IC-Tools

Date	
Slides	
Courses	IDP
Lecture video link	

For one of the tools for the testnet, which is `nns_dev_testnet.sh` , in its doc (testnet/tools/nns-tools/README.md), it is mentioned that

```
Needs to be run on zh1-spm22.zh1.dfinity.network. (Ideally, we'd
be able to run this locally; implementing that is feasible, but
we haven't done it yet.)
```

The following is the analysis on that.

What's this tool used for

This script creates a testnet with a mainnet state using a stable shared identity and modifies it in a few ways for development purposes.

1. Adds an application subnet.
2. Sets CMC default subnet list to that application subnet.
3. Creates a cycles wallet for our shared principal on the application subnet.
4. Configures SNS-W to create SNS's on application subnet, and to respond to our principal's wallet.
5. Uploads the latest SNS Wasms into SNS-W canister

`nns_dev_testnet.sh` analysis

Steps:

I**nitial setup and argument checking**: The script starts by checking the number of arguments and verifies that `dfx`, is installed. Also creates a custom identity

1. First, they create the NNS State deployment (`nns_state_deployment.sh`)using the special identity.

`nns_state_deployment` analysis

- it is an internal tool. To use this script, the public key should be present on pyr07 backup pod.
- Steps
    1. Downloads some tools `ic-replay ic-recovery ic-admin sandbox_launcher canister_sandbox` from `https://download.dfinity.systems/ic/$GIT_HASH/release/$DOWNLOAD_NAME.gz`
    2. uses the `icos_deploy.sh` script to deploy the icos on the testnet.
    3. Fetches the NNS state from the backup pod. (located here:dev@zh1-pyr07.zh1.dfinity.network)
        a. It then ssh's into the nns node and copies the ic.json file.
    4. It provides 1 billion neuron so it can pass all the proposals instantly. It is being done through `ic_replay` tool that they have.
    5. Then provides the principal (i.e. the person deploying the testnet) with the one million neurons.
    6. Recover the NNS subnet to the first unassigned node. this is done by their recovery tool `ic_recovery` tool. (Not sure why they have to reassign the subnet, but I think this is based on how the nns functions when transferring neurons)
    7. Then they move the remaining unassigned nodes to new subnet so that it can be controlled by the new subnet.
    8. Then they test if the new nns works by creating a proposal with the `ic_admin` tool.
- From this script result output, they obtain the list of Subnets so that it can re-add them to the Registry topography

2. Then a new subnet is created from the unassigned node. In order to create a new subnet, A new proposal has to be passed to the nns, using the `ic_admin` tool.

```
$IC_ADMIN -s "$PEM" --nns-url "$NNS_URL" \
        propose-to-create-subnet \
        --summary "Creating a subnet" \
        --proposer "$NEURON_ID" \
        --subnet-type "$SUBNET_TYPE" \
        --replica-version-id "$REPLICA_VERSION" \
        ${UNASSIGNED_NODE_IDS[@]}
```

3. Sets CMC default subnets to one on testnet

```
$IC_ADMIN -s "$PEM" --nns-url "$NNS_URL" \
        propose-to-set-authorized-subnetworks \
        --proposer "$NEURON_ID" \
        --summary "Setting authorized subnetworks" \
        --subnets "$SUBNET_ID
```

4. Creates the cycles wallet for the test user created. It is completely done using `dfx` .

```
$(dfx ledger --network "$NNS_URL" create-canister "$PRINCIPAL" --amount 10000 \
        | grep "Canister created" \
        | sed 's/.*"\(.*\)"/\1/') # get the CanisterId in quotes

# Request to install must be made to here...
dfx identity --network "$SUBNET_URL" deploy-wallet "$WALLET_CANISTER"
dfx identity --network "$SUBNET_URL" set-wallet "$WALLET_CANISTER"
```

5. Configure SNS-WASMs. (A **Service Nervous System**, or SNS, enables a service to run under the control of a decentralized community, allowing the dapp or service to perform community-based fundraising and gain the censorship resistance needed for advanced tokenization.)
6. Sets up the boundary nodes
7. Sets up a new XRC mock. (Exchange rate canister)

   The script uses the Bazel build system to build the mock XRC canister from source code located at `rs/rosetta-api/tvl/xrc_mock:xrc_mock_canister`. Then creates a new canister on the ledger, and then install the canister in the subnet.

Why it has to be running on that particular network?