Deploy a dapp on your Arbitrum rollup devnet

First, review theArbitrum integration ,Deploy an Arbitrum rollup devnet , andDeploy a smart contract to your Arbitrum rollup pages.

Dependencies

- a funded account to deploy your smart contract
- anArbitrum rollup devnet
- running

Setup and contract deployment

1. Clone thegm-portal
2. from Github and start the frontend:
3. bash
4. cd
5. HOME
6. git
7. clone
8. https://github.com/jcstein/gm-portal.git
9. cd
10. gm-portal
11. &&
12. git
13. checkout
14. arbitrum
15. cd
16. frontend
17. &&
18. yarn
19. &&
20. yarn
21. dev
22. cd
23. HOME
24. git
25. clone
26. https://github.com/jcstein/gm-portal.git
27. cd
28. gm-portal
29. &&
30. git
31. checkout
32. arbitrum
33. cd
34. frontend
35. &&
36. yarn
37. &&
38. yarn
39. dev
40. In a new terminal instance, set your private key for the faucet as a variable and the RPC URL you're using:
41. bash
42. export
43. PRIVATE_KEY
44. =
45. 0xb6b15c8cb491557369f3c7d2c287b053eb229daa9c22138887752191c9520659
46. export
47. ARB_RPC_URL
48. =
49. http://localhost:8547
50. export
51. PRIVATE_KEY
52. =
53. 0xb6b15c8cb491557369f3c7d2c287b053eb229daa9c22138887752191c9520659

export
ARB_RPC_URL
=
http://localhost:8547
Change into thegm-portal/contracts
directory in the same terminal and deploy the contract using Foundry:
bash
cd
HOME
/gm-portal/contracts
forge
script
script/GmPortal.s.sol:GmPortalScript
--rpc-url
ARB_RPC_URL
--private-key
PRIVATE_KEY
--broadcast
cd
HOME
/gm-portal/contracts
forge
script
script/GmPortal.s.sol:GmPortalScript
--rpc-url
ARB_RPC_URL
--private-key
PRIVATE_KEY
--broadcast
In the output of the deployment, find the contract address and set it as a variable:
bash
export
CONTRACT_ADDRESS
=<
your-contract-address-from-the-output-abov
e

export
CONTRACT_ADDRESS
=<
your-contract-address-from-the-output-abov
e

Interact with the contract

Next, you're ready to interact with the contract from your terminal!

1. Send a "gm" to the contract:
2. bash
3. cast
4. send
5. CONTRACT_ADDRESS
6. \
7. "gm(string)"
8. "gm"
9. \
10. --private-key PRIVATE_KEY
11. \
12. --rpc-url ARB_RPC_URL
13. cast
14. send
15. CONTRACT_ADDRESS
16. \
17. "gm(string)"
18. "gm"
19. \
20. --private-key PRIVATE_KEY

\
--rpc-url ARB_RPC_URL
Now that you've posted to the contract, you can read all "gms" (GMs) from the contract with this command:
bash
cast
call
CONTRACT_ADDRESS
"getAllGms()"
--rpc-url
ARB_RPC_URL
cast
call
CONTRACT_ADDRESS
"getAllGms()"
--rpc-url
ARB_RPC_URL
Next, query the total number of gms, which will be returned as a hex value:
bash
cast
call
CONTRACT_ADDRESS
"getTotalGms()"
--rpc-url
ARB_RPC_URL
cast
call
CONTRACT_ADDRESS
"getTotalGms()"
--rpc-url
ARB_RPC_URL
(Optional) In order to interact with the contract on the frontend, you'll need to fund an account that you have in your Ethereum wallet. Transfer to an external account with this command:
bash
export
RECEIVER
=<
receiver
ETH
addres
s

cast
send
--private-key
PRIVATE_KEY RECEIVER
--value
1
ether
--rpc-url
ARB_RPC_URL
export
RECEIVER
=<
receiver
ETH
addres
s

cast
send
--private-key
PRIVATE_KEY RECEIVER
--value
1
ether
--rpc-url
ARB_RPC_URL

87. TIP
88. If you are in a different terminal than the one you set the private key in, you may need to set it again.

Update the frontend

Next, you will need to update a few things before you can interact with the contract on the frontend:

1. Change the contract address ongm-portal/frontend/src/App.tsx
2. to your contract address
3. Match the chain info ongm-portal/frontend/src/main.tsx
4. with the chain config of your L2
5. If you changed the contract, update the ABI ingm-portal/frontend/GmPortal.json
6. fromgm-portal/contracts/out/GmPortal.sol/GmPortal.json
7. . This can be done with:

bash cd HOME cp

dev/gm-portal/contracts/out/GmPortal.sol/GmPortal.json

dev/gm-portal/frontend cd HOME cp

dev/gm-portal/contracts/out/GmPortal.sol/GmPortal.json

dev/gm-portal/frontend

Interact with the frontend

Now, login with your wallet that you funded, and post a GM on your GM portal! [] Edit this page on GitHub] Last updated:
Previous page Deploy a smart contract on Arbitrum rollupNext page Intro to OP Stack integration []