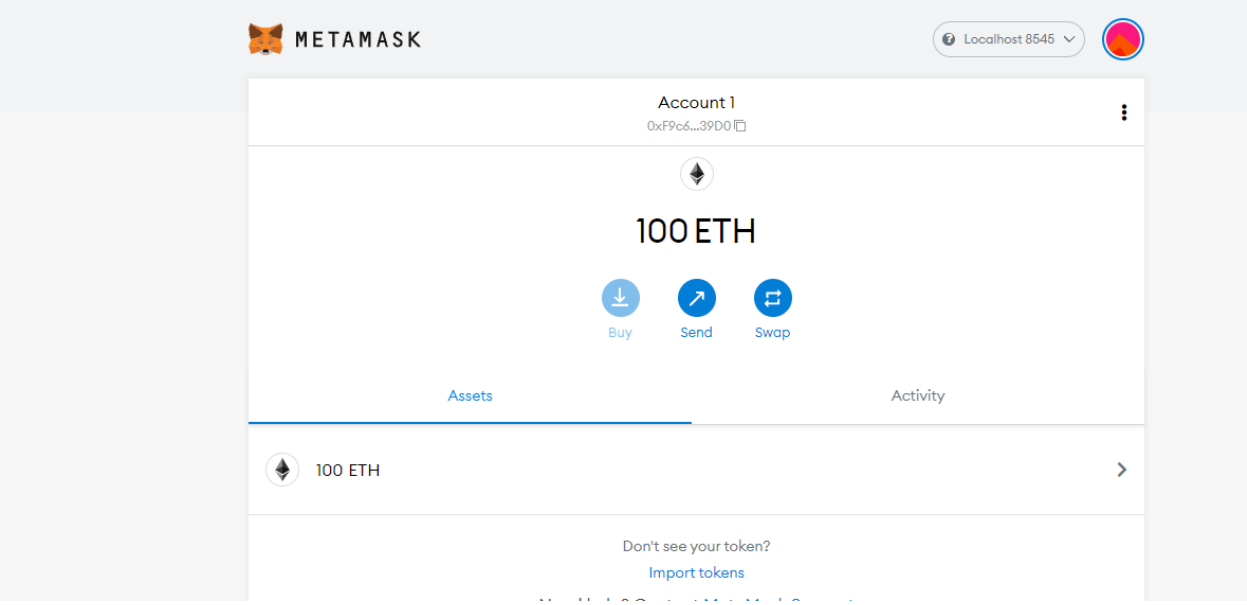
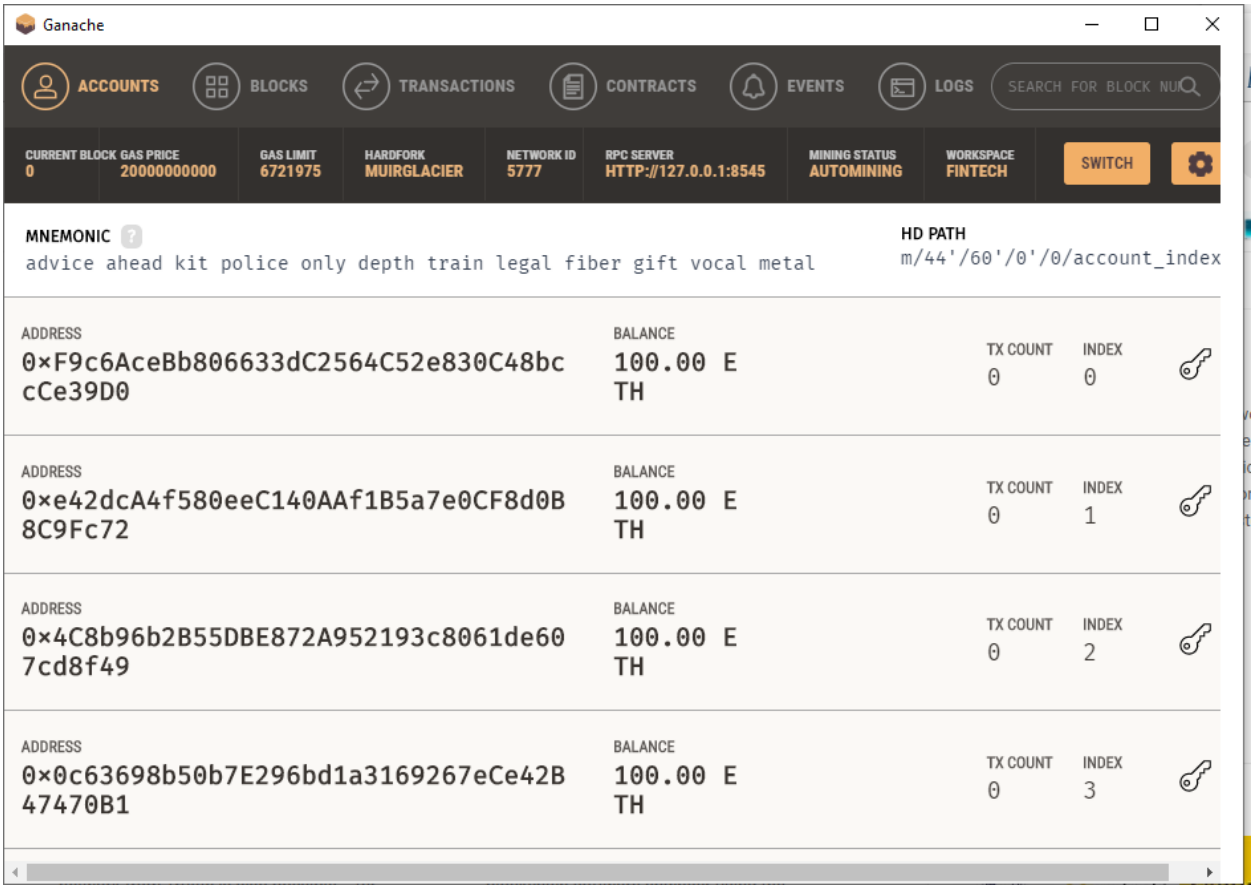
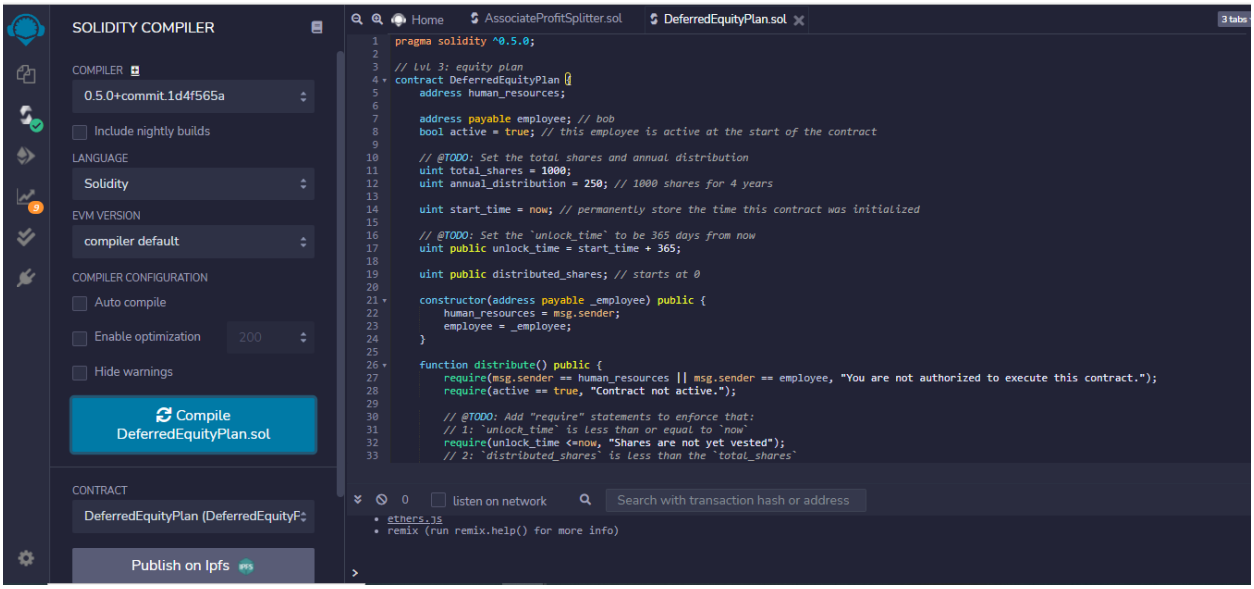
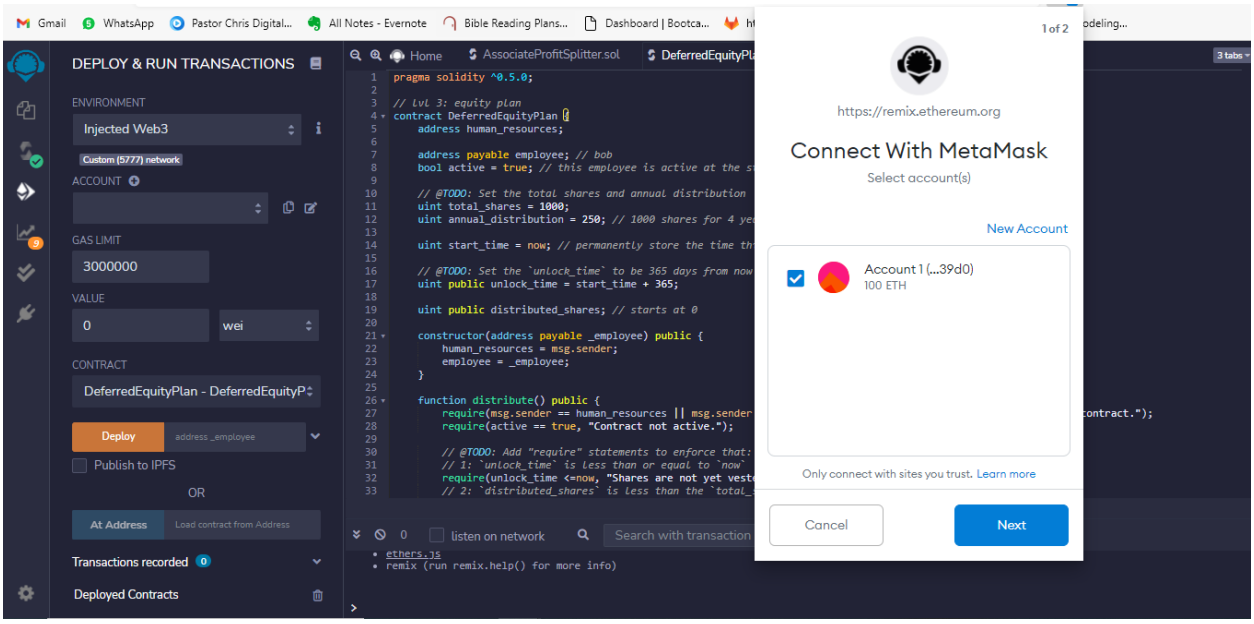

METAMASK
Localhost 8545
Account 1
0xF9c6...39D0
100 ETH
Buy
Send
Swap
Assets
Activity
100 ETH
Don't see your token?
Import tokens

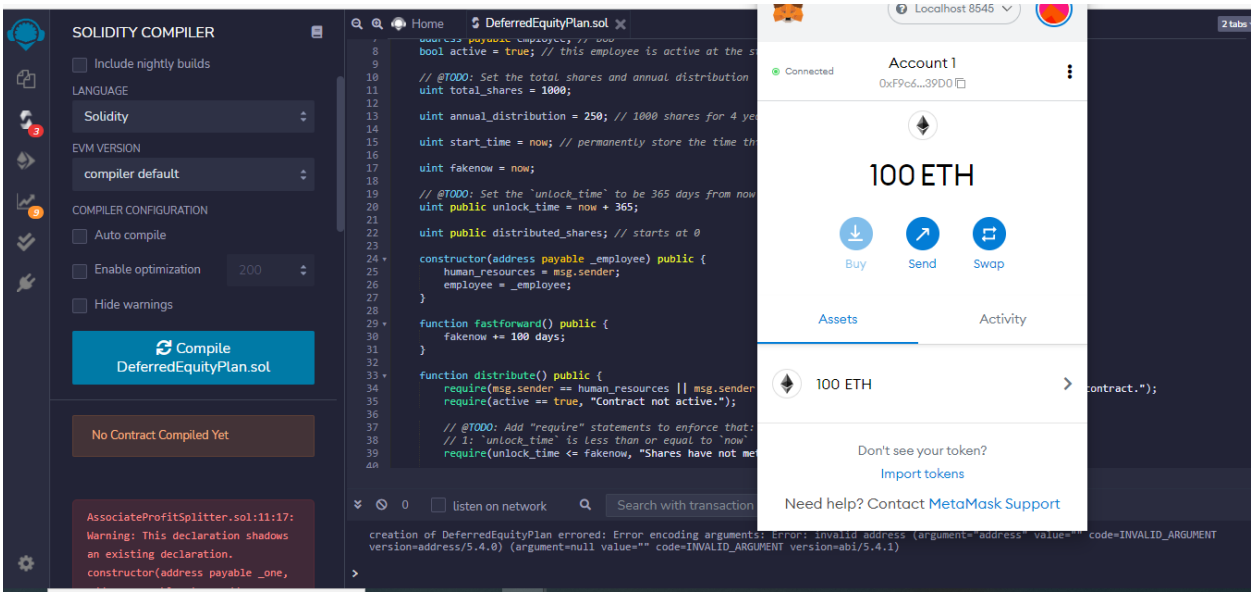

Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NU
CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MUIRGLACIER
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE FINTECH
SWITCH
MNEMONIC
advice ahead kit police only depth train legal fiber gift vocal metal
HD PATH
m/44'/60'/0'/0/account_index
ADDRESS 0x F9c6AceBb806633dC2564C52e830C48bccCe39D0
BALANCE 100.00 ETH
TX COUNT 0
INDEX 0
ADDRESS 0xe42dcA4f580eeC140AAf1B5a7e0CF8d0B8C9Fc72
BALANCE 100.00 ETH
TX COUNT 0
INDEX 1
ADDRESS 0x4C8b96b2B55DBE872A952193c8061de607cd8f49
BALANCE 100.00 ETH
TX COUNT 0
INDEX 2
ADDRESS 0x0c63698b50b7E296bd1a3169267eCe42B47470B1
BALANCE 100.00 ETH
TX COUNT 0
INDEX 3

Home AssociateProfitSplitter.sol

```sol
pragma solidity ^0.5.0;

// Lvl 1: equal split
contract AssociateProfitSplitter {
    // @TODO: Create three payable addresses representing `employee_one`, `employee_two` and `employee_three`.
