HOW TO DEPLOY ON OTHER TESTNETS (Part 2)

This guide will walk you through on how to deploy a smart contract on non-Ethereum, EVM compatible test networks.
You will learn how to deploy via *Remix IDE*, a web-based smart contract creation tool, and *Hardhat*, a local Ethereum development network.
You do not have to do both, but it may be helpful to try both to see what suits your deployment better.
From my personal experience, I have found Remix IDE to be a simpler deployment process, but Hardhat produced faster results and cheaper costs.

The test networks that will be used are:

- **Matic Mumbai (Ethereum Sidechain)**
  - **https://mumbai.polygonscan.com/**

- **Arbitrum Rinkeby (Ethereum Layer 2)**
  - **https://rinkeby-explorer.arbitrum.io/**

- **IOTA 2.0 Beta (Ethereum Virtual Machine)**
  - **https://explorer.wasp.sc.iota.org/**

In order to complete this tutorial, you must have completed Part 1 of this tutorial series, or have your own deployment-ready smart contract. If you would simply like to test and deploy you may use the code provided in the github repository.

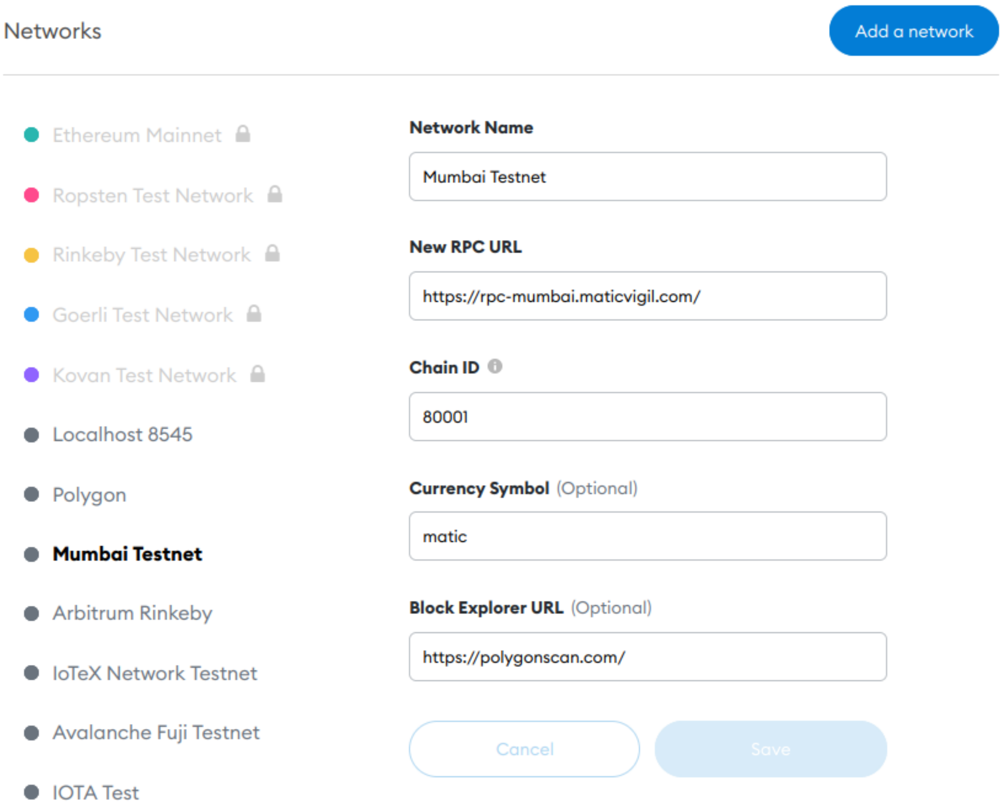
Add test networks to your Metamask

Matic Mumbai

Go to Add Network on your Metamask browser extension wallet.

Input the following:
Network Name: Mumbai Testnet
New RPC URL: https://rpc-mumbai.maticvigil.com/
Chain IDL 80001
Currency Symbol: MATIC
Block Explorer URL: https://polygonscan.com/

Your page should look like the following screenshot.



