Adding a DFK Chain Mainnet Node

Important Variables

DFK SubnetID: Vn3aX6hNRstj5VHHm63TCgPNaeGnRSqCYXQqemSqDd2TQH4qJ
DFK BlockchainID: q2aTwKuyzgs8pynF7UXBZCU7DejbZbZ6EUyHr3JQzYgwNPUPi
DFK VMID: mDV3QWRXfwgKUWb9sggkv4vQxAQR4y2CyKrt5pLZ5SzQ7EHBv
DFK FeeContractAddress: TBD

Create a AvalancheGo Node on the mainnet

In order to run an RPC node on the DFK Chain you need to be running a node on the Avalanche Primary network. You can follow instructions found here: https://docs.avax.network/build/tutorials/nodes-and-staking/run-avalanche-node#run-an-avalanche-node-and-send-funds

Note: You must build avalanchego from source in order to add a subnet-evm in a later step. Do not use the prebuilt binary or other options

Create the DFK Chain Subnet EVM

On your server you must clone the subnet-evm repository from defikingdoms: https://github.com/DefiKingdoms/subnet-evm

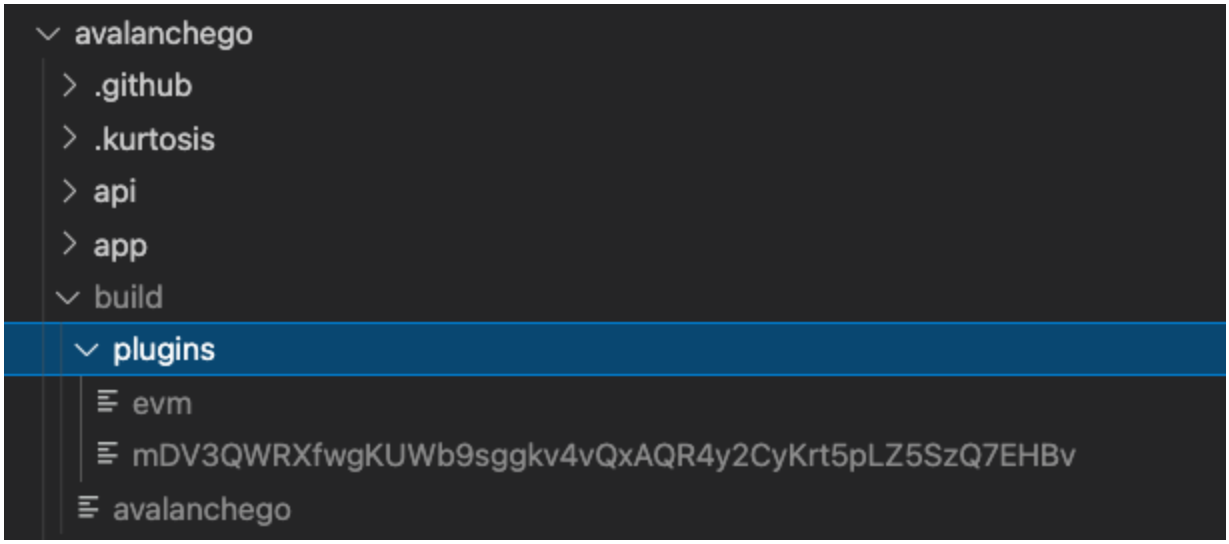
This evm is a fork of the Ava Labs Subnet EVM and is mostly identical.

Next run the build script to the evm and output it into the avalanchego build/plugins directory.

```
cd subnet-evm
./scripts/build.sh ../avalanchego/build/plugins/mDV3QWRXfwgKUWb9sggkv4vQxAQR4y2CyKrt5pLZ5SzQ7EHBv
cd ..
```

Note: the id “mDV3QWRXfwgKUWb9sggkv4vQxAQR4y2CyKrt5pLZ5SzQ7EHBv” is our custom VMID created with the subnet-cli by AVA Labs. This must not change.

