**TOMCAT Eris Environment**

This document is intended to aid in replication of End-To-End Testing.

*Table of Contents*

**Content** **Page**

IP Address 2

Current Chain 2

Nohup ChainKeeper 2

Locations in Eris of Dao,Gatekeeper,Oraclizers,etc. 2

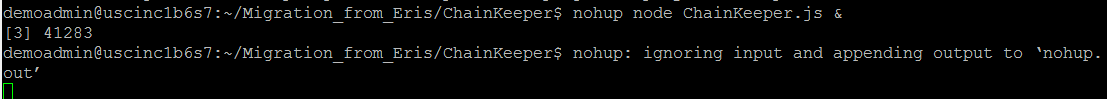
Deployment onto current Chain 2

Wallet Accounts for Testing 3

Adding to the DAO (updated with validator IP) 4

**IP ADDRESS**:VPN 10.100.98.218 (sometimes you will have to use this for cURL requests as the baseline IP *10.101.114.231* doesn’t always work)

This environment uses newchain4. No need to worry about turning the chain off/on. Chain keeper is running as shown below.



Here are locations in Eris.

|  |  |
| --- | --- |
| **Contract Item** | **Location** |
| BigchainOraclizer | /home/demoadmin/Migration\_from\_Eris/BigchainOraclizer |
| ChainKeeper | /home/demoadmin/Migration\_from\_Eris/ChainKeeper |
| Dao | /home/demoadmin/Migration\_from\_Eris/Dao |
| Gatekeeper | /home/demoadmin/Migration\_from\_Eris/gatekeeper |
| VerifyOraclizerEthereum | /home/demoadmin/Migration\_from\_Eris/VerifyOraclizerEthereum |

Deployment on Chain:

eris pkgs do --chain newchain4 --address BB6724FE1A9A2A6D4F95B9969EAC59FDA94838AD --compiler <https://compilers.monax.io:10114>

Accounts.json

You can find it in: */home/demoadmin/.eris/chains/newchain4*

Testing

We use these three accounts. They have been added to the Dao.

**Account 1**

String of public key is: 0373ecbb94edf2f4f6c09f617725e7e2d2b12b3bccccfe9674c527c83f50c89055

String of private key is: 66cdd508f950d08e02d8448c55a03c14e08e9a3447c4361105e14ad32ec3286a

Message hash string: 7624778dedc75f8b322b9fa1632a610d40b85e106c7d9bf0e743a9ce291b9c6f

sig string is: c21bb2c81d0ca548a59b61a6d9fdc3871b867a951963e059a1862f829e4f835a5a0584405f64288ca6315d871efc4d7c39a625c5eb01108f2ddb49e37a711515

**Account 2**

String of public key is: 03683536757fdb821c10810b51caa51a84fc1dfab5c17edbf5246f9713ffe31adf

String of private key is: f4822c62077945009cd816e479eef4ac371613d2d44164fc7ecaa93f180f2eb4

Message hash string: 7624778dedc75f8b322b9fa1632a610d40b85e106c7d9bf0e743a9ce291b9c6f

sig string is: 9a83145f3533169aea22d253ee30533768f112b0818d096716f7d6ef664d95ea4c8e2e548c57d0142b82ce85403d307cb817a9d9e7b64e088d8c4b08d551eab4

**Account 3**

String of public key is: 03a066efbb37f5fabfab05bf4a65e0dc376d0e3fb1c3d930d7f5ec6da3ac5bc237

String of private key is: 2eff28029dced258c4aca03415fd3ef201ec118e57241b758e82454a6bc75cd4

Message hash string: 7624778dedc75f8b322b9fa1632a610d40b85e106c7d9bf0e743a9ce291b9c6f

sig string is: 9337df4f95d3460dea440315c17666ebef89db10d19760ee3277c4fd1fafed7d1e3b1a8b7822bb8374de88b3b50305d7d94d14377b8b0945f1180a8db45bd0b4

**Updated Adding Validators to the Dao** (with validatorIP) –change these fields if desired to actual IP

Start The Application

*node Dao.js*

Set the host variable

*host=localhost*

Add Validators via cURL

curl --request POST --data '{"validatorAddr": "0373ecbb94edf2f4f6c09f617725e7e2d2b12b3bccccfe9674c527c83f50c89055", "validatorIP": "forTesting1"}' $host:8002/addValidator

curl --request POST --data '{"validatorAddr": "03683536757fdb821c10810b51caa51a84fc1dfab5c17edbf5246f9713ffe31adf", "validatorIP": "forTesting2"}' $host:8002/addValidator

curl --request POST --data '{"validatorAddr": "03a066efbb37f5fabfab05bf4a65e0dc376d0e3fb1c3d930d7f5ec6da3ac5bc237","validatorIP": "forTesting3"}' $host:8002/addValidator

If you want to see the list:

curl --request GET $host:8002/getList