Network Name
Mumbai Testnet

New RPC URL
https://rpc-mumbai.maticvigil.com/

Chain ID
80001

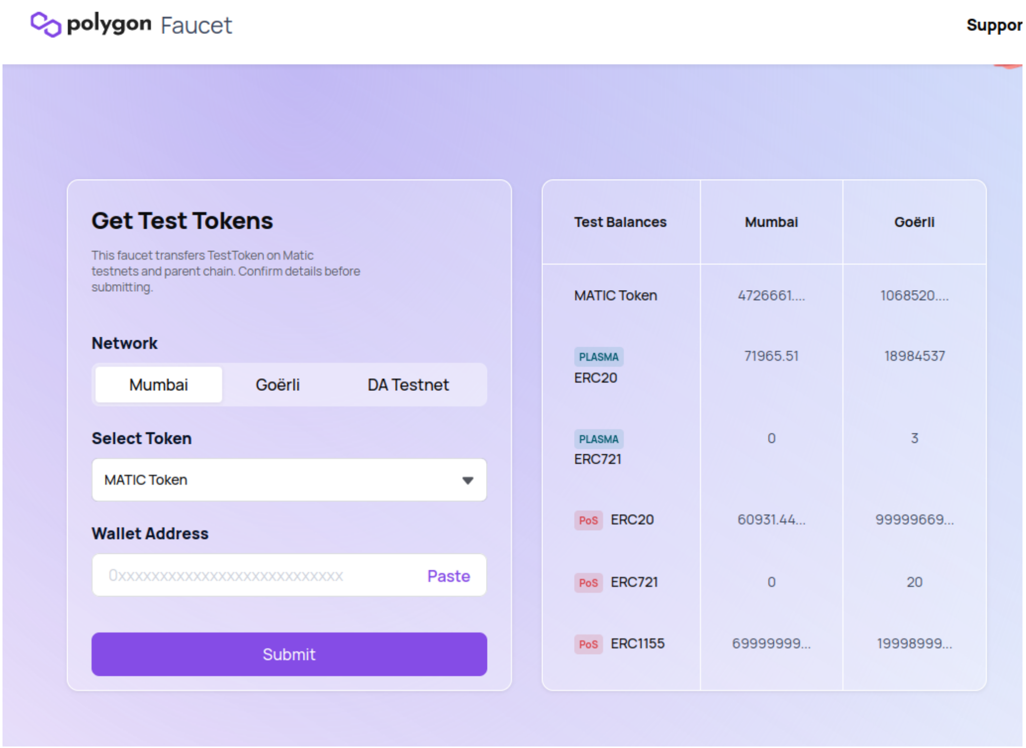
Currency Symbol (Optional)
matic

Block Explorer URL (Optional)
https://polygonscan.com/

Request MATIC test funds from the polygon faucet.
https://faucet.polygon.technology/

The settings to use are:

- Network: Mumbai
- Token: MATIC Token
- Copy and paste your wallet address



Get Test Tokens

This faucet transfers TestToken on Matic testnets and parent chain. Confirm details before submitting.

Network: Mumbai | Goërli | DA Testnet

Select Token: MATIC Token

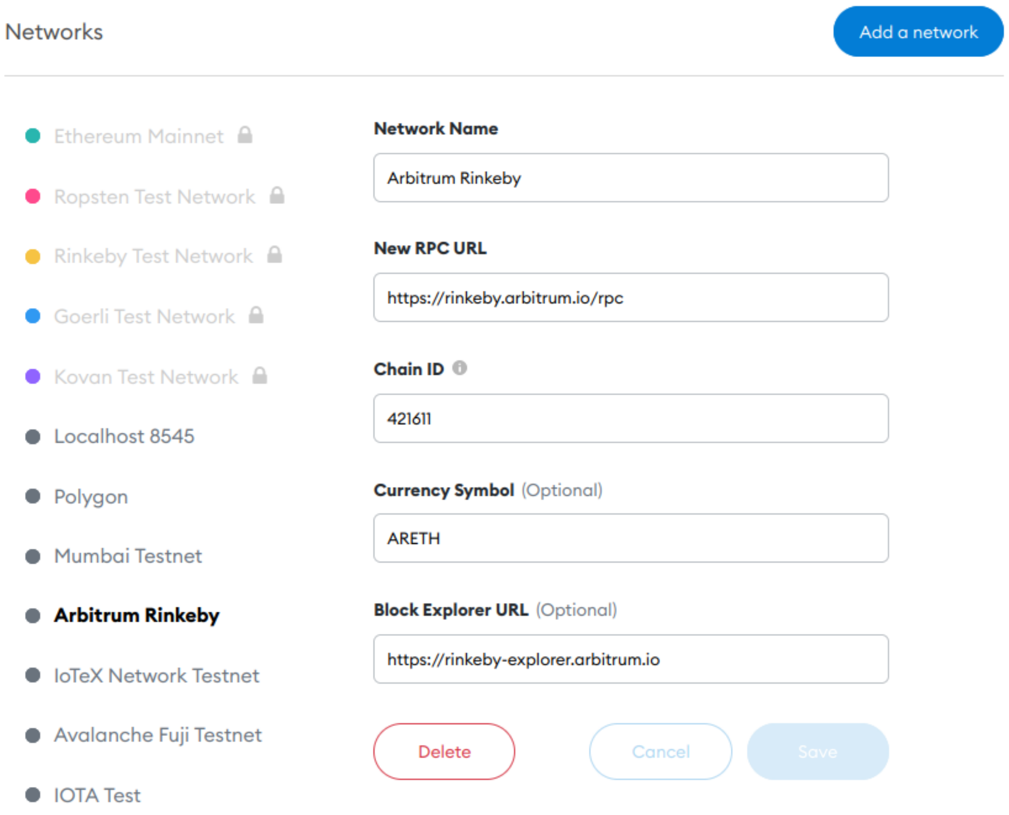
Wallet Address: 0xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Test Balances	Mumbai	Goërli
MATIC Token	4726661....	1068520....
PLASMA ERC20	71965.51	18984537
PLASMA ERC721	0	3
PoS ERC20	60931.44...	99999669...
PoS ERC721	0	20
PoS ERC1155	69999999...	19998999...

Arbitrum Rinkeby

Input the following.

