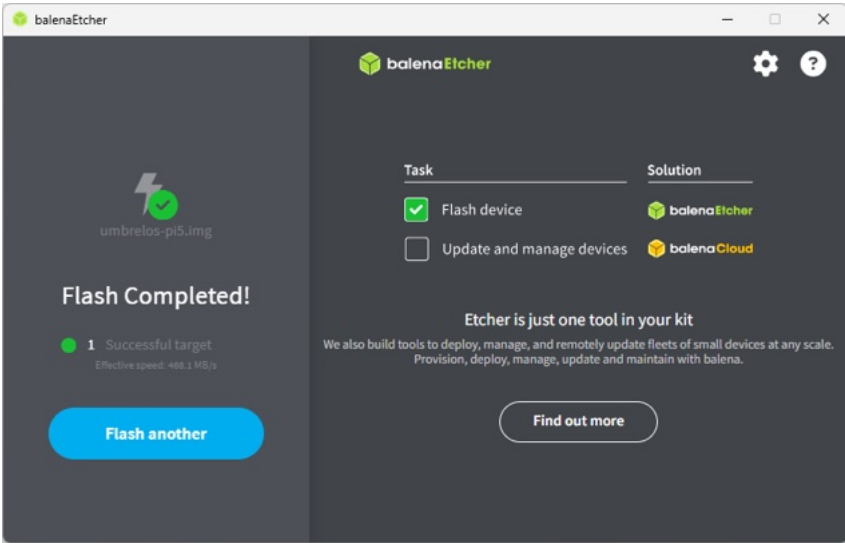


다음과 같은 화면이 뜨다가 'Flash Completed!'가 뜰 것이다. 뜨면 오른쪽 위 [x] 버튼을 누르고 SSD가 연결된 USB를 빼면 된다. 중간에 포맷하라는 알림창이 뜨면 무시하고 취소하면 된다.

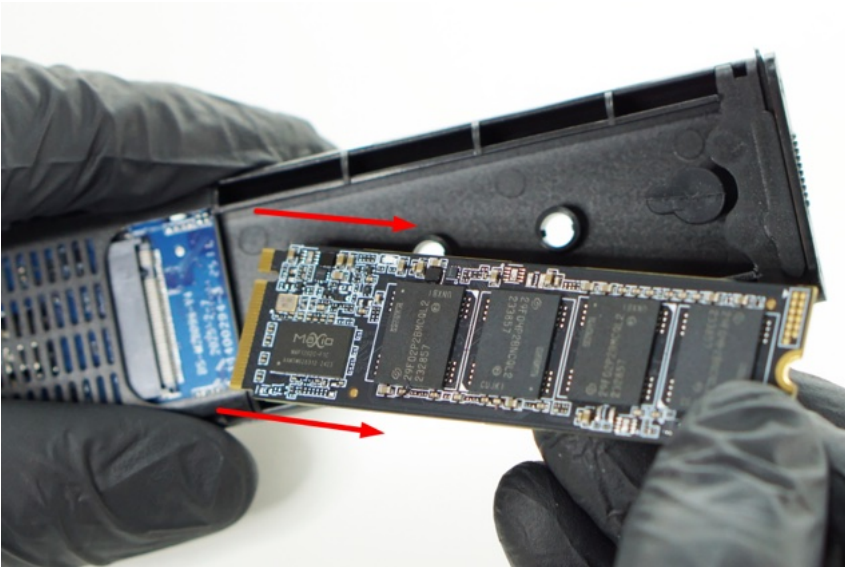
만약 Flash에 실패하면 SSD를 초기화하고 다시 플래싱해야 한다. 이때 일반적인 포맷 방법으로는 안 되는데, SSD 초기화 방법으로 검색하면 결과가 많이 나온다.

윈도우의 경우 윈도우 키 + R → cmd 입력 후 실행 → 터미널이 나오면 `diskpart` 입력 → `list disk` → SSD 디스크 번호 확인 → `select disk [디스크 번호]` → `clean`을 입력하면 된다. 그리고 다시 포맷하고 처음부터 진행하면 된다.



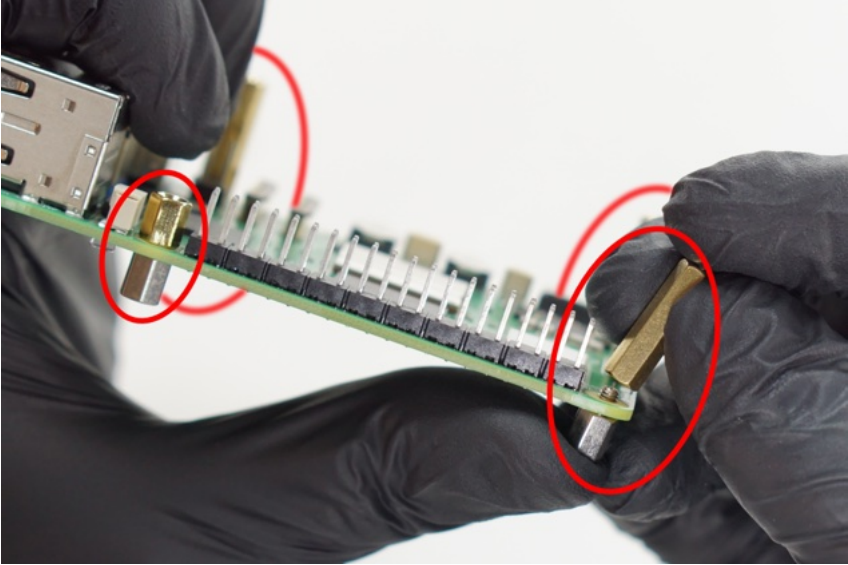


다 되었으면 컴퓨터에서 인클로저 USB를 빼고, 인클로저에서 SSD도 뺀다.

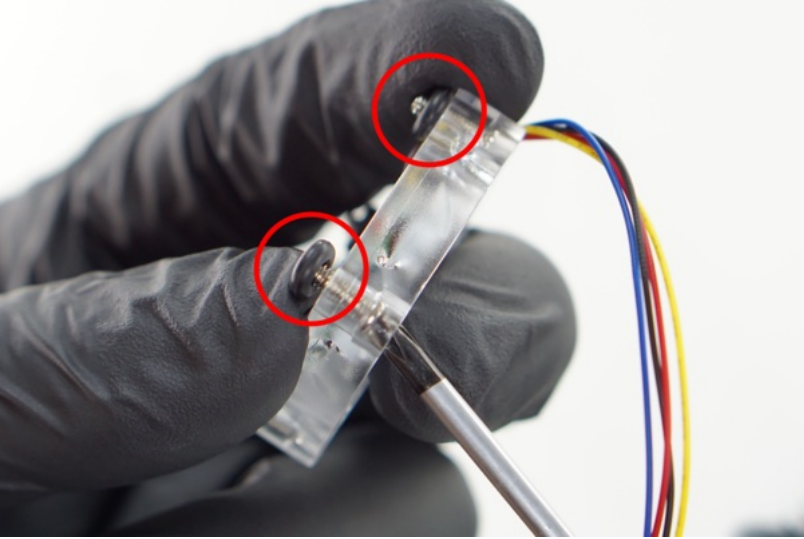


라즈베리파이5 조립

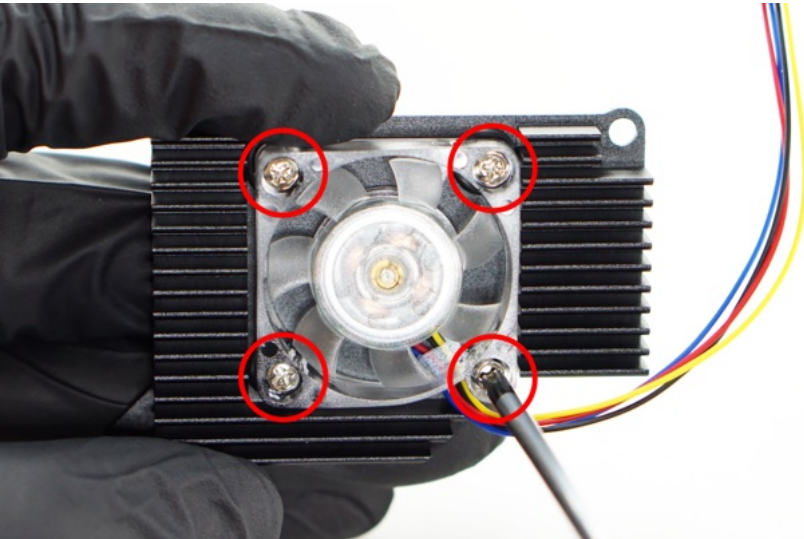
이제 라즈베리파이5 조립을 해보자. 먼저 육각 스페이서 나사를 조립한다. 아래는 은색 육각 나사, 위에는 금색 육각 나사를 조립한다. 이때, 랜선 연결 포트가 왼쪽에 오도록 위쪽에서 바라봤을 때 왼쪽 아래는 짧은 스페이서를, 나머지 3개는 긴 스페이서를 끼워야 한다.



이제 팬을 히트싱크 판에 연결해 주어야 한다. 먼저 팬에 나사를 넣고, 고무 와셔를 끼운다.



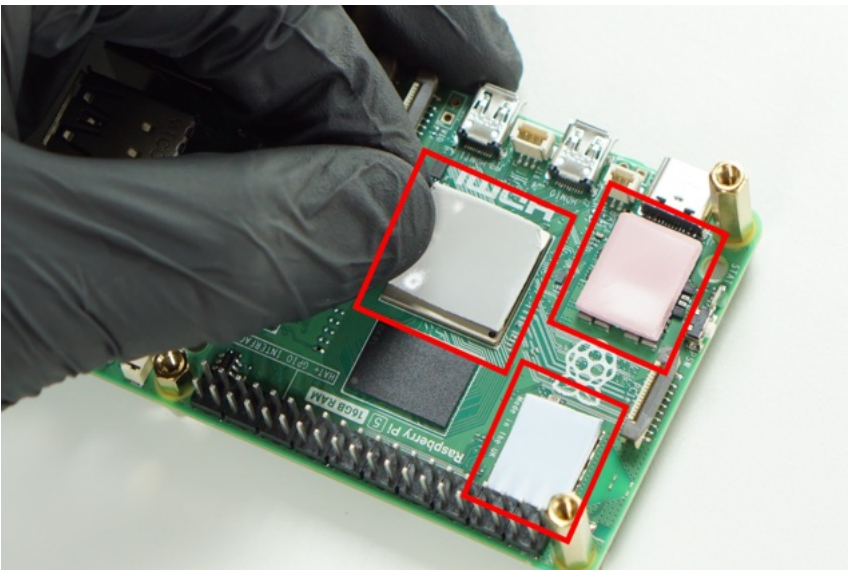
팬과 히트싱크 판을 연결한다.



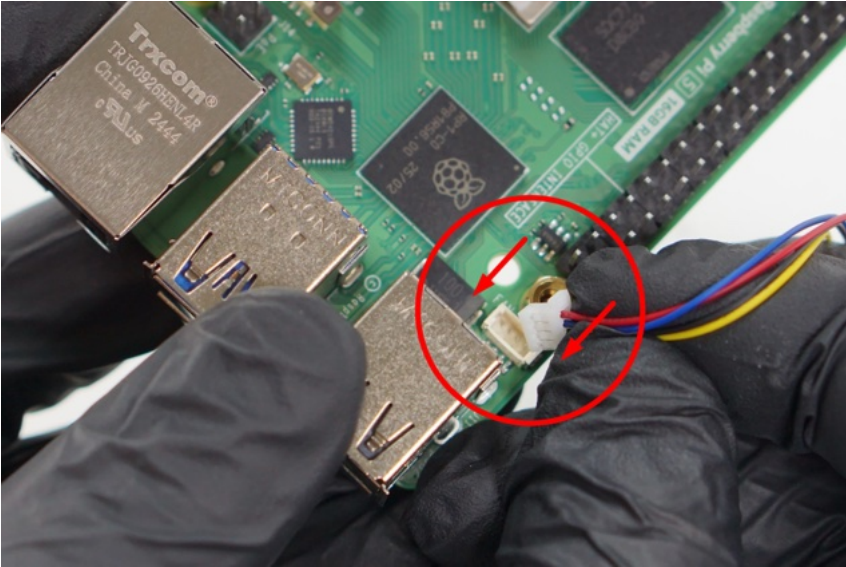
팬이 결합된 히트싱크 팬에 푸시 핀 2개를 끼워 넣는다.



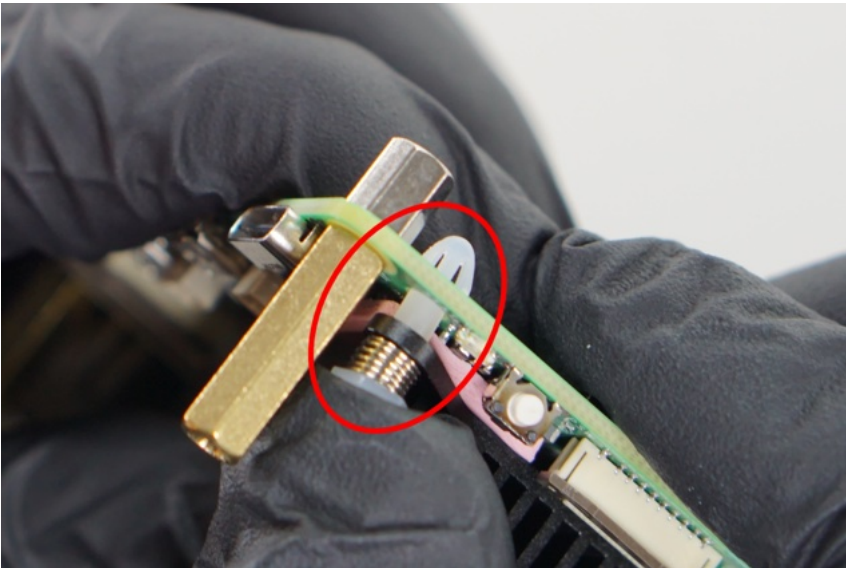
각 부품에 맞는 썬열패드를 붙인다.



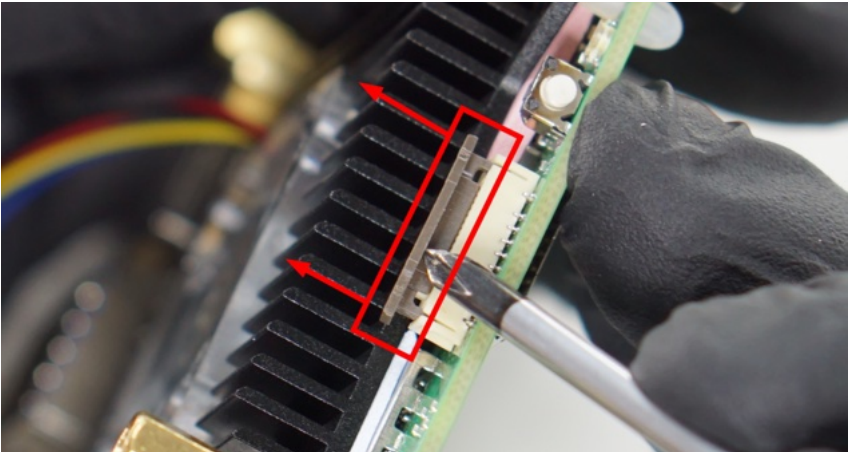
팬의 전원선을 라즈베리파이5 보드에 연결한다.



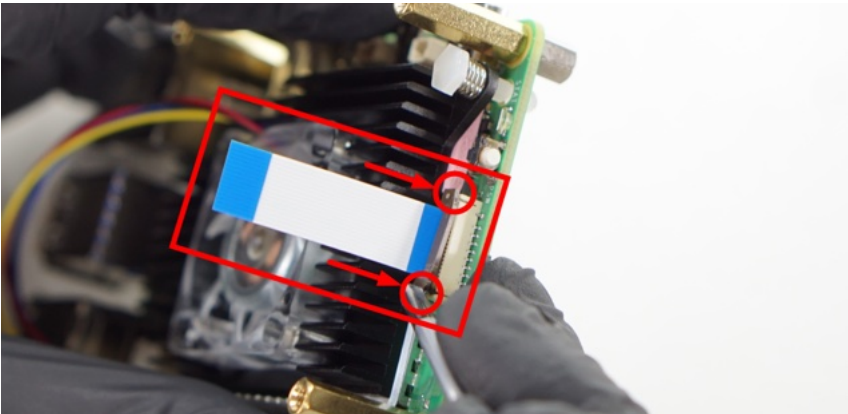
푸시 핀을 눌러 히트싱크 판과 라즈베리파이5 보드를 결합한다.



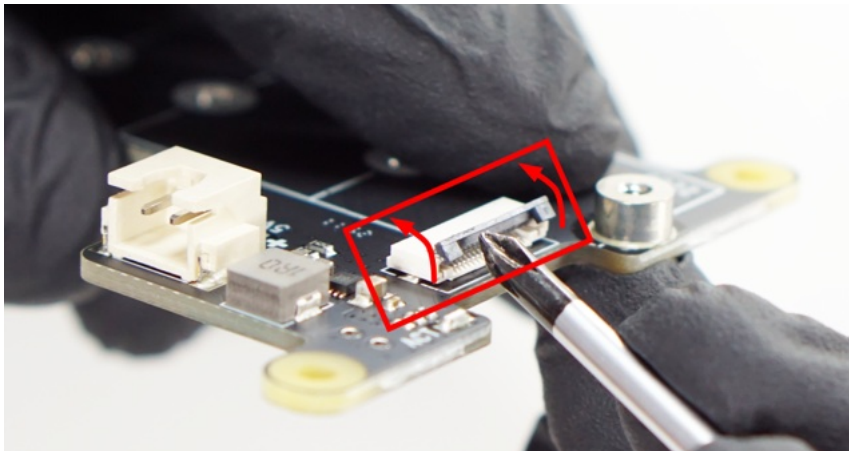
이제 SSD 쉴드와 라즈베리파이5 보드를 연결할 것이다. 먼저 케이블을 꽂을 포트를 열어준다. 검은색 덮개를 살짝만 위로 밀어주면 된다.



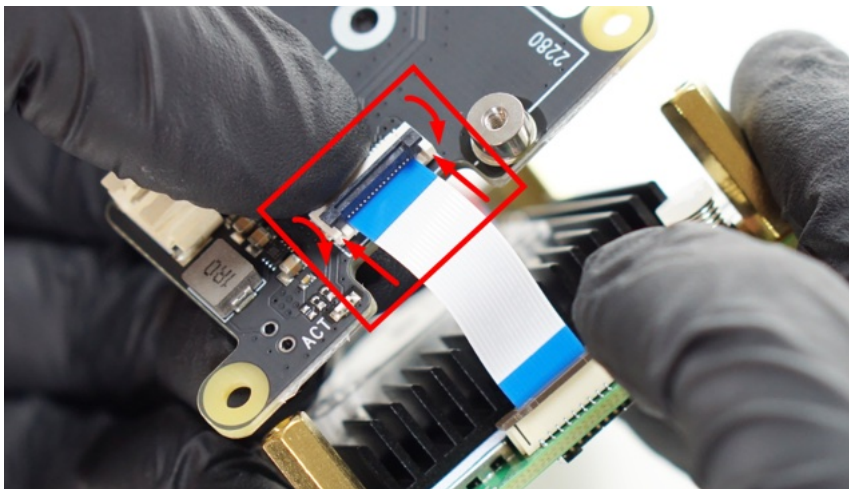
FFC 케이블을 끼운다. 이때 단자가 안쪽으로 오고, 파란색 부분이 바깥쪽으로 가도록 방향에 유의해서 넣는다. 넣고 나면 검은색 덮개를 다시 밀어서 끼워준다.



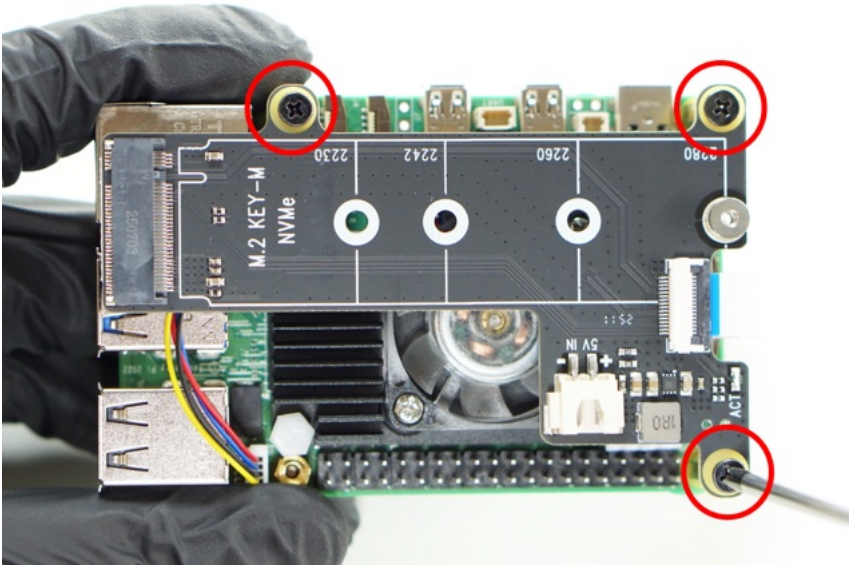
SSD 쉘드에 케이블을 꽂을 포트도 열어준다. 검정 덮개를 위로 밀어 열어주면 된다.



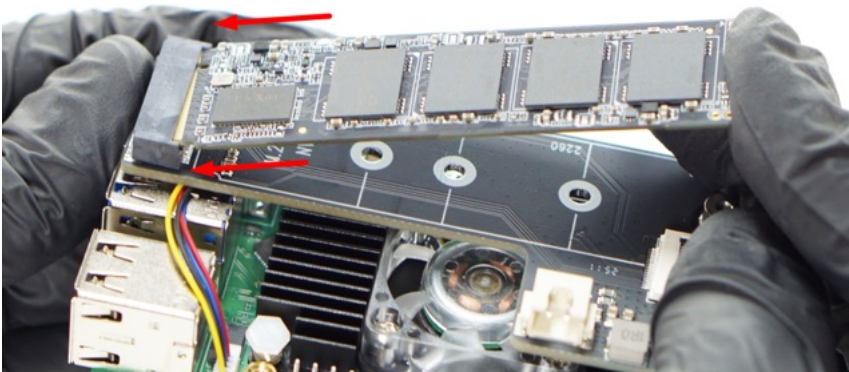
사진과 같이 파란색이 위로 오도록 케이블을 넣고 검은색 덮개를 눌러 끼워준다.



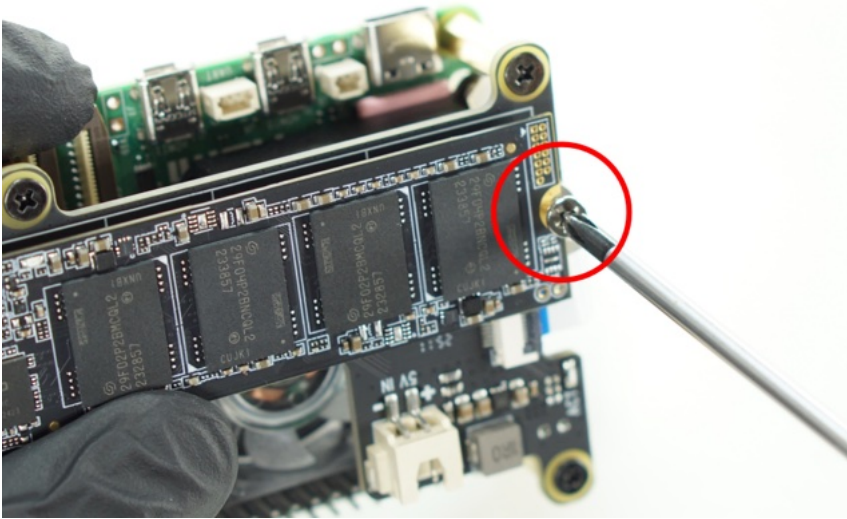
SSD 쉴드의 구멍을 스페이서 나사 구멍에 잘 맞추고 나사를 돌려 끼워준다.



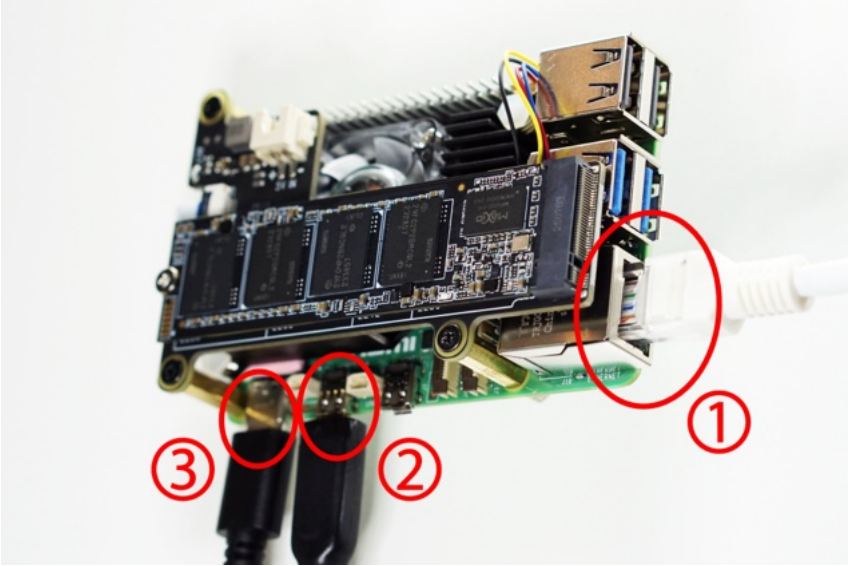
NVMe SSD를 SSD 쉴드에 끼워줄 것이다. SSD를 비스듬한 방향으로 세워 밀어넣는다.



나사를 조여 SSD를 쉘드에 고정한다.



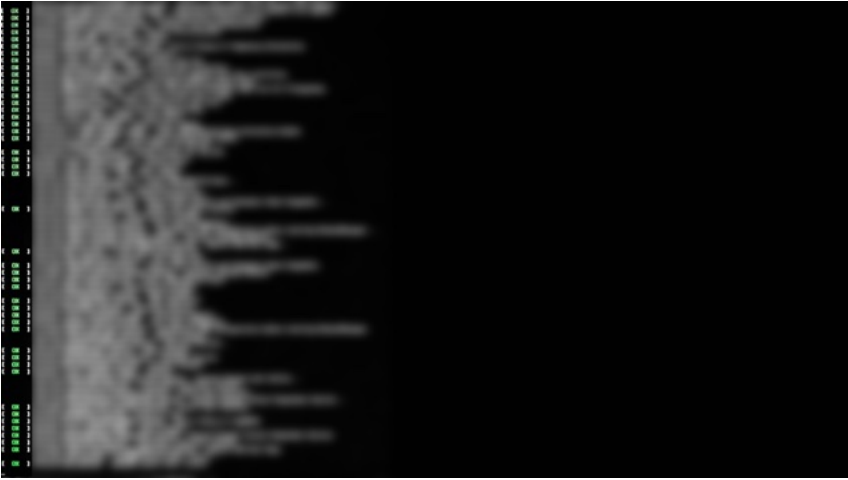
이제 기본적인 조립은 끝났고, 케이스 조립만 남았다. 케이스를 조립하기 전에 잘 구동되는지 확인하고 넘어갈 것이다. ① 랜선으로 공유기와 라즈베리파이5 기기를 연결하고, ② 마이크로HDMI선으로 모니터와 연결한다. 연결했으면 ③ 전원선을 연결하고 콘센트에 꽂는다.



부팅이 되면 이제 다음과 같은 화면이 보여야 한다. 참고로 라즈베리 파이5의 팬은 CPU 온도가 올라가면 가동되고, 온도가 낮을 때는 가동되지 않기도 한다. 만약 다음과 같은 화면이 안 나오고 부팅이 안 된다면 다음 절을 참고하라.



화면에 무언가 많이 지나갈 것이다.



다 지나가고 나면 `Your Umbrel is now accessible at:`이라는 문구가 뜰 것이다. 그 밑에 URL 두 개가 뜨는데 이 URL을 잘 기억하자. `http://192.168.0.???` 이것이 엄브렐 기기의 로컬 IP 주소다. 여기까지 완료되었다면 ‘케이스 조립’ 절로 넘어가면 된다.

```
Your Umbrel is now accessible at:  
http://umbrel.local  
http://192.168.0.???  
umbrel login: _
```

부팅이 안 될 경우

만약 발레나에처에서 성공적으로 플래싱이 되었다고 뻤는데도 부팅이 안 된다면 부트로더 문제일 가능성이 크다. 이때는 SSD로 부팅을 시도하도록 마이크로SD카드를 이용해 설정을 바꿔줘야 한다. 만약 부팅이 잘 되었다면 이 절을 진행할 필요는 없다.

컴퓨터에 마이크로SD카드를 꽂는다(포맷이 안 되어 있다면 FAT32 형식으로 포맷한다).



라즈베리파이 이미지를 다운로드해야 한다. 아래 웹사이트에 접속하여 스크롤을 내리고, 자신의 운영체제에 맞는 라즈베리파이 이미지를 다운로드한다.

<https://www.raspberrypi.com/software>



Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

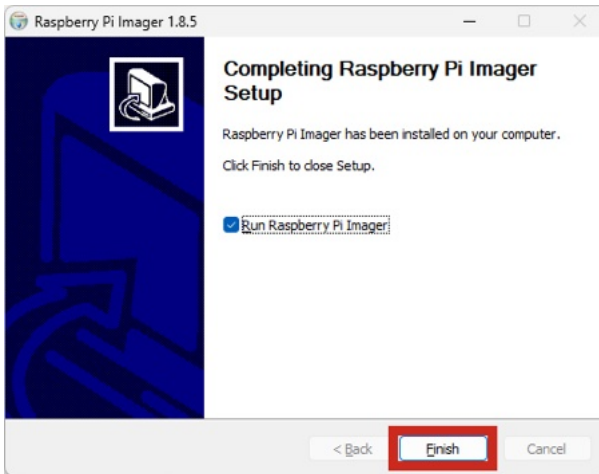
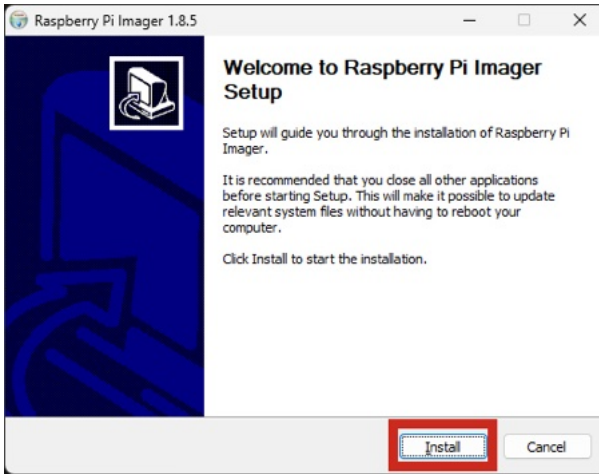
[Download for macOS](#)

[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.

A screenshot of the Raspberry Pi Imager v1.6.1 application window. The window has a red header with the Raspberry Pi logo and the text 'Raspberry Pi'. Below the header, there are three tabs: 'Raspberry Pi OS', 'Operating System', and 'Image'. Under the 'Raspberry Pi OS' tab, there are three buttons: 'CHOOSE DEVICE', 'CHOOSE OS', and 'CHOOSE STORAGE'. A 'WRITE' button is visible at the bottom right of the window.

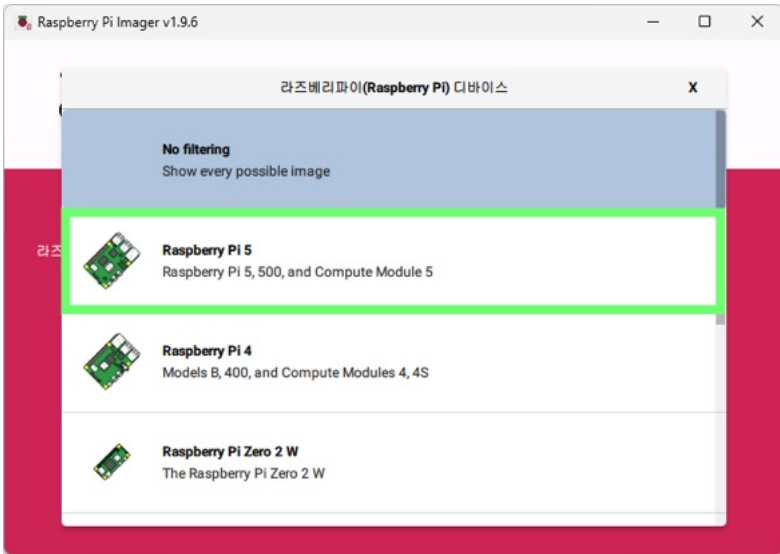
다운로드가 완료되면 파일을 실행하여 라즈베리파이 이미지를 설치한다.



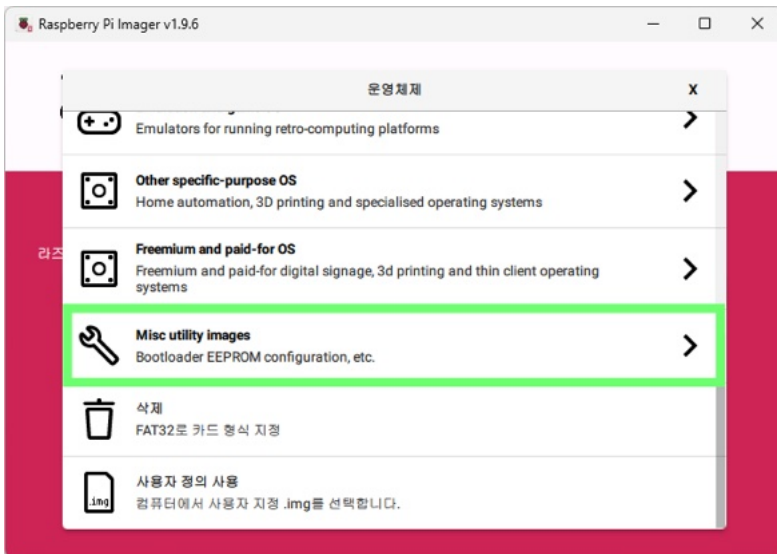
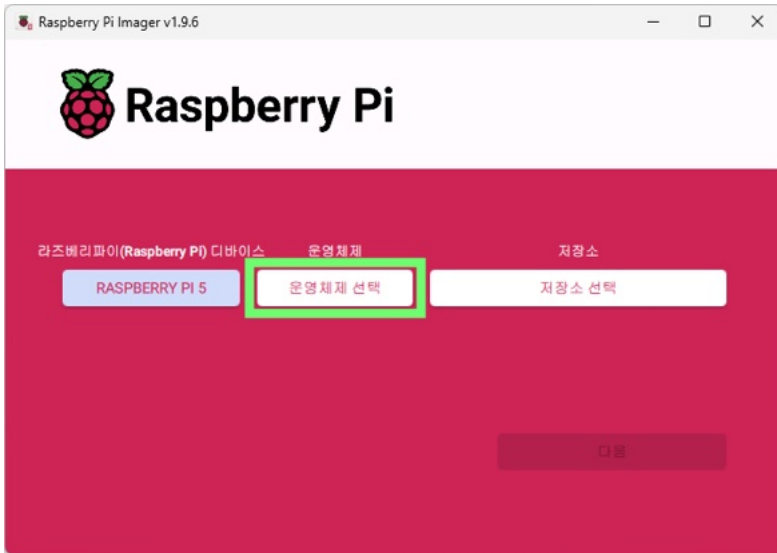
라즈베리파이 이미지를 실행한다. 먼저 [장치 선택]을 누른다.



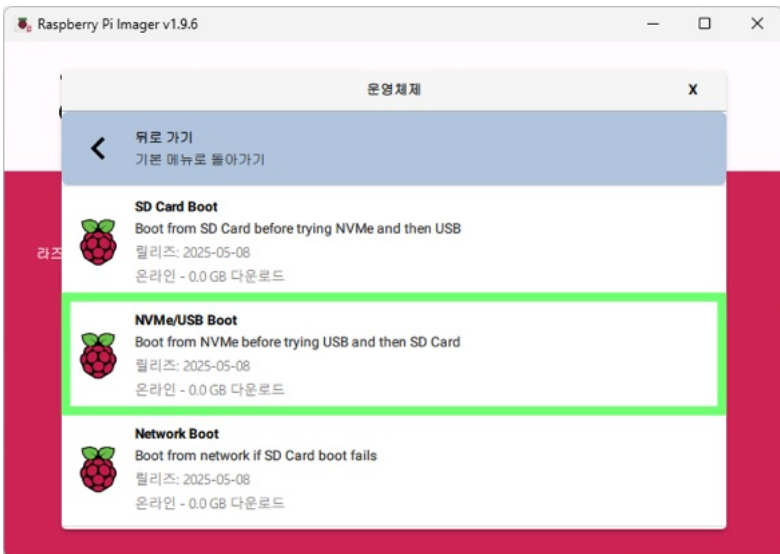
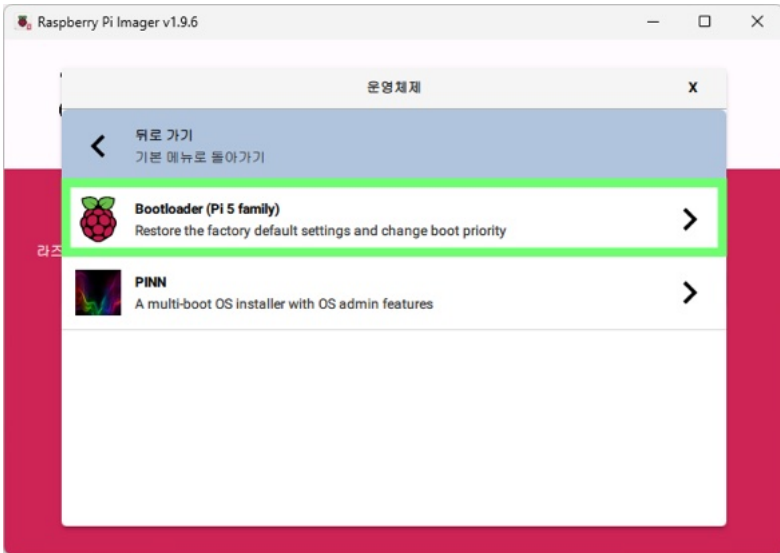
장치에서 [Raspberry Pi 5]를 선택한다.



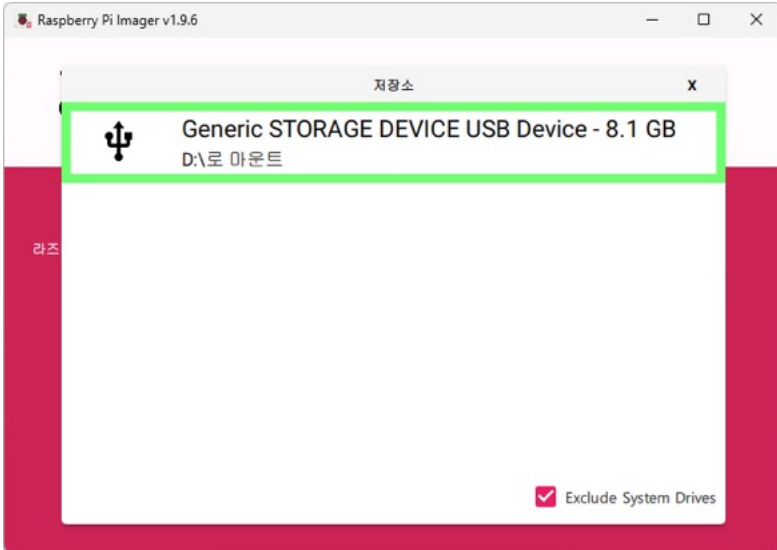
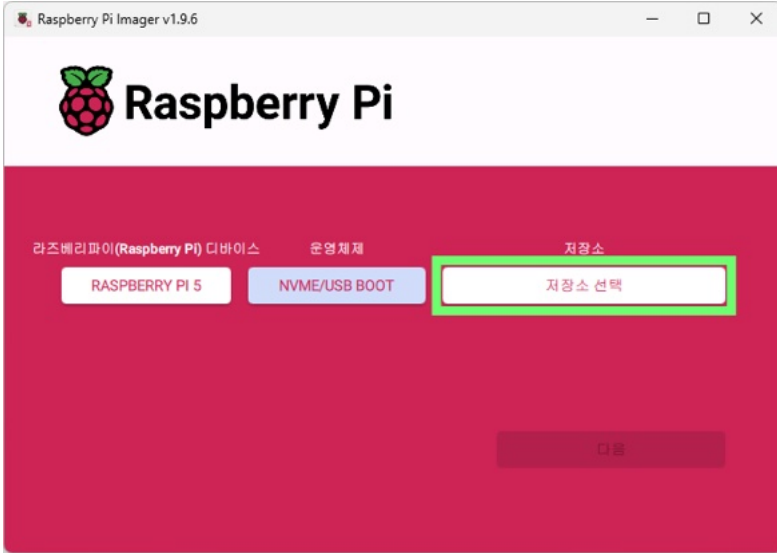
[운영체제 선택]을 누르고 스크롤을 아래로 내려 [Misc utility images]를 누른다.



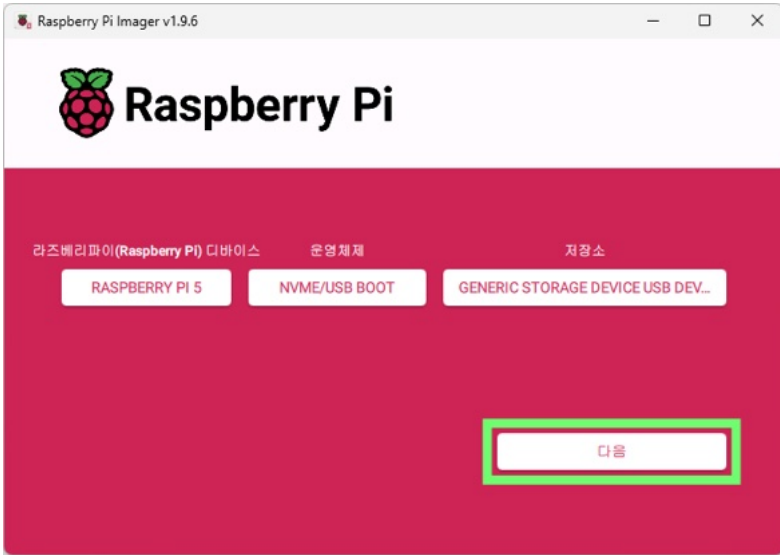
[Bootloader (Pi 5 family)]를 선택하고, [NVMe/USB Boot]를 선택한다.