The avalanchego build folder should look like this:



Create Config.file

Create a config.json file and put the following in it. This file can be anywhere but we will assume its in your home directory when we run avalanchego.

```
{
 "network-id":"mainnet",
 "http-host":"",
 "public-ip": "[YOUR IP ADDRESS]",
 "health-check-frequency":"2s",
 "log-display-level":"INFO",
 "log-level":"INFO",
 "chain-config-dir":"/home/ec2-user/chain-config-dir",
 "whitelisted-subnets": "Vn3aX6hNRstj5VHHm63TCgPNaeGnRSqCYXQqemSqDd2TQH4qJ"
}
```

Note: The above config assumes you are on amazon linux 2 and have a /home/ec2-user/ directory. Change this if you are using a different OS change the chain config dir

Create the chain config folder and file

Now we need to tell avalanchego what parameters we want for the new chains RPC endpoint. To do so we will make a chain config folder at /home/ec2-user/chain-config-dir. This can be anywhere you want but the previous config file we made is pointing here.

In that directory you need to make another folder with the DFK Blockchain ID: q2aTwKuyzgs8pynF7UXBZCU7DejbZbZ6EUyHr3JQzYgwNPUPi

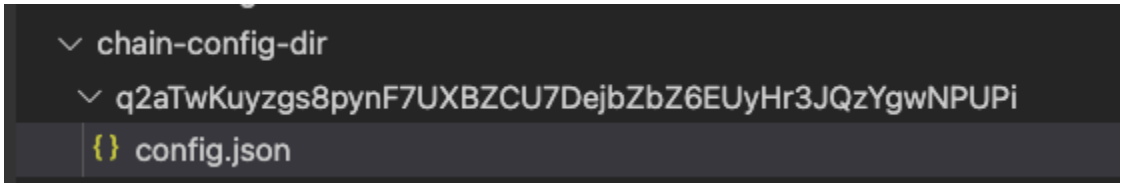
In that folder make a config.json file and at the very minimum you need to add this configuration to it:

```
{
        "feeRecipient": "TBD"
}
```

Note: If you are not going to validate the network this feeRecipient really does nothing but it is best practice and we ask partners to configure it to make transitioning any RPC to a validator an easier process to validate.

You can add a bunch of different parameters to the RPC with this file. All options are found here: https://docs.avax.network/build/references/avalanchego-config-flags/#c-chain-configs

Your folder structure should look like this:



Run AvalancheGo

Now that you have the node configured you can run avalanche go. If your node is not bootstrapped to the network it may take a few days to bootstrap.

```
./avalanchego/build/avalanchego --config-file=/home/ec2-user/config.json
```

Use the RPC

The RPC endpoint is now found at the following address: http://localhost:9650/ext/bc/q2aTwKuyzgs8pynF7UXBZCU7DejbZbZ6EUyHr3JQzYgwNPUPi/rpc

(Optional) Set up reverse proxy

Rather than having to use that long path every time to access the rpc, we set up a nginx reverse proxy on the server. The nginx configuration could look something like this to add SSL and shorten the path.

```
    server {

        server_name rpc.dfkchain.com;

        location / {
            proxy_pass
http://localhost:9650/ext/bc/q2aTwKuyzgs8pynF7UXBZCU7DejbZbZ6EUyHr3JQzYgwNPUPi/rpc;
        }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/rpc.dfkchain.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/rpc.dfkchain.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}


    server {
    if ($host = rpc.dfkchain.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        listen  80 default_server;
        listen  [::]:80 default_server;

        server_name rpc.dfkchain.com;
    return 404; # managed by Certbot
}
```

(Optional) Become a Validator

After setting up an RPC node the user can request to become a DFK Chain validator and start earning. Visit this portal to start the application: https://validator.dfkchain.com and click on apply now.

Once approved next steps will be provided to you.