    address payable employee_one;
    address payable employee_two;
    address payable employee_three;


    constructor(address payable _one, address payable _two, address payable _three) public {
        employee_one = _one;
        employee_two = _two;
        employee_three = _three;
    }

    function balance() public view returns(uint) {
        return address(this).balance;
    }

    function deposit() public payable {
        // @TODO: Split `msg.value` into three
        uint amount = msg.value/3

        // @TODO: Transfer the amount to each employee

        employee_one.transfer(uint amount);
        employee_two.transfer(uint amount);
        employee_three.transfer(uint amount);

        // @TODO: take care of a potential remainder by sending back to HR (`msg.sender`)
        msg.sender.transfer(msg.value - (amount *3));
    }
```




DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Injected Web3
Custom (5777) network
ACCOUNT
GAS LIMIT
3000000
VALUE
0
wei
CONTRACT
DeferredEquityPlan - DeferredEquityP
Deploy
address _employee
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 0
Deployed Contracts
Home
AssociateProfitSplitter.sol
DeferredEquityPl
pragma solidity ^0.5.0;
// lvl 3: equity plan
contract DeferredEquityPlan {
address human_resources;
address payable employee; // bob
bool active = true; // this employee is active at the s
// @TODO: Set the total shares and annual distribution
uint total_shares = 1000;
uint annual_distribution = 250; // 1000 shares for 4 ye
uint start_time = now; // permanently store the time th
// @TODO: Set the `unlock_time` to be 365 days from now
uint public unlock_time = start_time + 365;
uint public distributed_shares; // starts at 0
constructor(address payable _employee) public {
human_resources = msg.sender;
employee = _employee;
}
function distribute() public {
require(msg.sender == human_resources || msg.sender
require(active == true, "Contract not active.");
// @TODO: Add "require" statements to enforce that:
// 1: `unlock_time` is less than or equal to `now`
require(unlock_time <=now, "Shares are not yet veste
// 2: `distributed_shares` is less than the `total_
contract.");
listen on network
Search with transaction
ethers.js
remix (run remix.help() for more info)
1 of 2
https://remix.ethereum.org
Connect With MetaMask
Select account(s)
New Account
Account 1 (...39d0)
100 ETH
Only connect with sites you trust. Learn more
Cancel
Next


SOLIDITY COMPILER
Include nightly builds
LANGUAGE
Solidity
EVM VERSION
compiler default
COMPILER CONFIGURATION
Auto compile
Enable optimization
200
Hide warnings
Compile DeferredEquityPlan.sol
No Contract Compiled Yet
AssociateProfitSplitter.sol:11:17:
Warning: This declaration shadows
an existing declaration.
constructor(address payable _one,
Home
DeferredEquityPlan.sol
bool active = true; // this employee is active at the s
// @TODO: Set the total shares and annual distribution
uint total_shares = 1000;
uint annual_distribution = 250; // 1000 shares for 4 ye
uint start_time = now; // permanently store the time th
uint fakenow = now;
// @TODO: Set the `unlock_time` to be 365 days from now
uint public unlock_time = now + 365;
uint public distributed_shares; // starts at 0
constructor(address payable _employee) public {
human_resources = msg.sender;
employee = _employee;
}
function fastforward() public {
fakenow += 100 days;
}
function distribute() public {
require(msg.sender == human_resources || msg.sender
require(active == true, "Contract not active.");
// @TODO: Add "require" statements to enforce that:
// 1: `unlock_time` is less than or equal to `now`
require(unlock_time <= fakenow, "Shares have not me
contract.");
listen on network
Search with transaction
creation of DeferredEquityPlan errored: Error encoding arguments: Error: invalid address (argument="address" value="" code=INVALID_ARGUMENT
version=address/5.4.0) (argument=null value="" code=INVALID_ARGUMENT version=abi/5.4.1)
Localhost 8545
Connected
Account 1
0xF9c6...39D0
100 ETH
Buy
Send
Swap
Assets
Activity
100 ETH
Don't see your token?
Import tokens
Need help? Contact MetaMask Support