[저장소 선택]을 누르고, 방금 꼽은 마이크로SD카드를 선택한다.



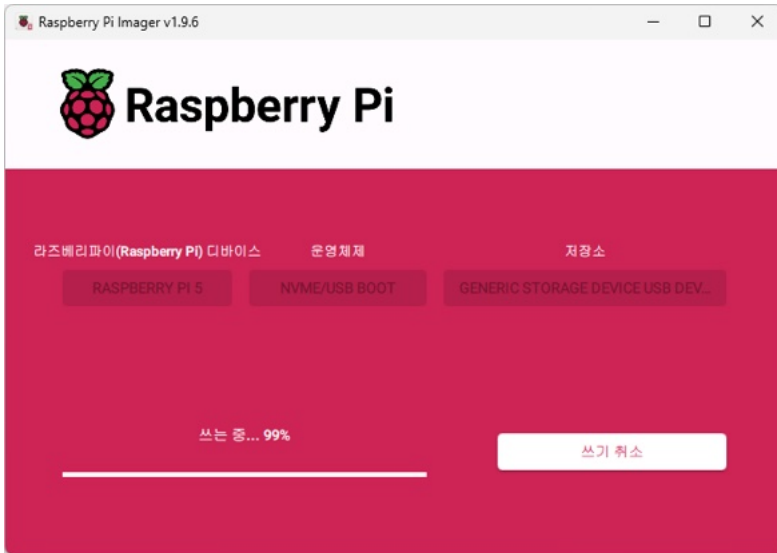
다 되었으면 [다음]을 누른다.



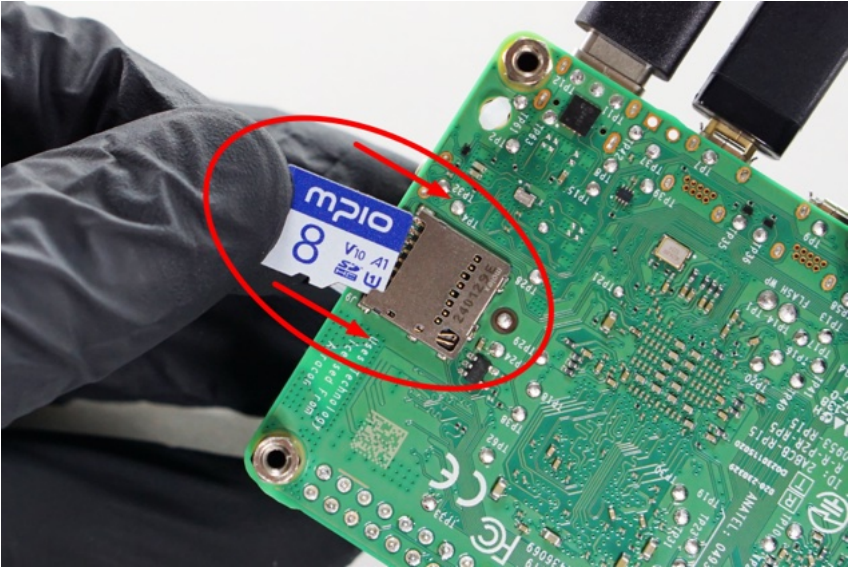
데이터가 지워질 거라는 안내문이 나오면 [예]를 누른다.



이제 SSD로 부팅을 시도하도록 만드는 마이크로SD카드가 만들어졌다. 컴퓨터에서 마이크로SD카드를 분리한다.



이 마이크로SD카드를 라즈베리파이5 보드 뒷면에 있는 포트에 꽂고 부팅을 하면 부팅이 잘 될 것이다. 이 과정은 한 번만 거치면 되므로, 다음부터는 마이크로SD카드를 제거하고 부팅해도 자동으로 SSD를 이용해 부팅을 시도하게 된다.



케이스 조립

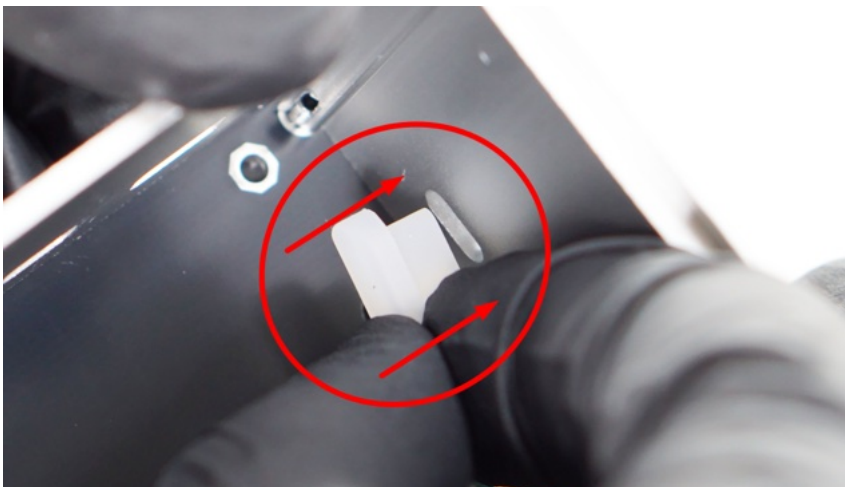
부팅이 잘 되고 `Your Umbrel is now accessible at:`이라는 문구가 떴다면, 이제 케이스를 조립해야 한다. 먼저 파워 버튼을 눌러 라즈베리 파이5를 종료하고, 연결된 선을 모두 뽑는다. 파워 버튼은 다음 사진과 같은 위치에 있다.



케이스를 조립하자. 밑판을 먼저 나사로 조립해 준다.



파워 버튼을 조립해 준다.



조립했던 라즈베리파이5를 넣어준다. 파워 버튼이 잘 맞닿게 비스듬히 넣어주고, 각 구멍들이 포트와 잘 맞도록 위치를 조정한다.



위치를 조정했으면 밑판과 라즈베리파이5 보드를 나사로 연결해 준다.



하단에 고무 받침대를 붙여준다.



상판을 나사로 연결해 준다.



조립이 다 되었다. 랜선을 꽂고, 전원선을 연결하면 된다. 지금부터는 모니터를 연결할 필요가 없다. 여기까지 잘 따라왔다면 이제 ‘엠프렐 설정 및 풀 노드 동기화’ 장으로 넘어가면 된다.



| 노트북에 엠브렐OS 설치하기

남는 노트북에 엠브렐OS 설치하기

남는 노트북이나 데스크톱이 있다면 여기에 엠브렐OS를 설치할 수도 있다. 풀 노드는 보통 24시간 가동되는 것을 전제로 하기 때문에 남는 노트북이나 데스크톱을 이용해 홈 서버를 구축한다면 저전력 미니 PC 나 라즈베리파이5보다는 전기료가 많이 나올 것이다. 그러나 풀 노드를 장만하는 것에 비유적인 부담을 느끼는데 남는 노트북, 데스크톱 등이 있다면 하나의 옵션으로 고려할 수 있다.

모든 블록 데이터를 저장하려면 노트북에 2TB의 저장 용량이 필요하지만, 저장 용량이 그것보다 부족하다면 프루닝(가지치기)을 할 수도 있다. 프루닝 노드는 제네시스 블록부터 이어지는 모든 블록을 검증하지만, 일정 용량 이상이 되면 오래된 블록 데이터를 삭제하는 노드다. 프루닝 노드는 UTXO 세트를 풀 노드와 똑같이 보관하기 때문에 풀 노드 처럼 블록이나 거래를 검증할 수 있다.

준비물은 OS 설치에 필요한 USB 메모리카드만 있으면 된다. 여기서 8GB 용량의 SD카드를 이용해 설치했다.

노트북은 유선 인터넷에 연결되어야 한다. 처음에 엠브렐OS를 설치하고 나면 와이파이를 자동으로 잡지 못하기 때문이다. 노트북에 랜선을 꽂는 곳이 없다면 랜선을 USB로 바꿔주는 어댑터도 있으니 그러한 어댑터를 구매해야 한다.

노트북에 엠브렐OS를 설치하고 나면 노트북은 풀 노드 용도로만 쓸 수 있으며, 기존에 노트북에 저장되어 있던 파일들은 모두 사라진다. 따

라서 노트북에 엠브렐OS를 설치하기 전에 노트북에 있는 중요한 파일들을 다른 저장 장치에 옮겨놓는 등의 백업을 하자.

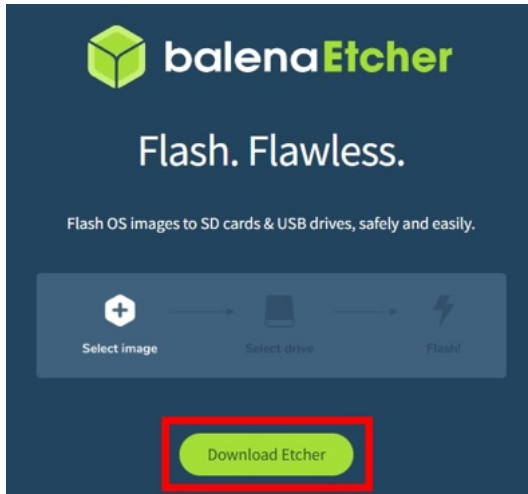
OS 설치용 USB 만들기

먼저 USB 저장 장치를 OS 설치용으로 만들기 위해 발레나에처로 구워야 한다. 아래 링크에서 발레나에처 프로그램을 다운로드한다.

<https://etcher.balena.io/>



[Download Etcher]를 누른다.



자신의 컴퓨터 운영체제에 맞는 버전을 다운로드한다.

DOWNLOAD

Download Etcher

ASSET	OS	ARCH	
ETCHER FOR WINDOWS (X86 X64) (INSTALLER)	WINDOWS	X86 X64	Download
ETCHER FOR MACOS	MACOS	X64	Download
ETCHER FOR MACOS (ARM64)	MACOS	ARM64	Download
ETCHER FOR LINUX X64 (64-BIT) (ZIP)	LINUX	X64	Download
ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)	LINUX	X86	Download

설치가 완료되었으면 발레나에처를 실행한다.



balenaEtcher-1.1
9.25.Setup

이제 깃허브에서 움브렐OS iso 파일을 다운로드할 것이다. 아래 링크에 접속한다.

<https://github.com/getumbrel/umbrel/wiki/install-umbrelOS-on-x86-Systems>



[latest umbrelOS USB installer]를 누르면 iso 파일이 다운로드된다.

Install umbrelOS on x86 Systems

Luke Childs edited this page on Aug 1, 2024 - [3 revisions](#)

Note

umbrelOS is specifically built for [Umbrel Home](#). Support for other devices is best-effort and not guaranteed. Compatibility issues may arise due to the differences in hardware, drivers, and such.

Minimum System Requirements

- **CPU:** Dual-core 64-bit Intel or AMD processor. (Quad-core or higher recommended)
- **RAM:** 4GB. (8GB+ recommended)
- **Storage:** 32GB. (Storage needs depend on your use case. For instance, running a Bitcoin node typically requires 1TB or more, while file/media storage needs can be estimated based on your cloud services usage.)

How to install

Step 1: Prepare a bootable USB stick

- Download the [latest umbrelOS USB installer](#) on your computer.
- Download and install [Balena Etcher](#) on your computer.
- Plug a USB stick (at least 4GB) into your computer. Ensure that it doesn't contain any important data as it will be formatted.
- Use Balena Etcher to flash the `umbrelOS-amd64-usb-installer.iso` file to the USB stick.

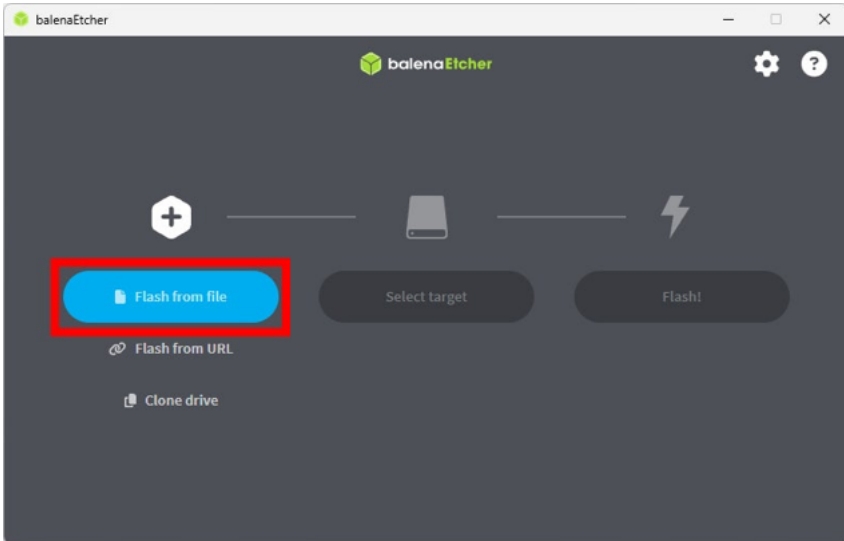


umbrelOS-amd64-usb-installer.iso

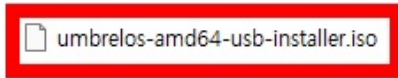
컴퓨터에 USB 저장 장치를 꽂는다(USB 포맷이 안 되어 있다면 FAT32 형식으로 포맷한다).



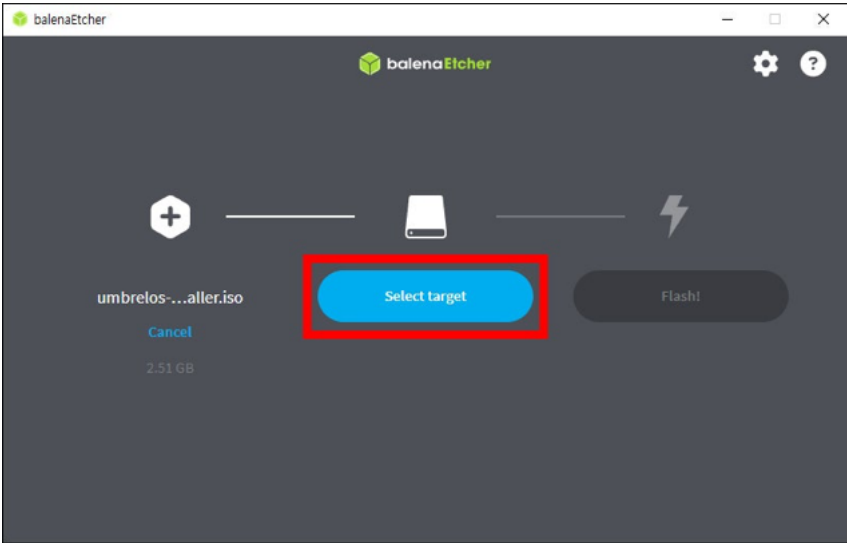
발레나에처를 실행하면 다음과 같은 화면이 나온다. [Flash from file]을 누른다.



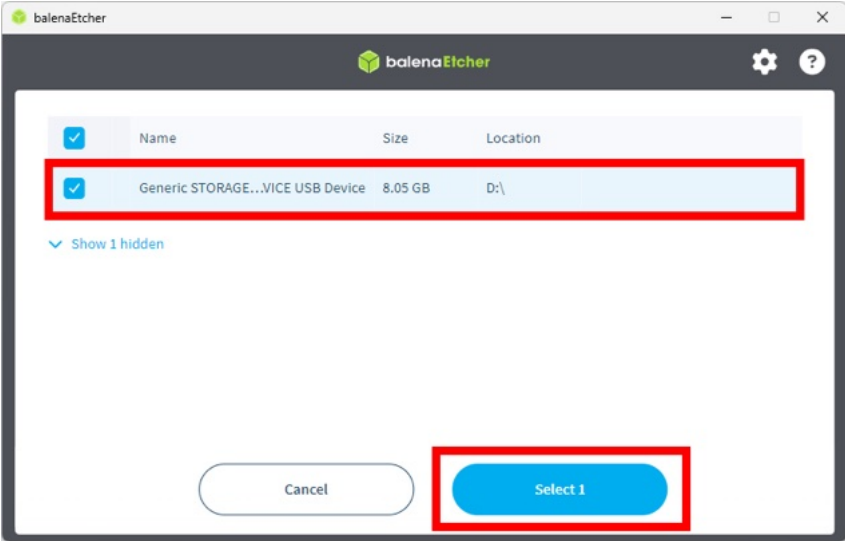
iso 파일을 선택하는 창이 뜰 것이다. 아까 다운로드했던 엮브렐OS의 iso 파일을 찾아서 더블 클릭하자.



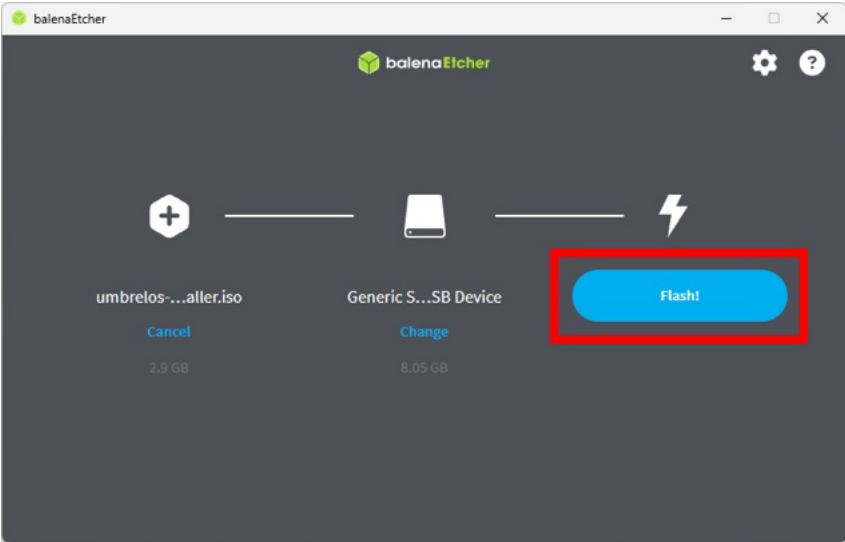
[Select target]을 누른다.



조금 전 쫓은 USB 저장 장치를 선택한다.



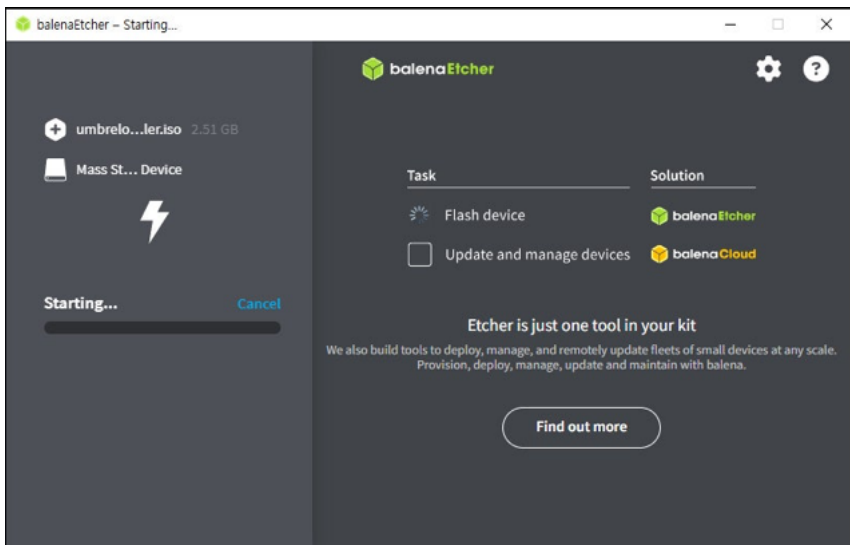
이제 [Flash!]를 누른다.

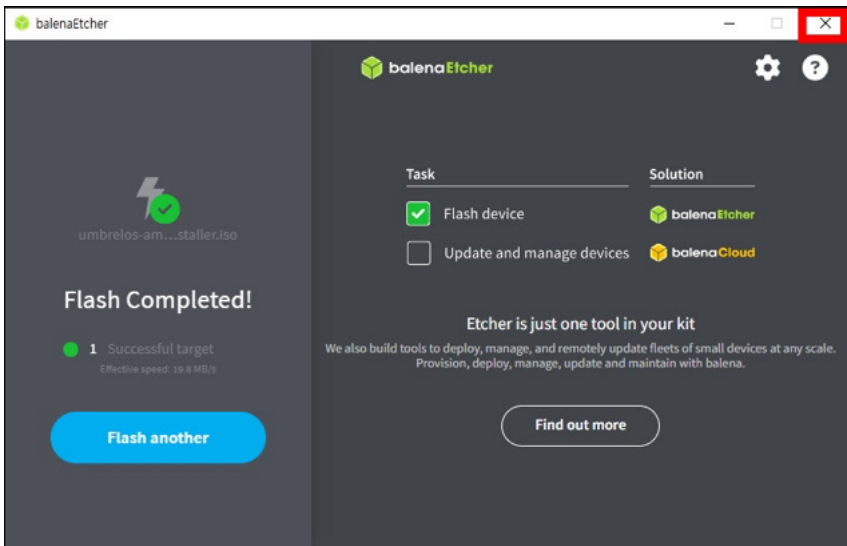


다음과 같은 화면이 뜨다가 'Flash Completed!'가 뜰 것이다. 뜨면 오른쪽 위 [x] 버튼을 누르고 USB를 빼면 된다. 중간에 포맷하라는 알림창이 뜨면 무시하고 취소하면 된다.

만약 플래싱에 실패하면 USB를 초기화하고 다시 플래싱해야 한다. 이때 일반적인 포맷 방법으로는 안 되는데, 부팅 USB 초기화 방법으로 검색하면 결과가 많이 나온다.

윈도우의 경우 윈도우 키 + R → cmd 입력 후 실행 → 터미널이 나오면 `diskpart` 입력 → `list disk` → 마이크로SD카드 디스크 번호 확인 → `select disk [디스크 번호]` → `clean`을 입력하면 된다. 그리고 다시 포맷하고 처음부터 진행하면 된다.



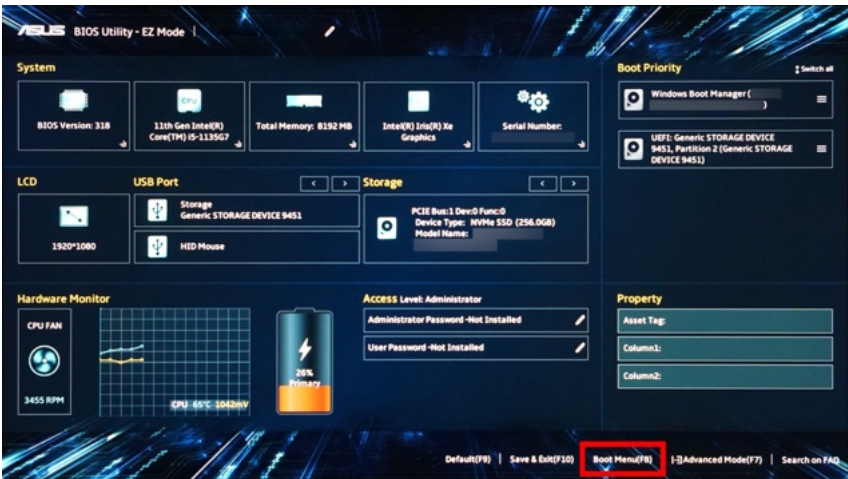


노트북에 엠브렐OS 설치하기

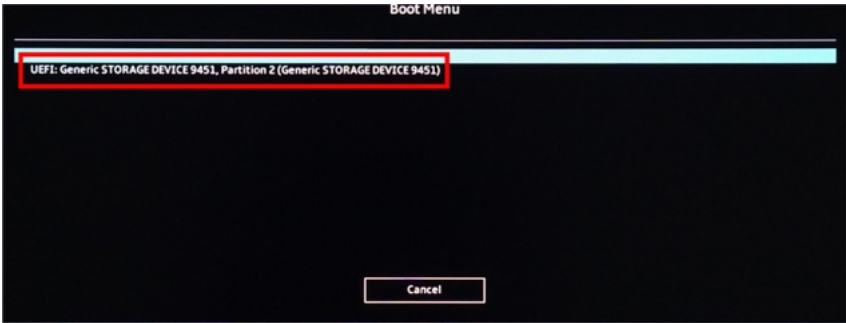
노트북에 엠브렐OS 설치용 USB를 꽂는다. 그다음에는 바이오스 설정 창으로 진입한다.

바이오스 설정 창으로 진입하기 위해서는 노트북을 켜자마자 F2, F7, F10, Delete 키 중 하나를 연타해야 한다. 이 중 어떤 키인지는 노트북 기종에 따라 다르다. 잘 모르겠으면 인터넷에 자신의 노트북 모델과 함께 '바이오스 진입'이라고 검색하면 나온다.

바이오스에 진입했으면 부트 메뉴를 찾아야 한다. 필자의 노트북에서는 F8을 누르면 부트 메뉴로 들어갈 수 있었다.



부트 메뉴에서 우리가 쫓은 엠브렐OS 설치용 USB 장치를 찾아 더블 클릭하거나, 방향키로 이동한 뒤 엔터 키를 누른다.



그러면 엠브렐OS 설치가 시작된다. 화면에 무언가 많이 지나갈 것이다.



우산 모양이 뜨면 OS를 어디에 설치할지 저장공간을 선택하는 창이 나온다. 여기서 노트북의 내장 저장 장치를 선택해야 한다. 다른 하나는 OS 설치용 USB다. 다음 사진에서는 노트북의 내장 저장 장치 번호에 해당하는 '2'를 입력하고 엔터 키를 눌렀다.

```
      ,###GGGGGGGGG!#Sp
    ,##GGG!W""^'  '""%GGG#S,
  ,#GGG"          "lGG#o
 #GG!^          'GG#
 ,#GGb         \GGG,
 lGG"         "GGG
 #GGG!GGG!#p, ,p##lGG!#p, ,p##ll#GGGG
 lGGG!W""""~GGGGGG#""""~lGGGGG#W""""~lGGGG#
 ""          ""          ""

umbre!OS USB Installer

Installing umbre!OS will wipe your entire storage device.

1) sda      Generic STORAGE DEVICE          7.5G
2) mmc0n1   [REDACTED] 238.5G
Select a storage device by number to install umbre!OS on: 2
```

무언가 뜨다가 umbre!OS has been installed! 문구와, Press any key to shutdown 문구가 뜨면 엔터를 누른다.

```
      ,###GGGGGGGGG!#Sp
    ,##GGG!W""^'  '""%GGG#S,
  ,#GGG"          "lGG#o
 #GG!^          'GG#
 ,#GGb         \GGG,
 lGG"         "GGG
 #GGG!GGG!#p, ,p##lGG!#p, ,p##ll#GGGG
 lGGG!W""""~GGGGGG#""""~lGGGGG#W""""~lGGGG#
 ""          ""          ""

umbre!OS USB Installer

Installing umbre!OS will wipe your entire storage device.

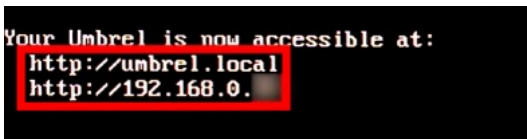
1) sda      Generic STORAGE DEVICE          7.5G
2) mmc0n1   [REDACTED] 238.5G
Select a storage device by number to install umbre!OS on: 2
installing umbre!OS on: mmc0n1
21126791168 bytes (21 GB, 20 GiB) copied, 196 s, 108 MB/s
0*2504261 records in
0*2504261 records out
21474936400 bytes (21 GB, 20 GiB) copied, 203.651 s, 105 MB/s

umbre!OS has been installed!
Press any key to shutdown, remember to remove the USB drive before turning the device back on..
```

노트북이 자동으로 켜다 켜질 것이다. 다시 무언가가 많이 지나갈 것이다. 만약 Finished로 시작하는 문구가 나왔는데 시간이 많이 지나도 다음 화면으로 넘어가지 않으면 기기를 재부팅해 보자.



다 지나가고 나면 `Your Umbrel is now accessible at:`이라는 문구가 뜰 것이다. 그 밑에 URL 두 개가 뜨는데 이 URL을 잘 기억하자. `http://192.168.0.???` 이것이 이 노트북의 로컬 IP 주소다.



노트북에서 엠브렐OS 설치용 USB를 제거한다. 노트북은 24시간 365일 켜져 있으면서 풀 노드의 역할을 할 것이다.

이제 같은 공유기에서 접속한 일반 PC나 스마트폰에서 umbrel에 접속할 수 있다.

노트북에서 umbrel 화면 띄우기

노트북에는 모니터가 내장되어 있다. 그런데 umbrelOS는 미니 PC나 라즈베리파이 등의 저전력 홈 서버 용도로 나온 OS이므로 모니터에 GUI(우리가 일반적으로 보는 화면)를 띄우지 않는다. CLI(컴퓨터 프로그래머들이 자주 쓸 것 같은, 명령어를 입력하는 화면)만 띄운다. umbrel이 설치된 노트북과 같은 공유기에 연결된 다른 스마트폰이나 컴퓨터 기기에서는 웹 브라우저에 `umbrel.local` 또는 `umbrel.` 또는 `192.168.0.???` 등의 로컬 IP 주소를 입력하면 umbrel 화면이 나온다(접속이 안 되면 `http://`도 함께 입력해 보면 된다). 만약 노트북에서도 이러한 화면이 나오게 하려면 몇 가지 설정이 필요하다.


노트북의 다음 화면에서 계정 이름과 비밀번호를 입력한다. 초기 설정은 계정 이름과 비밀번호 모두 'umbrel'이다. 비밀번호를 입력할 때는 입력이 안 되고 있는 것처럼 보여도 입력이 되고 있는 것이니, 정확하게 umbrel을 입력하고 엔터 키를 누르면 된다.

```
Your Umbrel is now accessible at:  
http://umbrel.local  
http://192.168.0.  
  
umbrel login: umbrel  
Password:
```

우산 모양이 나오면 이제 명령어를 입력할 수 있다. 여기에 정확하게 다음 명령어를 입력한다. 줄 바꿈은 'Shift' + 'Enter' 키를 눌러서 할

수 있으며, ‘\’ 문자는 보통 키보드에서 ‘₩’ 키를 누르면 입력된다. 줄 바꿈은 잘 보이도록 검정 배경으로 표시했다. 오타에 주의하여 입력하자. ‘EOF’는 모두 영어 대문자다. 만약 오타가 있는 상태로 이미 엔터를 눌렀다면 그냥 다시 한번 입력하고 엔터 키를 누르면 된다.

```
sudo apt update && \
sudo apt install --no-install-recommends -y xorg lxde-core
  lightdm chromium unclutter fonts-noto-cjk && \
mkdir -p ~/.config/lxsession/LXDE/ && \
cat > ~/.config/lxsession/LXDE/autostart << EOF
@xset s off
@xset -dpms
@xset s noblank
@unclutter -idle 1
@chromium --noerrdialogs --disable-sessio
  n-crashed-bubble --disable-infobars \
  --incognito --kiosk http://localhost
EOF
```



각 명령어를 간단히 설명해 보겠다. `sudo`는 관리자 권한으로 명령어를 실행하는 것을 말한다. `apt update`는 `apt` 패키지를 업데이트하는 것이다.

두 번째 줄의 `apt install`은 설치 명령어다. `--no-install-recommends`는 추천 패키지를 설치하지 않고 최소 구성만 설치하겠다는 뜻이다. `-y`는 설치 과정에서 묻는 질문에 자동으로 `y`를 입력하게 하는 것이다. `xorg`는 리눅스에서 GUI를 가능하게 해주는 기본 엔진이라 생각하면 된다. `lxde-core`는 데스크톱 GUI다. `lightdm`은 부팅 후 GUI 세션을 띄우게 하는 디스플레이 매니저다. `chromium`은 크롬 브라우저의 오픈소스 버전인 ‘크로미움’이다. 이 브라우저를 통해 엠프렐 화

면을 띄울 것이다. `unclutter`는 일정 시간 마우스 움직임이 없으면 커서를 숨겨주는 프로그램이다. `fonts-noto-cjk`는 구글의 Noto CJK 폰트다. 여기에 한글 폰트가 포함되어 있는데 이걸 설치하지 않으면 엠프렐에서 한글이 모두 깨져서 보이게 된다.

그다음 줄의 `mkdir`은 디렉토리, 즉 폴더를 생성하는 것이다.

`cat`은 파일 편집 명령어다. EOF까지 입력된 여러 줄을 파일에 그대로 입력해 저장하라는 뜻이다. EOF 전까지의 내용들이 `autostart` 파일로 들어간다.

`@xset s off`는 스크린세이버 기능을 끄는 것이다. 스크린세이버는 일정 시간이 지나면 검은 화면을 출력하는 화면보호기 기능이다. `@xset -dpms`는 모니터 절전 기능을 끄는 것이다. 모니터가 꺼지지 않게 한다. `@xset s noblank`는 빈 화면으로 전환되는 기능도 끄는 것을 말한다. 이 설정들은 시간이 지나도 화면이 꺼지지 않게 하는 설정이다. 만약 화면 꺼짐이 있게 설정하고 싶으면 `@xset`으로 시작하는 세 줄을 다음 두 줄로 바꾸면 된다. 여기서 시간 단위는 초 단위이다. 예를 들어 3시간 동안 입력이 없을 때 화면이 꺼지게 만들고 싶다면 다음과 같이 10,800을 입력하면 된다($60\text{초/분} \times 60\text{분/시간} \times 3\text{시간} = 10,800\text{초}$). 이렇게 하면 시간이 지나 모니터가 꺼져도 엠프렐OS는 계속 돌아간다.

```
@xset s 10800 10800
@xset dpms 0 0 10800
```

`@unclutter -idle 1`은 마우스를 1초 동안 움직이지 않으면 커서를 화면에서 숨기겠다는 뜻이다.



중간에 키보드 레이아웃을 물어보는 창이 나온다. 일반적인 쿼티 표준 키보드 배열을 사용한다면 1을 입력하고 엔터 키를 누르면 된다. 드보락이나 콜맥 배열을 사용하는 경우 해당 숫자를 누르고 엔터 키를 누르면 된다.

```
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is instal
debconf: falling back to frontend: Readline
Configuring keyboard-configuration
-----
Please select the layout matching the keyboard fo

  1. English (US)
  2. English (US) - Cherokee
  3. English (US) - English (Colemak)
  4. English (US) - English (Colemak-DH ISO)
  5. English (US) - English (Colemak-DH)
  6. English (US) - English (Dvorak)
  7. English (US) - English (Dvorak, Macintosh)
Keyboard layout: 1
Progress: [ 71%] |#####
```

완료되면 다음 명령어를 입력한다. 오타에 주의하라. `systemctl`에서 마지막 문자는 알파벳 `L` 소문자다.

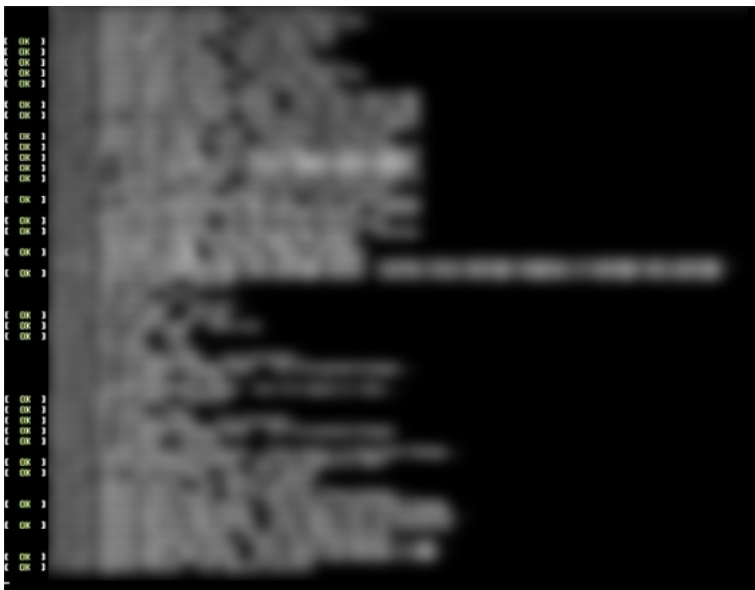
```
sudo systemctl set-default graphical.target && sudo reboot
```



`sudo systemctl set-default graphical.target`은 부팅되었을 때 기본적으로 GUI로 시작하라는 명령어고, `sudo reboot`는 재부팅하라는 명령어다.

```
Setting up
Setting up
Setting up
Setting up
Processing
Processing
Processing
Processing
Processing
Processing
Processing
Setting up
umbrel@umbrel:~$ sudo systemctl set-default graphical.target && sudo reboot
```

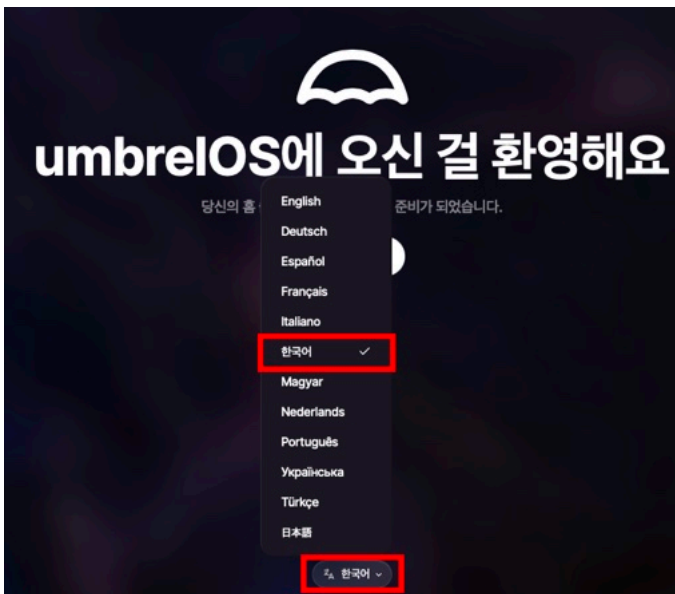
무언가 많이 지나가고 컴퓨터가 재부팅된다.



그러면 그래픽 화면(GUI)과 함께 로그인 창이 나온다. 여기에 계정 이름과 비밀번호 모두 'umbrel'을 입력하면 된다. 참고로 엠브렐에서 계정 설정을 한 뒤에는 계정 이름은 'umbrel', 비밀번호는 엠브렐에서 설정한 로그인 비밀번호를 입력하면 된다.



언어 설정이 한국어가 아니라면 한국어로 변경한다.



이제 노트북에서 엠브렐OS 웹 화면을 띄울 수 있게 되었다.



만약 CPU나 램 용량이 부족하거나 전력 절감 등의 이유로 다시 GUI (그래픽 화면)를 삭제하고 싶다면 다음과 같이 하면 된다. 먼저 'Ctrl' + 'Alt' + 'F2' 키를 눌러 CLI 화면(명령줄 화면)으로 들어가야 한다(안 된다면 F2 대신 F1-F6을 눌러보자). 그리고 다음 명령어를 입력하면 된다.

```
sudo apt purge -y xorg lxde-core lightdm  
chromium unclutter fonts-noto-cjk && \n  
sudo apt autoremove -y && \n  
sudo systemctl set-default multi-user.target && \n  
sudo systemctl isolate multi-user.target
```



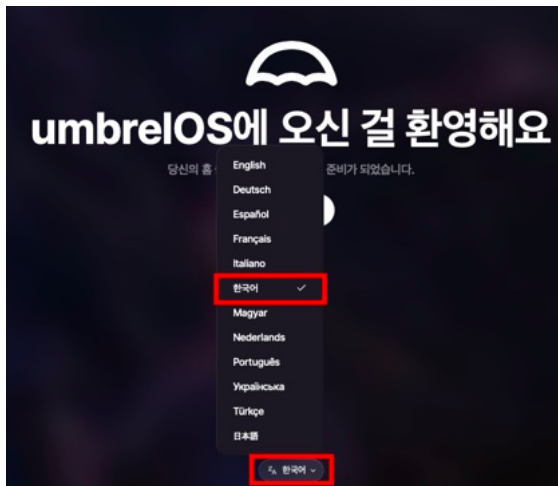
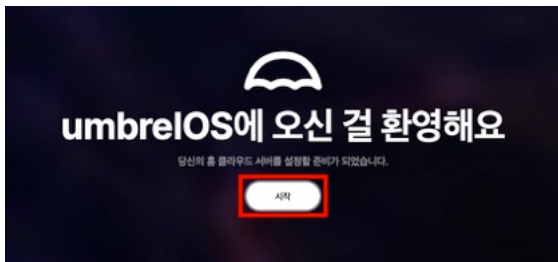
이렇게 해서 노트북에 엠브렐OS를 설치하고, 그래픽 화면을 모니터에 출력하는 방법까지 알아보았다.

| 엠브렐 설정 및 풀 노드 동기화

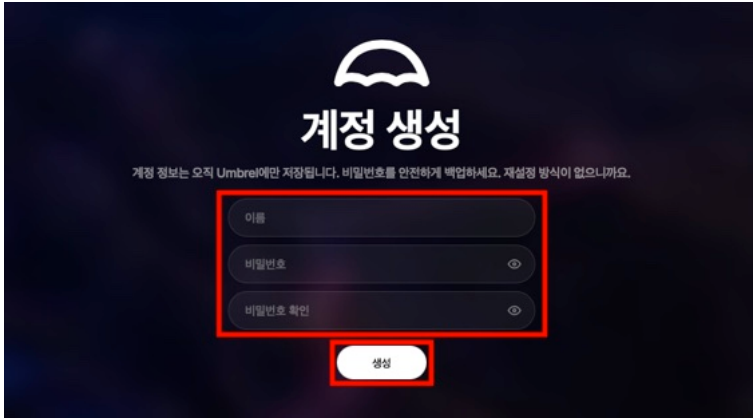
엠브렐 설정 및 업데이트

이제 사파리, 크롬 등의 웹브라우저에서 umbrel.local 혹은 umbrel. 혹은 엠브렐 기기의 로컬 IP 주소인 192.168.0.???를 입력한다. 만약 안 넘어간다면 http://까지 함께 입력해 보자.

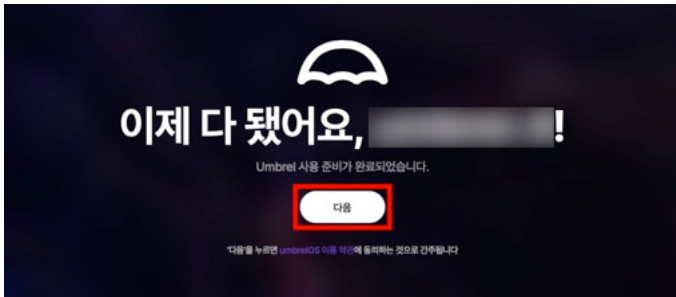
‘umbrelOS에 오신 걸 환영해요’가 나오면 [시작]을 누르고, 언어 설정이 한국어가 아니라면 한국어로 변경한다.