Network Name: Arbitrum Rinkeby
New RPC URL: https://rinkeby.arbitrum.io/rpc
Chain IDL 421611
Currency Symbol: ARETH
Block Explorer URL: https://rinkeby-explorer.arbitrum.io



Networks

- Ethereum Mainnet
- Ropsten Test Network
- Rinkeby Test Network
- Goerli Test Network
- Kovan Test Network
- Localhost 8545
- Polygon
- Mumbai Testnet
- **Arbitrum Rinkeby**
- IoTeX Network Testnet
- Avalanche Fuji Testnet
- IOTA Test

Network Name
Arbitrum Rinkeby

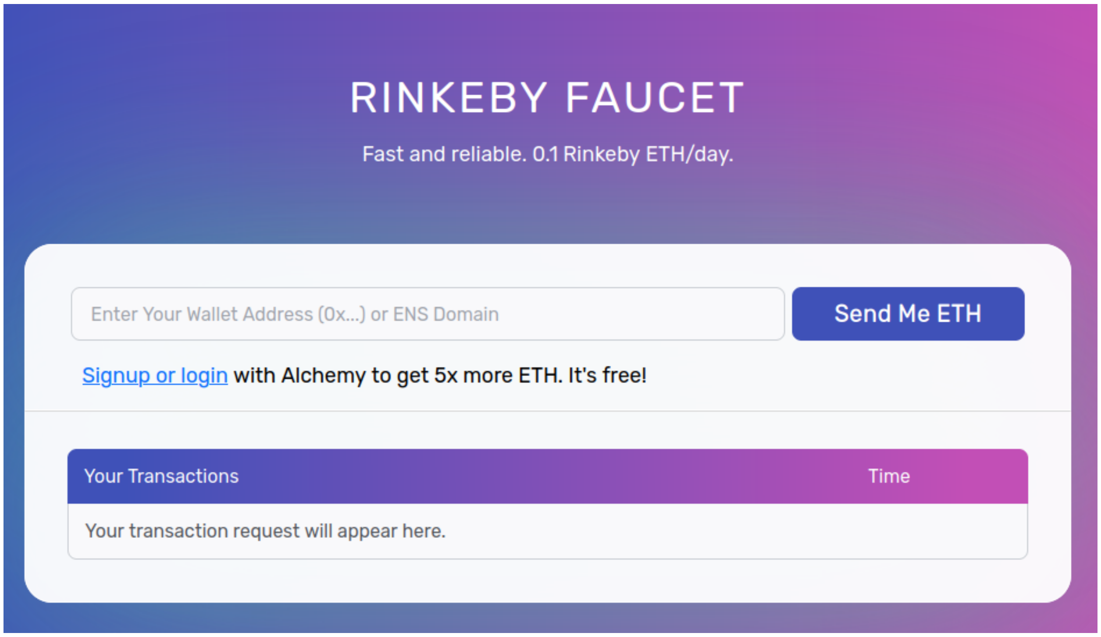
New RPC URL
https://rinkeby.arbitrum.io/rpc

Chain ID
421611

Currency Symbol (Optional)
ARETH

Block Explorer URL (Optional)
https://rinkeby-explorer.arbitrum.io

Request Rinkeby faucet funds from https://rinkebyfaucet.com/



RINKEBY FAUCET

Fast and reliable. 0.1 Rinkeby ETH/day.

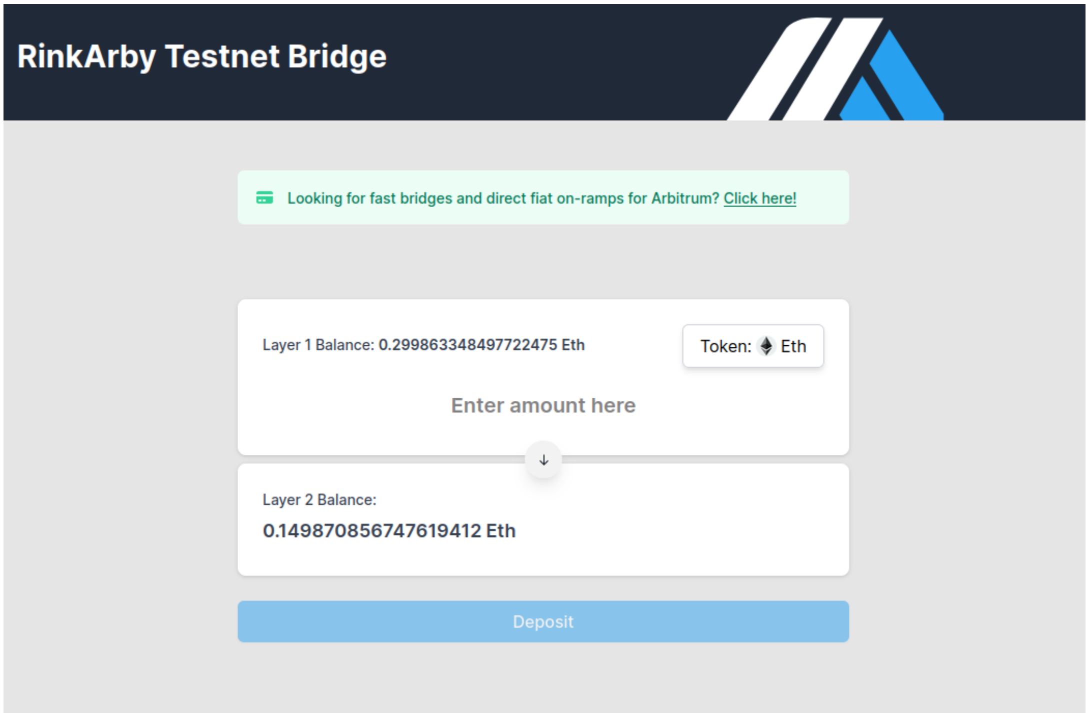
Enter Your Wallet Address (0x...) or ENS Domain