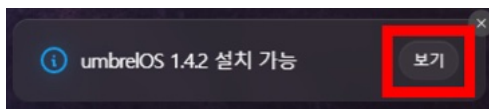
‘계정 생성’에서 엮브렐 계정 이름과 비밀번호, 비밀번호 확인을 입력하고 [생성]을 누른다. 비밀번호를 잊어버리면 복구가 불가능하니 잊어버리지 않도록 꼭 주의하자.



[다음]을 누르면 엮브렐 홈 화면으로 넘어간다.



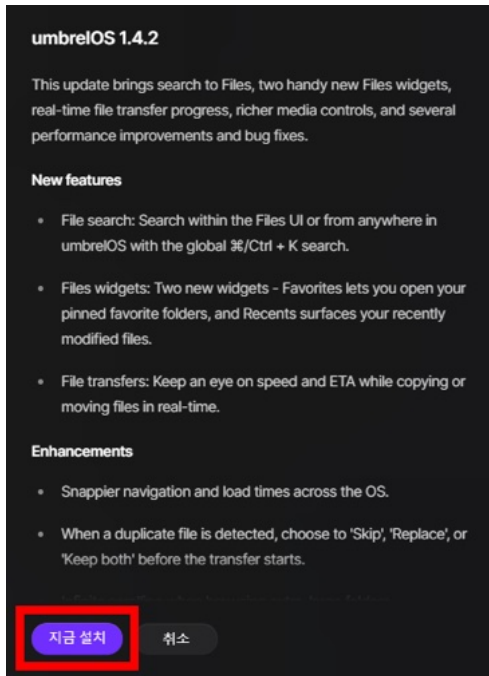
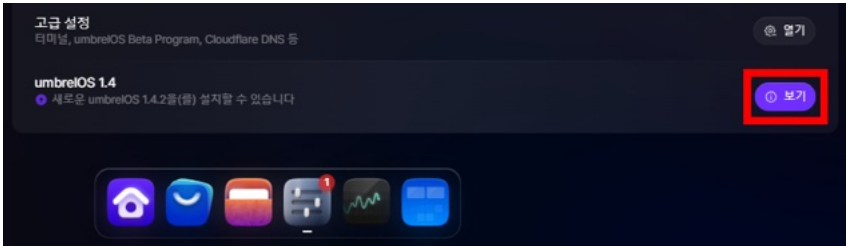
엮브렐에 처음 접속하면 오른쪽 위에 다음과 같은 업데이트 알림창이 나온다.



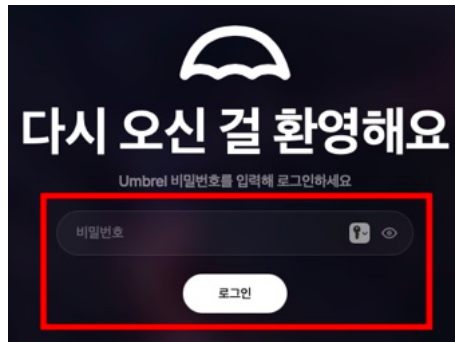
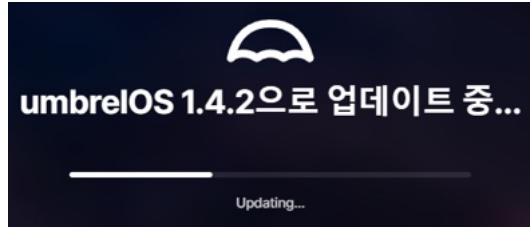
만약 이런 알림창이 뜨지 않는다면 화면 하단 설정 버튼 → 스크롤 내리면 있는 umbrella 버전 → [업데이트 확인]을 누르면 된다.



‘최신 umbrelOS를 사용 중입니다’라는 문구가 나왔다면 이미 최신 버전인 것이므로 다음 절로 넘어가도 된다. 최신 버전이 아니라면 새로운 버전을 설치할 수 있다는 문구가 나온다. [보기] → [지금 설치]를 누른다.



업데이트가 완료되면 다시 로그인하는 창이 뜰 것이다. 비밀번호를 입력하고 로그인한다. 한국어 설정이 해제되었다면 로그인한 뒤 설정에서 다시 언어를 한국어로 바꿔주면 된다.



비트코인 노드(코어) 또는 노츠 설치

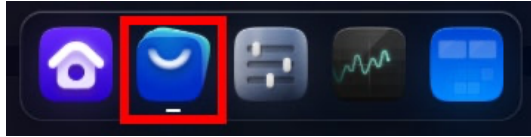
이제 비트코인 클라이언트를 설치할 것이다. 업브렐에서는 비트코인 노드(비트코인 코어)와 비트코인 노츠 중 하나를 선택해 설치하거나 둘 다 설치할 수 있다(여기서 리브레는 논외로 한다). 비트코인 코어는 그 초기 버전을 사토시 나카모토가 만들었다. 노츠는 메모 정책이나 릴레이 정책을 사용자가 좀 더 자유롭게 설정할 수 있다. 다양한 설정을 제공하는 만큼 좀 더 CPU와 램을 많이 차지한다. 일각에서는 이러한 릴레이 정책이나 메모 정책이 소용이 없다고 비판한다. 채굴자가 아닌 이상 메모, 릴레이 정책만으로는 최종 블록 반영에 영향력이 크지 않아 기기 자원만 낭비된다는 것이 주요 주장이다. 노츠의 사용자 선택권은 다팀 DATUM과 결합할 때 강력해지는데, 자신이 원하는 블록 템플릿대로 채굴할 수 있게 되기 때문이다. 예를 들어 오디널스 거래들을 스패밍으로 인식해 블록 템플릿에 포함시키지 않고 채굴하는 것 등이 가능하다. 노츠는 개발자 한 명에 대한 의존도가 크다는 점에서 장기 안정성이 불안하다고 보는 시각도 있다.

코어는 노츠만큼의 선택권을 보장하지 않는 대신 노츠보다 가볍다. 비트코인 풀 노드 중 가장 많이 쓰이고 있으며, 호환성과 안정성이 가장 검증되어 있다. 가장 많은 개발자들이 코드 리뷰에 참여하므로 오류 발견이나 보완 속도가 빠르다. 거꾸로 실험적 기능이나 새로운 정책 채택에 보수적이다. 또한, 사용자가 메모, 릴레이 정책을 세부적으로 제어하기가 어렵다.

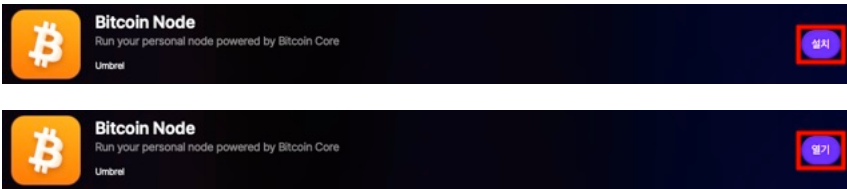
만약 이후에 다팀으로 채굴기까지 연결할 계획이 있다면, 노츠로만 연결이 되니 참고하여 진행하면 된다. 어느 것이 좋다고보다는 장단점이 분명하니 사용자가 선택하면 된다. 필자는 둘 다 운영한다. 무엇보다

다 비트코인 풀 노드 클라이언트가 다양화되는 것은 비트코인 네트워크에 좋은 일이다.

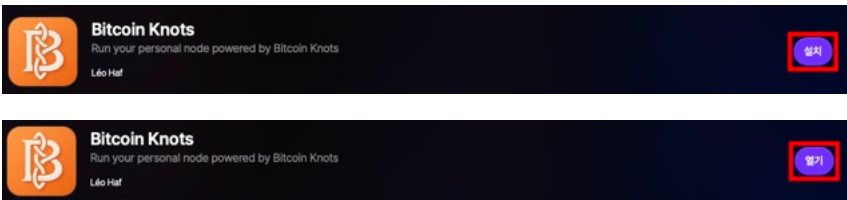
이제 비트코인 코어 또는 비트코인 노츠를 설치하는 방법을 알아보자. 빠른 실행에서 앱스토어를 누른다.



비트코인 코어를 설치할 경우, '비트코인 노드(Bitcoin Node)'를 찾아 [설치] → [열기]를 누른다.



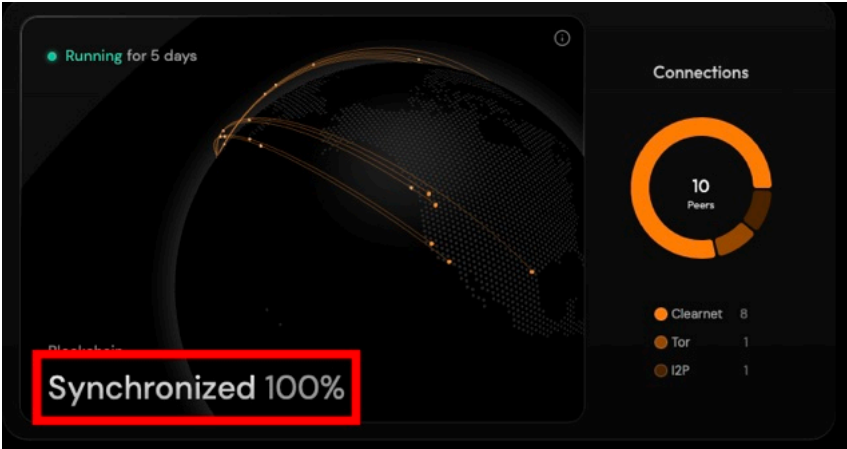
비트코인 노츠를 설치할 경우, '비트코인 노츠(Bitcoin Knots)'를 찾아 [설치] → [열기]를 누른다.



설치가 완료되면 자동으로 동기화를 시작한다. 이는 IBD (초기 블록 다운로드) 과정인데, 다른 노드에게서 1번 블록부터 현재 블록까지 다운로드하는 일이다. 참고로 0번 블록인 제네시스 블록은 다른 노드에게서 받아오는 데이터가 아니라 앱을 설치할 때 기본적으로 있는 블록이다.



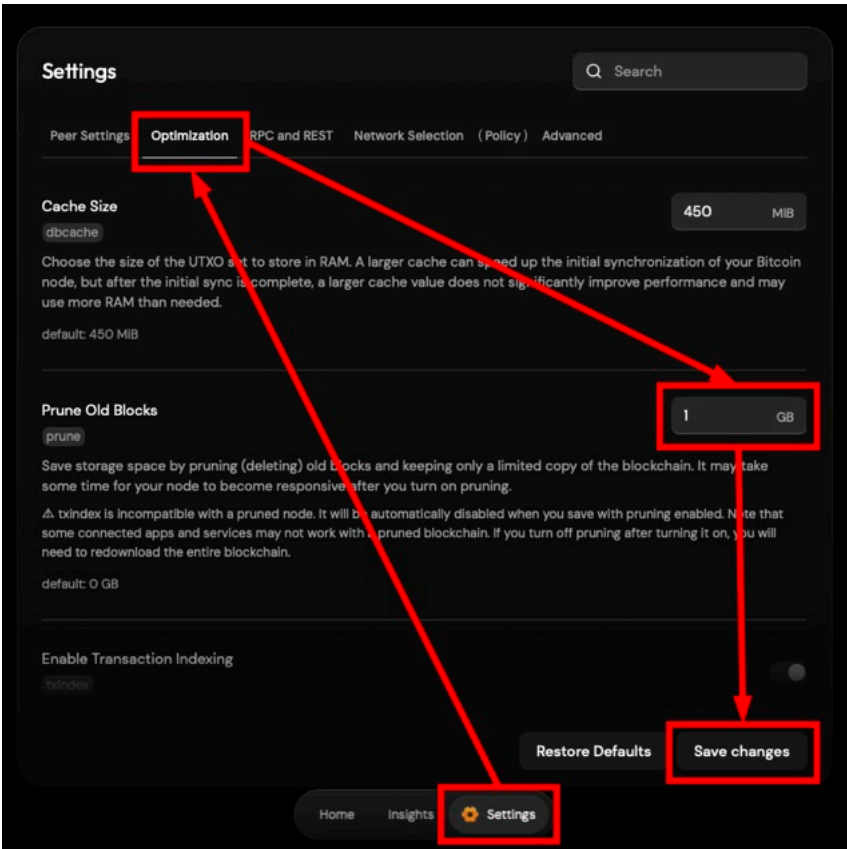
블록 데이터가 크기 때문에 100% 동기화가 될 때까지 인터넷 환경에 따라 하루-일주일 정도 걸릴 수 있다.



가지치기(프루닝) 설정

저장 공간이 2TB가 넘는다면 블록 데이터를 여유롭게 모두 보관할 수 있다. 그러나 노트북에 엠브렐OS를 설치한 앞의 사례처럼 블록 데이터를 모두 보관할 공간이 없을 수도 있다. 이때는 프루닝 설정을 통해 오래된 블록 데이터를 삭제할 수 있다. 이렇게 해도 검증 과정은 모두 거치기 때문에 동기화 시간은 똑같이 걸린다. UTXO 세트도 모두 보관하므로 프루닝 노드는 거래나 블록 검증 과정을 일반 풀 노드와 똑같이 수행한다. 단지 비트코인의 모든 거래 역사를 보관하지 않을 뿐이다. 또한, 프루닝 노드의 경우 이후에 진행할 일렉터스 연동 과정 등에서 문제가 발생할 수 있다.

프루닝은 하단에서 [Settings] → [Optimization] → 'Prune Old Blocks'에서 원하는 블록 데이터 용량을 입력하면 된다. 1GB를 입력하면 최신 블록 1GB만 제외하고 나머지 블록 데이터는 삭제한다. 원하는 블록 데이터 용량을 입력했으면 아래 [Save changes]를 누른다.

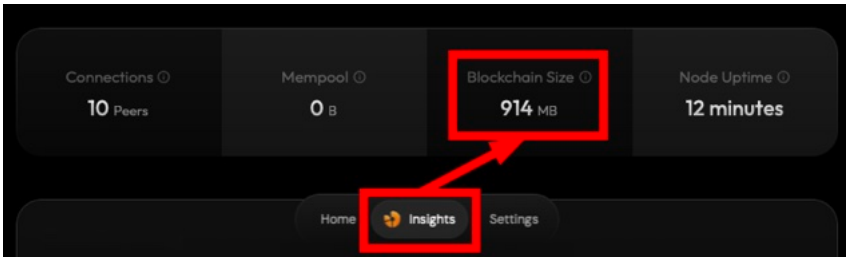


잠시 기다리면 노드가 재시작되고 왼쪽 상단에 바뀐 설정이 적용되었다는 알림창이 나온다.

☀ Hang tight, Bitcoin Knots is restarting...

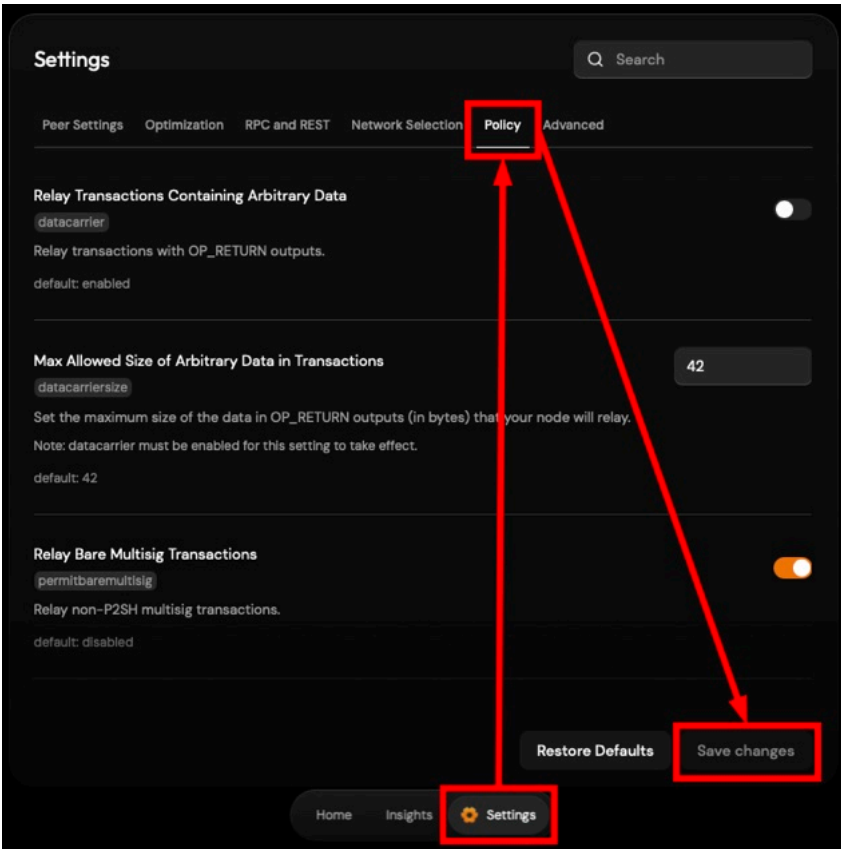
✔ Settings applied

하단 [Insights]를 누르면 저장된 블록 데이터의 크기를 볼 수 있는데, 프루닝을 설정한 경우 설정한 용량을 넘지 않는 것을 알 수 있다.



노츠의 사용자 정책 설정

하단 [Settings]에 들어가 보면 노츠에는 코어에 없는 [Policy]라는 설정 창이 하나 더 있다. 이 설정이 무엇인지 보면 코어와 노츠의 정확한 차이를 이해할 수 있을 것이다.



풀 노드는 일정한 규칙에 따라 거래나 블록을 검증한다. 그런데 여기에는 두 가지 종류의 검증 규칙이 있다. 하나는 합의 규칙consensus rules이라고 하는 것들이다. 이는 모든 네트워크의 참여자들이 동일하게 적용하는 것으로, 이를 지키지 않는 노드가 생성한 블록이나 거래는 비트코인 네트워크 전체가 거부한다. 이러한 합의 규칙에는 이중 지불 불가, 블록 크기 4 Mwu 제한 등이 있다. 즉, 합의 규칙은 정직하게 행동하는 네트워크 노드 전체가 동일하게 적용한다. 그래서 모든 풀 노드가 보관

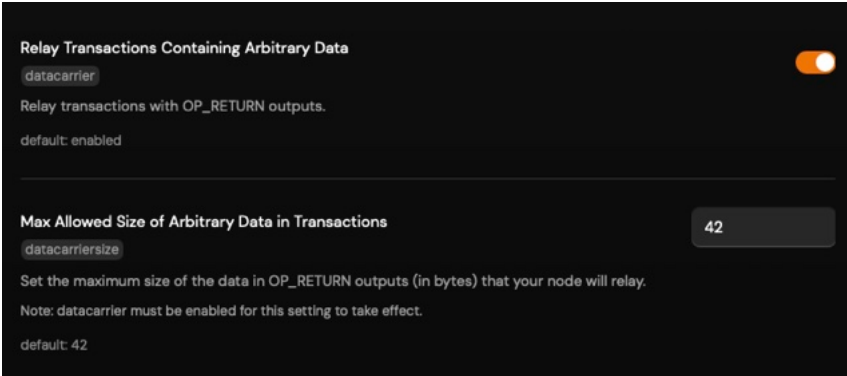
하는 블록 데이터는 동일하다. 일시적으로 분기가 발생할 수 있지만 결국 리오그^{re-org}되어 동일해진다.

블록 데이터와 달리 풀 노드들의 맴플 공간은 제각각이다. 풀 노드에는 검증에 통과했지만 아직 블록에 실리지 않은 거래들을 모아놓는 공간이 있는데, 이를 맴플이라고 한다고 했다. 맴플을 기반으로 만들어진 다음 블록 후보를 블록 템플릿이라 하는데, 채굴기는 이러한 블록 템플릿을 가져와서 채굴을 한다. 맴플은 풀 노드마다 모두 달라도 되기 때문에 풀 노드는 어떤 거래를 맴플에 넣을지, 안 넣을지 정할 수 있다. 이렇게 풀 노드가 개별적으로 어떤 거래를 맴플에 넣을지 말지, 그래서 어떤 거래를 주변에 전파할지 말지 선택하는 규칙을 정책^{policy}이라 한다. 주로 맴플 정책, 릴레이 정책이라고 이야기한다. 다음 후보 블록을 만드는 것도 정책이다.

정책은 합의 규칙을 준수하는 선에서만 설정할 수 있다. 즉, 합의 규칙보다 더 엄격한 규칙만 적용할 수 있고, 더 느슨한 규칙은 적용할 수 없다. 더 느슨한 규칙은 합의 규칙을 위반할 수 있기 때문이다. 예를 들어보자. 비트코인의 합의 규칙에 의하면 블록의 용량 크기가 4 Mwu 이하여야 한다. 그러면 블록 템플릿을 만드는 정책에서는 블록의 용량 크기가 2 Mwu 이하가 되도록 설정할 수 있다. 2 Mwu 이하(정책)여도 4 Mwu 이하(합의 규칙)이기 때문이다. 그러나 정책이 2 Mwu이어도 채굴된 블록이 3 Mwu이었다면 이는 받아들여야 한다. 합의 규칙은 준수해야 하기 때문이다. 즉, 정책은 맴플에 담을지 말지, 주변에 전파할지 말지에만 적용되고 이미 다른 누군가가 채굴한 블록에는 적용되지 않는다. 반대로 정책을 6 Mwu 이하로 해서 합의 규칙보다 느슨하게 하는

것은 의미가 없다. 왜냐하면 반드시 준수되어야 하는 합의 규칙이 4 MWu 이하이기 때문이다.

이제 노트에서 어떤 사용자 정책들을 설정할 수 있는지 알아보자.



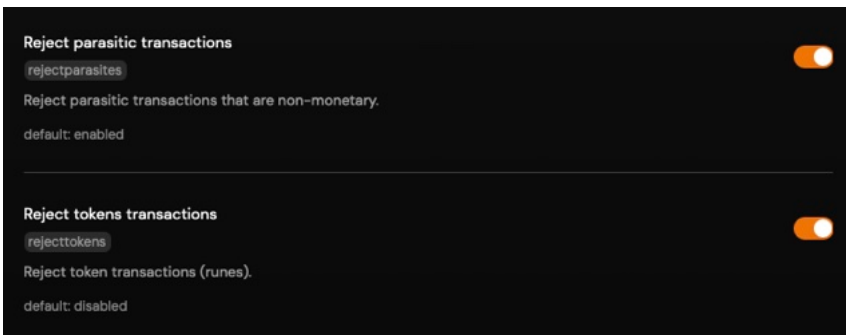
‘Relay Transactions Containing Arbitrary Data (datacarrier)’는 OP_RETURN 출력이 포함된 거래를 메모에서 허용할지 말지, 주변에 전파할지 말지 설정하는 것이다. 합의 규칙에서는 OP_RETURN 출력은 블록 데이터에는 실리지만 UTXO 세트에는 실리지 않는다. 즉, OP_RETURN은 합의상 허용된다. 이 토글 스위치를 켜면 OP_RETURN 출력이 포함된 거래를 메모에 넣고, 주변에 전파한다.

‘Max Allowed Size of Arbitrary Data in Transactions (datacarriersize)’는 OP_RETURN 뒤에 오는 데이터 크기의 상한을 설정할 수 있는 메모, 릴레이 정책이다. 합의 규칙 차원에서는 OP_RETURN 뒤에 오는 데이터 크기에 대한 상한이 없지만 스크립트의 크기가 1만 바이트 이하여야 하고, 거래 크기가 블록 크기 제한을 초과할 수 없다는 합의 규칙이 있어서 이 데이터가 1만 바이트를 넘을 수는 없다. 코어에서는 이 데이터 상한 정책이 83바이트였는데 이 정책 때문에 OP_RETURN을 통해 데이터를 쓰지 않고 스팸이 자꾸 UTXO

세트로 가서 지저분해지고, 일관된 뎀플 상태(수수료 예측) 등이 불가능해진다는 등 여러 가지 이유로 이 정책을 없앤다고 공지해서 논란이 일어난 적이 있다. 노트에서는 이 크기 제한 정책이 기본적으로 42바이트다.



‘Relay Bare Multisig Transactions (permitbaremultisig)’는 P2SH로 감싸지 않은 P2MS 출력을 뎀플에서 허용할지 말지 설정하는 옵션이다. 멀티시그는 여러 개의 키를 이용해 출력을 잠그는 것을 말한다. 여러 개의 공개키가 들어가니 출력도 길다. 그래서 이 긴 출력을 해시로 한 번 돌려 짧게 만드는데, 이것이 P2SH이다. 합의 규칙은 P2SH로 감싸지 않은 P2MS도 허용한다. 이것을 끄면 P2SH로 감싸지 않은 P2MS 출력은 뎀플에 넣지 않고, 전파하지도 않는다.

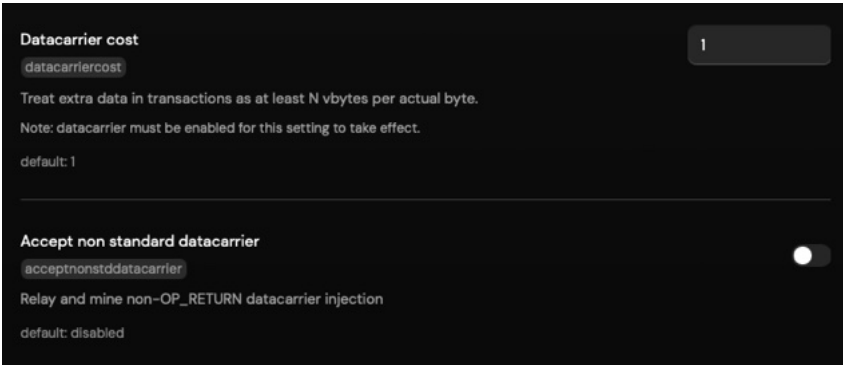


‘Reject parasitic transactions (rejectparasites)’는 금전적 이동이 없는 기생적 거래를 거부하는 정책이다. 합의 규칙에서는 허용한다. 예를 들어 오디널스-인스크립션 거래 대부분은 트랜잭션을 아주 적은 양의 비트코인과 함께 전송한다. 그중에서 비트코인 전송은 아예 없고, 데이터만 실려있는 거래들도 있는데 이러한 거래를 거부하는 것이다. 노츠에서는 기본적으로 이 옵션이 켜져 있어서 기생적 거래를 맴플에서 거부한다.

‘Reject tokens transactions (rejecttokens)’는 룬과 같은 토큰 프로토콜 관련 거래를 거부하는 정책이다. 합의 규칙에서는 허용한다. 노츠에서는 기본적으로 이 옵션이 꺼져 있는데, 이 옵션을 켜면 토큰 프로토콜 관련 거래를 맴플에서 거부한다.

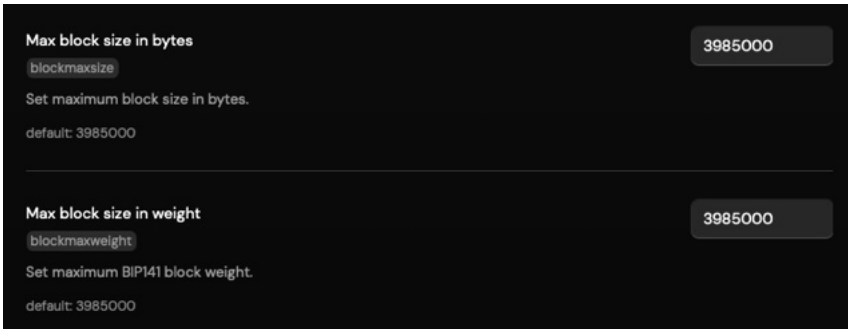


‘Permit Bare Pubkey (permitbarepubkey)’는 공개키 출력을 허용하는 정책이다. 합의 규칙에서도 허용한다. 지금은 거의 공개키 출력을 쓰지 않고 주소 출력을 쓴다. ‘bc1q’, ‘bc1p’ 등으로 시작하는 것이 주소인데, 주소가 제안되기 전에는 공개키를 바로 사용했었다. 노츠에서는 기본적으로 허용하지 않는다.



‘Datacarrier cost (datacarriercost)’는 OP_RETURN 뒤에 오는 데이터 크기의 가중치를 몇으로 할지 설정할 수 있는 옵션이다. 1은 실제 바이트와 동일하게 간주하는 것이다. 만약 2로 설정하면 OP_RETURN 뒤에 오는 데이터를 일반 데이터보다 2배 더 큰 것으로 간주한다. 예를 들어 이 값이 2일 때 OP_RETURN 뒤에 80바이트의 데이터가 오면, 이 데이터를 160 vB (가상 바이트)로 간주한다. 노트에서 기본값은 1이다.

‘Accept non standard datacarrier (acceptnonstddatacarrier)’는 OP_RETURN 이외의 방식으로 임의의 데이터를 삽입하는 거래를 전파할지 여부이다. 노트에서 기본값은 꺼져 있다.



'Max block size in bytes (blockmaxsize)'는 블록 템플릿의 최대 블록 크기를 설정하는 것이다. 블록 템플릿 관련 정책이므로 채굴기를 연결하지 않을 경우 아무 영향이 없는 정책이다. 합의 규칙에 의하면 블록 크기는 $4Mw$ 이하여야 한다. 모든 데이터가 증인(witness) 칸에 들어갈 경우 블록 데이터가 4MB까지 될 수 있지만, 이는 불가능한데 정말로 모든 데이터가 증인 칸에 들어갈 수는 없기 때문이다. 노츠에서 기본값은 3.985 MB인데 더 낮은 값으로 설정할 수도 있다.

'Max block size in weight (blockmaxweight)'는 블록 템플릿의 최대 블록 무게를 설정하는 것이다. 합의 규칙에서는 $4 Mw_u$ 이하여야 한다. 노츠에서 기본값은 3.985 MW_u 인데 이것보다 작은 값을 설정할 수 있다. 이 설정도 블록 템플릿 정책이므로 채굴기를 연결하지 않을 경우 아무 영향이 없다.

이렇듯 노츠에서는 사용자가 다양한 정책을 설정할 수 있다. 대신 코어보다 더 무겁다.

| 외부에서 엮브렐 접속하기

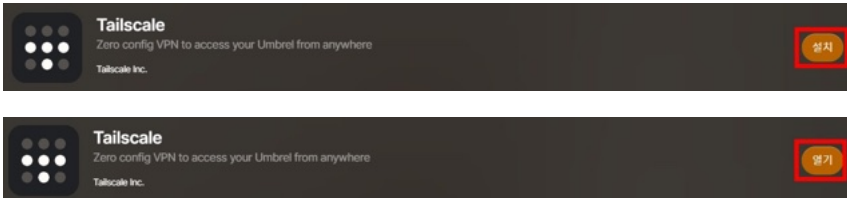
테일스케일 설치 및 연결

이제 외부에서 엮브렐에 접속하는 방법을 알아보자. 엮브렐은 기본적으로 로컬 네트워크를 통해 접속할 수 있다. 무슨 뜻이냐면 같은 공유기를 쓰고 있을 때만 다른 기기에서 엮브렐에 접속할 수 있다는 뜻이다.

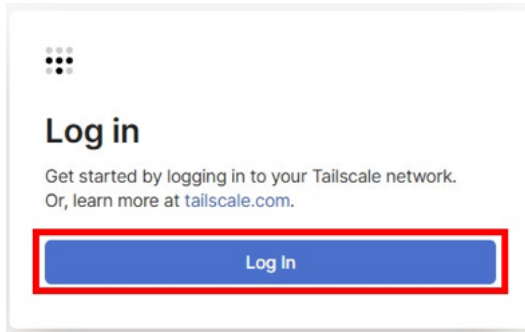
그러면 외부에 있을 때 스마트폰으로 엮브렐 기기에 접속하려면 어떻게 해야 할까? 스마트폰과 엮브렐 기기가 마치 로컬 네트워크에 묶여있는 것처럼 해주는 가상의 네트워크가 필요하다. 이를 가상의 사설 네트워크라고 해서 VPN(virtual private network)이라 한다.

테일스케일은 P2P VPN으로, 엮브렐 기기와 스마트폰을 연결하기에 아주 좋은 VPN이다. 지금부터 테일스케일을 설치하고, 외부에서 엮브렐에 접속해 보도록 하자.

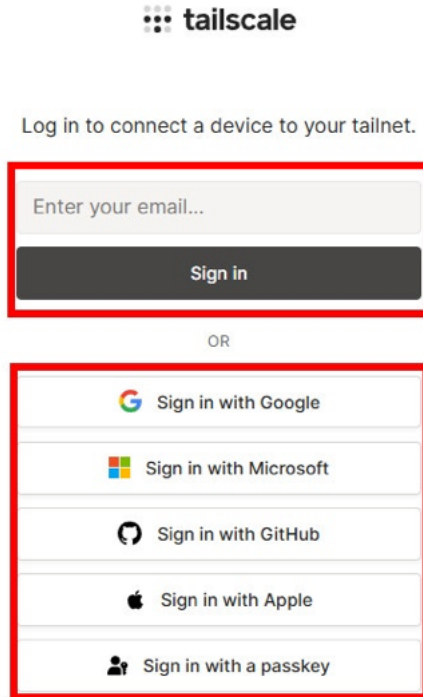
먼저 엮브렐에 테일스케일을 설치한다.



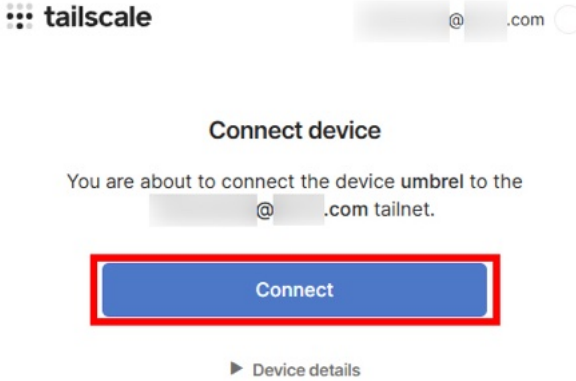
테일스케일을 열면 로그인하라는 창이 나온다. 로그인 버튼을 누른다.



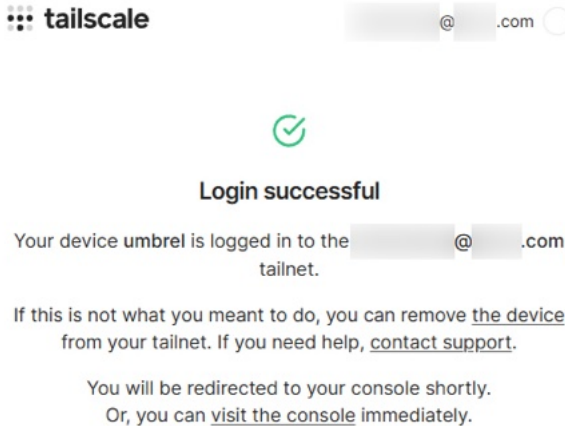
이메일 주소를 입력해 가입해도 되고, 구글이나 애플 계정 등을 이용해 가입해도 된다.



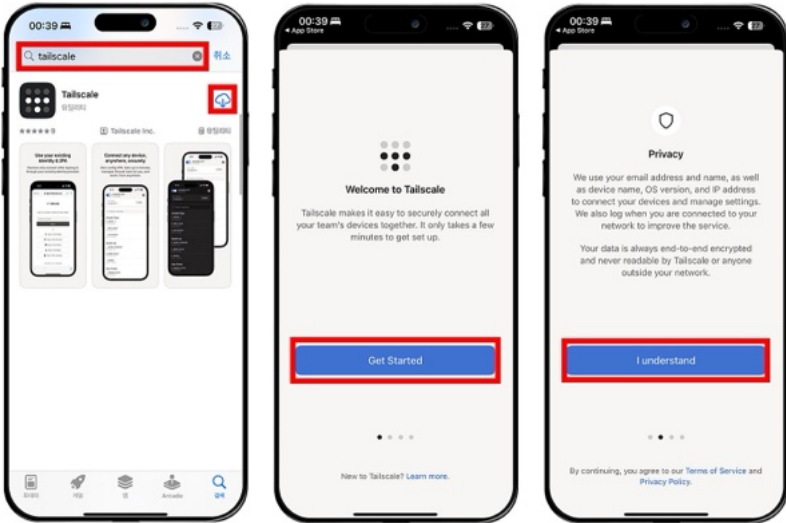
가입이 되면 Connect device 창이 나온다. [Connect]를 누른다.



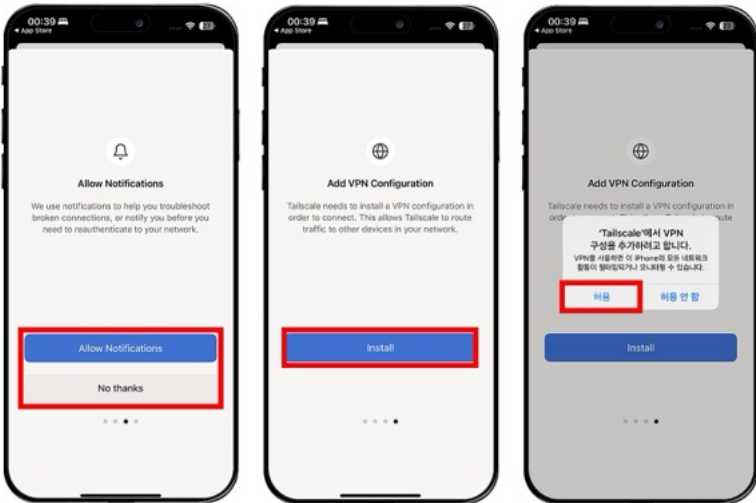
그러면 로그인에 잘 되었다는 알림창이 뜬다. 이제 테일스케일에 엮 브렐 기기 연결은 끝났다.



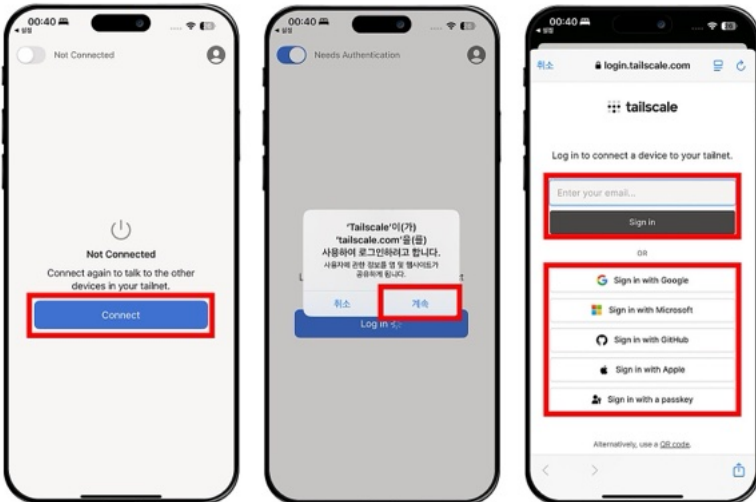
이제 스마트폰도 테일스케일에 연결해야 한다. 먼저 앱스토어 혹은 구글 플레이스토어에서 'Tailscale'을 검색하고 앱을 다운로드한다. 앱을 실행한 후 [Get Started] → [I understand]를 누른다.



알림을 받을지 말지를 선택하고 [Install]을 누른다. 그러면 테일스케일 앱이 VPN 구성을 추가한다는 알림창이 뜨면 [허용]을 누른다.

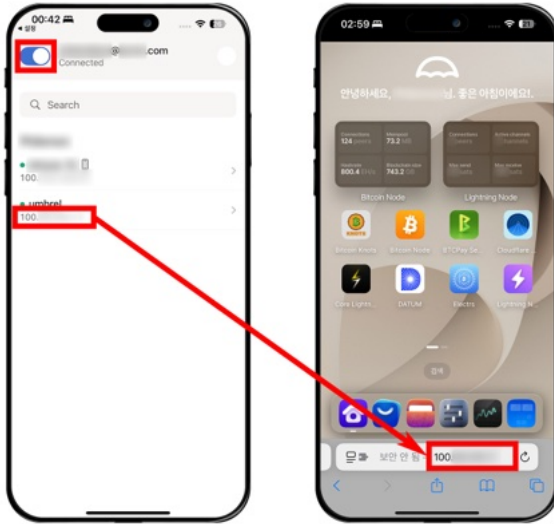


그리고 나서 [Connect]를 누른다. 앰브렐 기기에서 로그인했던 똑같은 계정으로 로그인한다.



잘 연결되었다면 로그인했을 때 왼쪽 위에 토글 스위치가 파란색으로 되어있을 것이다. 스마트폰과 엀브렐 기기 왼쪽에 초록색 점이 있는지 확인하자.

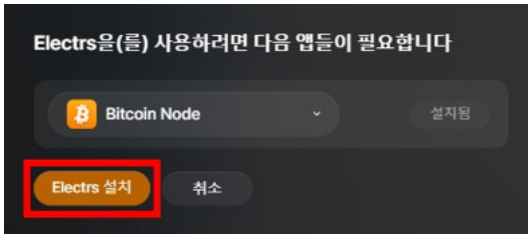
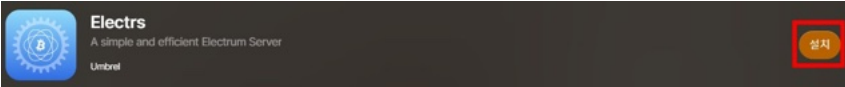
이제 테일스케일을 통해 원격으로도 엀브렐에 접속할 수 있게 되었다. 엀브렐 밑에 있는 100.????.????.???의 주소를 웹 브라우저 창에 입력해 보자. 나의 엀브렐에 접속할 수 있게 되는 것을 알 수 있다.



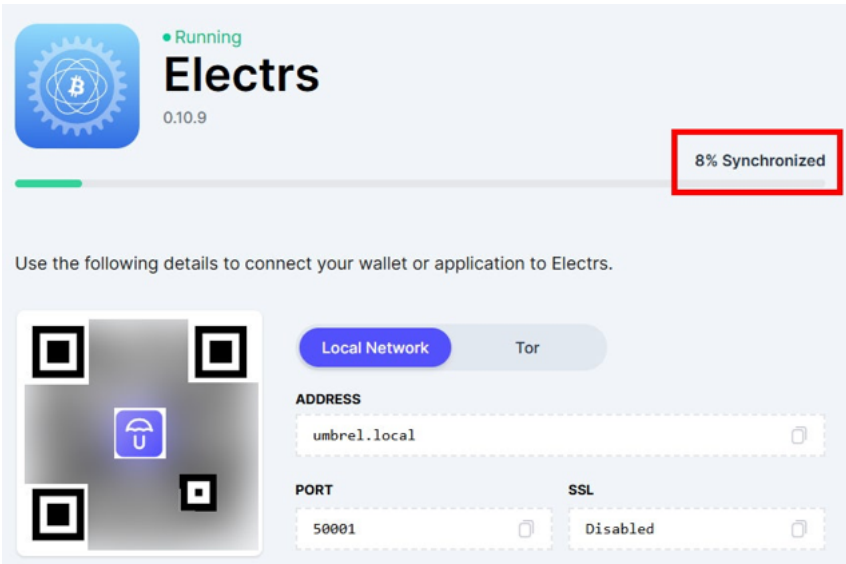
| 위치-온리 지갑과 자신의 풀 노드 연동하기

일렉터스(Electrs) 설치

풀 노드 동기화가 완료되었다면 이제 자신의 풀 노드와 위치-온리 지갑을 연결해 보자. 먼저 일렉트럼 서버인 일렉터스를 설치할 것이다.




일렉터스를 깔고 나면 비트코인 코어와 동기화를 시작한다. 이 동기화가 100%가 될 때까지 기다린다. 하루-이틀 정도 걸릴 수 있다.



The screenshot shows the Electrs interface with the following details:

- Running
- Electrs**
- 0.10.9
- 8% Synchronized (highlighted in a red box)
- Use the following details to connect your wallet or application to Electrs.
- Local Network (selected) / Tor
- ADDRESS: umbrel.local
- PORT: 50001
- SSL: Disabled



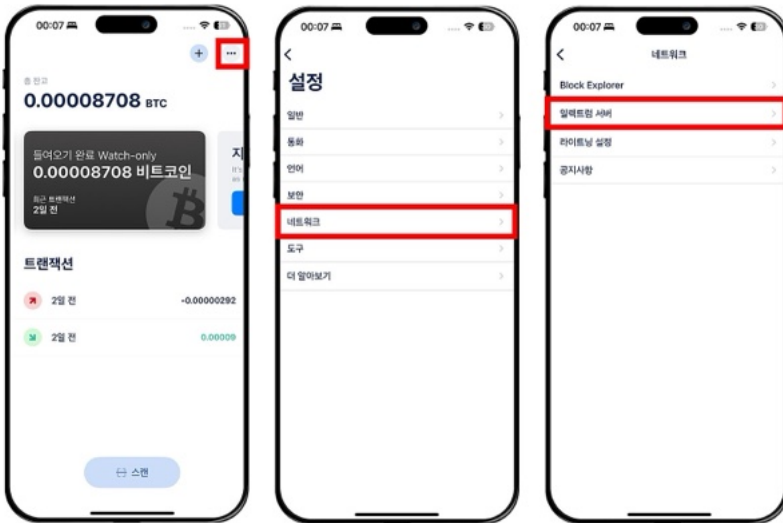
The screenshot shows the Electrs interface with the following details:

- Running
- Electrs**
- 0.10.9
- 100% Synchronized (highlighted in a red box)

블루월렛과 자신의 풀 노드 연결하기

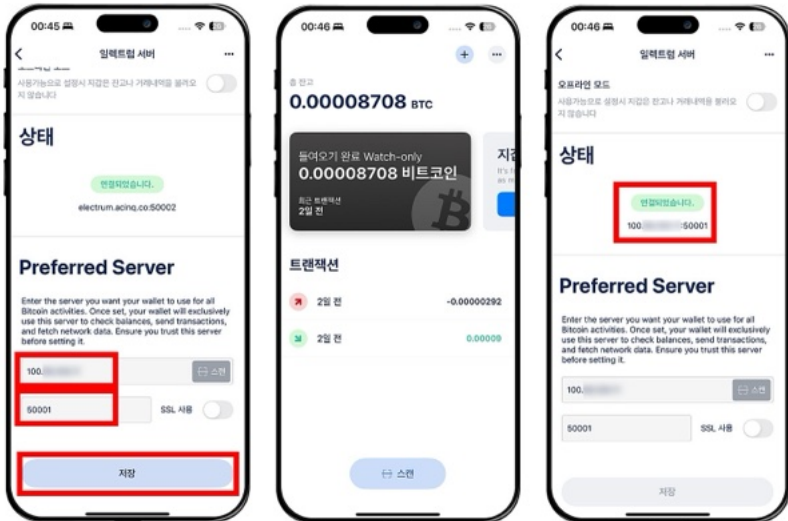
이제 테일스케일도 준비되었고, 일렉트럼 서버 동기화도 끝났으니 워치-온리 지갑과 나의 풀 노드를 연결할 수 있다.

블루월렛부터 시작해 보자. 블루월렛과 풀 노드를 연결할 때는 테일스케일이 켜져 있어야 한다. 블루월렛 오른쪽 위 점 세 개 버튼 → [네트워크] → [일렉트럼 서버]로 들어간다.



아래에 테일스케일에서 봤던 엠브렐 풀 노드의 주소를 입력하면 된다. 포트 번호는 '50001'을 입력한다. 저장을 누르고 앱을 아예 꺾다가 켜다. 백그라운드에서도 돌아가지 않도록 완전히 종료했다가 다시 실행해야 한다.

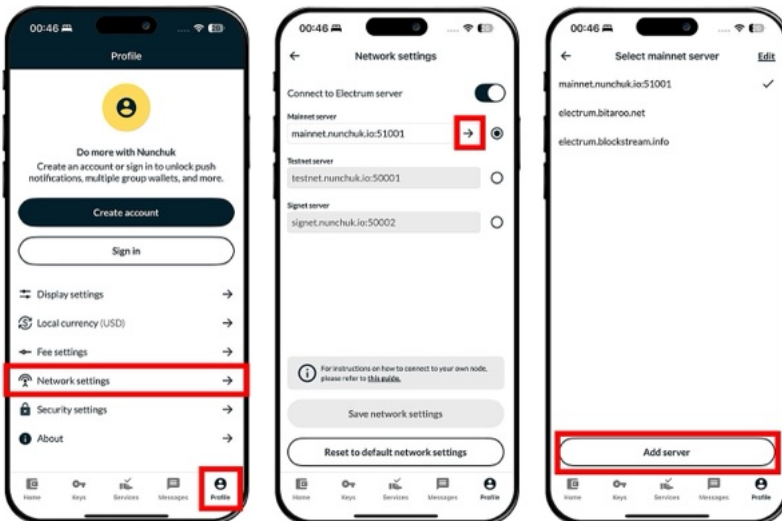
앱을 재시작한 뒤 다시 설정 → [네트워크] → [일렉트럼 서버]에 들어가 보면 테일스케일에서 제공하는 내 풀 노드 IP 주소와 잘 연결된 것을 알 수 있다.



넛척과 자신의 풀 노드 연결하기

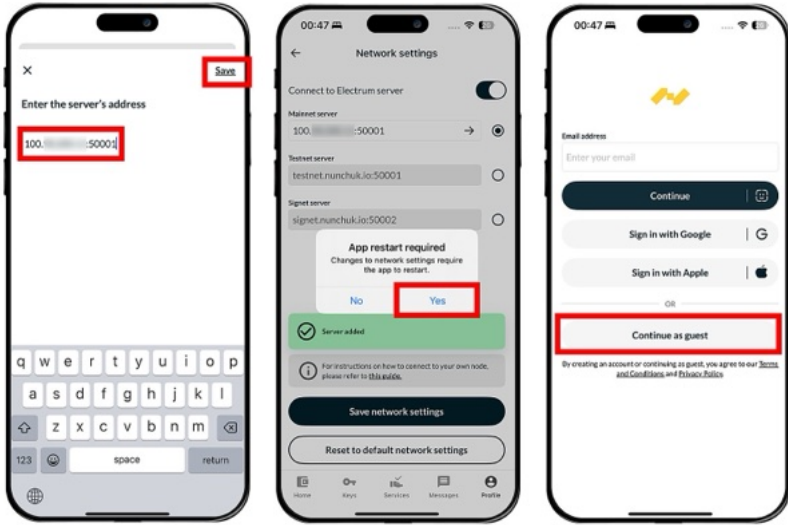
이번에는 넛척과 나의 풀 노드를 연결해 보자. 넛척과 풀 노드를 연결할 때도 테일스케일 연결이 켜져 있어야 한다. 아래 탭에서 [Profile] 탭을 선택한다.

[Network settings]를 누르고, 메인넷 서버 옆에 있는 화살표를 누른다. [Add server]를 누른다.

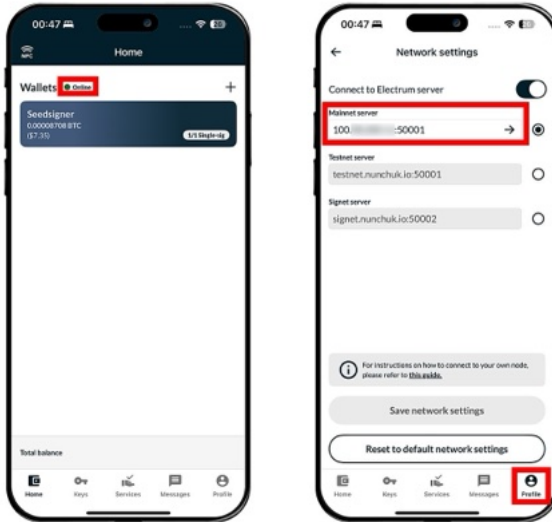


입력창이 나타나면 테일스케일에서 보여줬던 엠브렐 기기의 IP 주소를 적고, 그 뒤에 ‘:50001’을 붙인다. 그러면 100.???.???.???:50001 이런 형태가 될 것이다. 그다음 [Save]를 누른다.

앱을 재시작해야 한다는 알림이 뜨면 [Yes]를 누른다. 앱이 재시작되면 다시 [Continue as guest]를 누른다.

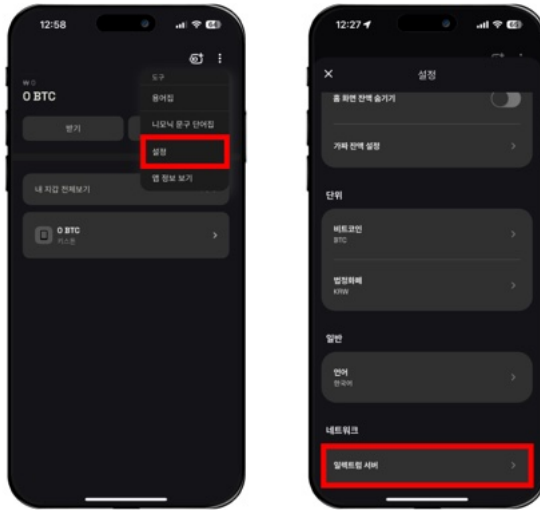


지갑 옆에 초록색 점과 함께 'Online' 표시가 뜨는 것을 알 수 있다. 다시 네트워크 설정에 들어가 보면 우리가 설정했던 엠브렐 기기의 IP 주소로 잘 연결되어 있는 것을 확인할 수 있다.

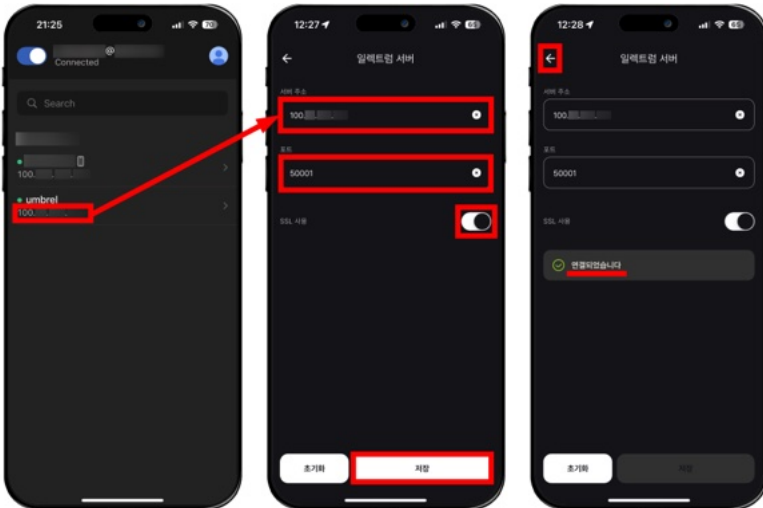


코코넛 월렛과 자신의 풀 노드 연결하기

이번에는 코코넛 월렛과 나의 풀 노드를 연결해 보자. 코코넛 월렛과 풀 노드를 연결할 때도 테일스케일 연결이 켜져 있어야 한다. 우측 상단 점 세 개 → [설정]을 누르고, 스크롤을 내려 [일렉트럼 서버]를 누른다.



테일스케일에서 봤던 엄브렐의 IP 주소를 입력한다. 포트 번호는 '50001' 그대로 둔다. 'SSL 사용'도 그냥 둔다. 테일스케일 VPN은 와이어가드로 중단간 암호화되므로 SSL을 켜든 끄든 크게 상관없다. 그다음 [저장]을 누른다. 잘 연결되면 '연결되었습니다'라는 알림이 나온다.



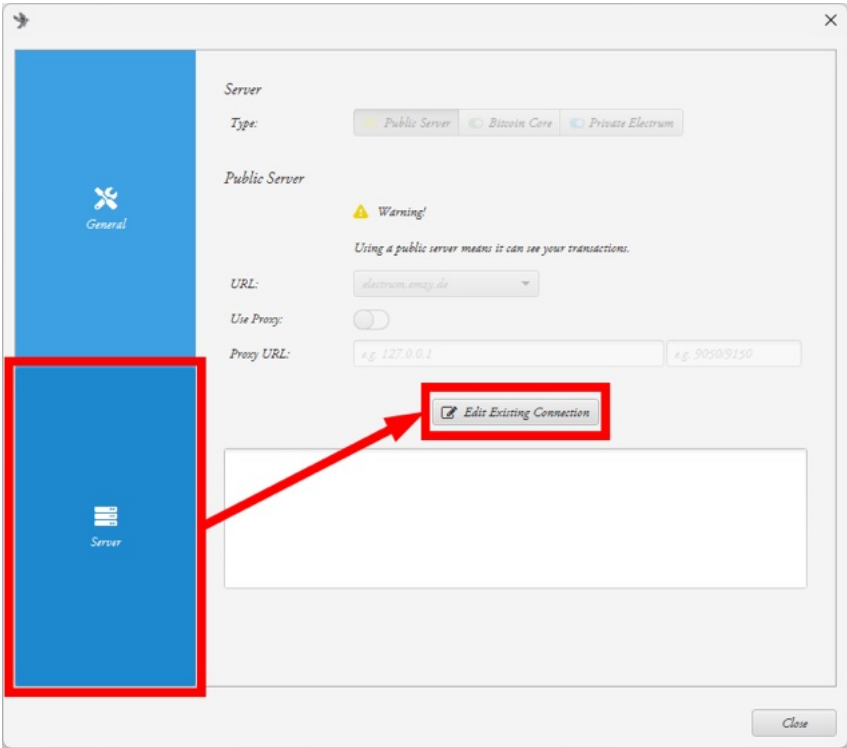
스패로우와 자신의 풀 노드 연결하기

데스크톱에서 사용하는 위치-온리 지갑인 스패로우도 로컬 네트워크에서 자신의 일렉트럼 서버와 연결할 수 있다. PC가 엠브렐 기기와 같은 공유기에 연결되어 있어야 한다.

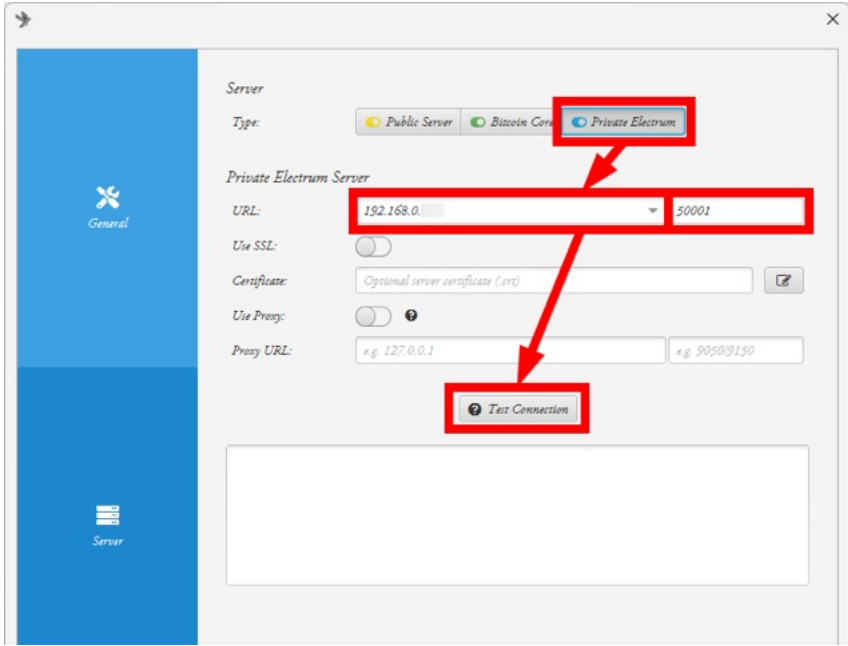
스패로우 좌측 상단에서 [File] → [Settings]에 들어간다. 맥에서는 [Sparrow] → [Settings]로 들어가면 된다.



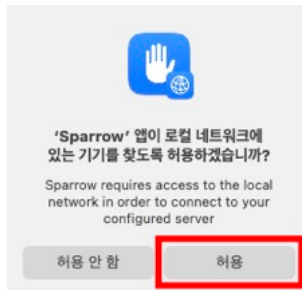
[Server]를 선택하고, [Edit Existing Connection]을 누른다.



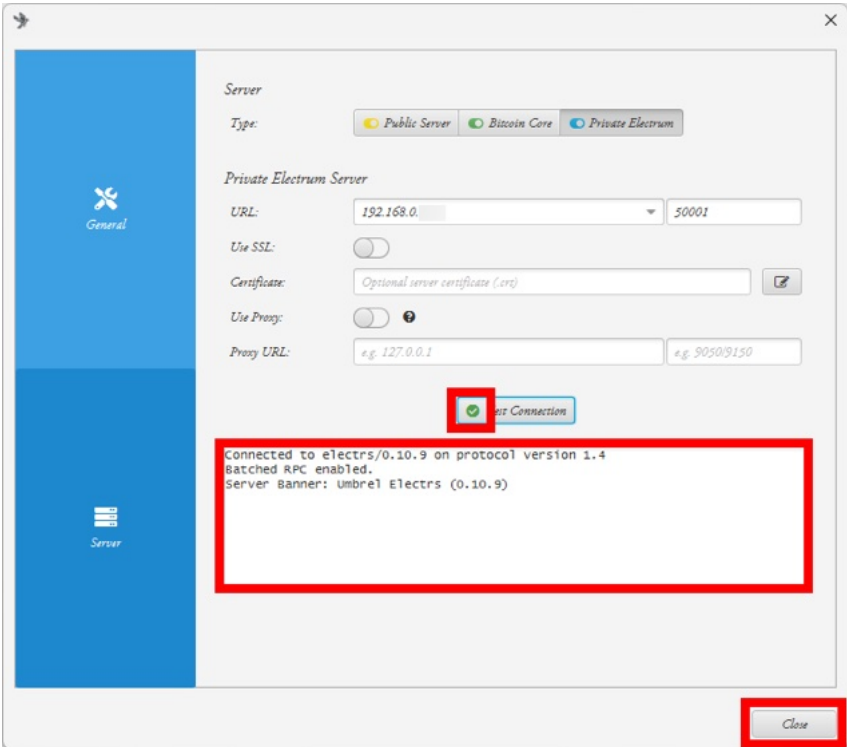
상단에서 [Private Electrum]을 선택한다. 아래 ‘URL’ 입력창에 자신의 엄브렐 기기의 로컬 IP 주소를 입력하고, 뒤에 포트 번호는 ‘50001’을 입력한다. 그다음에 [Test Connection]을 누른다.



맥의 경우 다음과 같이 로컬 네트워크에 있는 기기를 찾으려 허용할지 묻는 창이 나올 수도 있는데 그러면 [허용]을 누른다.



잘 연결되면 ‘Test Connection’ 버튼 왼쪽에 초록색 체크 표시가 나오고 아래 연결 상태가 나온다. 연결되었으면 [Close]를 누른다. 그러면 앞으로 자신의 풀 노드(일렉트럼 서버)를 통해 잔액을 조회하고 새 거래를 전파할 것이다.



연결이 잘 되면 스패로우 하단에 파란색 토글 스위치가 켜져 있는 것을 볼 수 있다.



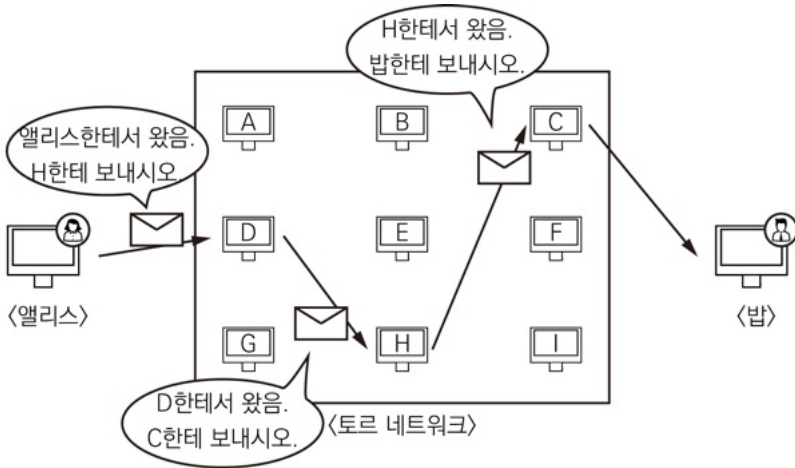
이로써 테일스케일을 이용해 워치-온리 지갑들과 내 풀 노드를 연결하는 과정을 살펴보았다. 앱의 문제로 인해 종종 연결이 안 되기도 하므로 여러 워치-온리 지갑 앱을 교차로 사용하는 것을 권장한다. 이제 토르를 통해 내 풀 노드와 워치-온리 지갑을 연결해 보도록 하겠다.

토르를 이용해 자신의 풀 노드와 워치-온리 지갑 연결하기

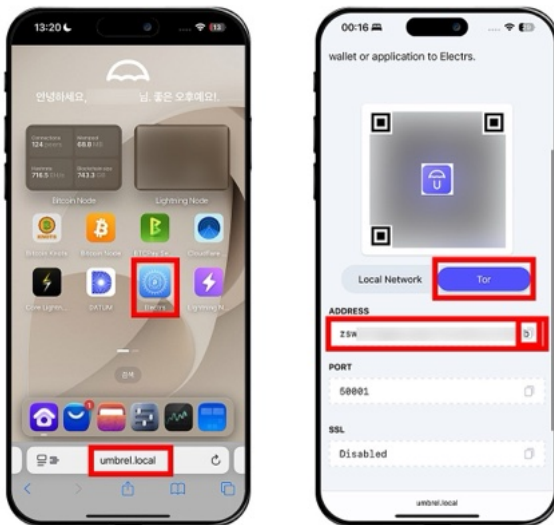
토르의 숨김 서비스를 이용하면 IP 주소를 노출하지 않고도 토르 네트워크에서 접속 가능한 .onion 주소를 생성할 수 있다. 일렉트럼 서버인 일렉터스에서도 토르를 통해 연결할 수 있다.

워치-온리 지갑과 풀 노드를 연동할 때 토르 네트워크를 사용하는 것의 장점은 테일스케일 같은 VPN을 쓰지 않고도 일렉트럼 서버에 연결할 수 있다는 점이다. 제3자에게 나의 IP 주소 등을 제공할 필요가 없게 된다.

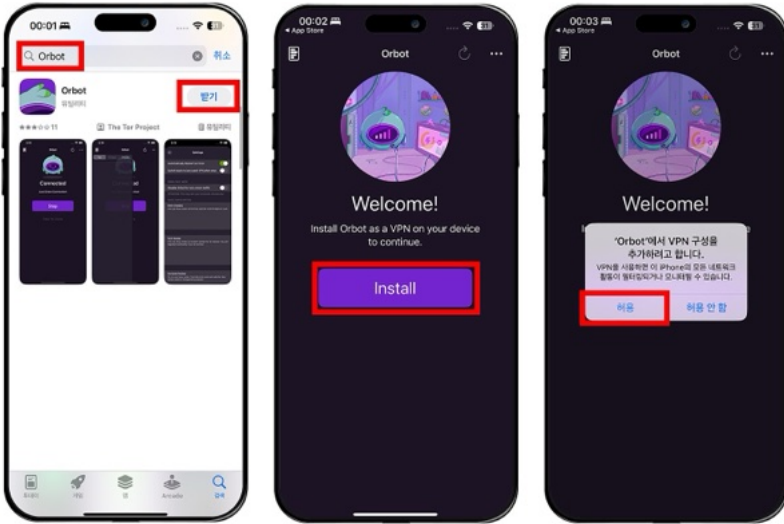
단점도 명확하다. 토르는 데이터를 주고받을 때 최종 목적지가 어디인지 모르게 하기 위해 입구 노드, 중계 노드, 출구 노드 총 3개의 노드를 거쳐 가게 된다. 입구 노드는 데이터가 어디서 온 건지 알 수 있지만 어디로 가는지 알 수 없다. 출구 노드는 데이터의 최종 목적지를 알 수 있지만 어디서 온 건지는 알 수 없다. 어쨌든 여러 개의 노드를 통해서 가다 보니 토르는 느리다.



이제 토르 네트워크를 통해 위치-온리 지갑과 일렉트럼 서버를 연결하는 방법을 알아보자. 스마트폰에서 엄브렐에 접속한 뒤 일렉트럼 앱을 누른다. 네트워크를 토르로 선택하면 아래에 숨김 서비스 주소가 나온다. 이 주소를 복사한다.



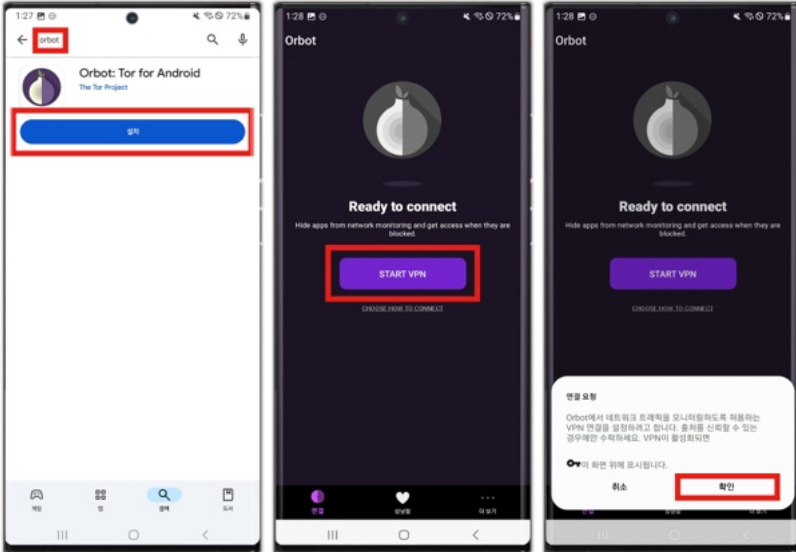
이제 앱스토어나 구글 플레이스토어에 가서 'Orbot' 앱을 다운로드한다. Orbot은 스마트폰에서 토르 네트워크를 사용할 수 있게 해주는 앱이다. 먼저 아이폰에서 Orbot을 설정하는 방법을 알아보자. Orbot 앱이 켜지면 [Install]을 누르고 VPN 구성을 [허용]한다.



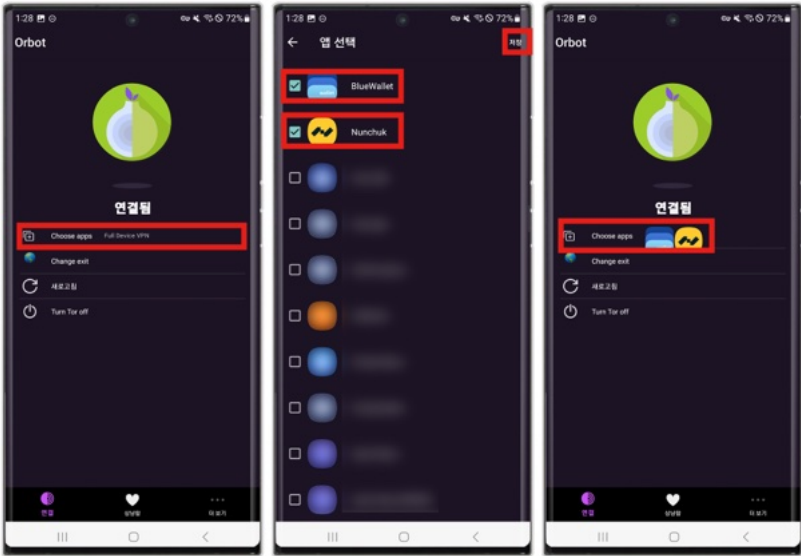
다 되면 [Start]를 누른다. 잠시 기다리면 'Connected'가 보이게 된다. 이제 토르 네트워크에 연결 가능한 상태가 되었다. 워치-온리 지갑을 다 사용하고 Orbot을 끌 때는 밑에 있는 [Stop] 버튼을 누르면 된다.



안드로이드에서는 다음과 같이 하면 된다. 구글 플레이스토어에서 ‘Orbot’을 검색하고 앱을 다운로드한다. Orbot 앱이 켜지면 [START VPN]을 누르고 VPN 연결 요청에 [확인]을 누른다.



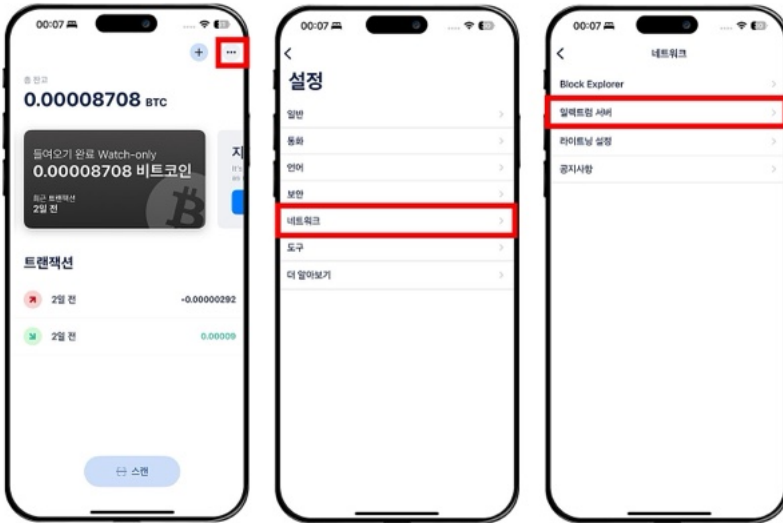
‘접속 중’이라는 문구와 함께 뜨는 로딩 창이 지나가면, ‘연결됨’이라는 문구가 보인다. [Choose apps]를 누르고 블루월렛과 넉척을 찾아 선택한 후 오른쪽 위 [저장]을 누른다. 이제 블루월렛과 넉척 앱은 토르 네트워크로 연결하는 상태가 되었다. 위치-온리 지갑을 다 사용하고 Orbot을 끌 때는 밑에 있는 [Turn Tor off] 버튼을 누르면 된다.



블루월렛에서 토르를 이용해 위치-온리 지갑 연결하기

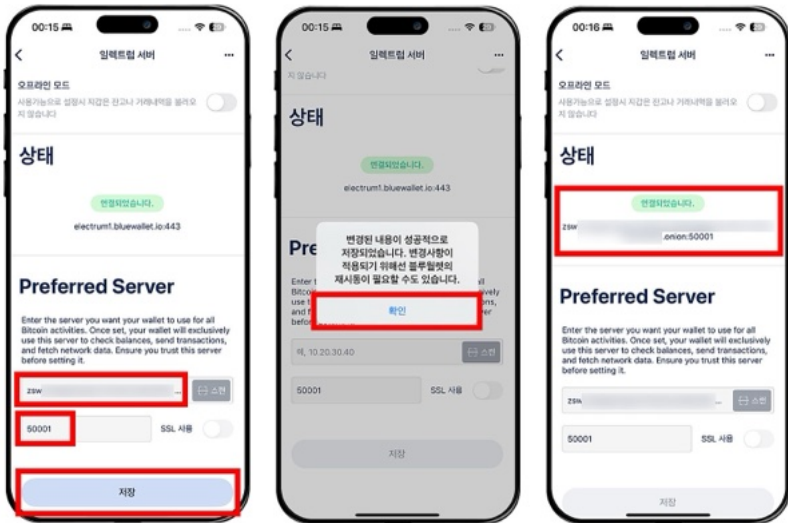
이제 블루월렛에서 토르를 이용해 자신이 운영하는 일렉트럼 서버와 연결해 보자. 연결할 때마다 Orbot이 켜져 있어야 한다.

블루월렛에 들어가 오른쪽 위 점 세 개 → [네트워크] → [일렉트럼 서버]를 누른다.



아래에 엠브렐 일렉트스에서 복사했던 토르 숨김 주소를 붙여넣기 한다. 포트 번호는 50001을 입력한다. 그다음 [저장]을 누른다.

이제 블루월렛을 완전히 재시작해야 한다. 백그라운드에서도 돌아가지 않도록 완전히 종료했다가 블루월렛을 다시 켜보자. 다시 네트워크로 들어가면 토르 네트워크를 통해 내 풀 노드의 일렉트럼 서버와 연결된 것을 알 수 있다.

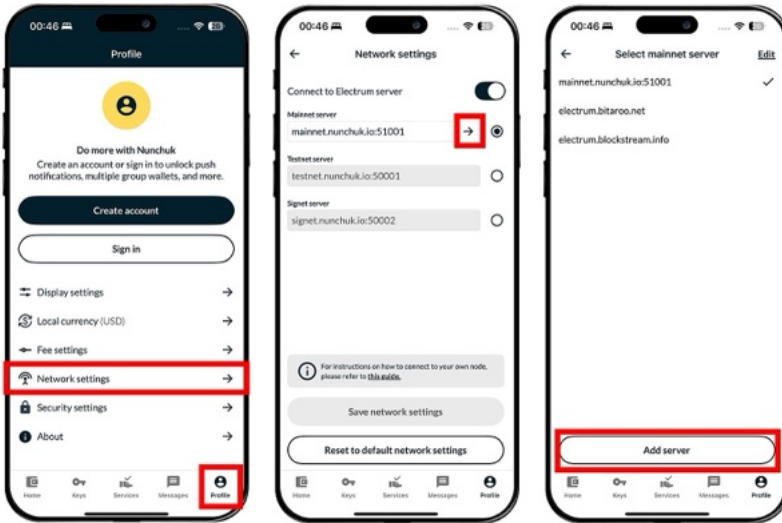


토르 네트워크는 느리므로 워치-온리 앱을 다 사용했으면 Orbot을 끄는 것이 좋다. 그리고 사용할 때 다시 Orbot을 켜면 된다.

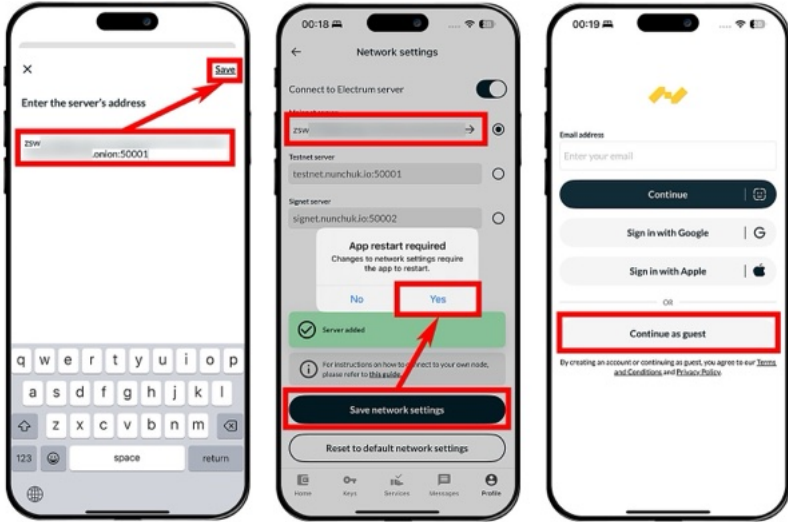
넉척에서 토르를 이용해 워치-온리 지갑 연결하기

이번에는 넉척에서 토르를 이용해 자신이 운영하는 일렉트럼 서버와 연결해 보자. 마찬가지로 연결할 때마다 Orbot이 켜져 있어야 한다.

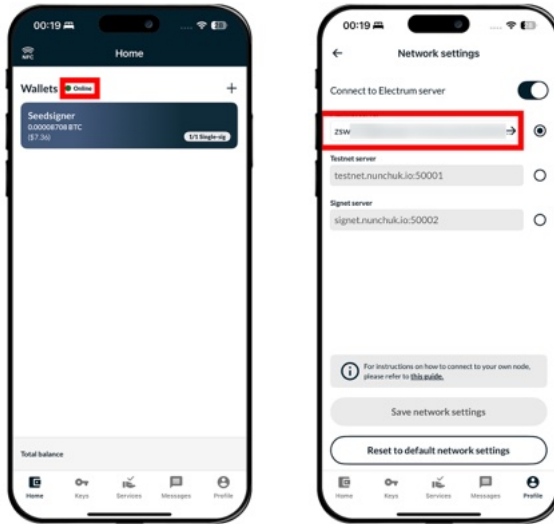
하단 탭에서 [Profile] → [Network settings] → 'Mainnet server'에 있는 화살표 → [Add server]를 누른다.



여기에 임브렐의 일렉터스에서 복사했던 토르 주소를 붙여넣기 하고 그 뒤에 ':50001'을 붙여준다. 반드시 포트 번호인 :50001까지 붙여줘야 한다. 이후 [Save] → [Save network settings] → [Yes]를 눌러 넉척 앱을 완전히 재시작한다.



넌척 앱을 재시작하고 잠시 기다리면 왼쪽 위에 초록색 점과 함께 'Online'이라는 표시가 뜰 것이다. 그러면 내 일렉트럼 서버와 잘 연결되었다는 뜻이다. 다시 네트워크 설정에 들어가 내 일렉터스의 토르 숨김 주소로 잘 설정되어 있는지 확인해 보자.



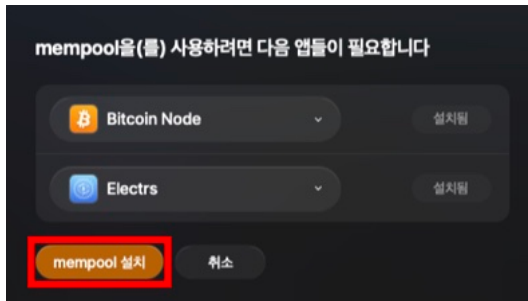
이로써 토르 네트워크를 이용해 넌척과 내 풀 노드의 일렉터스를 연결해 보는 과정도 끝났다. 토르 네트워크는 느리므로 워치-온리 앱을 다 사용했으면 Orbot을 끄는 것이 좋다. 그리고 사용할 때 다시 Orbot을 켜면 된다.

| 뎀풀과 RPC 명령어

뎀풀 앱 연결하기

비트코인을 사용하면 수수료 상황 조회 등의 이유로 뎀풀 웹사이트 (<https://mempool.space>)를 자주 이용하게 된다. 눈치챘겠지만 뎀풀 웹사이트도 누군가의 풀 노드로부터 정보를 받아오는 것이다. 따라서 여기에 자신과 관련 있는 txid를 검색하면 어떤 IP 주소에서 어떤 txid를 검색했는지 뎀풀 웹사이트와 연결된 풀 노드 운영자는 알 수 있게 된다. 자신이 운영하는 풀 노드를 통해 뎀풀 웹사이트와 똑같은 익스플로러 UI를 볼 수도 있는데 이렇게 하면 프라이버시 침해를 줄일 수 있다. 이 방법에 대해 알아보자.

엠프렐의 앱스토어에서 ‘mempool’을 설치한다. 만약 코어가 아닌 노드를 운영한다면 다음 앱들이 필요하다는 알림창에서 ‘Bitcoin Node’ 대신 ‘Bitcoin Knots’를 선택하면 된다. 설치가 완료되면 [열기]를 누른다.





그러면 자신의 풀 노드를 이용해 네트워크 상황을 조회할 수 있다. 왼쪽은 아직 채굴되지 않은 예상 블록이므로 자기 풀 노드의 멤풀 공간을 조회해 보여주는 것이고, 오른쪽은 채굴된 블록이므로 자기 풀 노드의 블록 저장 공간을 조회해 보여주는 것이다.

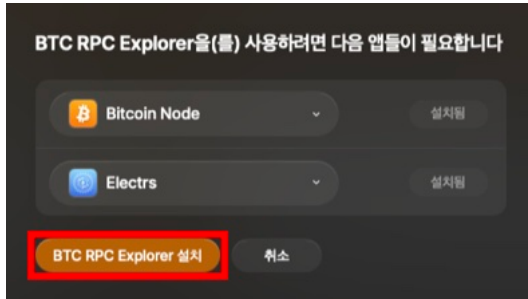


RPC 익스플로러 사용하기

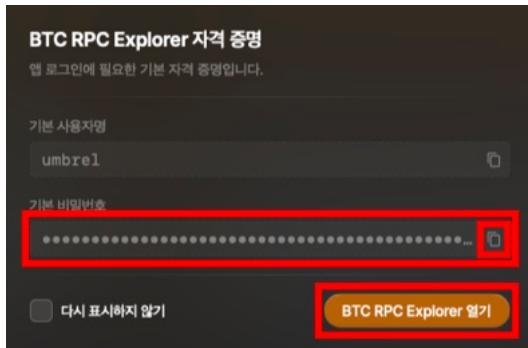
비트코인 코어 또는 노츠에서는 RPC 명령어를 이용할 수 있다. RPC 명령어를 이용하면 풀 노드의 여러 정보를 조회할 수 있다. 엠브렐OS에서는 각 앱의 터미널에 들어가서 `bitcoin-cli`를 이용해 RPC 명령어를 이용할 수 있다. 하지만 'RPC 익스플로러' 앱을 이용하면 좀 더 쉽게 명령어를 이용할 수 있다(터미널에서 RPC 명령어를 입력하는 방법은 뒤에서 알아볼 것이다).

먼저 엠브렐 앱스토어에서 'BTC RPC Explorer' 앱을 찾아 설치한다. 만약 코어가 아닌 노츠를 운영한다면 다음 앱들이 필요하다는 알림

창에서 'Bitcoin Node' 대신 'Bitcoin Knots'를 선택하면 된다. 설치가 완료되면 [열기]를 누른다.



그러면 사용자명과 비밀번호를 보여줄 것이다. 비밀번호 옆의 복사 버튼을 눌러 비밀번호를 복사한다. 그다음 [BTC RPC Explorer]를 누른다.