Signup or login with Alchemy to get 5x more ETH. It's free!

Your Transactions	Time
Your transaction request will appear here.	

Arbitrum Rinkeby testnet requires another step where you must bridge the faucet funded ETH to Arbitrum via https://bridge.arbitrum.io/

Connect your metamask wallet, change your network to Arbitrum Rinkeby, then deposit your Layer 1 ETH to Layer 2.



RinkArby Testnet Bridge

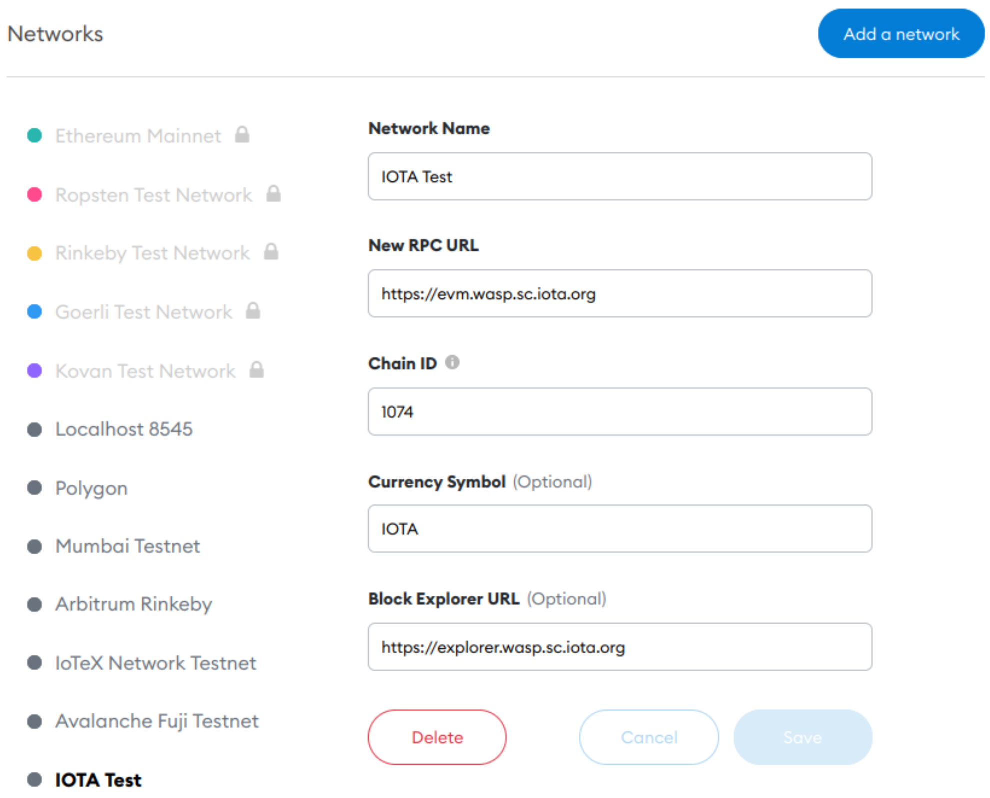
Looking for fast bridges and direct fiat on-ramps for Arbitrum? Click here!

Layer 1 Balance: 0.299863348497722475 Eth

Layer 2 Balance:
0.149870856747619412 Eth

IOTA 2.0 EVM

Network Name: IOTA EVM (optional)
New RPC URL: https://evm.wasp.sc.iota.org
Chain IDL 1074
Currency Symbol: IOTA
Block Explorer URL: https://explorer.wasp.sc.iota.org



Network Name: IOTA Test

New RPC URL: https://evm.wasp.sc.iota.org

Chain ID: 1074

Currency Symbol (Optional): IOTA

Block Explorer URL (Optional): https://explorer.wasp.sc.iota.org

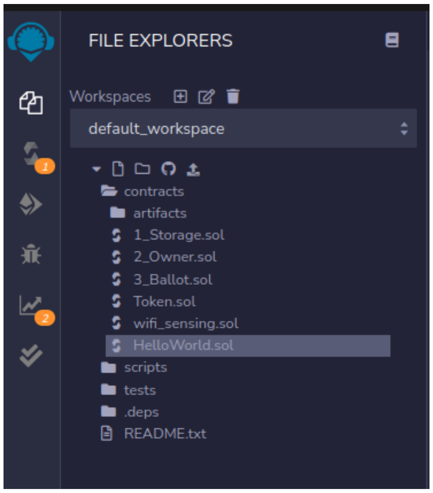
IOTA EVM is currently in Beta access (as of March 2022), so transaction fees are set to 0. You do not need faucet funds to deploy at this time.

Remix IDE

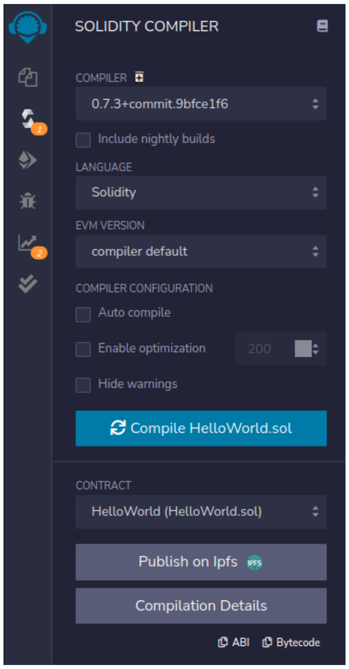
https://remix.ethereum.org/

Testing different networks with Remix IDE is simple.

Step 1: Upload your contract code to the File Explorer workspace.



Step 2: Compile your contract. Remix will automatically detect the necessary Solidity compiler version.


SOLIDITY COMPILER

COMPILER: 0.7.3+commit.9bfce1f6

□ Include nightly builds

LANGUAGE: Solidity

EVM VERSION: compiler default

COMPILER CONFIGURATION

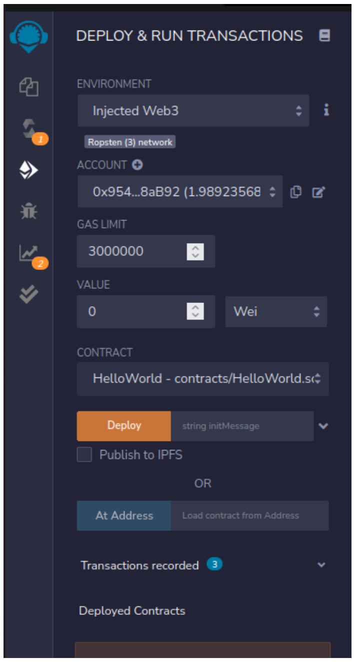
□ Auto compile

□ Enable optimization 200

□ Hide warnings

CONTRACT: HelloWorld (HelloWorld.sol)

Step 3: Change the network in your Metamask wallet, and set your deploy environment to Injected Web3.



DEPLOY & RUN TRANSACTIONS

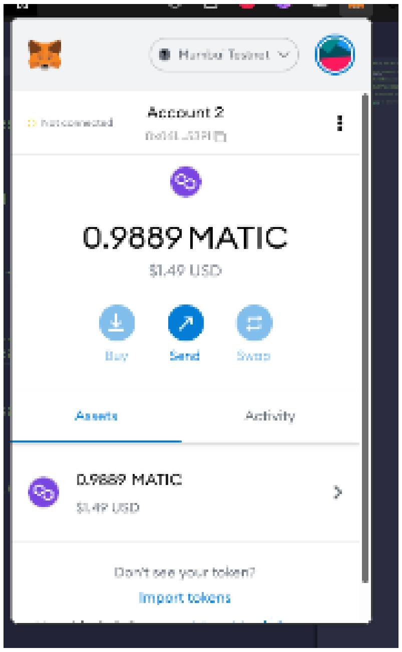
ENVIRONMENT
Injected Web3
Ropsten (3) network

ACCOUNT
0x954...8aB92 (1.98923568

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
HelloWorld - contracts/HelloWorld.sol