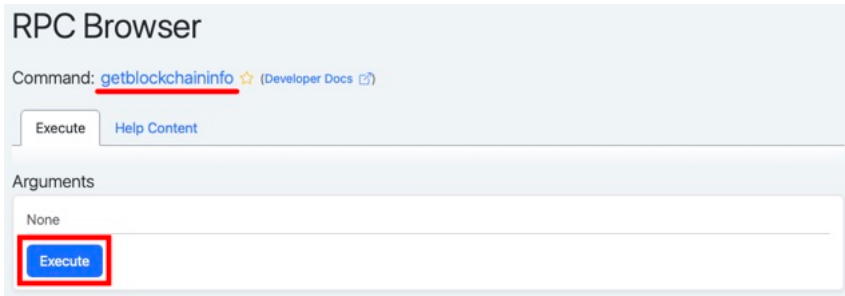
로그인하라는 창이 나올 것이다. 조금 전에 봤던 대로 사용자명예 ‘umbrel’을 입력하고, 비밀번호를 붙여넣기 한다. 그다음 [로그인]을 누른다.

192.168.0. :3002에 로그인
암호가 암호화되지 않은 상태로 전송됩니다.

스크롤을 조금 아래로 내려 [RPC Browser]를 누른다.

그러면 쓸 수 있는 RPC 명령어들이 나온다. 여기서 몇 가지 조회 관련 RPC 명령어만 설명하고 넘어가겠다.

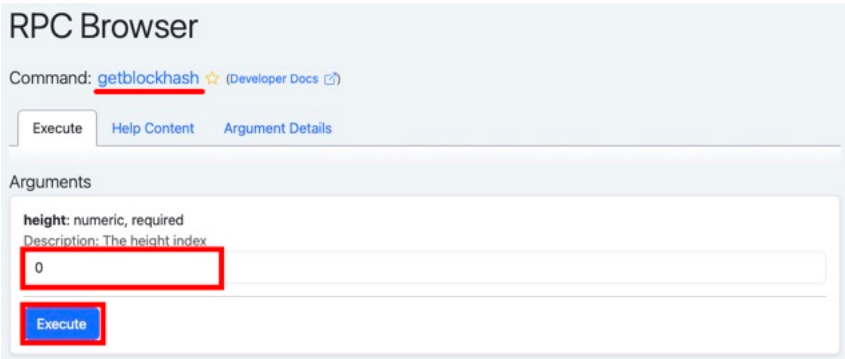
RPC 명령어를 실제로 사용해 보자. 먼저 [getblockchaininfo]를 찾아 눌러보자(Blockchain 섹션에 있다). 그리고 [Execute]를 누른다.



그러면 자신의 블록 데이터 상태가 나온다. 보통 이 명령어가 IBD가 끝나고 가장 먼저 입력해 보는 명령어다.

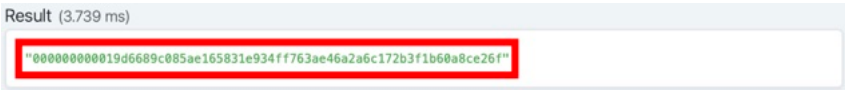


이번엔 RPC 명령어를 이용해 제네시스 블록의 정보를 조회해 보고 여기에 새겨진 문구를 찾아보자. [getblockhash]를 찾아 누른다 (Blockchain 섹션에 있다). 입력창에 0을 입력하고, [Execute]를 누른다. 제네시스 블록의 높이는 0이다.



The screenshot shows the 'RPC Browser' interface. At the top, the command 'getblockhash' is entered, with a link to 'Developer Docs'. Below the command bar, there are three buttons: 'Execute', 'Help Content', and 'Argument Details'. The 'Arguments' section is expanded, showing the parameter 'height: numeric, required' with the description 'The height index'. The input field for 'height' contains the value '0'. Below the input field, the 'Execute' button is highlighted with a red box.

그러면 0번 블록의 해시값이 나온다. 이 값을 복사한다.



The screenshot shows the 'Result' section of the RPC Browser. The result is displayed as a long hexadecimal string: '00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a0ce26f'. The entire result string is enclosed in a red rectangular box.

이제 뒤로 가서 [getblock] 명령어를 찾아본다(Blockchain 섹션에 있다). 여기에 방금 복사했던 해시값을 붙여넣는다. 그리고 아래 입력창에는 '1'을 적는다. '0'을 입력하면 직렬화된 HEX 값으로 출력하는 것이고, '1'을 입력하면 보기 편한 JSON 형식으로 출력한다.

RPC Browser

Command: [getblock](#) ☆ (Developer Docs [🔗](#))

[Help Content](#) [Argument Details](#)

Arguments

blockhash: string, required
Description: The block hash

verbosity: numeric, optional, default=1
Description: 0 for hex-encoded data, 1 for a JSON object, 2 for JSON object with transaction data, and 3 for JSON object with transaction data including prevout information for inputs

그러면 블록 정보가 나온다. 'tx' 아래에 있는 해시값이 제네시스 블록 코인베이스 거래의 txid다. 이를 복사한다.

```
Result (2.732 ms)
{
  "hash": "00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f",
  "confirmations": 910795,
  "height": 0,
  "version": 1,
  "versionHex": "00000001",
  "merkleroot": "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",
  "time": 1231006505,
  "mediantime": 1231006505,
  "nonce": 2083236893,
  "bits": "1d00ffff",
  "target": "0000000000000000000000000000000000000000000000000000000000000000",
  "difficulty": 1,
  "chainwork": "0000000000000000000000000000000000000000000000000000000000000000100010001",
  "nTx": 1,
  "nextblockhash": "00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048",
  "strippedsize": 285,
  "size": 285,
  "weight": 1140,
  "tx": {
    "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"
  }
}
```

이제 뒤로 가서 [getrawtransaction] 명령어를 찾아 누른다 (Rawtransactions 섹션에 있다). 여기에 방금 복사한 txid를 붙이고 [Execute]를 눌러본다. 보통 아래에 있는 블록 해시값도 입력해야 하지만, 소비되지 않은 출력은 블록 해시값을 입력하지 않아도 된다. 입력하지 않으면 UTXO 세트에서 찾기 때문이다(-txindex 속성의 기본값이 0인데, 이 값이 0이면 블록 해시값을 입력하지 않았을 때 UTXO 세트에서 찾는다).

RPC Browser

Command: [getrawtransaction](#) ☆ (Developer Docs)

[Execute](#) [Help Content](#) [Argument Details](#)

Arguments

txid: string, required
Description: The transaction id

verbosity: numeric, optional, default=0
Description: 0 for hex-encoded data, 1 for a JSON object, and 2 for JSON object with fee and prevout

blockhash: string, optional
Description: The block in which to look for the transaction

[Execute](#)

그러면 에러가 뜨는 것을 볼 수 있다. 이는 제네시스 블록은 검증 대상이 아니라서 여기에 있는 코인베이스 출력이 UTXO 세트에 들어가지 않기 때문이다. 그래서 제네시스 블록에 있는 채굴 보상은 영원히 쓸 수가 없다. 비트코인에는 철저히 사전 채굴이 없었던 것이다. 이는 터미널에서 명령어를 입력해 보면 좀 더 자세한 설명이 나온다. 그러면 제네시스 블록에 새겨진 문구는 어떻게 확인할 수 있을까?

```
Result (2.016 ms)
{
  "Error": "No response from node."
}
```

다시 뒤로 가서 [getblock]을 찾는다(Blockchain 섹션에 있다). 여기에 제네시스 블록의 해시값을 입력한다. 제네시스 블록의 해시값은 [getblockhash] 명령어를 이용해 찾을 수 있었다. 그리고 아래에 '2'를 입력한다. '2'를 입력하면 블록에 실려있는 거래 정보까지 함께 보여준다.

RPC Browser

Command: [getblock](#) (Developer Docs)

Execute Help Content Argument Details

Arguments

blockhash: string, required
Description: The block hash

verbosity: numeric, optional, default=1
Description: 0 for hex-encoded data, 1 for a JSON object, 2 for JSON object with transaction data, and 3 for JSON object with transaction data including prevout information for inputs

Execute

그러면 결과값 중 'coinbase' 아래에 어떤 HEX값이 있는 것을 볼 수 있다. 인터넷에 'HEX 디코딩'이라 검색하고 아무 웹사이트에 들어가 이 값을 복사, 붙여넣기 해보자. 그러면 제네시스 블록에 새겨진 문구를 발견할 수 있을 것이다.

```
Result (1.98 ms)
{
  "hash": "00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f",
  "confirmations": 918795,
  "height": 0,
  "version": 1,
  "versionHex": "00000001",
  "merkleroot": "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",
  "time": 1231006505,
  "mediantime": 1231006505,
  "nonce": 2083236893,
  "bits": "1d00ffff",
  "target": "0000000000000000000000000000000000000000000000000000000000000000",
  "difficulty": 1,
  "chainwork": "0000000000000000000000000000000000000000000000000000000000000000",
  "nTx": 1,
  "nextblockhash": "0000000039a8e6886ab5951d76f411475428afc90947ee320161bbf18eb048",
  "strippedsize": 285,
  "size": 285,
  "weight": 1140,
  "tx": [
    {
      "txid": "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",
      "hash": "4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",
      "version": 1,
      "size": 204,
      "vsize": 204,
      "weight": 816,
      "locktime": 0,
      "vin": [
        {
          "coinbase":
            "04ffff00100104455468052054696d65732030332f4a616e2f32303039204368616e6365666c6672206f6e206272696e6b206f662027
            365636f6e64206261696c6f757420666f722062616e6b73",
          "sequence": 4294967295
        }
      ]
    }
  ]
}
```

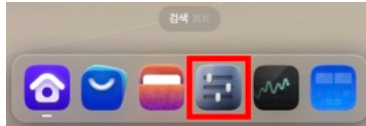
Output

00 The Times 03/Jan/2009 Chancellor on brink of second bailout for banks

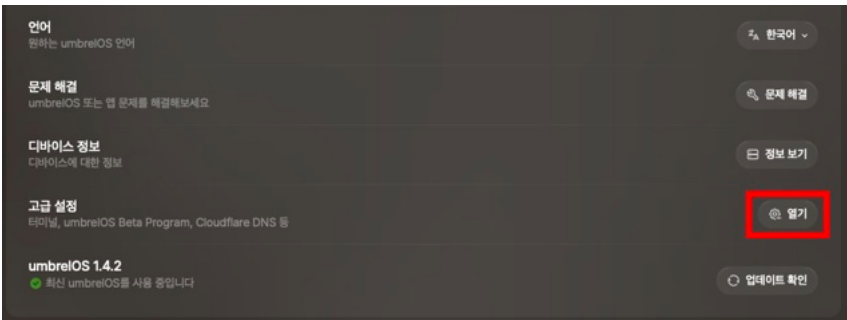
터미널에서 RPC 명령어 사용하기

RPC 익스플로러 앱을 이용하면 RPC 명령어를 쉽게 이용할 수 있었다. 이번에는 조금 어렵지만 터미널에서 RPC 명령어를 이용하는 방법을 알아보자.

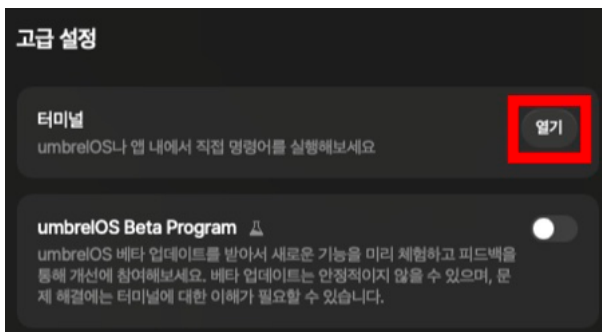
연브렐 홈 화면의 하단 탭에서 설정 버튼을 누른다.



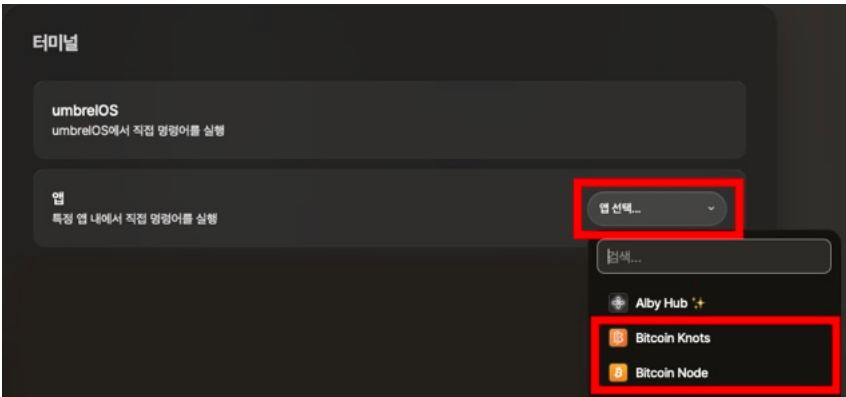
스크롤을 내리고 '고급 설정' 옆의 [열기]를 누른다.



'터미널' 옆의 [열기]를 누른다.



[앱 선택]에서 코어 운영자는 [Bitcoin Node]를, 노즈 운영자는 [Bitcoin Knots]를 누른다.



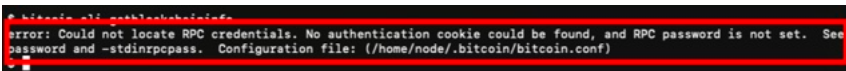
그러면 터미널이 나온다. 터미널에서는 앞에 `bitcoin-cli`를 붙이고 명령어를 쓰면 된다. 여기에 다음 명령어를 입력해 본다.

```
bitcoin-cli getblockchaininfo
```



```
$ bitcoin-cli getblockchaininfo
```

코어에서는 다음과 같은 오류가 날 수도 있다. 이는 엄브렐에서 코어를 설치하면 일반적으로 코어를 설치했을 때의 폴더 경로와 조금 달라져서 그렇다. RPC 명령어를 로컬에서 호출할 때는 `.cookie` 파일에 임시 RPC 사용자 이름과 비밀번호가 있어서 따로 권한 부여 없이 이용할 수 있다. 따라서 `.cookie` 파일이 있는 경로를 제대로 지정해 주어야 한다.



경로를 제대로 지정해서 명령어를 써보자. 경로를 지정할 때는 - datadir=를 사용하면 된다.



```
bitcoin-cli -datadir=/data/bitcoin  
getblockchaininfo
```

```
$ bitcoin-cli -datadir=/data/bitcoin getblockchaininfo  
{  
  "chain": "main",  
  "blocks": 910793,  
  "headers": 910793,  
  "bestblockhash": "00000000000000000209da7a5c7a22455fff2abd1662823b2e6875d072fcc",  
  "bits": "17022cb3",  
  "target": "0000000000000000022cb30000000000000000000000000000000000000000000",  
  "difficulty": 129435235580344.8,  
  "time": 1755634260,  
  "mediantime": 1755631257,  
  "verificationprogress": 0.9999955146114813,  
  "initialblockdownload": false,  
  "chainwork": "0000000000000000000000000000000000000000000000000000000000000000",  
  "size_on_disk": 774544685481,  
  "pruned": false,  
  "warnings": [  
  ]  
}
```

매번 이렇게 경로를 지정해서 bitcoin-cli를 사용하는 것은 불편한 일일 것이다. 그때는 별칭을 등록해 두면 된다(명령어 단축키를 등록하는 거라고 생각하면 된다). 별칭을 등록할 때는 alias를 사용하면 된다. 다만 등록된 별칭은 터미널을 껐다 켜면 없어진다. 다음 명령어는 bitcoin-cli를 입력하면 자동으로 bitcoin-cli -datadir=/data/bitcoin으로 바꾸라는 뜻이다.



```
alias bitcoin-cli="bitcoin-cli  
-datadir=/data/bitcoin"
```

```
$ alias bitcoin-cli="bitcoin-cli -datadir=/data/bitcoin"
```

별칭을 등록해 놓으면 다음과 같이 bitcoin-cli만 써도 제대로 된 경로를 통해 명령어를 호출한다.



bitcoin-cli getblockchaininfo

```
$ bitcoin-cli getblockchaininfo
{
  "chain": "main",
  "blocks": 910793,
  "headers": 910793,
  "bestblockhash": "0000000000000000000000000000000000000000000000000000000000000000",
  "bits": "17022cb3",
  "target": "0000000000000000000000000000000000000000000000000000000000000000",
  "difficulty": 129435235580344.0,
  "time": 1755694260,
  "mediantime": 1755631257,
  "verificationprogress": 0.9999952664687813,
  "initialblockdownload": false,
  "chainwork": "0000000000000000000000000000000000000000000000000000000000000000",
  "size_on_disk": 774544685481,
  "pruned": false,
  "warnings": [
  ]
}
```

노츠에서는 경로 문제 외에도 다음과 같은 오류가 날 수 있다. 이것은 RPC 명령어를 호출하는 포트가 잘못되어서 그렇다. 코어나 노츠나 일반적으로 외부 통신에는 8333 포트를, 내부 RPC 호출에는 8332를 쓰지만 임브렐OS에서는 코어와 노츠 포트 구분을 위해 노츠는 외부 통신에는 9333, 내부 RPC 호출에는 9332를 쓴다.

```
$ bitcoin-cli getblockchaininfo
error: timeout on transient error: Could not connect to the server 127.0.0.1:8332
Make sure the bitcoind server is running and that you are connecting to the correct RPC port.
Use "bitcoin-cli -help" for more info.
$
```

따라서 -datadir=를 이용해 경로를 제대로 지정해 주어야 할 뿐만 아니라 -rpcport=를 이용해 포트 번호도 제대로 적어줘야 한다. 경로와 포트를 제대로 지정해서 써보자.



bitcoin-cli -rpcport=9332 -datadir=/data/bitcoin
getblockchaininfo

| 도달 가능한 노드 되기

인바운드 연결을 허용하는 노드를 도달 가능한 노드(reachable node)라고 한다. 도달 가능한 노드가 되는 것은 진정으로 비트코인 네트워크에 기여하기 위해 풀 노드를 돌린다는 뜻이다. 이렇게 되면 IBD (초기 블록 다운로드)를 하는 노드들에게 자기 노드의 데이터를 나눠줄 수 있게 된다.

대한민국은 인터넷 강국으로 전 지역에서 빠른 속도로 인터넷을 이용할 수 있다. 따라서 대한민국에서 풀 노드를 구축하고, 도달 가능한 노드가 된다면 비트코인 네트워크에 크게 기여할 수 있다.

인지하고 넘어가야 할 사실이 있다. 도달 가능한 노드는 IP 주소가 노출된다. IP 주소가 노출되면 해당 풀 노드가 어느 지역에 있는지와 어느 인터넷 서비스를 이용하는지 대략적으로 알 수 있다. 프라이버시 수준이 낮아지는 것이다. 또한, IP 주소가 노출되면 해커의 표적이 될 수도 있다. 하지만 많은 노드들이 도달 가능한 상태가 된다면 이런 문제는 희석될 수 있다. 10명 속에 숨은 1명을 찾는 것과 1,000명 속에 숨은 1명을 찾는 것을 생각해 보면 된다.

과거 한국에서 도달 가능한 풀 노드를 극소수만 돌리던 시절에는 프라이버시 문제가 컸다. 비트코이너들은 가뜰이나 프라이버시를 중요하게 생각하는데 누가 도달 가능한 노드가 되려고 했겠는가? 그러나 그러한 문제를 알고서도 비트코인 네트워크에 기여하기 위해 희생했던 소수가 있었기에 현재 우리가 마음 놓고 도달 가능한 노드를 돌릴 수 있다. 도달 가능한 풀 노드를 돌리는 것이 당연시되고 이런 사람들이 많아진다면 앞서 말한 우려들은 사라질 것이다. 한 사람이 도달 가능한 풀 노

드를 운영함으로써 다음 사람은 더 마음 놓고 도달 가능한 풀 노드가 될 수 있는 것이다.

인지해야 할 또 다른 점은 인바운드 허용 노드가 되면 최대 125개 노드와 연결될 수 있으므로 인터넷 사용 환경에 따라 인터넷 속도가 느려질 수도 있다는 것이다. 이때는 공유기나 모뎀에서 풀 노드 기기에 대한 QoS 설정을 하는 것이 필요하다.

도달 가능한 노드 설정은 반드시 프라이버시와 네트워크 설정에 대해 이해하고 있는 사람만 도전하기를 권고한다.

자기 노드가 도달 가능한 노드인지 확인해 보기

먼저 자신의 풀 노드가 도달 가능한 노드인지 아닌지 확인해 보자.

다음 웹사이트에 접속한다.

<https://bitnodes.io/>



스크롤을 조금 아래로 내려보면 [CHECK NODE] 버튼이 있다. 자신의 외부 IP 주소는 자동으로 입력이 된다. 버튼을 누르고 조금 기다려보면 결과가 나온다. 엠브렐OS에서 코어의 경우 포트 번호는 8333이고, 노츠의 경우 포트 번호는 9333이다.

JOIN THE NETWORK

Be part of the Bitcoin network by running a Bitcoin full node, e.g. [Bitcoin Core](#).

<input type="text"/>	8333	<input type="button" value="CHECK NODE"/>
----------------------	------	---

Use this tool to check if your Bitcoin client is currently accepting incoming connections from other nodes. Port must be between 1024 and 65535.

⦿ Please wait, this can take up to 30 seconds.

코어의 경우 다음과 같이 ‘<IP 주소>:8333/버전/블록 높이’가 뜬다면 이미 도달 가능한 노드라는 뜻이다. 노츠의 경우 <IP 주소>:9333/버전/블록 높이’가 뜬다면 이미 도달 가능한 노드라는 뜻이다.

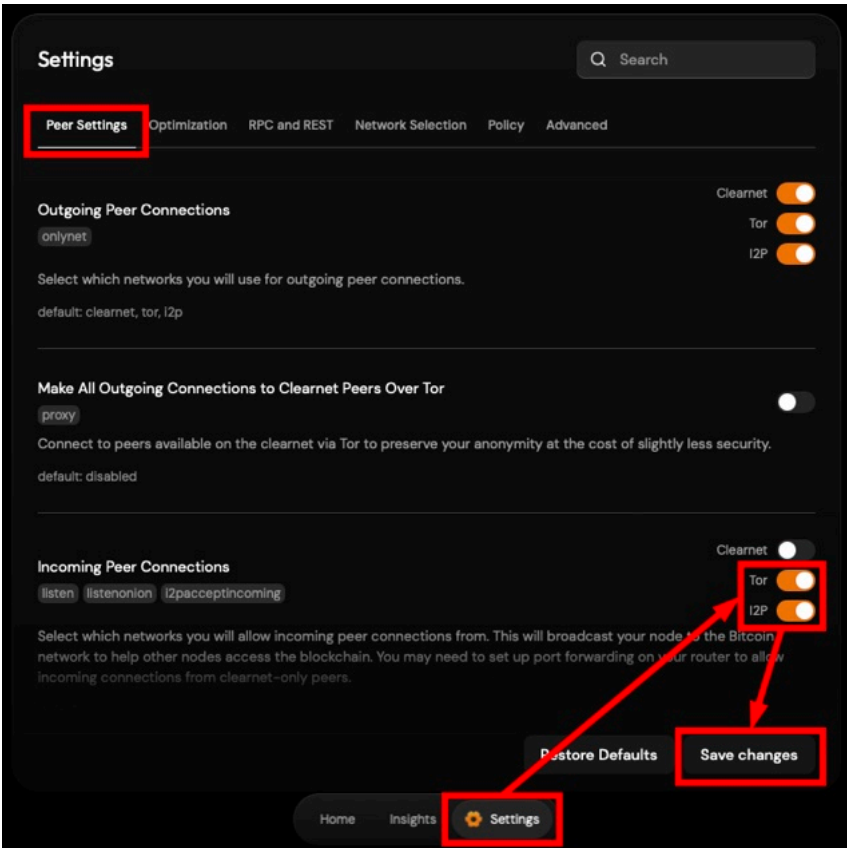
```
● :8333/Satoshi28.0.0/(Height:881575)
```

하지만 다음과 같이 ‘<IP 주소>:8333 is unreachable’(코어) 또는 ‘<IP 주소>:9333 is unreachable’(노츠)이라고 뜨면 도달 가능한 노드가 아니라는 뜻이다.

```
● :8333 is unreachable.
```

익명 네트워크에서 도달 가능한 노드 되기

토르나 I2P와 같은 익명 네트워크에서는 도달 가능한 노드가 되기 쉽다. 비트코인 코어 혹은 비트코인 노츠에서 하단 [Settings] → 'Incoming Peer Connections'에서 [Tor] 또는 [I2P]의 토글 스위치를 켜주면 된다. 그다음 [Save changes]를 누르면 설정이 끝난다.



✔ Settings applied

도달 가능한 노드가 되었는지 확인해 보자. RPC 명령어 중 `getnetworkinfo`를 입력한다. RPC 명령어를 사용하는 방법은 앞의 ‘RPC 익스플로러 사용하기’와 ‘터미널에서 RPC 명령어 사용하기’ 절을 참고하라.



```
bitcoin-cli getnetworkinfo
```

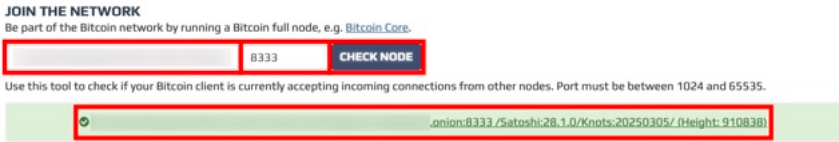
```
$ bitcoin-cli getnetworkinfo
```

RPC 명령어를 입력하고 스크롤을 내리면 `localaddresses`가 있다. 여기에 `.onion`으로 끝나는 주소가 토르 주소이고, `.b32.i2p`로 끝나는 주소가 I2P 주소다. 여기서 ‘score’가 1 이상이면 해당 네트워크로 리처블 노드가 되었다는 뜻이다.

```
"localaddresses": [
  {
    "address": " ",
    "port": 8333,
    "score": 4,
    "address": " ".onion"
  },
  {
    "address": " ",
    "port": 0,
    "score": 4,
    "address": " ".b32.i2p",
  }
],
"warnings": [
]
```

토르 네트워크는 도달 가능한 노드 여부를 `bitnodes` 웹사이트에서 도 조회할 수 있다. 왼쪽 입력창에 ‘`.onion`’으로 끝나는 토르 주소를 입력하고 포트 번호에 8333을 입력한 뒤(엠프렐에서 토르 연결은 코어나 노츠나 포트 번호가 8333으로 동일하다) [CHECK NODE]를 누르면 된다. 그러면 아래에 도달 가능한 노드라고 나오는 것을 볼 수 있다. 참고로 I2P 네트워크는 내부 네트워크인데, `bitnodes`는 클리어넷 기반

웹사이트이므로 클리어넷과 토르 네트워크에서의 도달 여부만 확인할 수 있다.



클리어넷에서 도달 가능한 노드 되기

클리어넷에서 도달 가능한 노드가 되려면 포트 포워딩 설정이 필요하다. 그러나 반드시 포트 8333 혹은 9333을 열 필요는 없다. 다른 포트 번호로 열고 bitnodes.io에서 조회 한 번만 해도 된다. 다른 포트 번호를 쓰면 인터넷 서비스 업체가 비트코인 풀 노드를 돌리고 있는 사람을 추적하기가 더 어려워진다. 여기서는 8333 포트를 여는 것을 기준으로 작성하겠다. 노츠의 경우 여기에 나오는 모든 포트 번호를 9333으로 바꿔 진행하면 된다.

와이파이가 보급된 현대에 공유기를 안 쓰는 사람은 거의 없을 것이다. 포트를 열려면 공유기에서 설정을 해야 한다. 만약 공유기를 안 쓰고 벽에서 나오는 랜선을 바로 사용한다면 모뎀에서 포트 포워딩 설정을 해야 한다. 대한민국에서는 주로 'TP링크'나 'IP타임' 공유기를 사용한다. 두 공유기에 한해 공유기 설정을 어떻게 해야 하는지 알아보자. 인터넷 서비스 업체의 공유기를 사용하는 경우도 크게 다르지 않다. 이 경우는 뒤에 짚막하게 서술했다.

다음은 엠브렐OS의 '비트코인 노드(코어)' 혹은 '비트코인 노츠'를 통해 풀 노드 운영을 하는 경우 클리어넷에서 도달 가능한 노드가 되는 방법이다.

TP링크 공유기: DHCP 서버 설정, 포트 포워딩

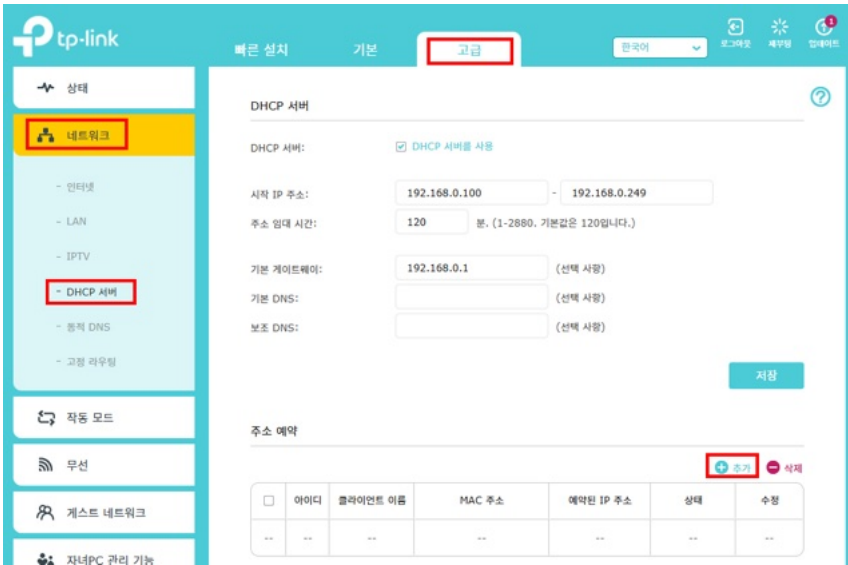
공유기 설정은 보통 인터넷 주소창에 192.168.0.1 혹은 192.168.1.1 을 입력하면 들어갈 수 있다. 만약 안 들어가진다면, 윈도우+R 키 → cmd 입력 후 실행 → **ipconfig** 입력 → 엔터를 누르면 IP 주소가 나타난다. 여기서 XXX.XXX.XXX.??? 주소가 나오면 ??? 부분만 1로 고치면 된다.



TP링크 공유기를 사용하는 경우 공유기 설정 비밀번호는 TP링크 계정의 비밀번호다. 와이파이 비밀번호가 아니다. 만약 WIFI 비밀번호를 바꾸는 등의 공유기 설정을 따로 한 적이 없다면 비밀번호는 'admin'이다.



로그인했다면 [고급] → [네트워크] → [DHCP 서버]에 들어간다. 그리고 '주소 예약' 밑에 있는 [추가]를 누른다.



'스캔' 버튼을 누르면 공유기에 연결되어 있는 여러 장치의 목록이 뜰 것이다. 그중 'umbrel'을 찾고 [선택]을 누른다.

DHCP 클라이언트 목록

아이디	클라이언트 이름	MAC 주소	IP 주소	조작
4				선택
5				선택
6	umbrel		192.168.	선택
7				선택
8				선택

< 1 2 >

그러면 엮브렐 기기의 MAC 주소와 로컬 IP 주소가 자동으로 채워질 것이다. [저장]을 누른다. 이때 로컬 IP 주소를 복사해 놓는다.

주소 예약

+ 추가 - 삭제

<input type="checkbox"/>	아이디	클라이언트 이름	MAC 주소	예약된 IP 주소	상태	수정
--	--	--	--	--	--	--

MAC 주소: 스캔

IP 주소: 192.168.

사용

취소
저장

이제 엮브렐 기기에 고정 로컬 IP 주소를 할당했다. '상태'가 전구 아이콘으로 표시된다면 설정이 잘 된 것이다.

주소 예약

+ 추가 - 삭제

<input type="checkbox"/>	아이디	클라이언트 이름	MAC 주소	예약된 IP 주소	상태	수정
<input type="checkbox"/>	1	umbrel	[MAC 주소]	192.168. [IP 주소]	💡	📄 🗑️

이제 포트 포워딩을 설정해 8333 포트(노츠는 9333)를 열어줘야 한다. 왼쪽의 [NAT 포워딩] 카테고리에 있는 [가상 서버]를 누른다.

The screenshot shows the TP-Link web interface. The top navigation bar includes '빠른 설치', '기본', and '고급' (highlighted with a red box). The left sidebar contains various settings categories, with 'NAT 포워딩' highlighted in yellow. Under 'NAT 포워딩', the sub-option '- 가상 서버' is highlighted with a red box. The main content area displays a table for '가상 서버' (Virtual Servers).

<input type="checkbox"/>	아이디	서비스 타입	외부 포트	내부 IP	내부 포트	프로토콜	상태	수정
--	--	--	--	--	--	--	--	--

[추가]를 누르고 각각의 항목을 입력한다. '서비스 타입'에는 'Bitcoin'을 입력하고, '외부 포트'와 '내부 포트' 모두 '8333'(노츠는 '9333')을 입력한다. '내부 IP'는 아까 DHCP 설정할 때 복사했던, 임브렐 기기의 로컬 IP 주소를 붙여넣는다. '프로토콜'은 'ALL'로 설정한다. 'TCP'로 설정해도 무방하다. [사용]을 체크하고 [저장]을 누른다.

가상 서버

+ 추가
 - 삭제

□	아이디	서비스 타입	외부 포트	내부 IP	내부 포트	프로토콜	상태	수정
--	--	--	--	--	--	--	--	--

서비스 타입:

외부 포트: (XX-XX 또는 XX)

내부 IP:

내부 포트: (XX 또는 빈 칸, 1-65535)

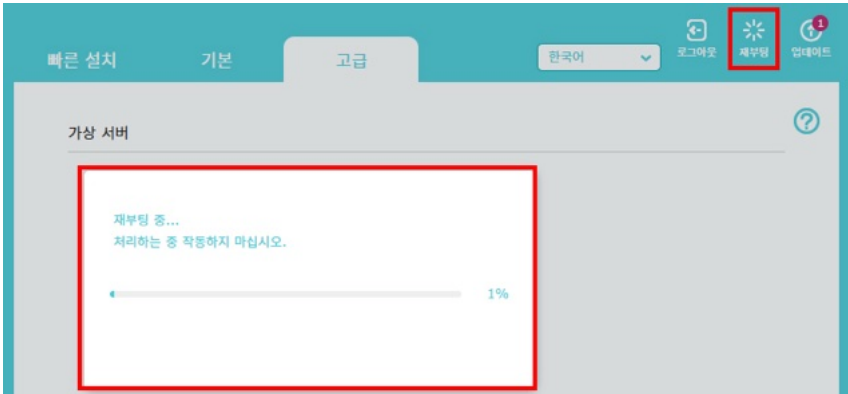
프로토콜: ▼

사용

기존 서비스 보기

취소
저장

여기까지 설정이 되었다면 공유기를 재부팅한다. 오른쪽 위 [재부팅]을 누른다. 재부팅되는 약 1-5분간 인터넷 사용이 안 될 수도 있다. 설정이 다 되었다면 ‘엠프렐 인바운드 연결 허용 및 방화벽 해제’ 절로 넘어가자.



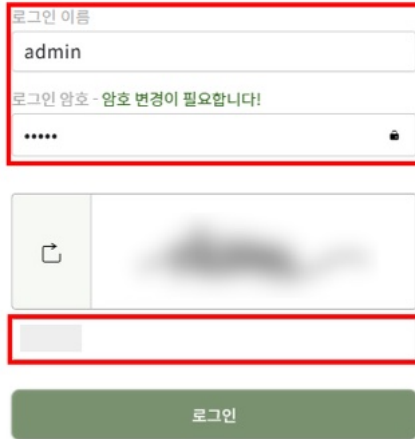
IP타임 공유기: DHCP 서버 설정, 포트 포워딩

IP타임 공유기를 쓰는 경우 포트 포워딩 방법에 대해 알아보자. 공유기 설정은 보통 인터넷 주소창에 192.168.0.1 혹은 192.168.1.1을 입력하면 들어갈 수 있다. 만약 안 들어가진다면, 윈도우+R 키 → cmd 입력 후 실행 → ipconfig 입력 → 엔터를 누르면 IP 주소가 나타난다. 여기서 XXX.XXX.XXX.??? 주소가 나오면 ??? 부분만 1로 고치면 된다.

192.168.0.1,

IP타임 공유기를 사용하는 경우 공유기 설정에 접속하려면 기존에 설정했던 관리자 ID와 비밀번호를 입력해야 한다. 와이파이 비밀번호가 아니다. 만약 WIFI 이름과 비밀번호를 바꾸는 등의 공유기 설정을 따로 한 적이 없다면 ID와 비밀번호는 모두 'admin'이다.

Login

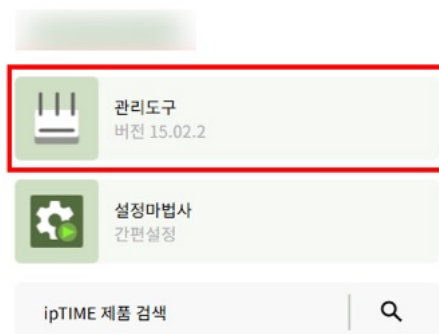


The login form consists of several elements:

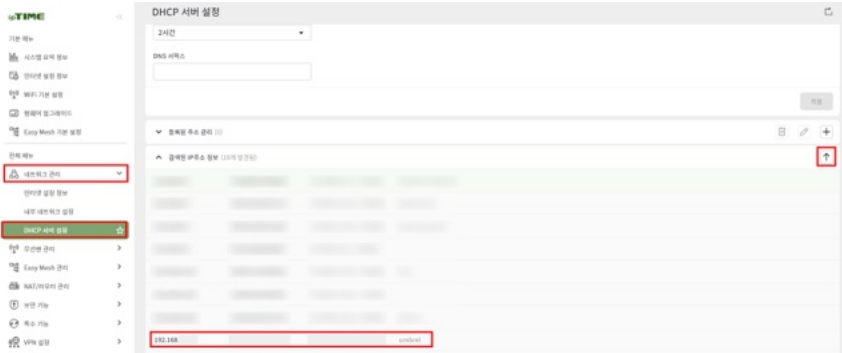
- A text input field labeled "로그인 이름" (Login Name) containing the text "admin".
- A text input field labeled "로그인 암호 - 암호 변경이 필요합니다" (Login Password - Password change is required) containing six dots ".....".
- A "로그인" (Login) button at the bottom.

Red boxes highlight the "로그인 이름" field, the "로그인 암호" field, and the "로그인" button.

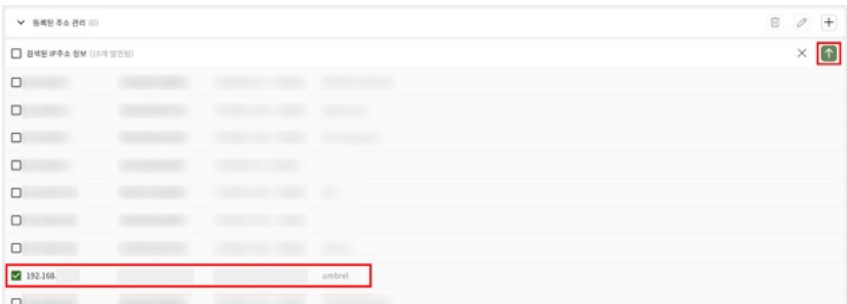
로그인했다면 [관리도구]를 누른다.



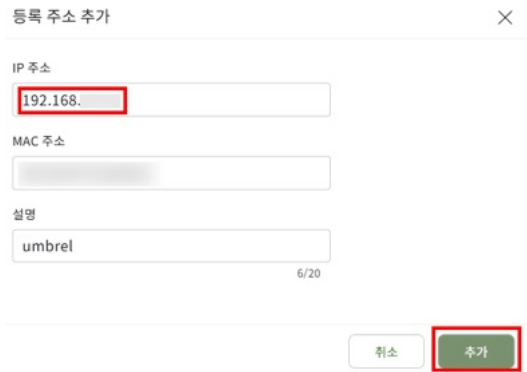
왼쪽 카테고리에서 [네트워크 관리] → [DHCP 서버 설정]에 들어간다. 스크롤을 내려서 ‘umbrel’이라고 쓰여 있는 기기를 찾는다. 오른쪽 위에 있는 화살표 버튼을 누른다.



엄브렐 기기를 체크하고, 다시 오른쪽 위에 있는 화살표 버튼을 누른다.



등록 주소 추가 창이 뜰 것이다. 이때 위에 뜨는 로컬 IP 주소는 메모해 놓는다. 그다음에 [추가]를 누른다.



등록 주소 추가

IP 주소
192.168.

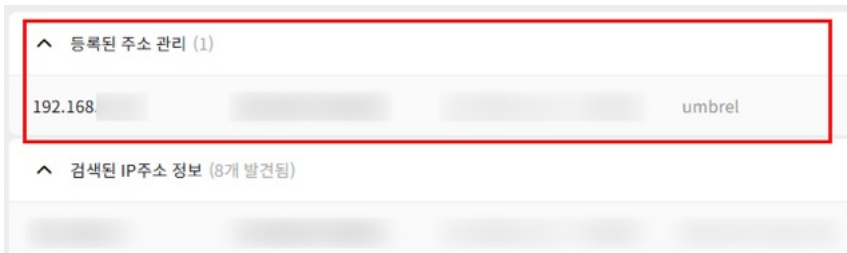
MAC 주소

설명
umbrel

6/20

취소 추가

이제 엄브렐 기기에 고정 로컬 IP 주소를 할당했다. ‘등록된 주소 관리’에 기기가 추가된 게 보인다면 잘 설정된 것이다.

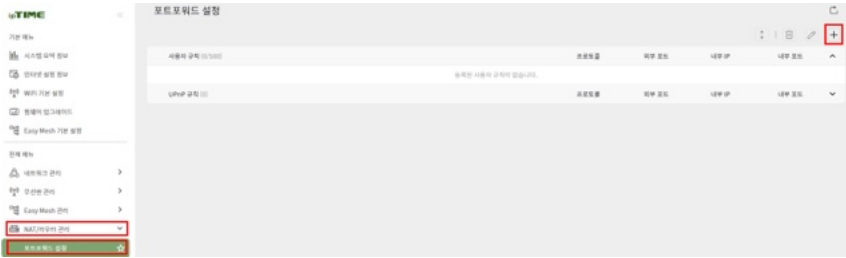


^ 등록된 주소 관리 (1)

192.168.		umbrel
----------	--	--------

^ 검색된 IP주소 정보 (8개 발견됨)

이제 포트 포워딩을 설정해 8333 포트(노츠는 9333)를 열어줘야 한다. 왼쪽의 [NAT/라우터 관리] 카테고리에 있는 [포트포워드 설정]을 누른다. 이후 오른쪽 위에 있는 [+] 버튼을 누른다.



그러면 규칙 추가 창이 뜰 것이다. 각각의 항목을 입력한다. ‘규칙 이름’에는 ‘Bitcoin’을 입력하고, ‘외부 포트’와 ‘내부 포트’ 모두 ‘8333~8333’(노츠는 ‘9333~9333’)을 입력한다. 내부 IP 주소는 아까 DHCP 설정할 때 메모했던, 엠브렐 기기의 로컬 IP 주소를 적는다. 프로토콜은 [TCP/UDP]로 설정한다. [TCP]로 설정해도 무방하다. 이후 [추가] 버튼을 누른다.



여기까지 설정이 되었다면 공유기를 재부팅한다. 왼쪽에서 [시스템 관리] → [기타 설정] → [공유기 즉시 재시작]을 눌러 공유기를 재부팅한다. 재부팅되는 약 1-5분간 인터넷 사용이 안 될 수도 있다. 설정이 다 되었다면 ‘엠프렐 인바운드 연결 허용 및 방화벽 해제’ 절로 넘어가자.



인터넷 서비스 업체의 공유기를 사용하는 경우

다음은 필자가 사람들과 함께 이 과정을 진행하며 몇 가지 알아낸 것들이다.

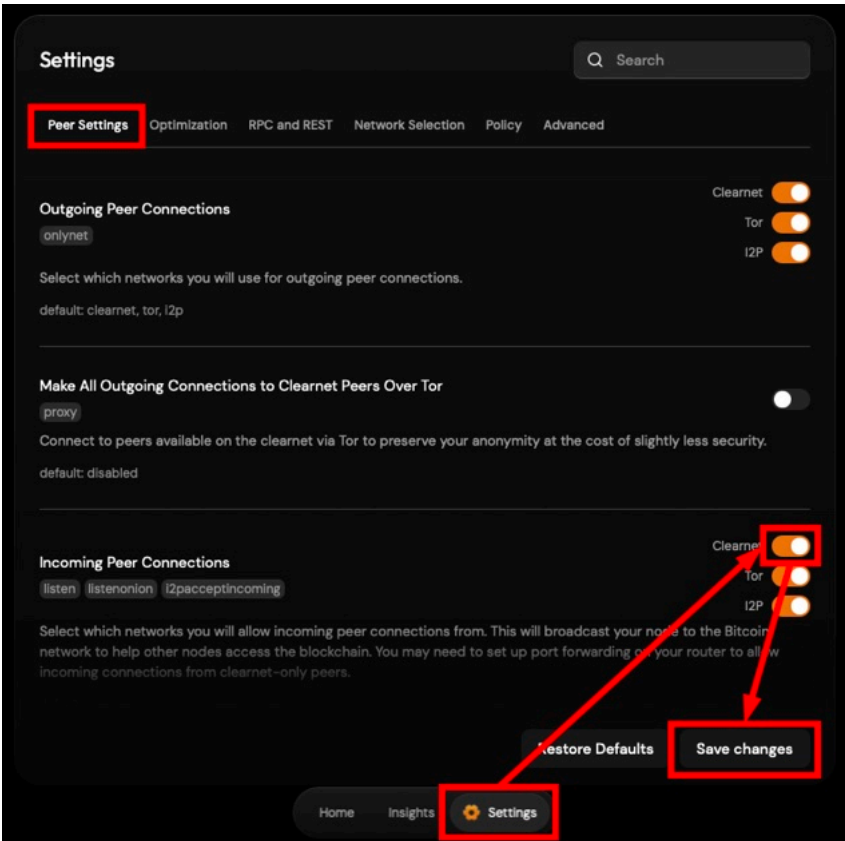
SKT 공유기의 경우, 공유기에서 포트 포워딩을 하고 모뎀에서도 공유기에 대한 포트 포워딩 설정을 해야 하는 경우가 많았다. 보통 모뎀 접속 주소는 192.168.55.1였으며, 아이디는 ‘admin’, 비밀번호는 모뎀 아래에 적혀있거나, 아래에 적혀있는 MAC 주소 뒤 6자리 + ‘_admin’ (??????_admin과 같은 형식)이었다. 여기서 공유기에 할당된 IP 주소 192.168.55.???를 알아낸 뒤 모뎀에서 공유기로 포트 포워딩 설정을 하면 대부분의 문제는 해결되었다. 잘 안된다면 SKT 브로드밴드 인터넷 서비스 업체에 문의하거나 인터넷에 ‘SKT 모뎀 포트 포워딩’으로 검색해 보라.

KT 공유기의 경우, 172.30.1.254로 공유기 설정 창에 접속하는 경우가 있었으며, 아이디는 'ktuser', 비밀번호는 'homehub' 또는 'megapp'(구형 모델)인 경우가 있었다. 설정 창에 진입한 후 [장치설정] → [네트워크 관리] → [수동 IP 할당] 설정에서 업브렐의 IP 주소를 할당하고, 포트 포워딩을 하면 되었다. 잘 안된다면 KT 인터넷 서비스 업체에 문의하거나 인터넷에 'KT 공유기 포트 포워딩'으로 검색해 보라.

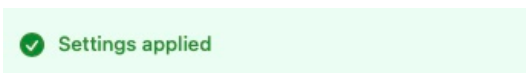
LG U+ 공유기의 경우, 접속 링크가 192.168.219.1로 되어있는 경우가 많았다. 공유기 비밀번호는 공유기 아랫면 혹은 뒷면 스티커에 있거나, 그런 게 없는 경우 아이디는 'user', 비밀번호는 'power'인 경우도 있었다. 설정 창에 진입한 후 [상태 정보] → [DHCP 할당 정보] → [DHCP 고정 할당]에서 풀 노드 기기를 추가하면 된다. 그다음에 [네트워크 설정] → [NAT 설정]에서 포트 포워딩 설정을 하면 되는 경우가 많았다. 잘 안된다면 LG U+ 인터넷 서비스 업체에 문의하거나 인터넷에 'LG U+ 포트 포워딩'으로 검색해 보라.

엠프렐 인바운드 연결 허용 및 방화벽 해제

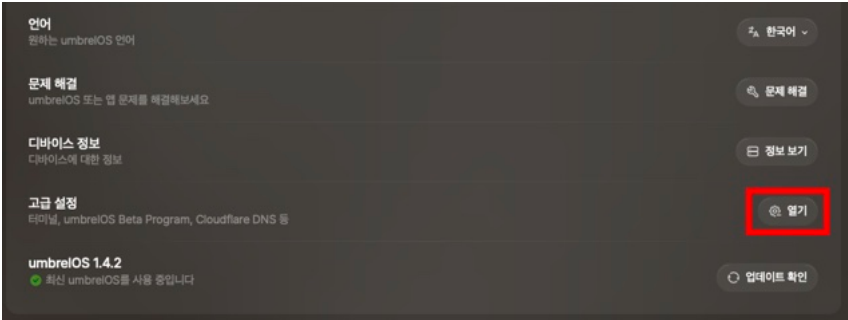
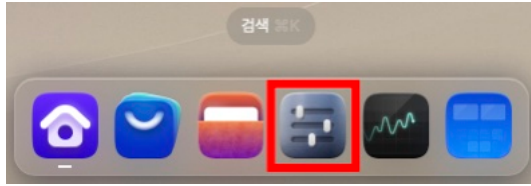
공유기 재부팅이 완료되었으면 엠프렐 기기로 접속한다. 코어 혹은 노츠 하단에서 [Settings] → [Peer Settings] → ‘Incoming Peer Connections’의 [Clearnet] 토글 스위치를 켜다. 그다음에 [Save changes]를 누르면 된다.



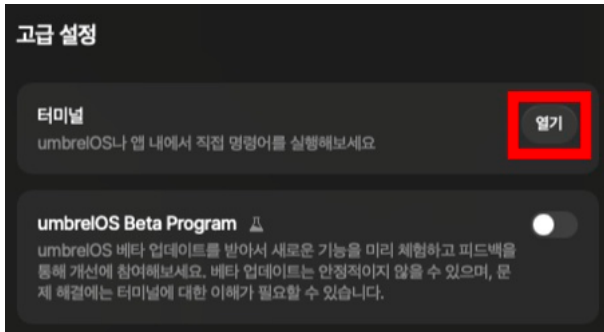
바뀐 설정이 잘 저장되면 우측 상단에 다음과 같은 알림이 나온다.



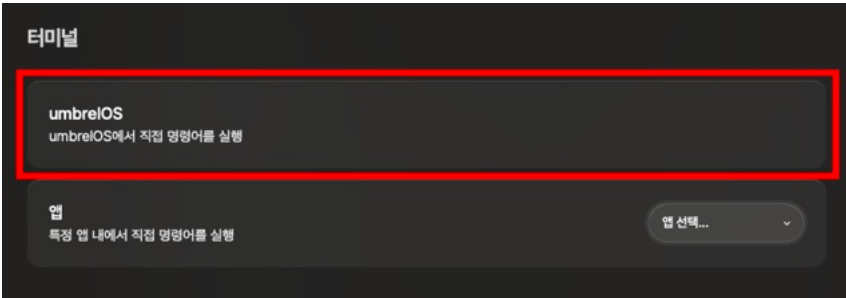
이제 엠브렐 홈 화면으로 나간다. 설정에 들어가 '고급 설정' 옆의 [열기]를 누른다.



'터미널' 옆에 있는 [열기]를 누른다.



[umbrelOS]를 누른다.



터미널 입력기가 뜰 것이다. 그러면 다음과 같이 입력한다(대문자 입력, 특히 대문자 I 입력에 주의하라).

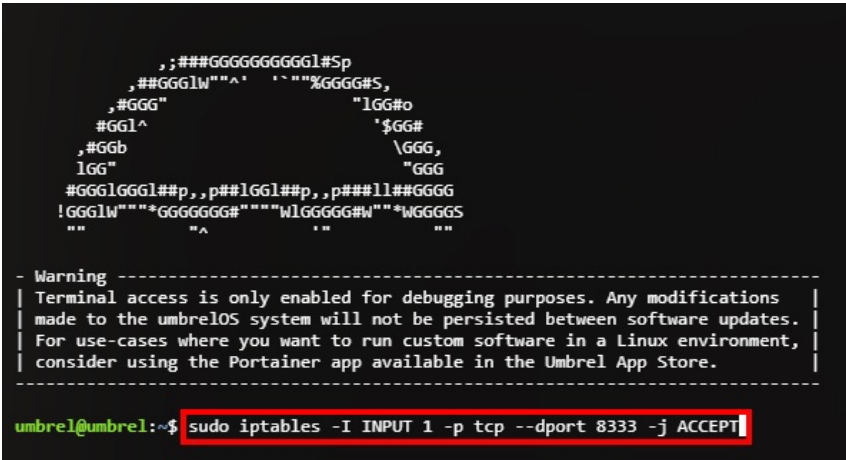


```
sudo iptables -I INPUT 1 -p tcp --dport 8333 -j ACCEPT
```

노츠는 다음과 같이 입력해야 한다.



```
sudo iptables -I INPUT 1 -p tcp --dport 9333 -j ACCEPT
```



입력한 후에 엔터를 누른다. 비밀번호를 입력하라는 창이 뜰 수도 있다. 그러면 엄브렐의 비밀번호를 입력하고 다시 엔터를 누르면 된다. 비밀번호가 입력이 안 되는 것 같아도 입력이 되고 있는 것이다.

```

,;###GGGGGGGGG#Sp
,##GGG1w""^'  ``"%GGG#S,
,#GGG"          "1GG#o
#GG1^          '$GG#
,#GGb          \GGG,
1GG"          "GGG
#GG1GGG1#p, ,p##1GG1#p, ,p##1l#GGGG
!GGG1w""*GGGGGGG#"*****1GGGGG#w""*WGGGGG
""          ""          ""          ""

- Warning -----
| Terminal access is only enabled for debugging purposes. Any modifications |
| made to the umbrelOS system will not be persisted between software updates. |
| For use-cases where you want to run custom software in a Linux environment, |
| consider using the Portainer app available in the Umbrel App Store. |
-----

umbrel@umbrel:~$ sudo iptables -I INPUT 1 -p tcp --dport 8333 -j ACCEPT
[sudo] password for umbrel:
umbrel@umbrel:~$

```

잘 입력이 됐다면 아무 메시지도 뜨지 않을 것이다. 여기까지 했다면 모든 설정이 끝났다. 참고로 엄브렐의 경우 OS 업데이트가 되거나 재부팅 할 때마다 방화벽이 다시 활성화된다. 이때는 터미널에 다시 들어가서 위에 있는 명령어를 입력해 주면 된다.

방화벽이 해제됐는지 확인하고 싶다면 아래 명령어를 입력하고, LISTEN이 뜨는지 확인하면 된다.

```
sudo ss -tulnp | grep 8333
```



노츠의 경우 아래 명령어를 입력하고 LISTEN이 뜨는지 확인하면 된다.

```
sudo ss -tulnp | grep 9333
```



도달 가능한 노드가 되었는지 확인하기

이제 도달 가능한 노드가 되었는지 확인해 보자. 다시 아래 웹사이트에 접속한다.

<https://bitnodes.io/>



노츠는 포트 번호를 9333으로 입력한 뒤 [CHECK NODE]를 눌러야 한다. 다음과 같은 화면이 뜬다면 도달 가능한 노드가 된 것이다. 축하한다. 앞으로 당신은 누군가 새로운 풀 노드를 운영할 때 블록 데이터를 나눠줌으로써 진정으로 비트코인 네트워크에 기여하게 되었다.

JOIN THE NETWORK

Be part of the Bitcoin network by running a Bitcoin full node, e.g. [Bitcoin Core](#).

Use this tool to check if your Bitcoin client is currently accepting incoming connections from other nodes. Port must be between 1024 and 65535.



/Satoshi:28.0.0/ (Height: 881575)

포트를 열어도 데이터를 주고받는 데는 최대 2시간이 걸릴 수 있다. 2시간 이후에도 도달 가능한 노드가 되지 않는다면 먼저 공유기를 재부팅하고, 업브렐도 재부팅해 보자. 그래도 안 된다면 쓰고 있는 인터넷 서비스 제공 업체 ISP (SK 브로드밴드, KT 인터넷, U+ 인터넷 등)에 전화해 모뎀 설정에서 포트 포워딩을 하는 방법이나 브릿지 연결 모드로 바꾸는 방법을 물어보면 된다. 물어보면 친절히 알려줄 것이다. 혹은 인터넷에 검색해도 된다. 모뎀이 인터넷 함 안에 들어가 있어서 안 보이는 경우도 있다. 특히 아파트에 이런 경우가 많다.

다시 한번 말하지만 도달 가능한 노드 설정은 반드시 프라이버시와 네트워크 설정에 대한 이해가 있는 사람만 도전하기를 권고한다.

| 윈도우OS에서 풀 노드 운영하기

윈도우OS에 비트코인 코어 설치하고 동기화하기

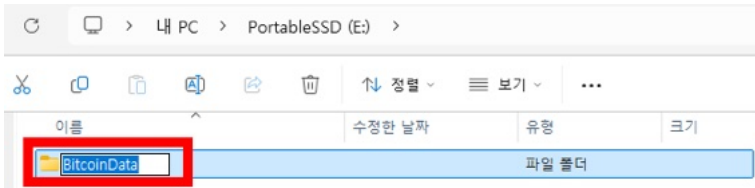
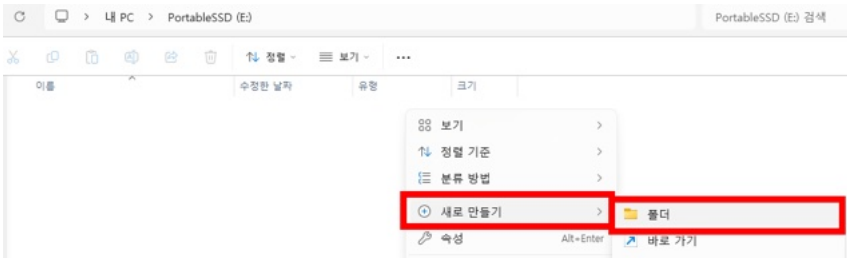
지금까지는 엠프렐OS를 이용해 풀 노드를 운영하고, 일렉트럼 서버를 이용해 자신이 이용하는 워치-온리 지갑들과 연동하는 방법을 알아보았다. 만약 새로운 기기를 장만할 여유가 없고, 남는 기기도 없는 경우 데스크톱OS인 윈도우OS나 맥OS에 비트코인 클라이언트(코어나 노츠)를 설치할 수도 있다. 그러나 이 방법을 사용하면 일렉트럼 서버를 사용하지 않으므로 데스크톱 지갑인 스페로우 워치-온리 지갑과만 연동할 수 있다.

비트코인 코어와 비트코인 노츠 중 무엇을 설치할지 먼저 선택하자. 두 클라이언트의 차이에 대해서는 ‘비트코인 노드(코어) 또는 노츠 설치’ 절을 참고하라. 이 절에서는 윈도우OS에서 비트코인 코어를 설치하는 방법에 대해 알아볼 것이다. 바로 뒤의 절에서 비트코인 노츠를 설치하는 방법에 대해서도 알아볼 것이다.

만약 설치할 데스크톱의 저장 공간에 여유가 없다면 2TB 이상의 외장 SSD를 구매하는 것이 좋다. 블록 데이터를 외장 SSD에 받을 수 있기 때문이다.



외장 SSD를 포맷했다면 새 폴더를 하나 만들고 폴더 이름을 영어로 쓴다.



이제 다음 웹사이트에 접속하고, [Download Bitcoin Core]를 누른다.

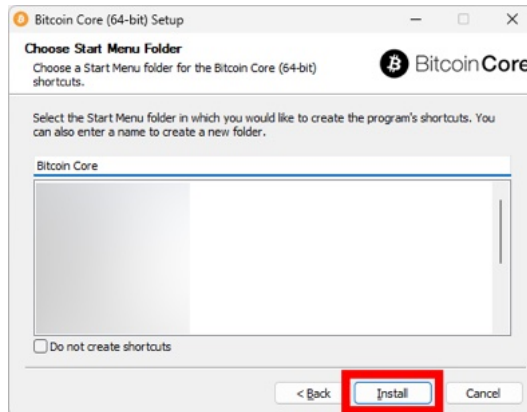
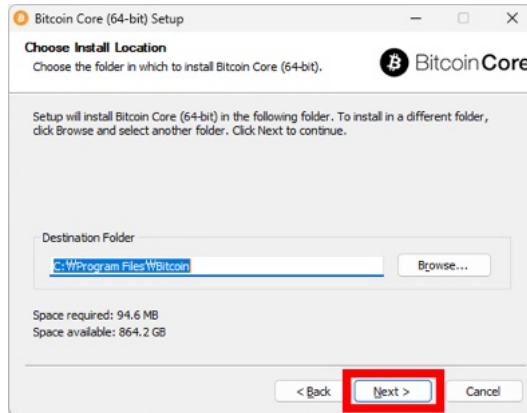
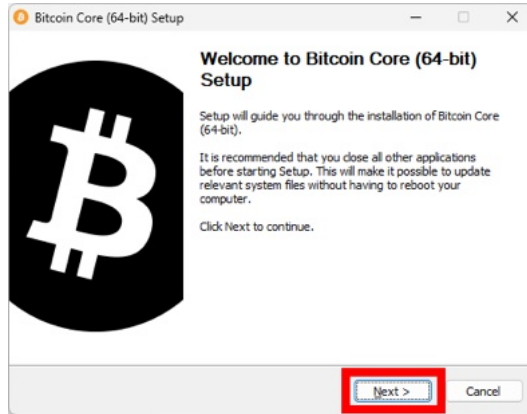
<https://bitcoincore.org/en/download/>



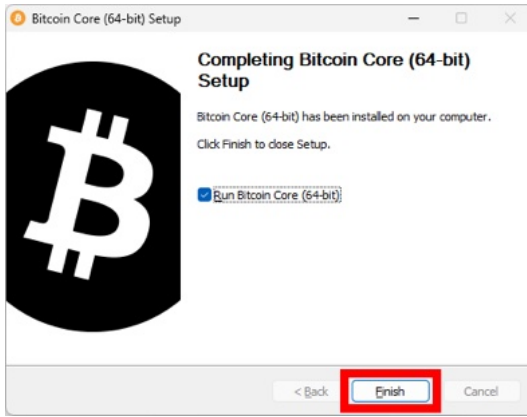
다운로드가 완료되면 파일을 실행한다.



설치 창이 나오면 [Next >] → [Next >] → [Install]을 누른다.



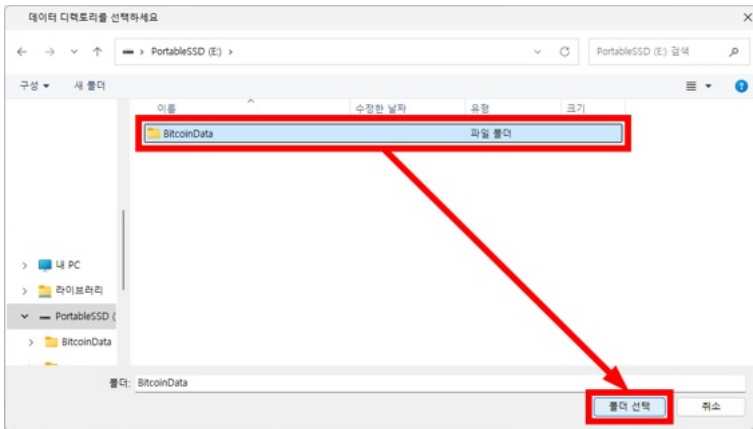
설치가 완료되고 [Finish]를 누르면 비트코인 코어 프로그램이 자동으로 실행된다.



비트코인 코어가 실행되면 [커스텀 데이터 폴더 사용:]을 선택하고 오른쪽의 [...] 버튼을 눌러 외장 SSD에서 만들었던 폴더를 선택한다. 이 폴더에 비트코인 블록 데이터가 설치될 것이다.

‘블록체인 스토리지를 다음으로 제한하기’ 옵션은 가지치기 설정이다. 이 옵션을 체크하면 설정된 용량의 최신 블록만 남기고 과거 블록 데이터는 지운다.

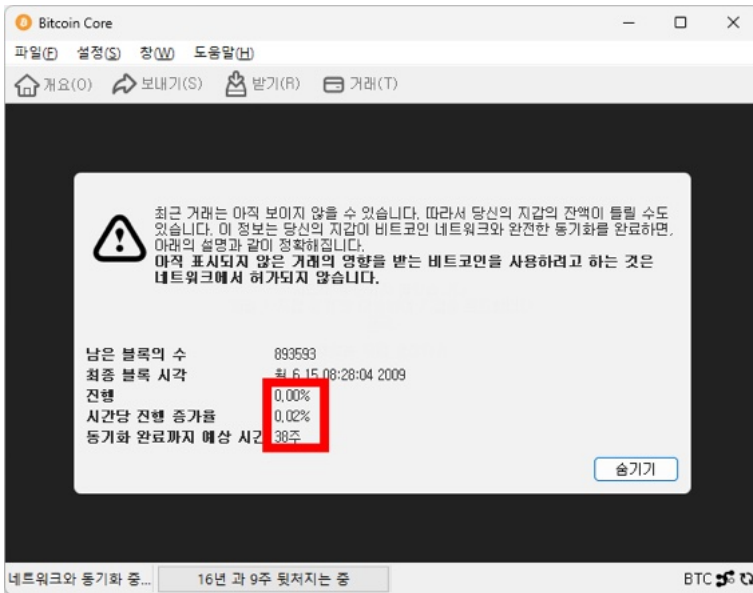
경로를 지정했으면 [OK]를 누른다.



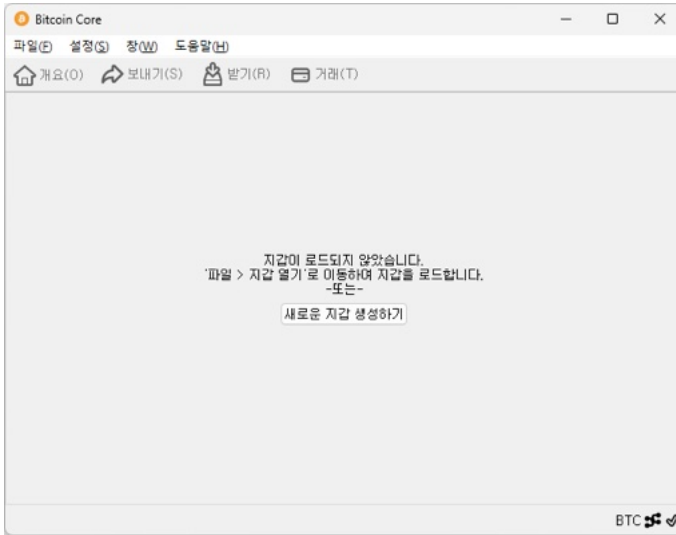
네트워크에서 액세스를 허용할지 묻으면 [허용]을 누른다.



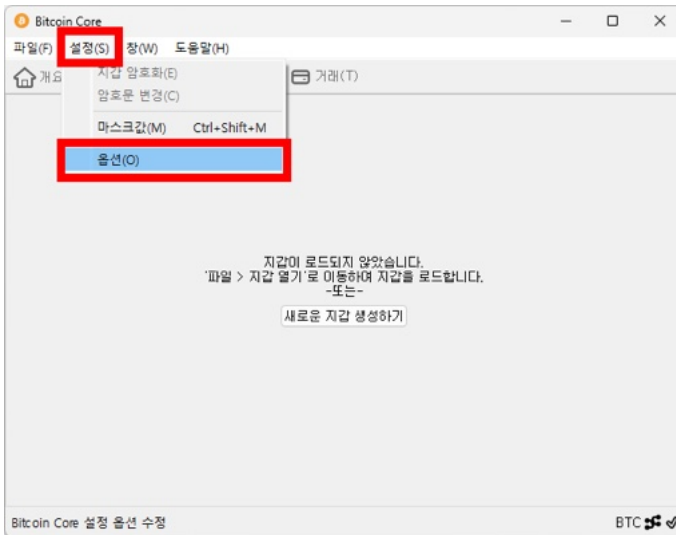
그러면 비트코인 코어가 동기화를 시작한다. 이 과정은 인터넷 환경에 따라 하루-일주일 정도 걸릴 수 있다.



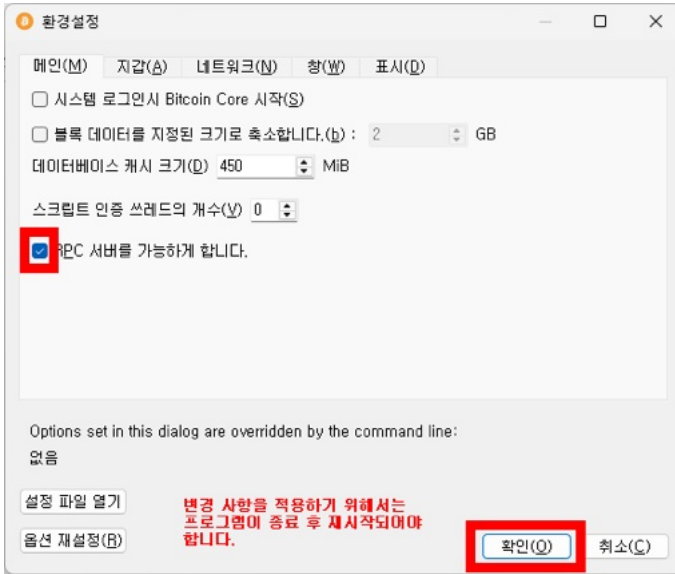
동기화가 완료되면 다음과 같은 화면이 나온다.



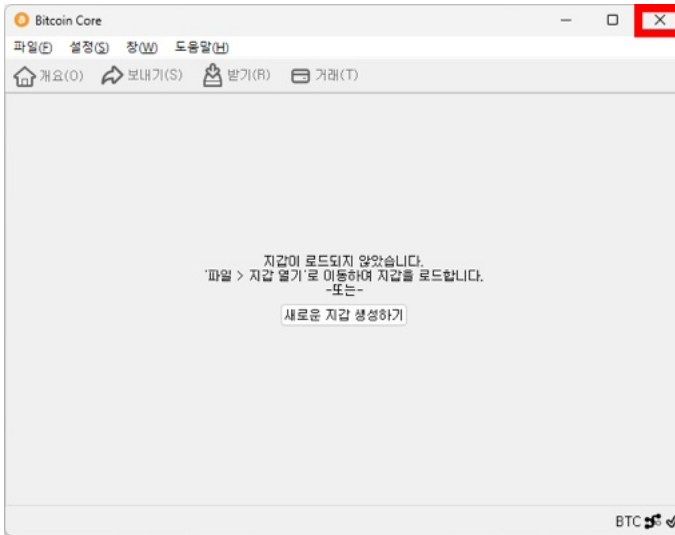
이후에 스페로우 지갑을 연결하기 위해 한 가지 설정을 켜놓고 가자. 상단 탭에서 [설정] → [옵션]을 누른다.

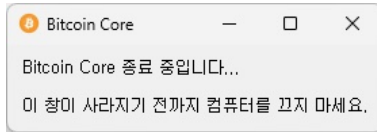


[RPC 서버를 가능하게 합니다.]에 체크하고 [확인]을 누른다.

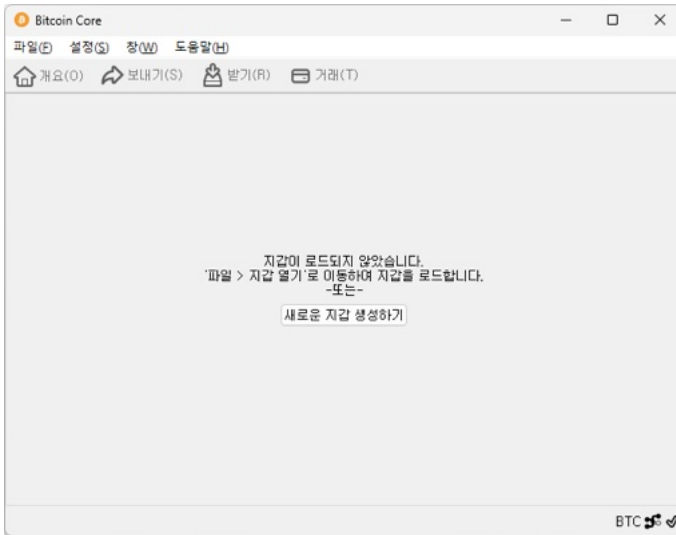


이제 비트코인 코어 프로그램을 닫는다.





비트코인 코어를 다시 실행한다. 그러면 바뀐 설정이 적용된다.



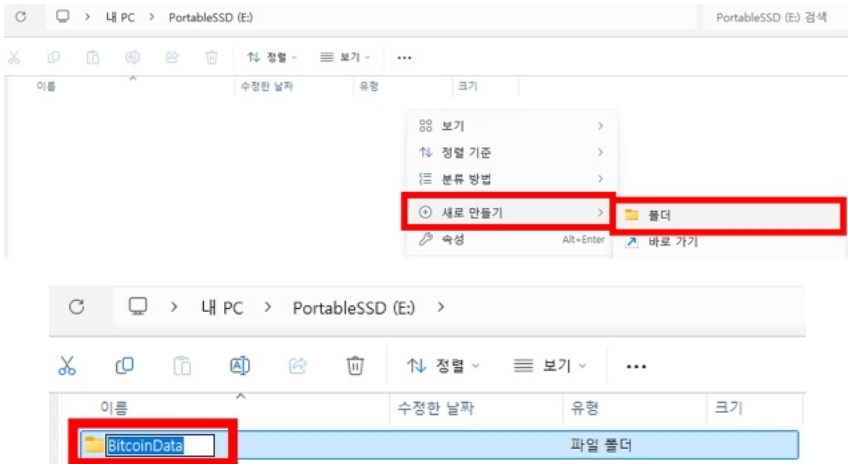
윈도우OS에 비트코인 노츠 설치하고 동기화하기

이 절에서는 윈도우OS에서 비트코인 노츠를 설치하는 방법에 대해 알아볼 것이다.

만약 설치할 데스크톱 저장 공간에 여유가 없다면 2TB 이상의 외장 SSD를 구매하는 것이 좋다. 블록 데이터를 외장 SSD에 받을 수 있기 때문이다.

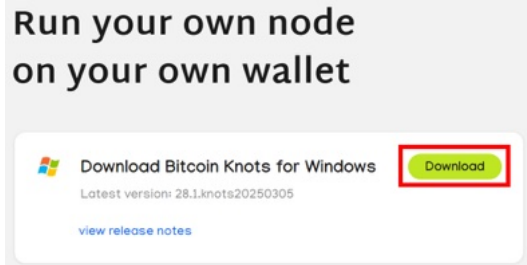


외장 SSD를 포맷했다면 새 폴더를 하나 만들고 폴더 이름을 영어로 쓴다.



이제 다음 웹사이트에 접속하고, [Download]를 누른다.

<https://bitcoinknots.org/>

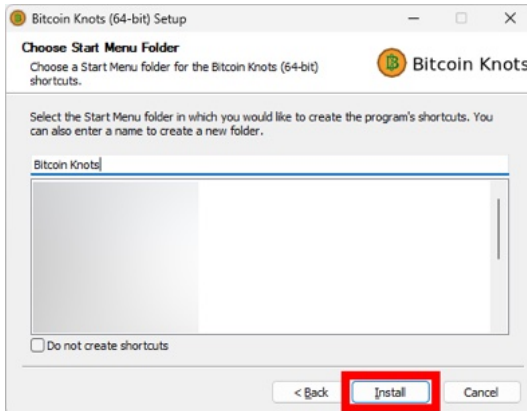
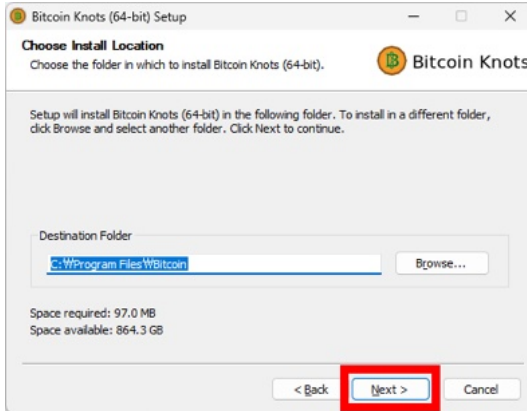
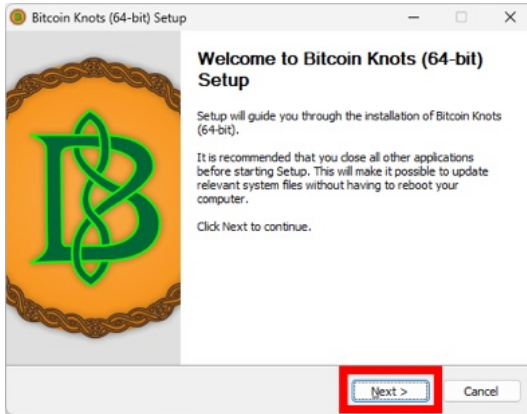


다운로드가 완료되면 파일을 실행한다.

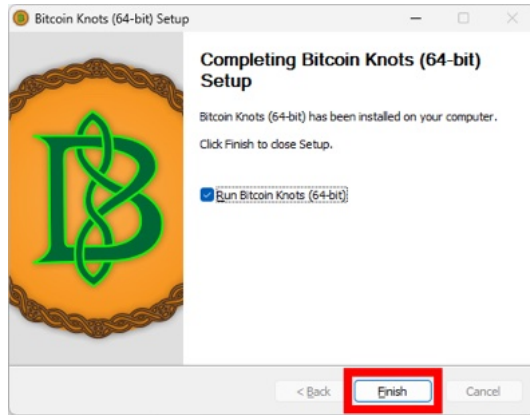


bitcoin-28.
1.knots202
50305-win
64-setup...

설치 창이 나오면 [Next >] → [Next >] → [Install]을 누른다.



설치가 완료되고 [Finish]를 누르면 비트코인 노드 프로그램이 자동으로 실행된다.



비트코인 노드가 실행되면 [커스텀 데이터 폴더 사용:]을 선택하고 오른쪽의 [...] 버튼을 눌러 외장 SSD에서 만들었던 폴더를 선택한다. 이 폴더에 비트코인 블록 데이터가 설치될 것이다.

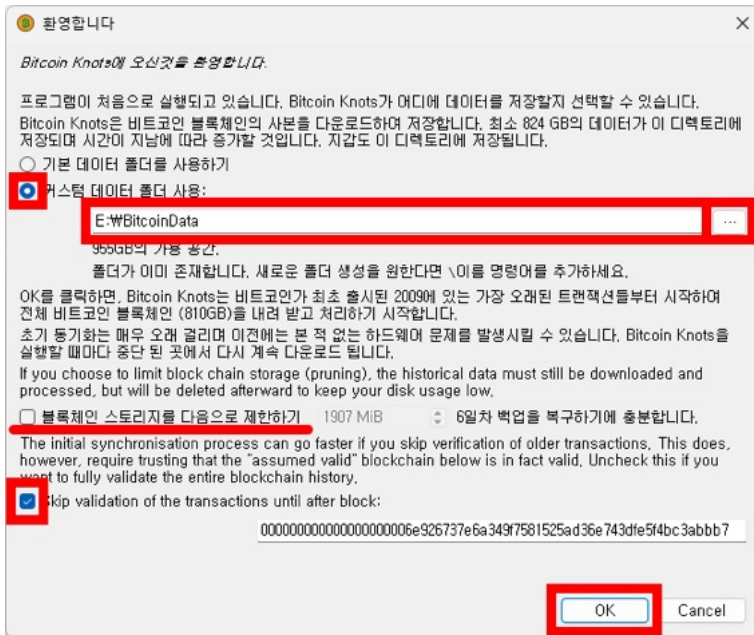
‘블록체인 스토리지를 다음으로 제한하기’ 옵션은 가지치기 설정이다. 이 옵션을 체크하면 설정된 용량의 최신 블록만 남기고 과거 블록 데이터는 지운다.

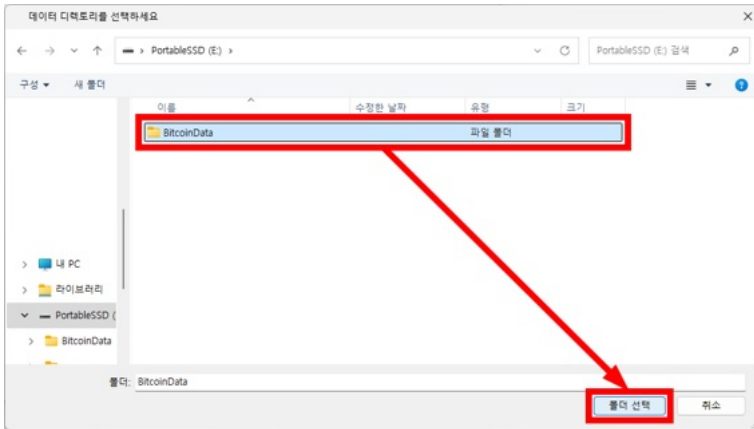
‘Skip validation of the transactions until after block:’ 체크박스는 특정 블록 이전까지의 거래 서명 검증을 건너뛰는 기능이다. 이 옵션을 켜면 해당 블록과 그 이전 블록들의 거래가 정상일 것이라고 가정한다는 뜻이다. 따라서 일정 부분 신뢰에 의존하게 된다. 다만 블록 간의 연결성이나 블록 헤더의 작업증명 같은 기본적인 유효성 검증은 여전히 수행하며, 블록 데이터 자체도 모두 다운로드한다. 단지 거래 검증만 생략하는 것이다. 이렇게 하면 동기화 속도가 빨라진다. 비트코인에

서는 어떤 거래가 블록에 포함되고, 그 위로 충분히 많은 블록이 쌓이면 사실상 되돌릴 수 없는 기록으로 여긴다. 즉, 해당 거래는 이미 수많은 노드들에 의해 검증되었다고 간주할 수 있다.

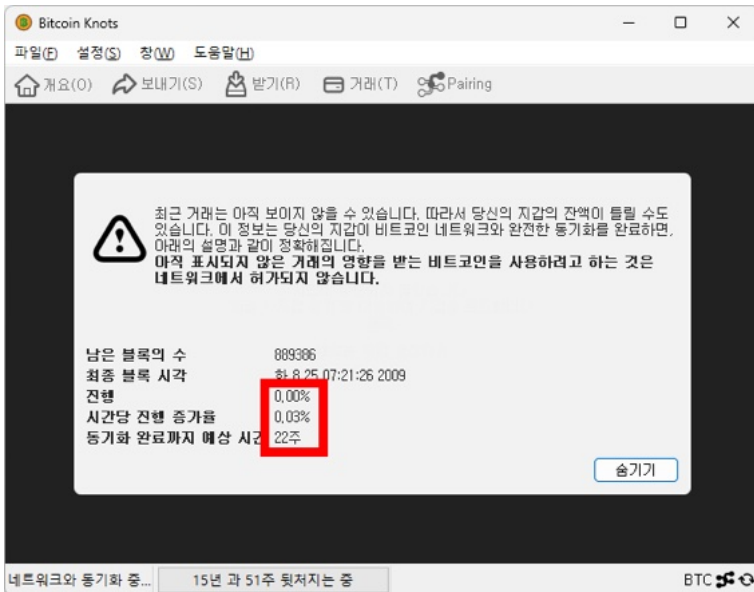
그러나 이런 신뢰를 두는 것이 내키지 않는다면, 체크박스를 해제하면 된다. 이 경우 제네시스 블록 이후부터 지금까지 온체인에서 이루어진 모든 거래의 서명을 직접 검증하게 된다. 다만 이 방식은 동기화가 매우 오래 걸릴 수 있다. 참고로 코어에서는 이런 체크박스 자체가 보이지 않고, 처음 설치할 때 이 설정이 무조건 켜져 있다.

경로를 지정했으면 [OK]를 누른다.

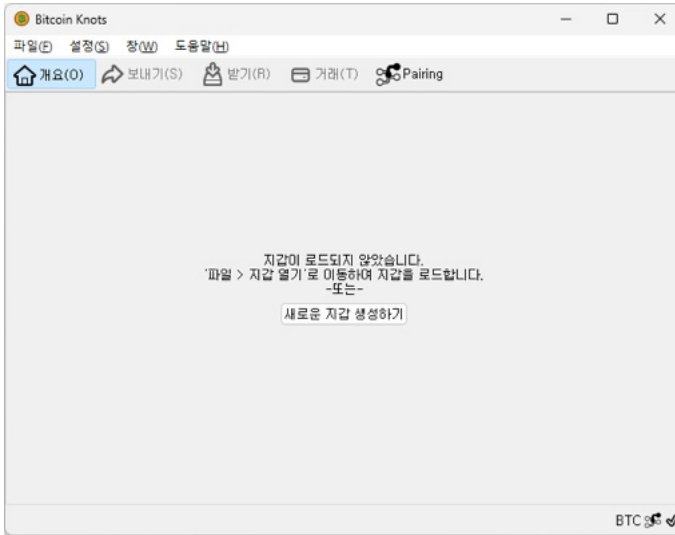




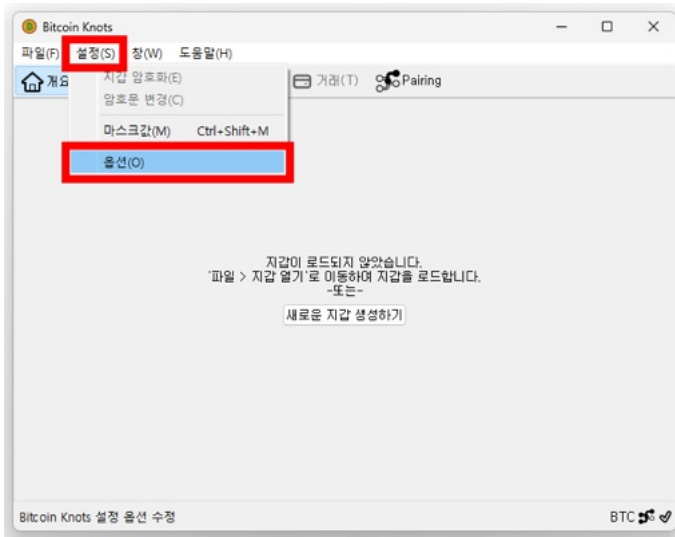
만약 네트워크에서 액세스를 허용할지 물으면 [허용]을 누른다. 비트코인 노츠가 켜지면 동기화를 시작한다. 이 과정은 인터넷 환경에 따라 하루-일주일 정도 걸릴 수 있다.