0.9889 MATIC
$1.49 USD

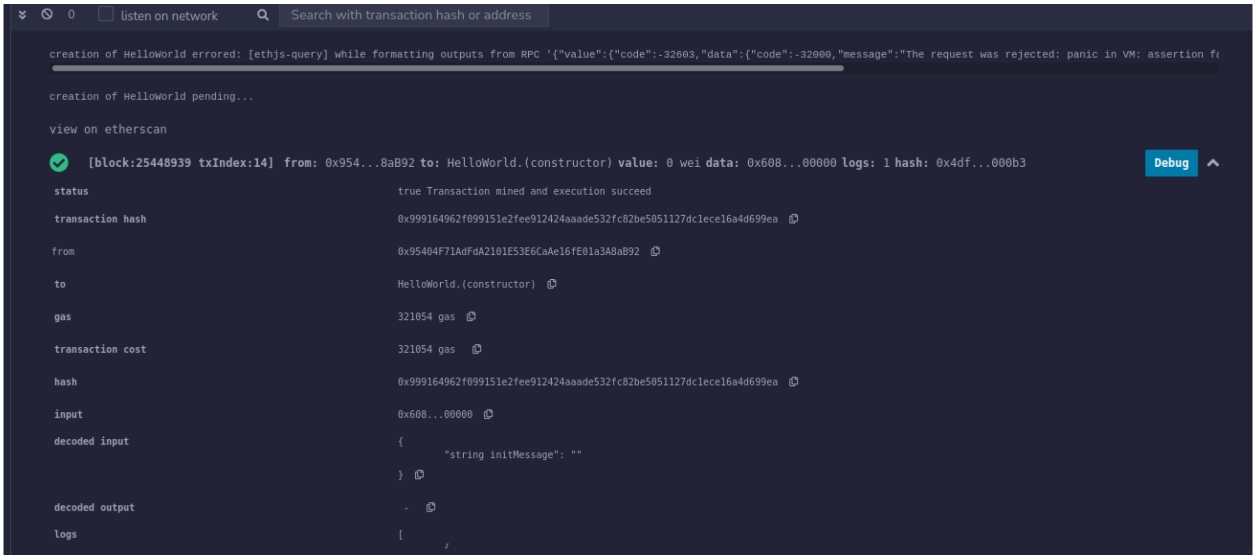
0.9889 MATIC
$1.49 USD

Remix will automatically detect the network in your Metamask wallet and deploy to the corresponding wallet using the required test funds.

If you do not have any faucet funds in your wallet, your transaction will not be processed due to not meeting the gas requirement.

Step 4: View your deployed contract in the output terminal.

You can copy your transaction hash and direct it to the respective block explorer to view it deployed on the test network.



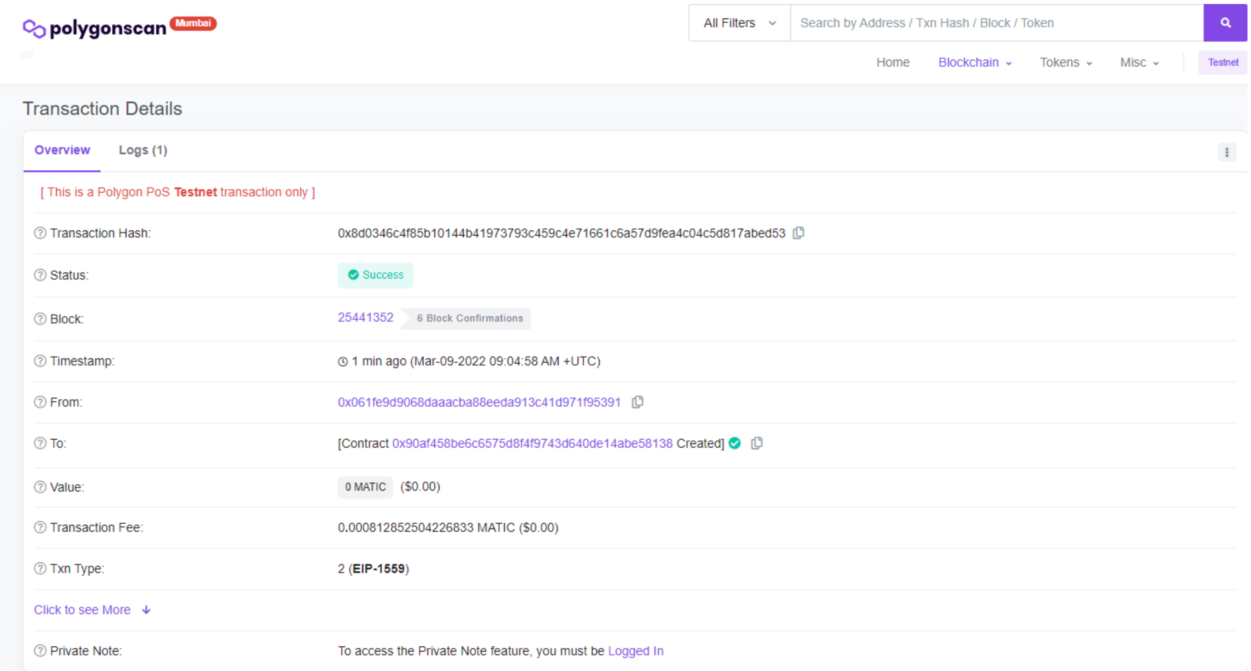
```
creation of HelloWorld errored: [ethjs-query] while formatting outputs from RPC '{"value":{"code":-32603,"data":{"code":-32000,"message":"The request was rejected: panic in VM: assertion fa

creation of HelloWorld pending...

view on etherscan

[block:25448939 txIndex:14] from: 0x954...8aB92 to: HelloWorld.(constructor) value: 0 wei data: 0x608...00000 logs: 1 hash: 0x4df...000b3
 status                      true Transaction mined and execution succeed
 transaction hash            0x999164962f099151e2fee912424aaade532fc82be5051127dc1ece16a4d699ea
 from                        0x95404F71AdFdA2101E53E6CaAe16fE01a3A8aB92
 to                          HelloWorld.(constructor)
 gas                         321054 gas
 transaction cost            321054 gas
 hash                        0x999164962f099151e2fee912424aaade532fc82be5051127dc1ece16a4d699ea
 input                       0x608...00000
 decoded input               {
                                     "string initMessage": ""
                             }
 decoded output               -
 logs                        [
```

This example deployed on the Matic Mumbai testnet.
The successful deployment can be verified on the Mumbai Polygon Scan site.