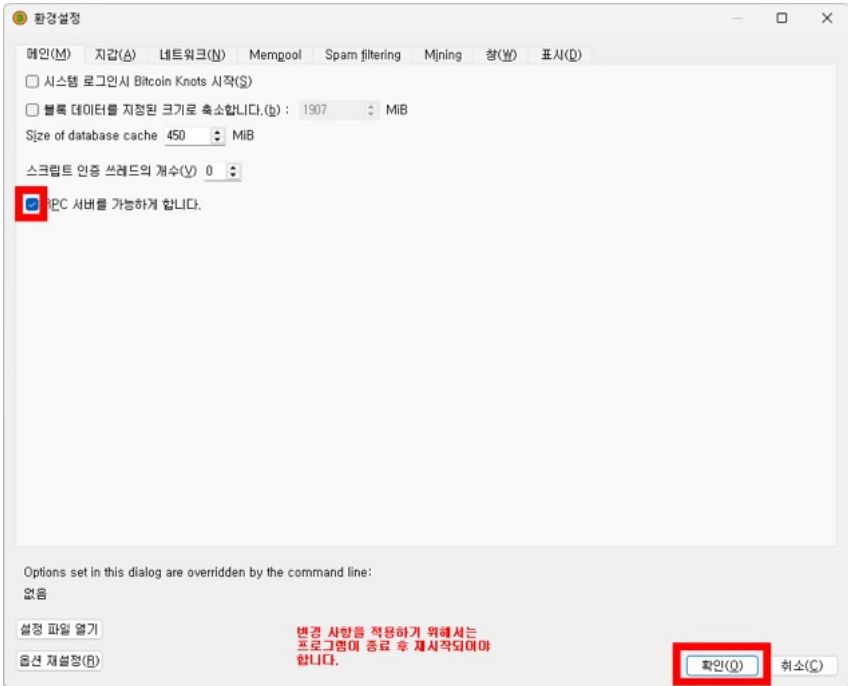
동기화가 완료되면 다음과 같은 화면이 나온다.



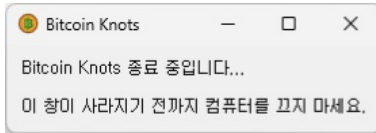
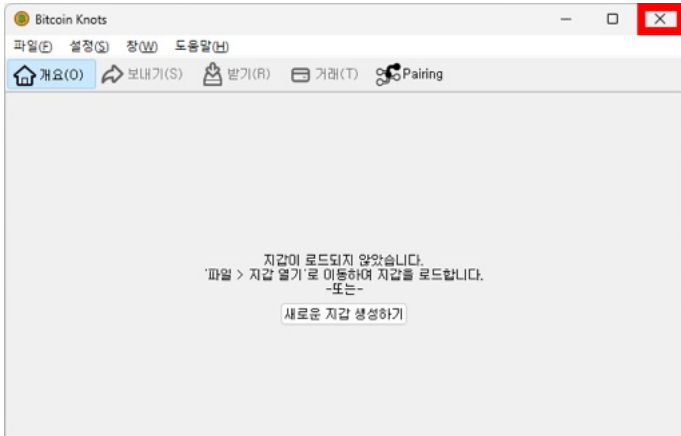
이후에 스패로우 지갑을 연결하기 위해 한 가지 설정을 켜놓고 가자. 상단 탭에서 [설정] → [옵션]을 누른다.



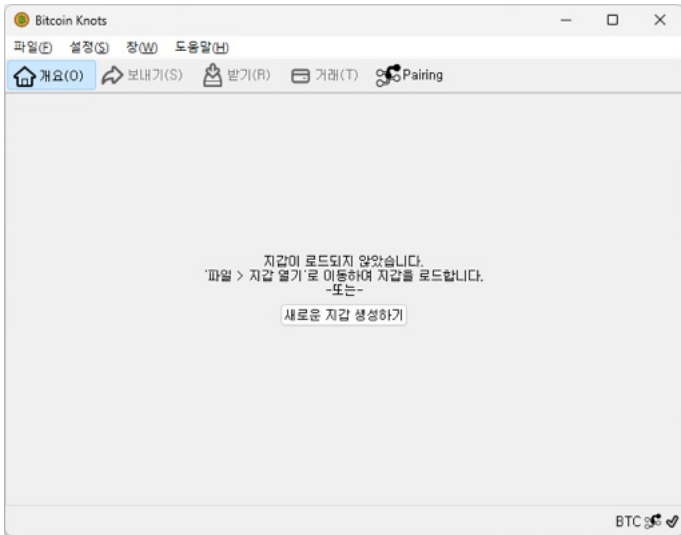
[RPC 서버를 가능하게 합니다.]에 체크하고 [확인]을 누른다.



이제 비트코인 노츠 프로그램을 닫는다.



비트코인 노츠를 다시 실행한다. 그러면 바뀐 설정이 적용된다.



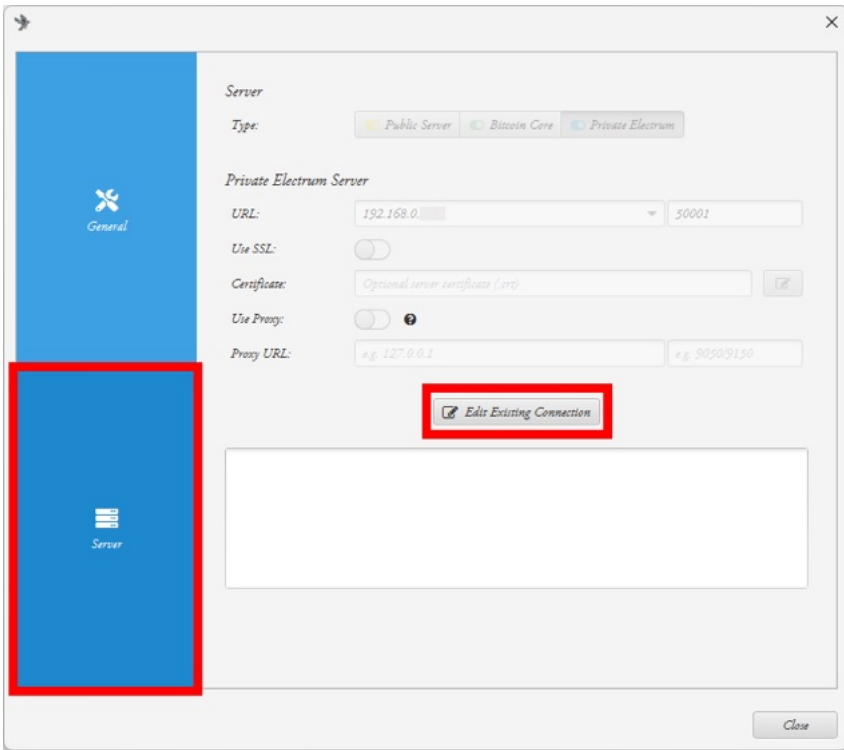
같은 기기에서 스페로우 지갑 연결하기

코어 혹은 노츠를 설치한 데스크톱 기기에서는 스페로우 위치-온리 지갑과 풀 노드를 연결할 수 있다.

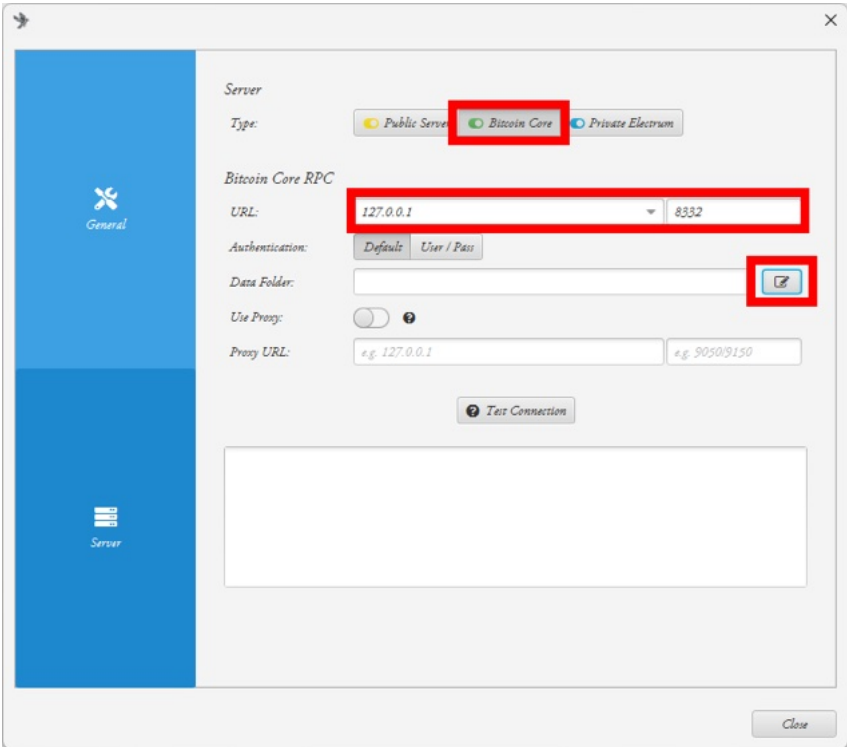
스페로우에서 좌측 상단 [File] → [Settings]를 누른다.



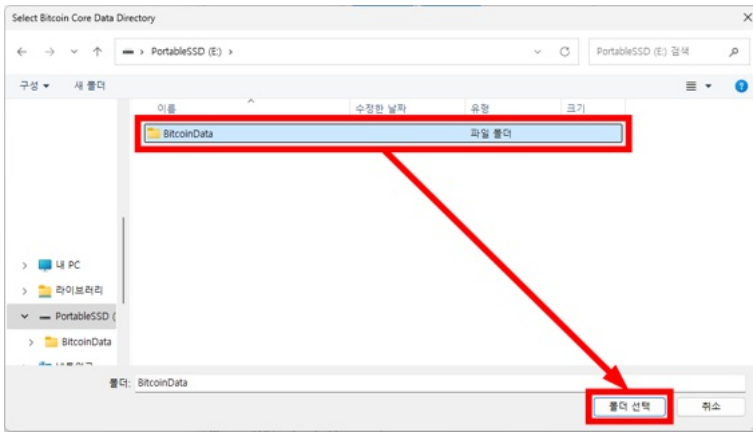
[Server]를 누르고, [Edit Existing Connection]을 누른다.



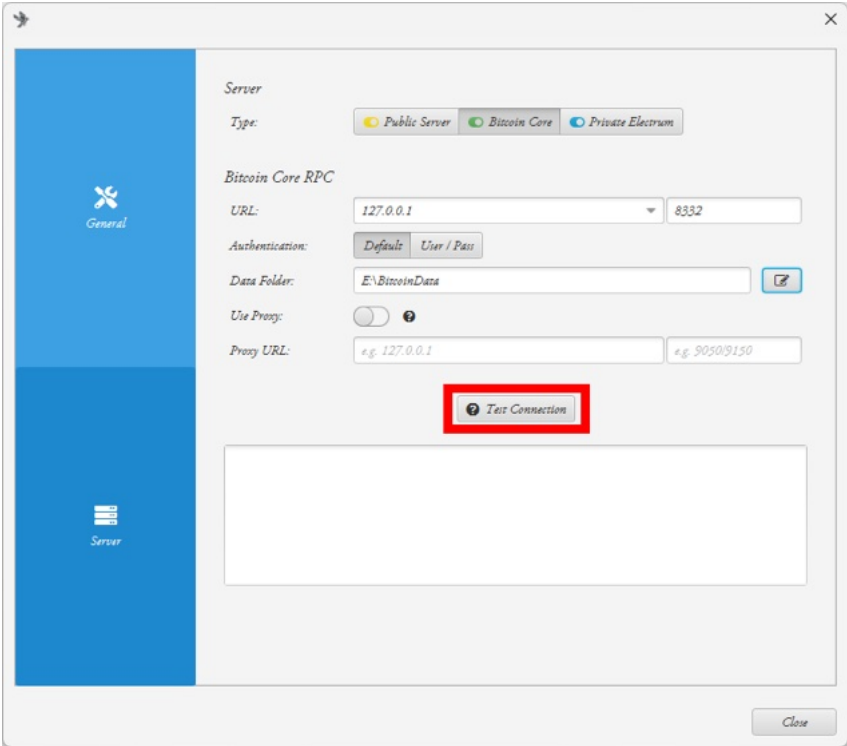
상단 가운데에 있는 [Bitcoin Core]를 누르고, 'Data Folder' 옆에 있는 버튼을 누른다. URL에 있는 127.0.0.1과 포트 번호 8332는 그대로 둔다. 참고로 127.0.0.1은 로컬 호스트의 IP 주소, 즉 현재 기기를 가리키는 주소다.



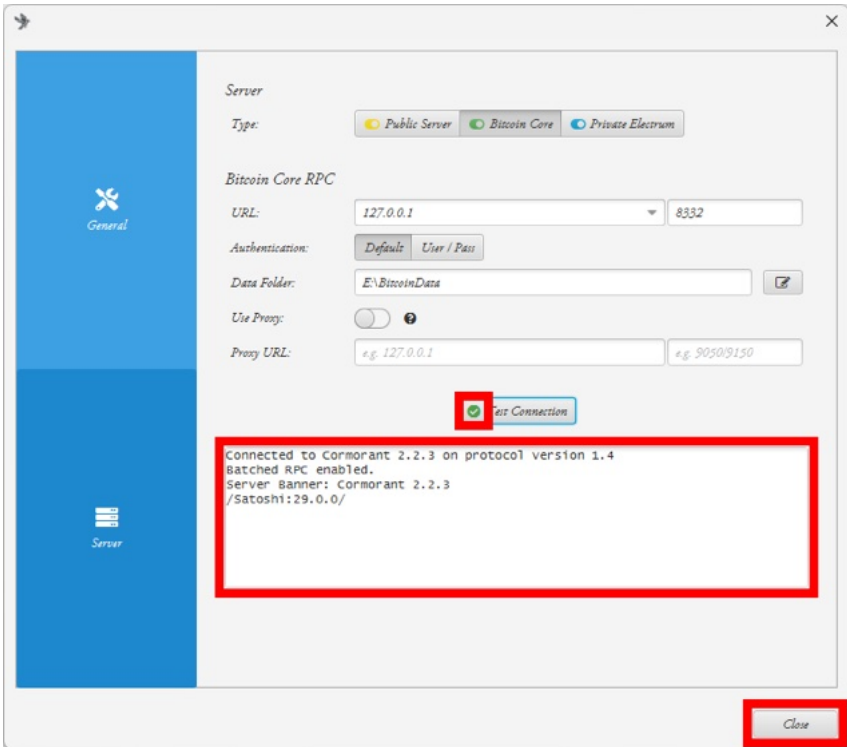
비트코인 코어나 노츠의 RPC 인터페이스를 이용해 풀 노드에 조회 명령을 보내려면 원래는 RPC 사용자 이름과 비밀번호 등을 미리 지정해 두어야 한다. 하지만 코어나 노츠가 설치된 기기에서는 .cookie 파일에 있는 임시 RPC 사용자 이름과 비밀번호를 통해 풀 노드와 계속 통신할 수 있다. 따라서 스페로우가 해당 파일을 찾을 수 있도록 비트코인 데이터가 저장된 폴더를 지정해 주어야 한다. 앞에서 외장 SSD에 비트코인 데이터를 저장했으므로 스페로우에서 이 폴더를 똑같이 선택한다.



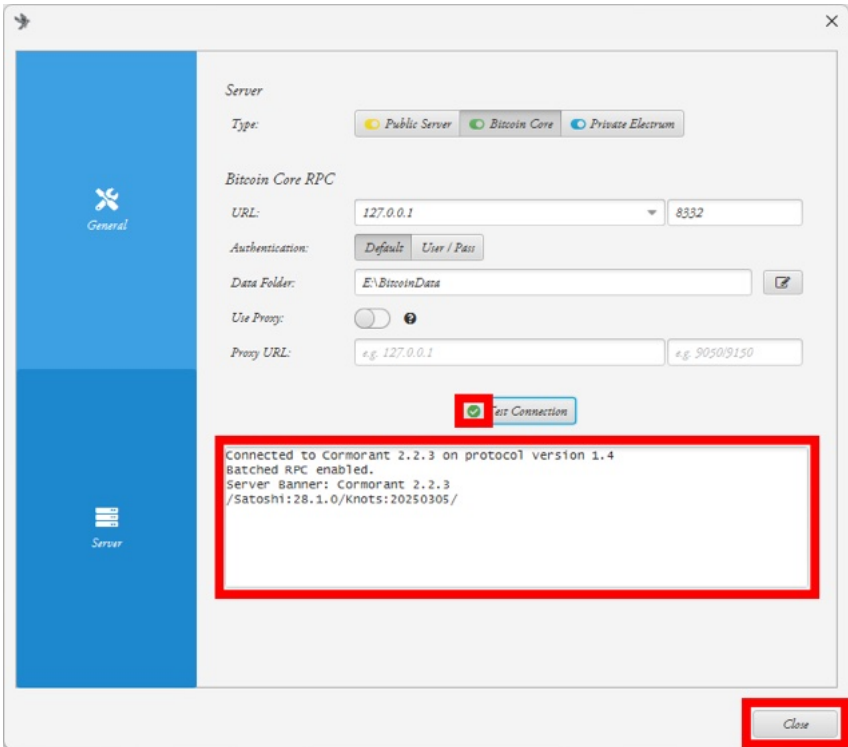
[Test Connection]을 누른다. 네트워크에서 스파로우 앱에 대한 접근 권한을 허용할지 묻으면 [허용]을 누른다.



비트코인 코어와 연결이 성공적으로 되면 초록색 체크 표시와 함께 다음 사진과 같은 문구가 나온다. [Close]를 누르면 된다.



비트코인 노츠와 연결이 성공적으로 되면 초록색 체크 표시와 함께 다음 사진과 같은 문구가 나온다. [Close]를 누르면 된다.



스패로우 지갑이 코어나 노츠와 잘 연결되었다면 스패로우 하단에 초록색 토글 스위치가 켜진다.



| 맥OS에서 풀 노드 운영하기

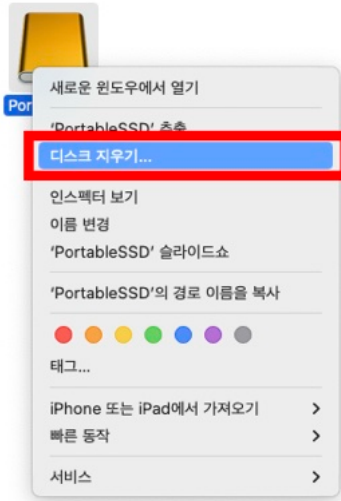
맥OS에 비트코인 코어 설치하고 동기화하기

이제 맥에서 비트코인 코어 또는 노츠를 설치하고, 스페로우 지갑과 연동하는 방법을 알아보겠다. 먼저 맥OS에 비트코인 코어를 설치하는 방법을 알아볼 것이다. 바로 뒤의 절에서 맥OS에 비트코인 노츠를 설치하는 방법도 알아볼 것이다.

만약 설치할 데스크톱 저장 공간에 여유가 없다면 2TB 이상의 외장 SSD를 구매하는 것이 좋다. 블록 데이터를 외장 SSD에 받을 수 있기 때문이다.



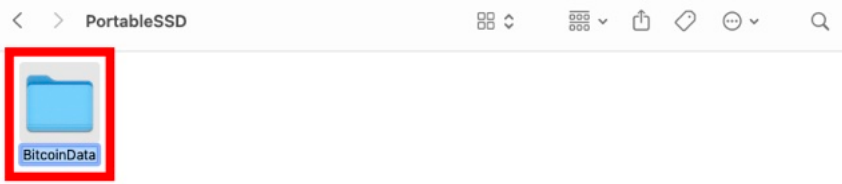
풀 노드는 쓰기가 반복적으로 일어난다. 외장 SSD의 포맷 형식 중 ExFAT 형식은 맥OS에서 쓰기를 할 때 오류가 잦은 것으로 알려져 있다. 따라서 외장 SSD를 포맷할 때는 APFS 형식을 선택한다. 외장 SSD를 우클릭하고 [디스크 지우기]를 선택한다.



포맷 형식으로 [APFS(Apple File System)]를 선택하고 [지우기]를 눌러 포맷한다.

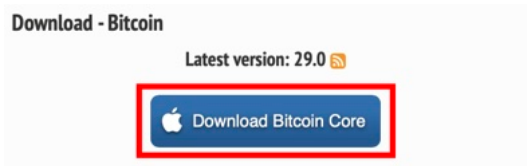


포맷이 완료되었다면 외장 SSD 안에 새 폴더를 하나 만들고 폴더 이름을 영어로 쓴다.

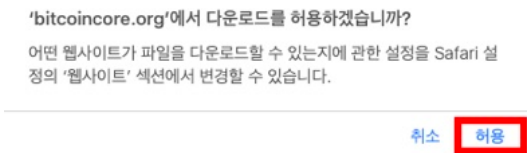


이제 다음 웹사이트에 접속하고, [Download Bitcoin Core]를 누른다.

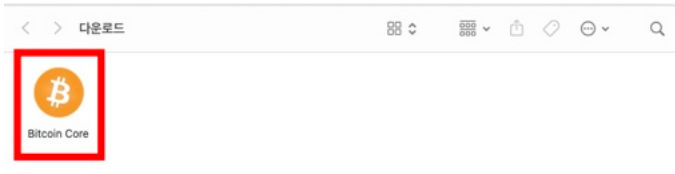
<https://bitcoincore.org/en/download/>



[허용]을 누른다.



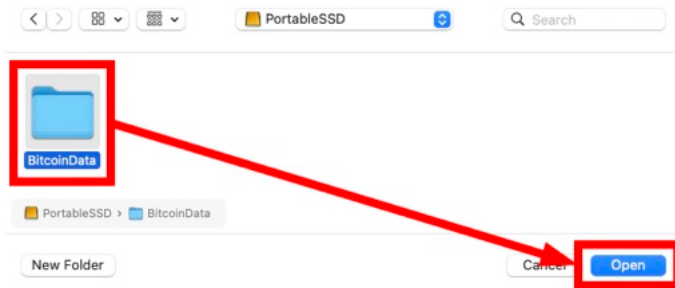
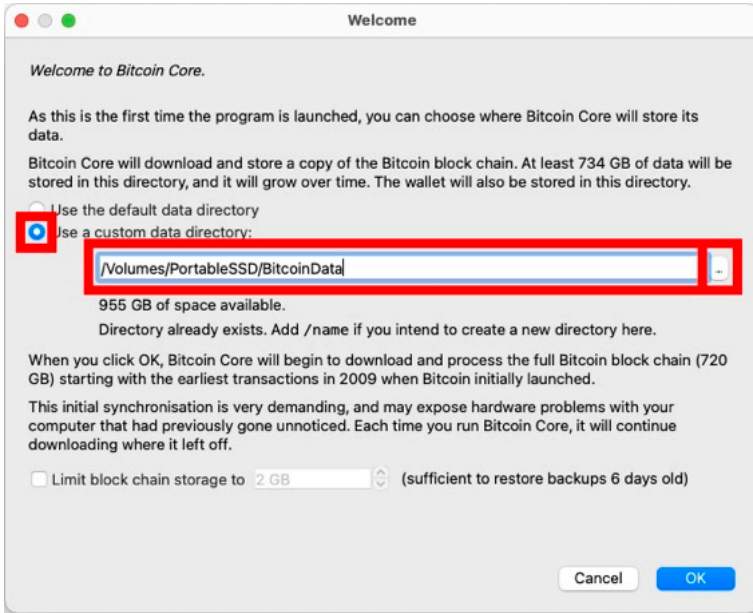
다운로드가 완료되면 파일을 실행한다.



[열기]를 누른다.

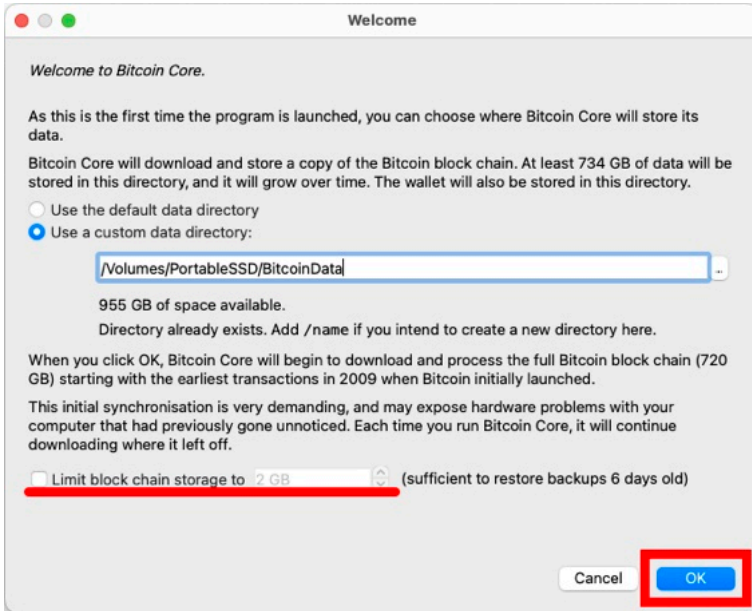


그러면 바로 비트코인 코어가 실행된다. 여기서 [Use a custom data directory:](커스텀 데이터 폴더 사용:)를 선택하고 오른쪽의 [...] 버튼을 눌러 외장 SSD에서 만들었던 폴더를 선택한다. 이 폴더에 비트코인 블록 데이터가 설치될 것이다.

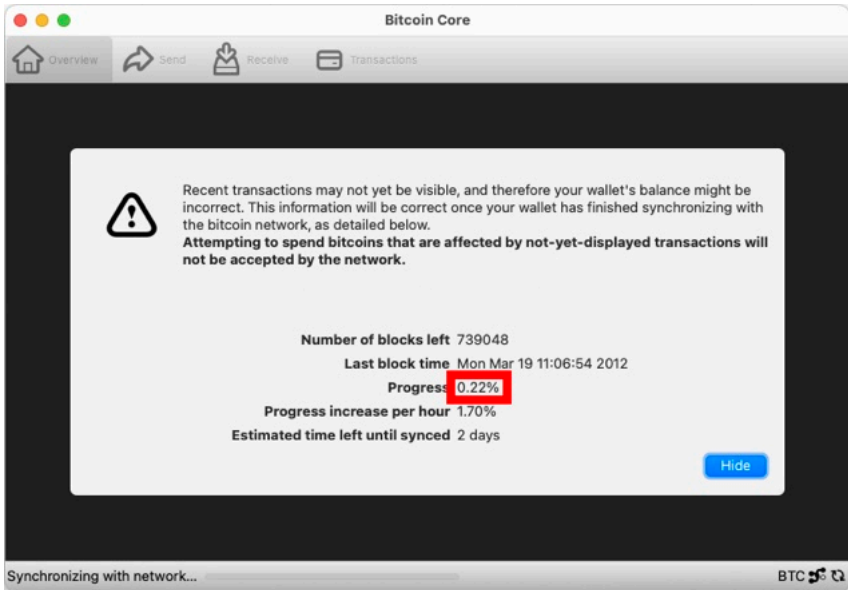


‘Limit block chain storage to (블록체인 스토리지를 다음으로 제한하기)’ 옵션은 가지치기 설정이다. 이 옵션을 체크하면 설정된 용량의 최신 블록만 남기고 과거 블록 데이터는 지운다.

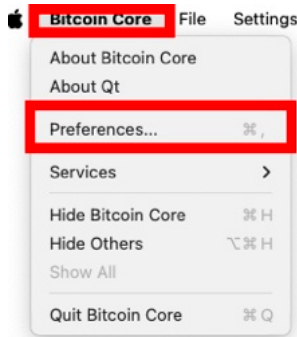
경로가 지정되었으면 [OK]를 누른다.



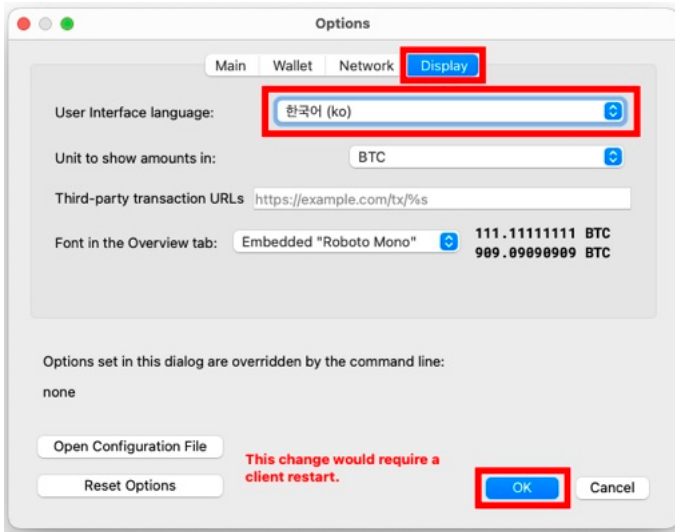
그러면 비트코인 코어가 동기화를 시작한다. 이 과정은 인터넷 환경에 따라 하루-일주일 정도 걸릴 수 있다.



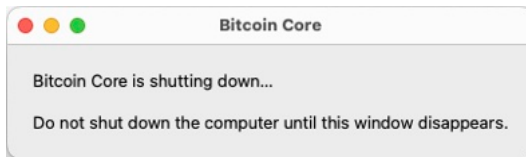
동기화가 완료되기 전에 먼저 코어의 언어 설정을 한국어로 바꾸자. 좌측 상단 [Bitcoin Core] → [Preferences]를 선택한다.



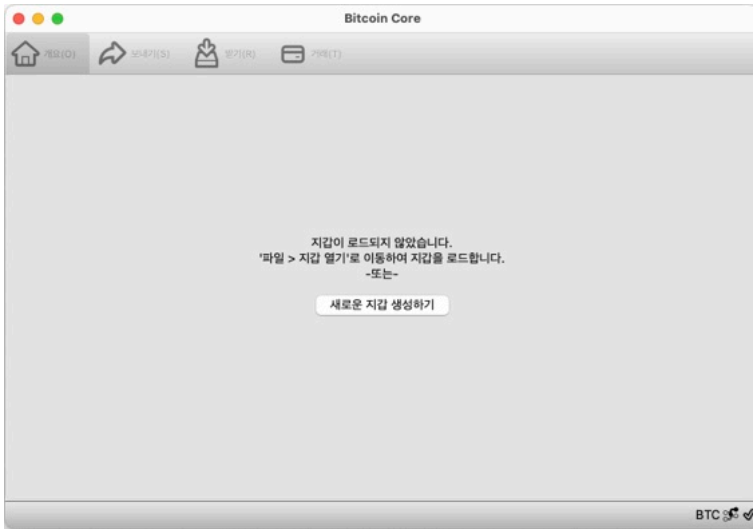
[Display] → ‘User Interface language’에서 [한국어 (ko)]를 선택한다. 그리고 [OK]를 누른다.



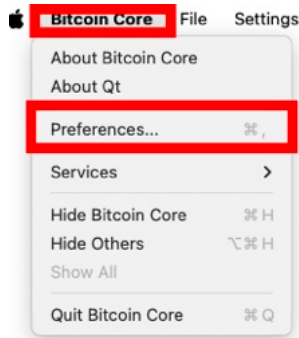
그리고 비트코인 코어에서 ‘Command’ + ‘Q’를 눌러 비트코인 코어를 완전히 종료한다.



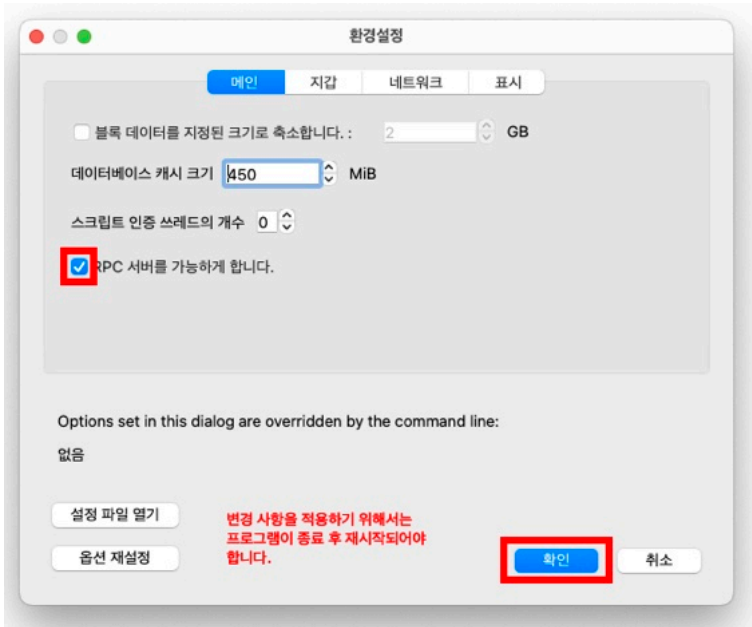
완전히 종료되었으면 다시 코어를 실행한다. 그러면 바뀐 설정이 적용된다. 동기화가 완료되면 다음과 같은 화면이 나온다.



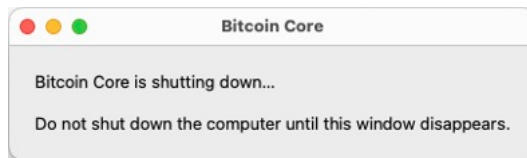
이후에 스패로우 지갑을 연결하기 위해 한 가지 설정을 켜놓고 가자. 상단 탭에서 [Bitcoin Core] → [Preferences]를 누른다.



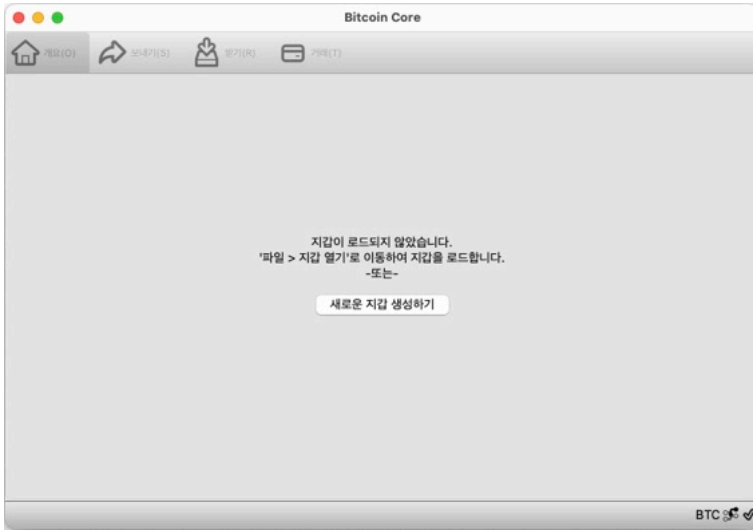
[RPC 서버를 가능하게 합니다.]에 체크하고 [확인]을 누른다.



그리고 비트코인 코어에서 'Command' + 'Q'를 눌러 비트코인 코어를 완전히 종료한다.



비트코인 코어를 다시 실행한다. 그러면 바뀐 설정이 적용된다.



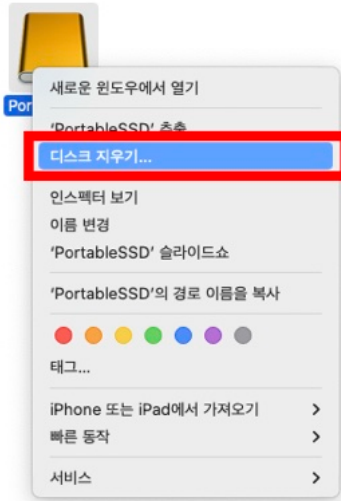
맥OS에 비트코인 노츠 설치하고 동기화하기

이 절에서는 맥OS에서 비트코인 노츠를 설치하는 방법에 대해 알아볼 것이다.

만약 설치할 데스크톱 저장 공간에 여유가 없다면 2TB 이상의 외장 SSD를 구매하는 것이 좋다. 블록 데이터를 외장 SSD에 받을 수 있기 때문이다.



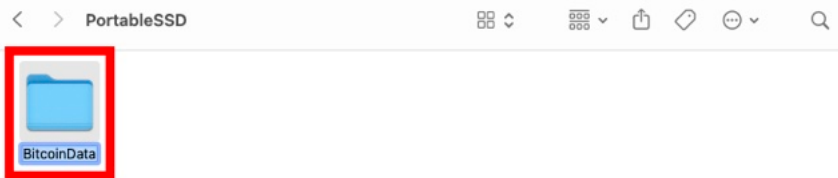
폴 노드는 쓰기가 반복적으로 일어난다. 외장 SSD의 포맷 형식 중 ExFAT 형식은 맥OS에서 쓰기를 할 때 오류가 잦은 것으로 알려져 있다. 따라서 외장 SSD를 포맷할 때는 APFS 형식을 선택한다. 외장 SSD를 우클릭하고 [디스크 지우기]를 선택한다.



포맷 형식으로 [APFS(Apple File System)]를 선택하고 [지우기]를 눌러 포맷한다.

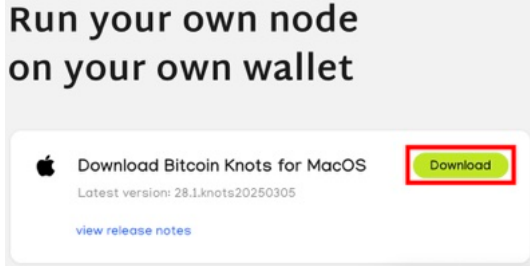


포맷이 완료되었다면 외장 SSD 안에 새 폴더를 하나 만들고 폴더 이름을 영어로 쓴다.

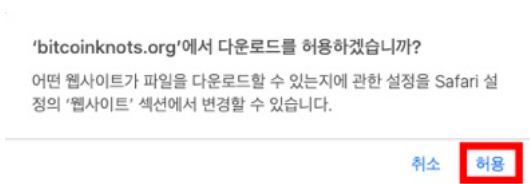


이제 다음 웹사이트에 접속하고, [Download]를 누른다.

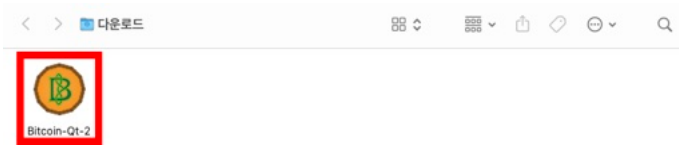
https://bitcoinknots.org/



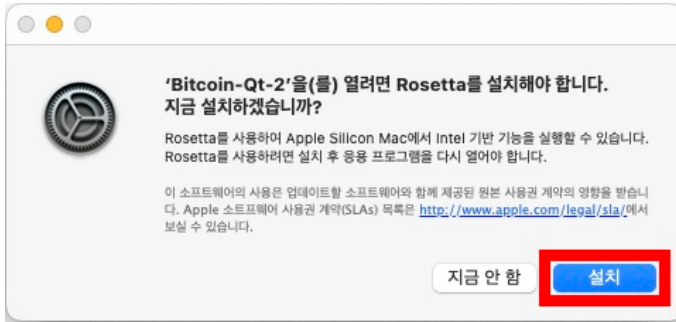
[허용]을 누른다.



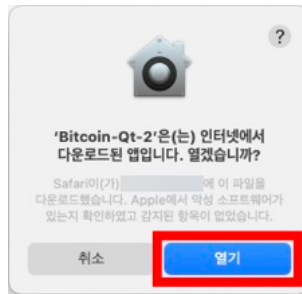
다운로드가 완료되면 파일을 실행한다.



로제타Rosetta를 설치한 적이 없다면 설치하라는 창이 나올 수 있다. 로제타는 애플 실리콘 칩(M1-M4 등)에서 인텔 칩용으로 만들어진 앱을 실행할 수 있게 해주는 프로그램이다. [설치]를 누르고, 안내에 따라 설치한다.



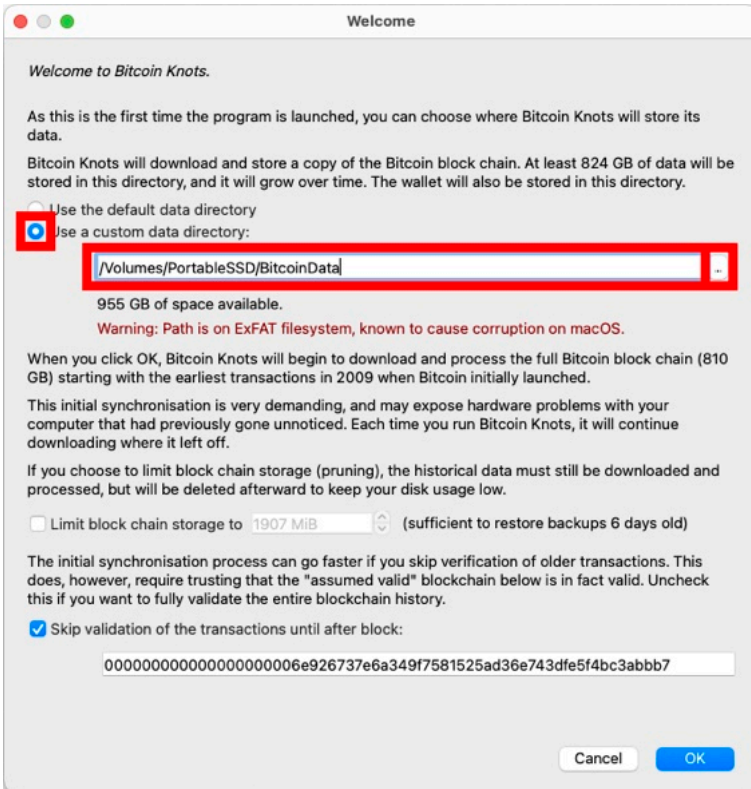
다시 노트를 실행하고, [열기]를 누른다.