Transaction Details

[This is a Polygon PoS **Testnet** transaction only]

Transaction Hash:	0x8d0346c4f85b10144b41973793c459c4e71661c6a57d9fea4c04c5d817abed53
Status:	Success
Block:	25441352 6 Block Confirmations
Timestamp:	1 min ago (Mar-09-2022 09:04:58 AM +UTC)
From:	0x061fe9d9068daaacba88eeda913c41d971f95391
To:	[Contract 0x90af458be6c6575d8f4f9743d640de14abe58138 Created] ✓
Value:	0 MATIC ($0.00)
Transaction Fee:	0.000812852504226833 MATIC ($0.00)
Txn Type:	2 (**EIP-1559**)

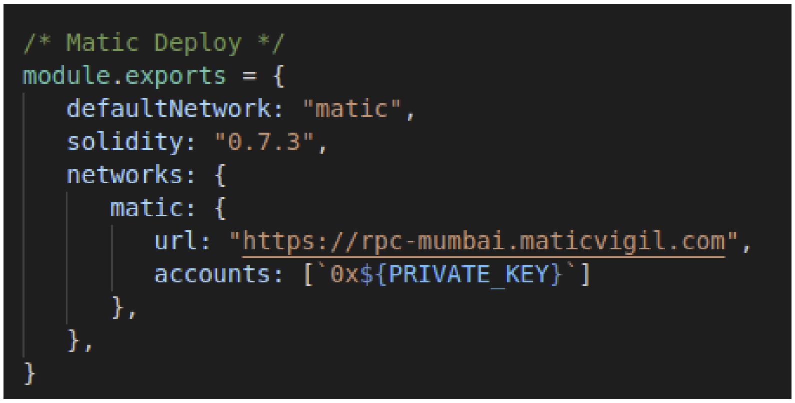
Click to see More

Private Note:	To access the Private Note feature, you must be Logged In

Hardhat

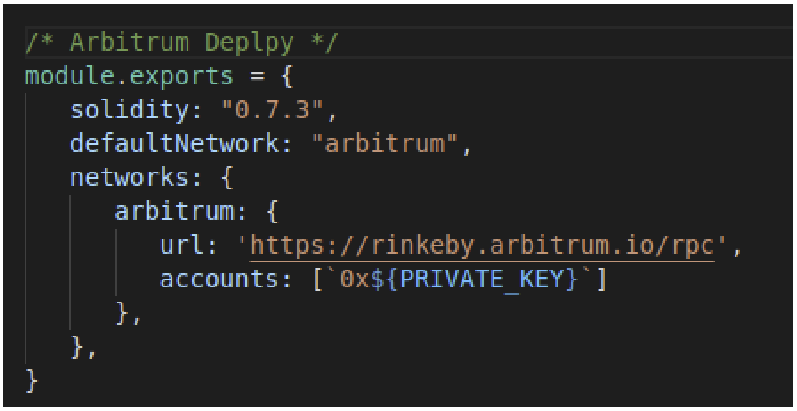
Hardhat deployment on different testnets is simple.
Go to your ***hardhat.config.js*** file and modify the ***RPC URL*** of your ***module.exports*** configuration.

Matic Mumbai:



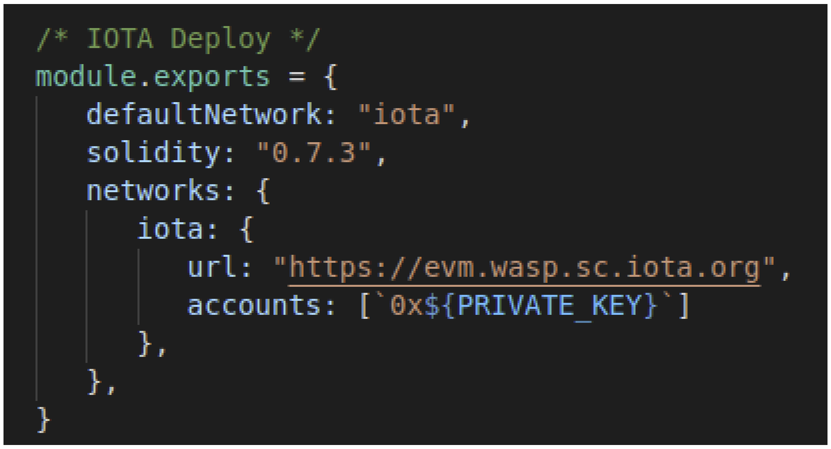
```
/* Matic Deploy */
module.exports = {
  defaultNetwork: "matic",
  solidity: "0.7.3",
  networks: {
    matic: {
      url: "https://rpc-mumbai.maticvigil.com",
      accounts: [`0x${PRIVATE_KEY}`]
    },
  },
}
```

Arbitrum Rinkeby:



```
/* Arbitrum Deplpy */
module.exports = {
  solidity: "0.7.3",
  defaultNetwork: "arbitrum",
  networks: {
    arbitrum: {
      url: 'https://rinkeby.arbitrum.io/rpc',
      accounts: [`0x${PRIVATE_KEY}`]
    },
  },
}
```

IOTA EVM:



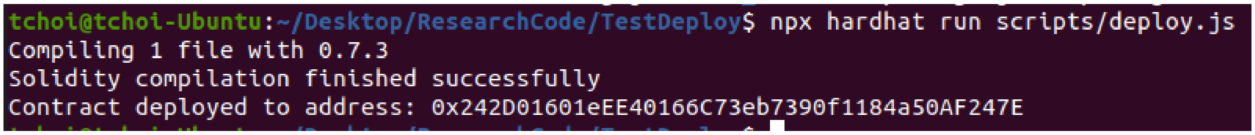
```
/* IOTA Deploy */
module.exports = {
  defaultNetwork: "iota",
  solidity: "0.7.3",
  networks: {
    iota: {
      url: "https://evm.wasp.sc.iota.org",
      accounts: [`0x${PRIVATE_KEY}`]
    },
  },
}
```

In your project terminal window,

```
npx hardhat run scripts/deploy.js
```

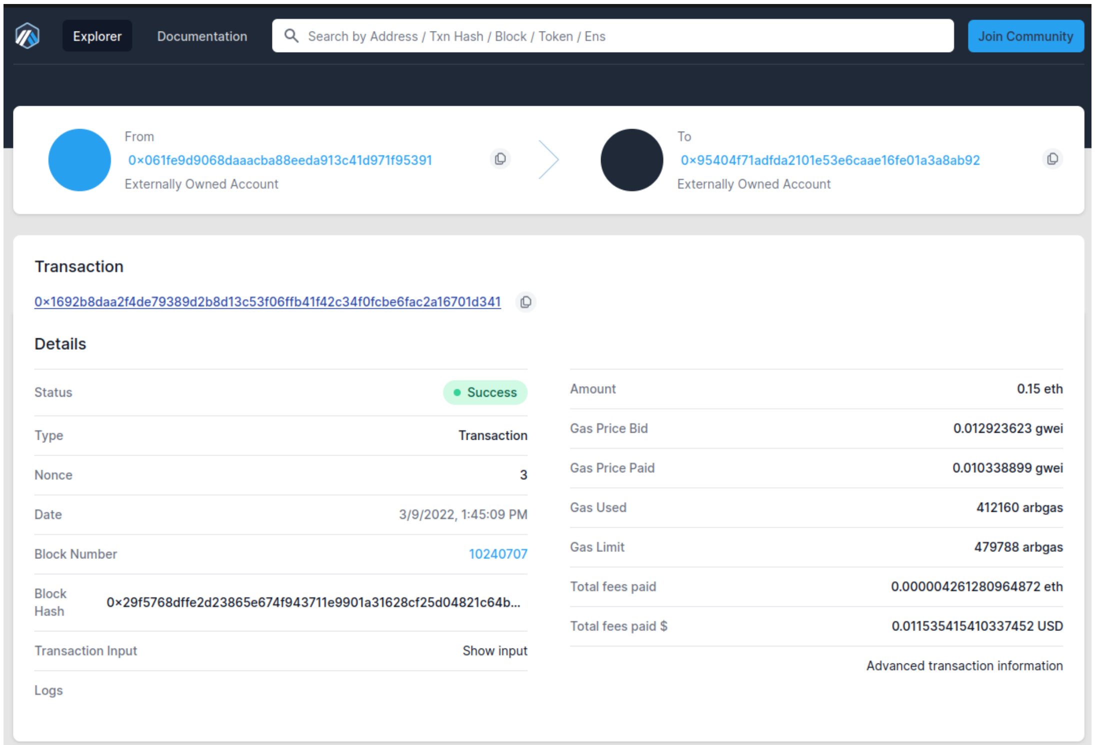
Hardhat will automatically detect the new RPC URL and deploy your contract to the detected test network.

Your terminal will output an address of where your contract is deployed.



```
tchoi@tchoi-Ubuntu:~/Desktop/ResearchCode/TestDeploy$ npx hardhat run scripts/deploy.js
Compiling 1 file with 0.7.3
Solidity compilation finished successfully
Contract deployed to address: 0x242D01601eEE40166C73eb7390f1184a50AF247E
```

Go to the respective Block Explorer URL pages and paste your address.
You will see your successfully deployed contract and the fees your transaction incurred.



From
0×061fe9d9068daaacba88eeda913c41d971f95391
Externally Owned Account

To
0×95404f71adfda2101e53e6caae16fe01a3a8ab92
Externally Owned Account

Transaction

0×1692b8daa2f4de79389d2b8d13c53f06ffb41f42c34f0fcbe6fac2a16701d341

Details

Status	Success
Type	Transaction
Nonce	3
Date	3/9/2022, 1:45:09 PM
Block Number	10240707
Block Hash	0×29f5768dffe2d23865e674f943711e9901a31628cf25d04821c64b...
Transaction Input	Show input
Logs	

Amount	0.15 eth
Gas Price Bid	0.012923623 gwei
Gas Price Paid	0.010338899 gwei
Gas Used	412160 arbgas
Gas Limit	479788 arbgas
Total fees paid	0.000004261280964872 eth
Total fees paid $	0.011535415410337452 USD

Advanced transaction information

Congratulations!

You have successfully done the following:

- Added Matic, Arbitrum, IOTA test networks to Metamask
- Received faucet funds of the different testnets
- Deployed your contract on the web via Remix IDE
- Deployed your contract via Hardhat
- Viewed and verified your deployed contract on the respective blockchains.