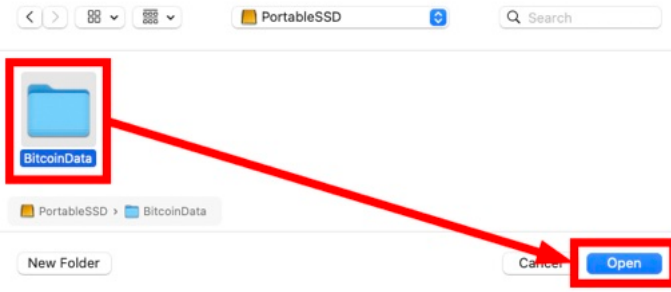


파일 접근 권한을 물으면 [허용]을 누른다.



그러면 바로 비트코인 노츠가 실행된다. 여기서 [Use a custom data directory:](커스텀 데이터 폴더 사용:)를 선택하고 오른쪽의 [...] 버튼을 눌러 외장 SSD에서 만들었던 폴더를 선택한다. 이 폴더에 비트코인 블록 데이터가 설치될 것이다.



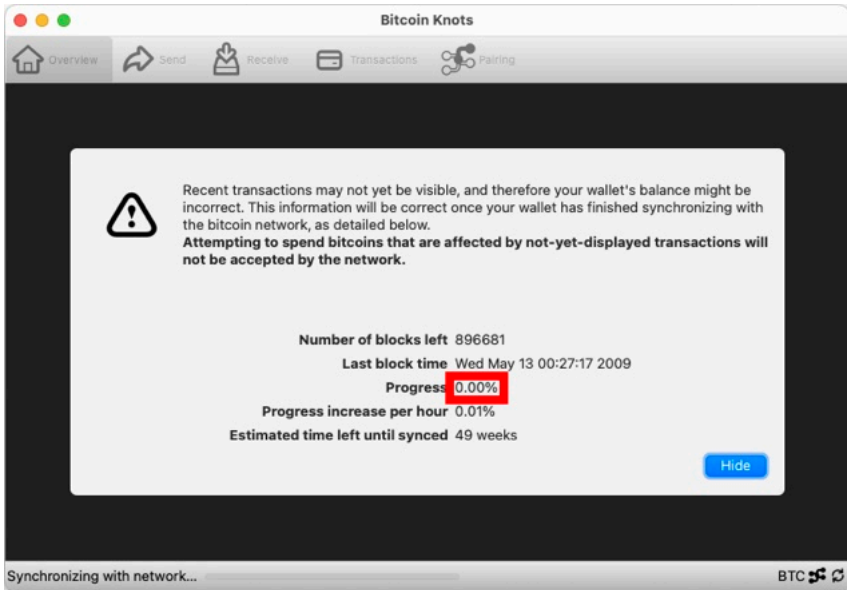


‘Limit block chain storage to (블록체인 스토리지를 다음으로 제한하기)’ 옵션은 가지치기 설정이다. 이 옵션을 체크하면 설정된 용량의 최신 블록만 남기고 과거 블록 데이터는 지운다.

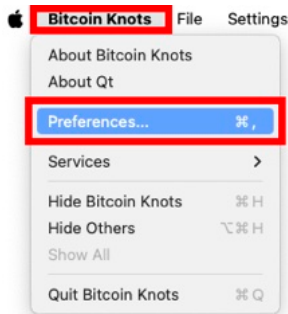
‘Skip validation of the transactions until after block:’ 체크박스는 특정 블록 이전까지의 거래 서명 검증을 건너뛰는 기능이다. 이 옵션을 켜면 해당 블록과 그 이전 블록들의 거래가 정상일 것이라고 가정한다는 뜻이다. 따라서 일정 부분 신뢰에 의존하게 된다. 다만 블록 간의 연결성이나 블록 헤더의 작업증명 같은 기본적인 유효성 검증은 여전히 수행하며, 블록 데이터 자체도 모두 다운로드한다. 단지 거래 검증만 생략하는 것이다. 이렇게 하면 동기화 속도가 빨라진다. 비트코인에서는 어떤 거래가 블록에 포함되고, 그 위로 충분히 많은 블록이 쌓이면 사실상 되돌릴 수 없는 기록으로 여긴다. 즉, 해당 거래는 이미 수많은 노드들에 의해 검증되었다고 간주할 수 있다.

그러나 이런 신뢰를 두는 것이 내키지 않는다면, 체크박스를 해제하면 된다. 이 경우 제네시스 블록 이후부터 지금까지 온체인에서 이루어진 모든 거래의 서명을 직접 검증하게 된다. 다만 이 방식은 동기화가

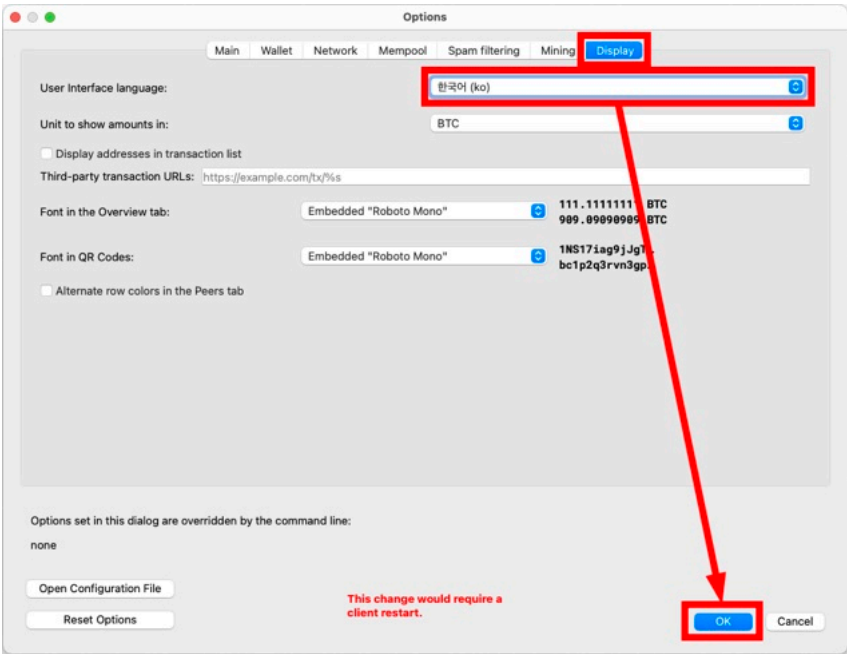
그러면 비트코인 노츠가 동기화를 시작한다. 이 과정은 인터넷 환경에 따라 하루-일주일 정도 걸릴 수 있다.



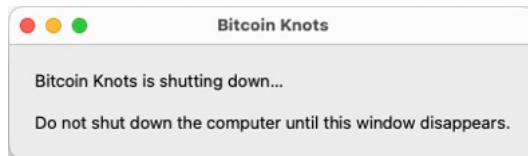
동기화가 완료되기 전에 먼저 노츠의 언어 설정을 한국어로 바꾸자. 좌측 상단 [Bitcoin Knots] → [Preferences]를 선택한다.



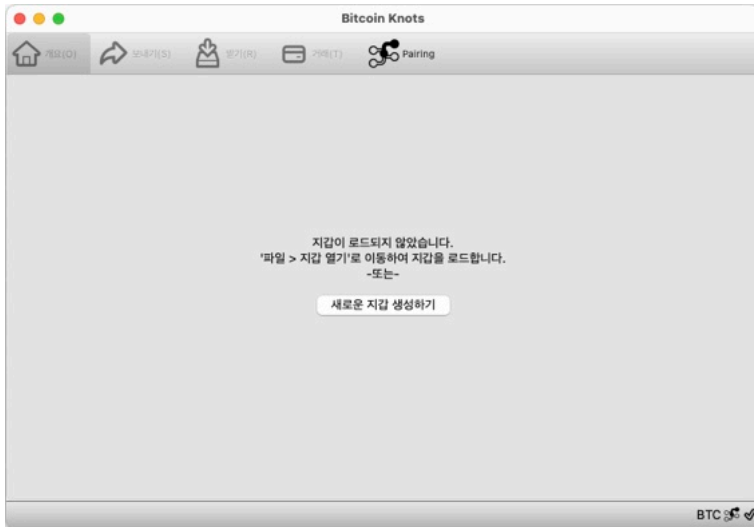
[Display] → ‘User Interface language’에서 [한국어 (ko)]를 선택한다. 그리고 [OK]를 누른다.



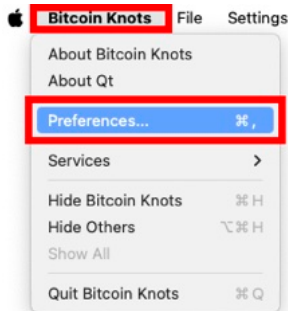
그리고 비트코인 노츠에서 ‘Command’ + ‘Q’를 눌러 비트코인 노츠를 완전히 종료한다.



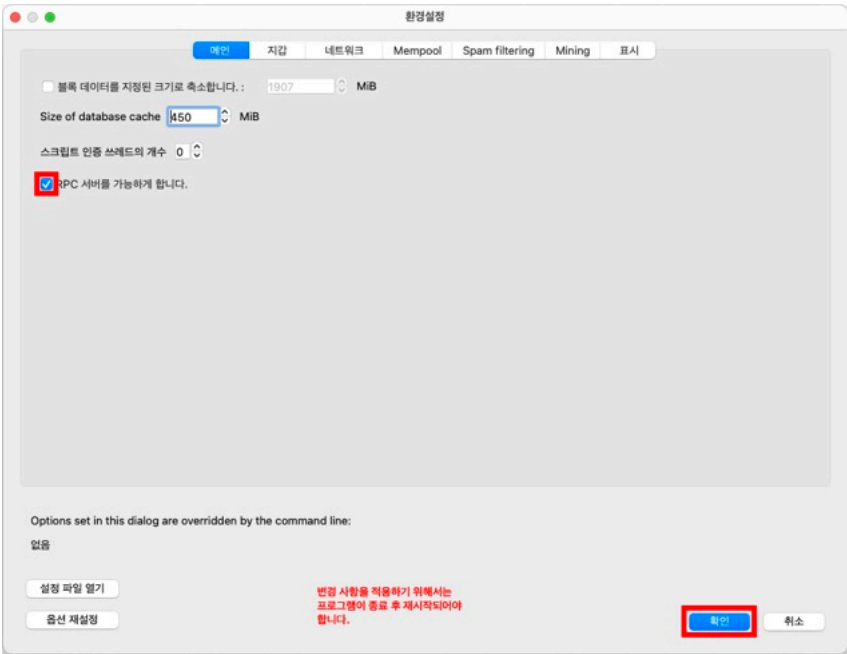
완전히 종료되었으면 다시 노트를 실행한다. 그러면 바뀐 설정이 적용된다. 동기화가 완료되면 다음과 같은 화면이 나온다.



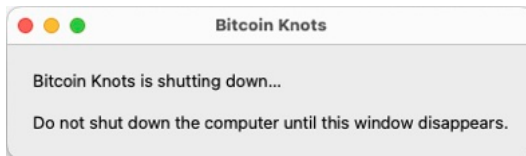
이후에 스페로우 지갑을 연결하기 위해 한 가지 설정을 켜놓고 가자. 상단 탭에서 [Bitcoin Knots] → [Preferences]를 누른다.



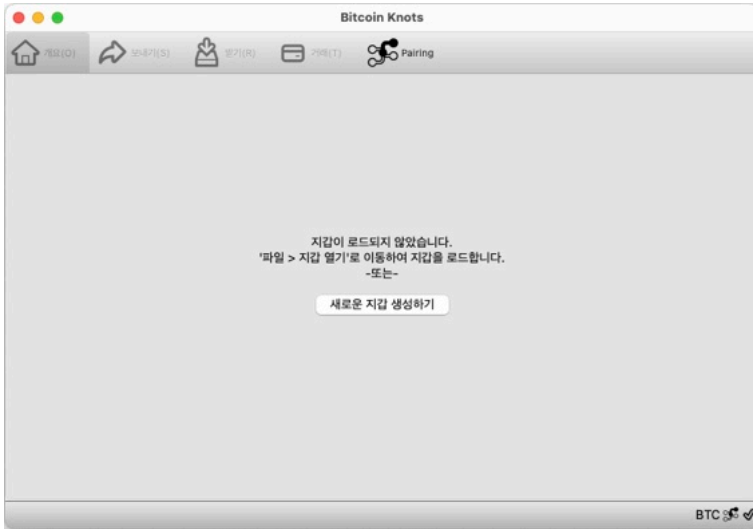
[RPC 서버를 가능하게 합니다.]에 체크하고 [확인]을 누른다.



그리고 비트코인 노트에서 'Command' + 'Q'를 눌러 비트코인 노트를 완전히 종료한다.



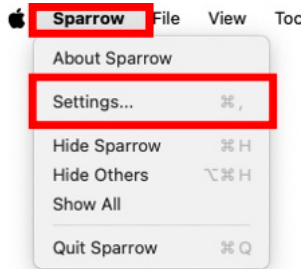
비트코인 노츠를 다시 실행한다. 그러면 바뀐 설정이 적용된다.



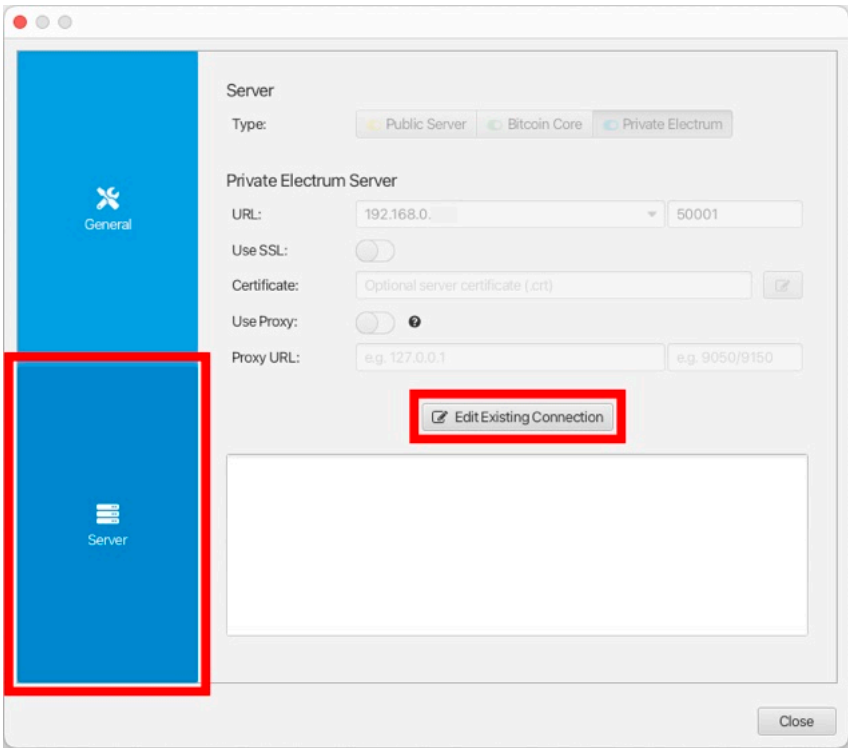
같은 기기에서 스파로우 지갑 연결하기

코어 혹은 노츠를 설치한 데스크톱 기기에서는 스파로우 위치-온리 지갑과 풀 노드를 연결할 수 있다.

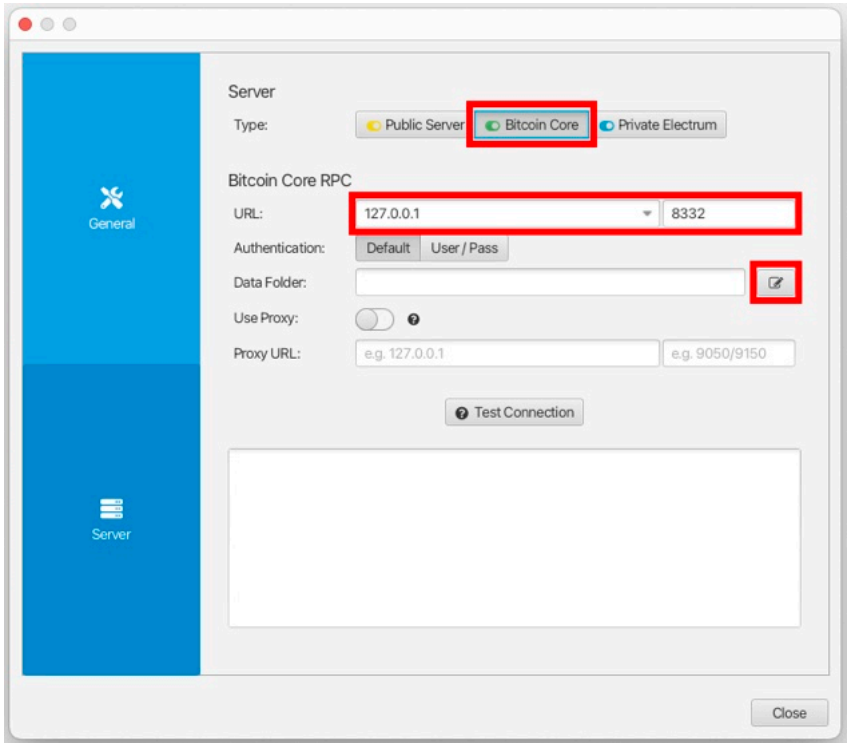
스파로우에서 좌측 상단 [Sparrow] → [Settings]를 누른다.



[Server]를 누르고, [Edit Existing Connection]을 누른다.



상단 가운데에 있는 [Bitcoin Core]를 누르고, 'Data Folder' 옆에 있는 버튼을 누른다. URL에 있는 127.0.0.1과 포트 번호 8332는 그대로 둔다. 참고로 127.0.0.1은 로컬 호스트의 IP 주소, 즉 현재 기기를 가리키는 주소다.

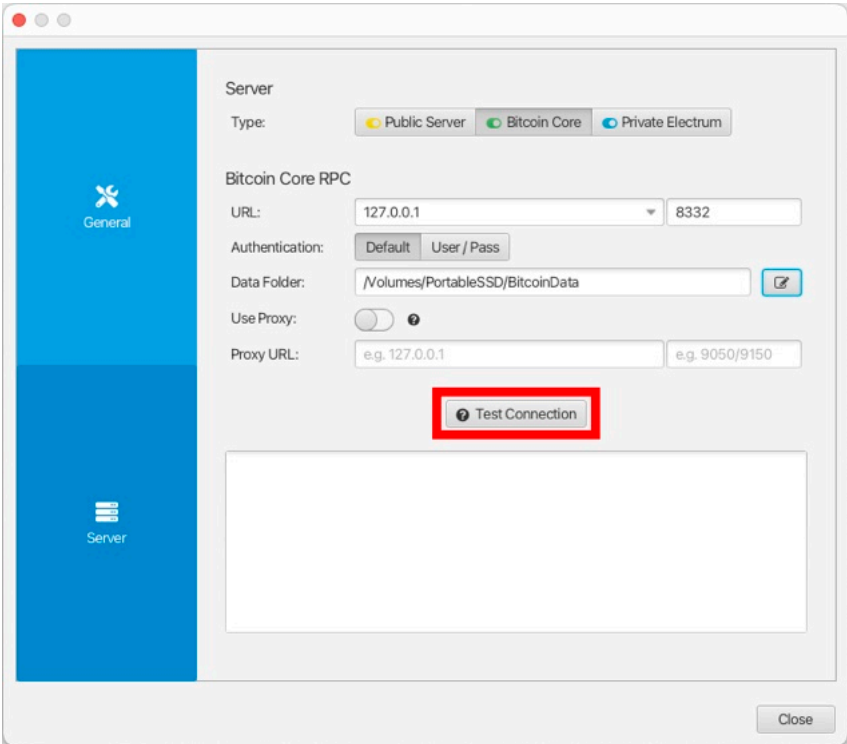


비트코인 코어나 노츠의 RPC 인터페이스를 이용해 풀 노드에 조회 명령을 보내려면 원래는 RPC 사용자 이름과 비밀번호 등을 미리 지정해 두어야 한다. 하지만 코어나 노츠가 설치된 기기에서는 .cookie 파일에 있는 임시 RPC 사용자 이름과 비밀번호를 통해 풀 노드와 계속 통신할 수 있다. 따라서 스페로우가 해당 파일을 찾을 수 있도록 비트코인

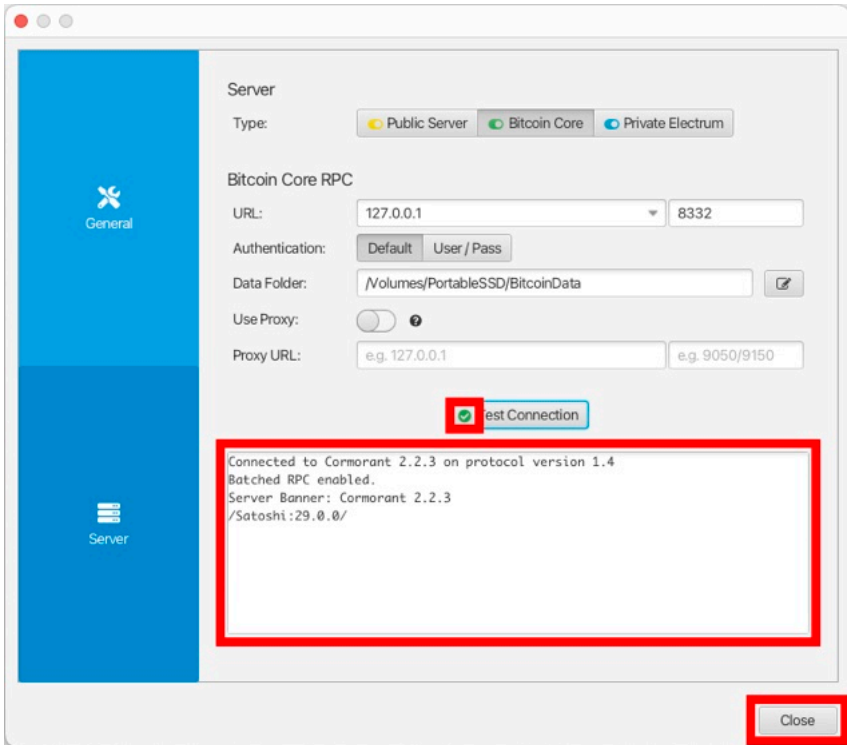
데이터가 저장된 폴더를 지정해 주어야 한다. 앞에서 외장 SSD에 비트코인 데이터를 저장했으므로 스페로우에서 이 폴더를 똑같이 선택한다.



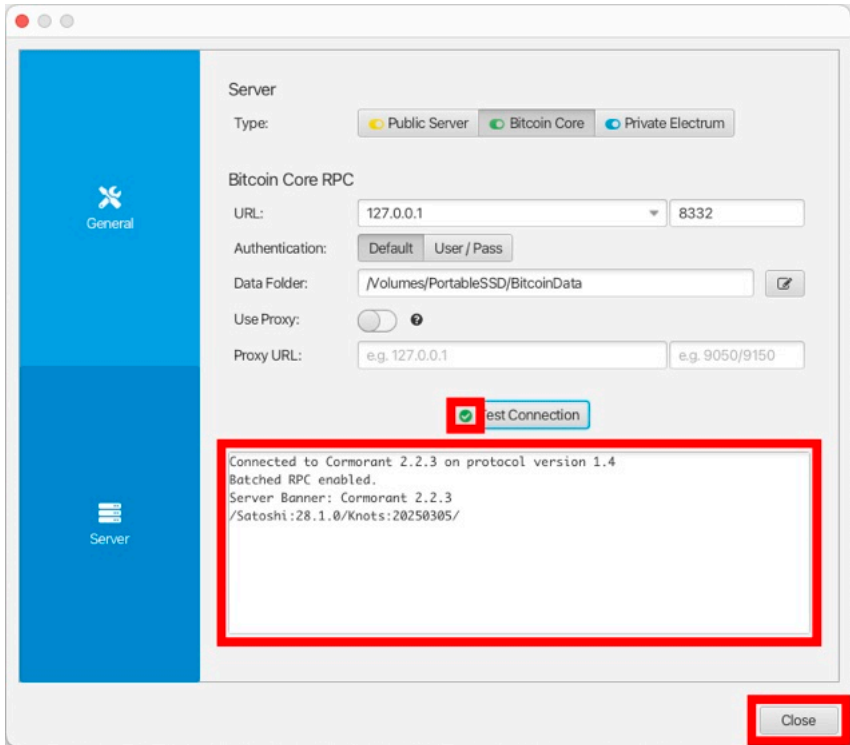
[Test Connection]을 누른다.



비트코인 코어와 연결이 성공적으로 되면 초록색 체크 표시와 함께 다음 사진과 같은 문구가 나온다. [Close]를 누르면 된다.



비트코인 노츠와 연결이 성공적으로 되면 초록색 체크 표시와 함께 다음 사진과 같은 문구가 나온다. [Close]를 누르면 된다.



스패로우 지갑이 코어나 노츠와 잘 연결되었다면 스패로우 하단에 초록색 토글 스위치가 켜진다.

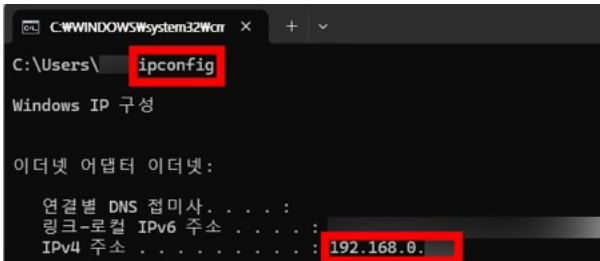


| 로컬 네트워크에서 스페로우 지갑과 비트코인 코어, 노츠 연결하기

코어, 노츠가 설치된 기기의 로컬 IP 주소 알아내기

만약 로컬 네트워크에 있는 다른 기기에서 스페로우 지갑을 사용하는데 비트코인 코어, 노츠와 연결하려면 어떻게 해야 할까? 쉽게 말하자면 컴퓨터 1과 컴퓨터 2가 같은 공유기에서 나오는 인터넷을 사용하는데, 컴퓨터 1에는 비트코인 코어나 노츠가 설치되어 있고 컴퓨터 2에 스페로우 지갑이 설치되어 있는 상황이다. 두 컴퓨터 모두 벽에서 바로 나오는 랜선이 아니라 같은 공유기를 사용한다는 가정하에 서술한다. 이때는 스페로우가 풀 노드의 정보를 조회하기 위해 정말로 원격으로 RPC를 사용하므로(원래 RPC의 뜻이 원격 프로시저 호출이다) RPC 사용자 이름과 비밀번호를 지정해 주어야 한다. 지금부터는 그 방법에 대해 알아볼 것이다.

먼저 코어 혹은 노츠가 설치된 컴퓨터 기기의 로컬 IP 주소를 알아내야 한다. 윈도우OS에서는 다음과 같이 하면 된다. ‘윈도우’ + ‘R’ 키를 누르고, ‘cmd’를 실행한다. 터미널에 `ipconfig`를 입력하면 로컬 IP 주소가 나온다. 이를 잘 기억한다.



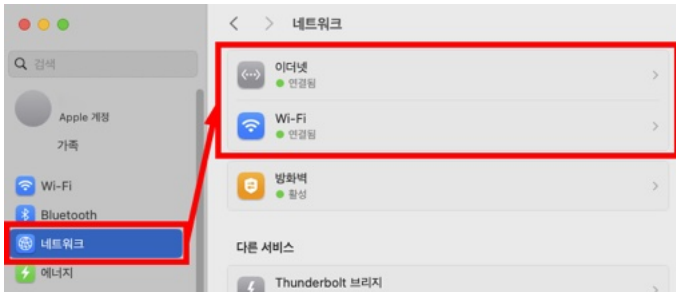
```
C:\WINDOWS\system32\cmd.exe
C:\Users\ > ipconfig

Windows IP 구성

이더넷 어댑터 이더넷:

    연결별 DNS 접미사 . . . . . :
    링크-로컬 IPv6 주소 . . . . . :
    IPv4 주소 . . . . . : 192.168.0.
```

맥OS에서는 다음과 같이 하면 된다. 설정에 들어가 [네트워크]를 누르고 랜선으로 연결되어 있다면 [이더넷]을, 와이파이로 연결되어 있다면 [Wi-Fi]를 누른다.



이더넷에 들어가면 바로 로컬 IP 주소가 나온다. 이를 잘 기억한다.



Wi-Fi에 들어갔을 때는 연결된 와이파이 옆의 [세부사항]을 누르고 스크롤을 아래로 내려보면 로컬 IP 주소가 나온다. 이를 잘 기억한다.



IP 주소	192.168.0.
라우터	192.168.0.1

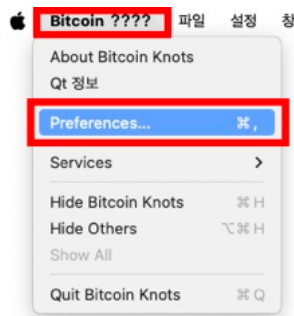
이 네트워크 지우기... 취소 확인

bitcoin.conf 파일 설정하기

로컬 IP 주소를 알아냈다면 코어 혹은 노트에서 설정에 들어가야 한다. 윈도우에서는 [설정] → [옵션]에 들어가면 된다.



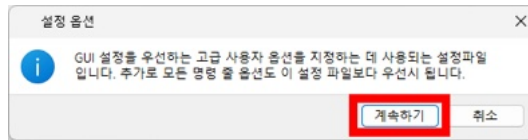
맥에서는 [Bitcoin Core] 혹은 [Bitcoin Knots] → [Preferences]에 들어가면 된다.



[설정 파일 열기]를 누른다.




[계속하기]를 누른다.



그러면 'bitcoin.conf'라는 설정 파일이 열린다. 윈도우에서는 메모장으로 열면 되고, 맥에서는 텍스트 편집기로 열면 된다.

여기에 다음을 입력한다.

```
server=1
rpcuser=[자신이 설정할 RPC 사용자 이름]
rpcpassword=[자신이 설정할 RPC 비밀번호:
  영어 대소문자, 특수문자 섞어서 반드시 길게]
rpcbind=[앞에서 확인했던 이 기기의 로컬 IP 주소]
rpccallowip=192.168.0.0/24
```



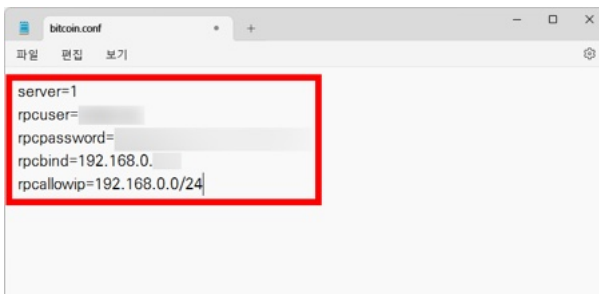
한 줄씩 설명해 보겠다. `server=1`은 RPC 서버를 가능하게 하는 옵션과 같은 기능이다.

`rpcuser`와 `rpcpassword`에는 각각 사용자 이름과 비밀번호를 설정하면 된다. 비밀번호는 꼭 영어 대소문자, 특수문자를 섞어서 길게 설정하자. 비밀번호를 직접 저장하지 않고 해시값을 저장하는 `rpcauth` 방식이 더 권장되나 여기서 다루지는 않겠다. RPC 사용자 이름과 비밀번호는 유출되지 않도록 잘 관리하자.

`rpcbind`는 어떤 주소로 명령어를 들을지 정하는 것이다. 그래서 코어 혹은 노츠가 설치된 기기의 로컬 IP 주소를 적는 것이다. 만약 코어 혹은 노츠가 설치된 기기에서도 스패로우를 사용할 것이라면 밑에 `rpcbind=127.0.0.1` 한 줄을 더 써주면 된다.

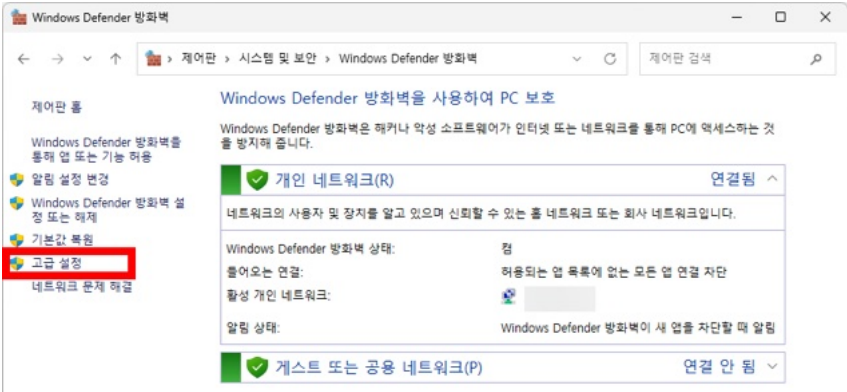
`rpccallowip`는 코어 혹은 노츠에 연결을 허용할 IP 주소를 지정하는 것이다. `192.168.0.0/24`는 192.168.0.0부터 192.168.0.255를 말하는 것으로, 로컬 네트워크에 있는 기기들만 접속을 허용하겠다는 뜻이다.

이렇게 입력하고 파일을 저장한다.



윈도우OS에 코어, 노츠가 설치되어 있는 경우 방화벽 해제

이제 코어 혹은 노츠가 설치된 기기에서 방화벽을 해제해야 한다. 윈도우에 코어 혹은 노츠가 설치되어 있다면 해당 기기에서 [제어판] → [시스템 및 보안] → [Windows Defender 방화벽]에 들어가서 [고급 설정]을 누른다.



좌측 탭에서 [인바운드 규칙]을 선택하고, 'Bitcoin Core' 혹은 'Bitcoin Knots'를 선택한다(노츠도 'Bitcoin Core'로 나올 수 있다). 그리고 우측의 [새 규칙]을 누른다.



[포트] → [다음]을 누른다.



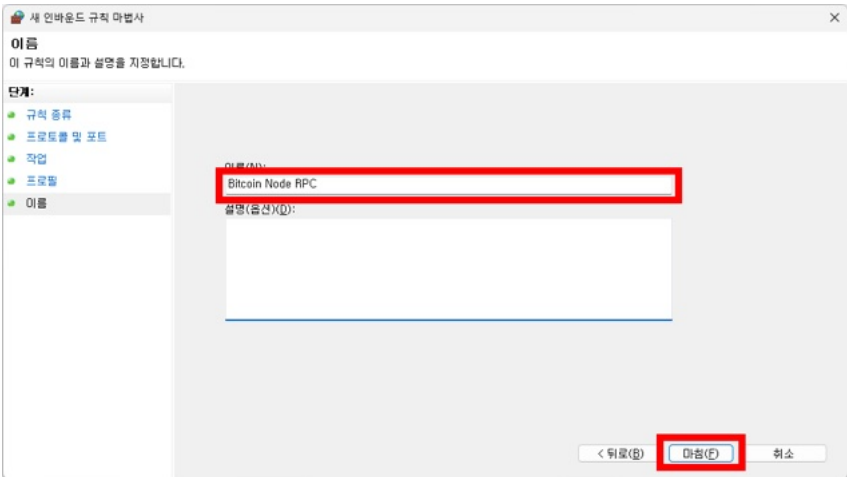
[TCP]를 선택하고 [특정 로컬 포트]에서 '8332'를 입력한다. 그리고 [다음]을 누른다.



[연결 허용]을 선택하고, [다음]을 누른다.



이름에 'Bitcoin Node RPC' 등을 입력하고 [마침]을 누른다.

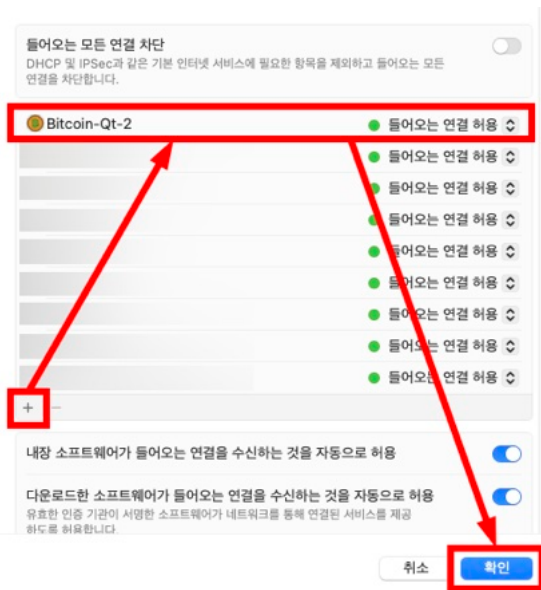


맥OS에 코어, 노츠가 설치되어 있는 경우 방화벽 해제

맥에서는 기본적으로 방화벽이 비활성화되어 있다. [설정] → [네트워크] → [방화벽]에서 확인할 수 있다. 비활성화되어 있다면 따로 바꿔줄 설정은 없다.



만약 맥에서 방화벽을 사용한다면 방화벽에서 [옵션...]을 누르고 [+] 버튼을 눌러 비트코인 코어 혹은 노츠 프로그램을 선택하고 [확인]을 누르면 된다.

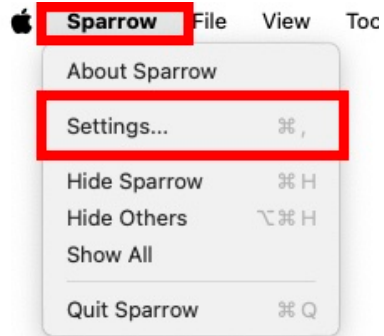


로컬 네트워크에서 스패로우 지갑 연결하기

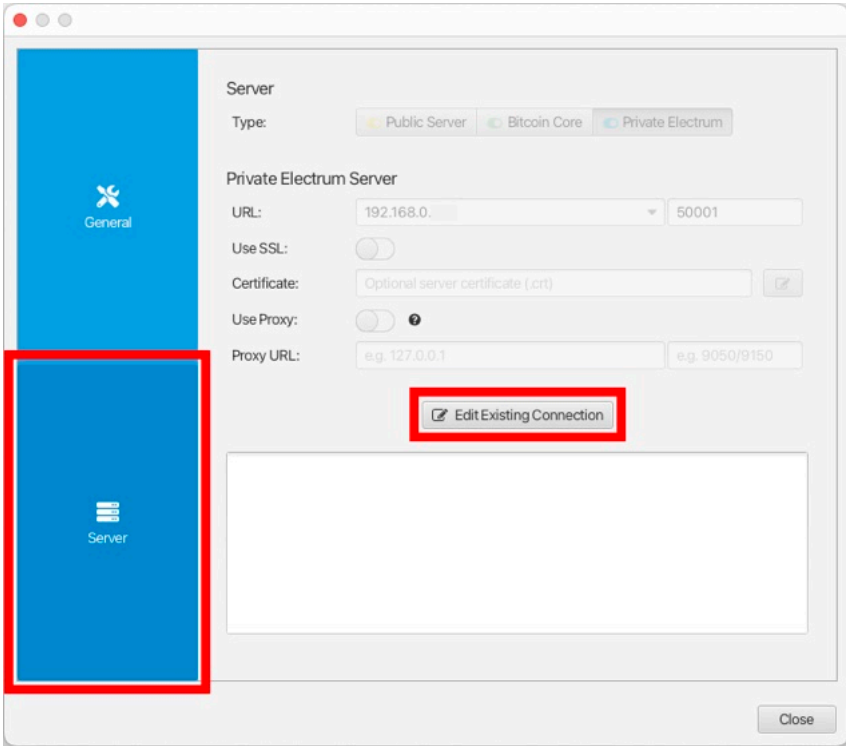
이제 로컬 네트워크에 있는 다른 컴퓨터의 스패로우 지갑에서 코어 혹은 노츠에 연결해보자.

윈도우: 스패로우에서 좌측 상단 [File] → [Settings]를 누른다.

맥: 스패로우에서 좌측 상단 [Sparrow] → [Settings]를 누른다.



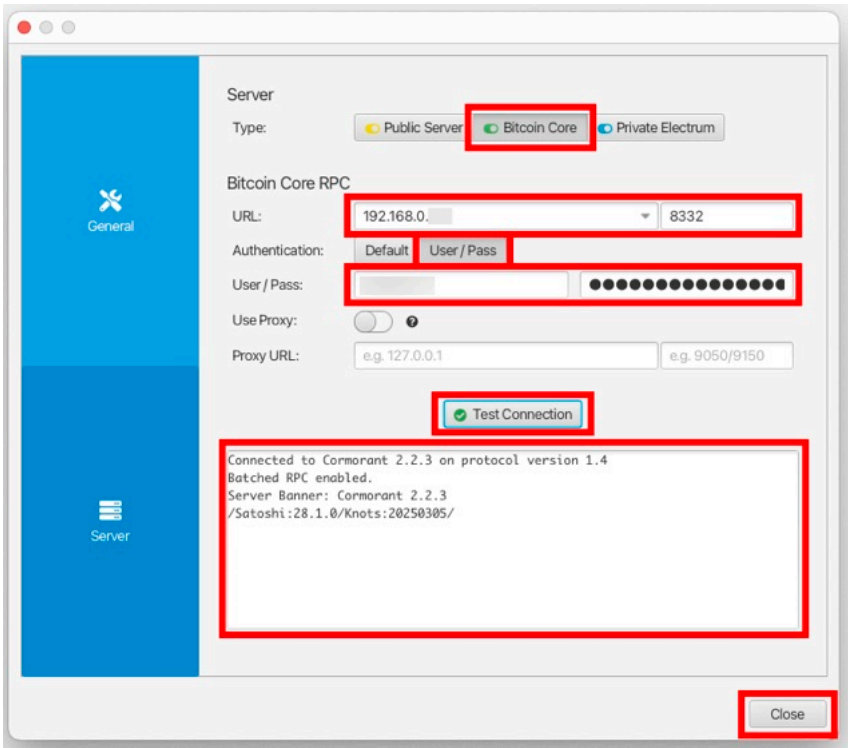
[Server]를 누르고, [Edit Existing Connection]을 누른다.



상단 가운데에 있는 [Bitcoin Core]를 누르고, 'URL'에 코어 혹은 노드가 설치된 기기의 로컬 IP 주소를 적는다. 포트 번호는 8332 그대로 둔다.

그 아래 'User / Pass'에 bitcoin.conf 파일에서 지정했던 RPC 사용자 이름과 비밀번호를 정확하게 입력한다.

잘 연결이 되면 [Test Connection]을 눌렀을 때 초록색 체크 표시와 함께 다음 사진과 같은 문구가 나온다. 그러면 [Close]를 누르고 스패우를 사용하면 된다.



이로써 윈도우OS와 맥OS에서 비트코인 코어, 노츠를 설치하고 스페로우 지갑과 연동하는 방법까지 모두 알아보았다.