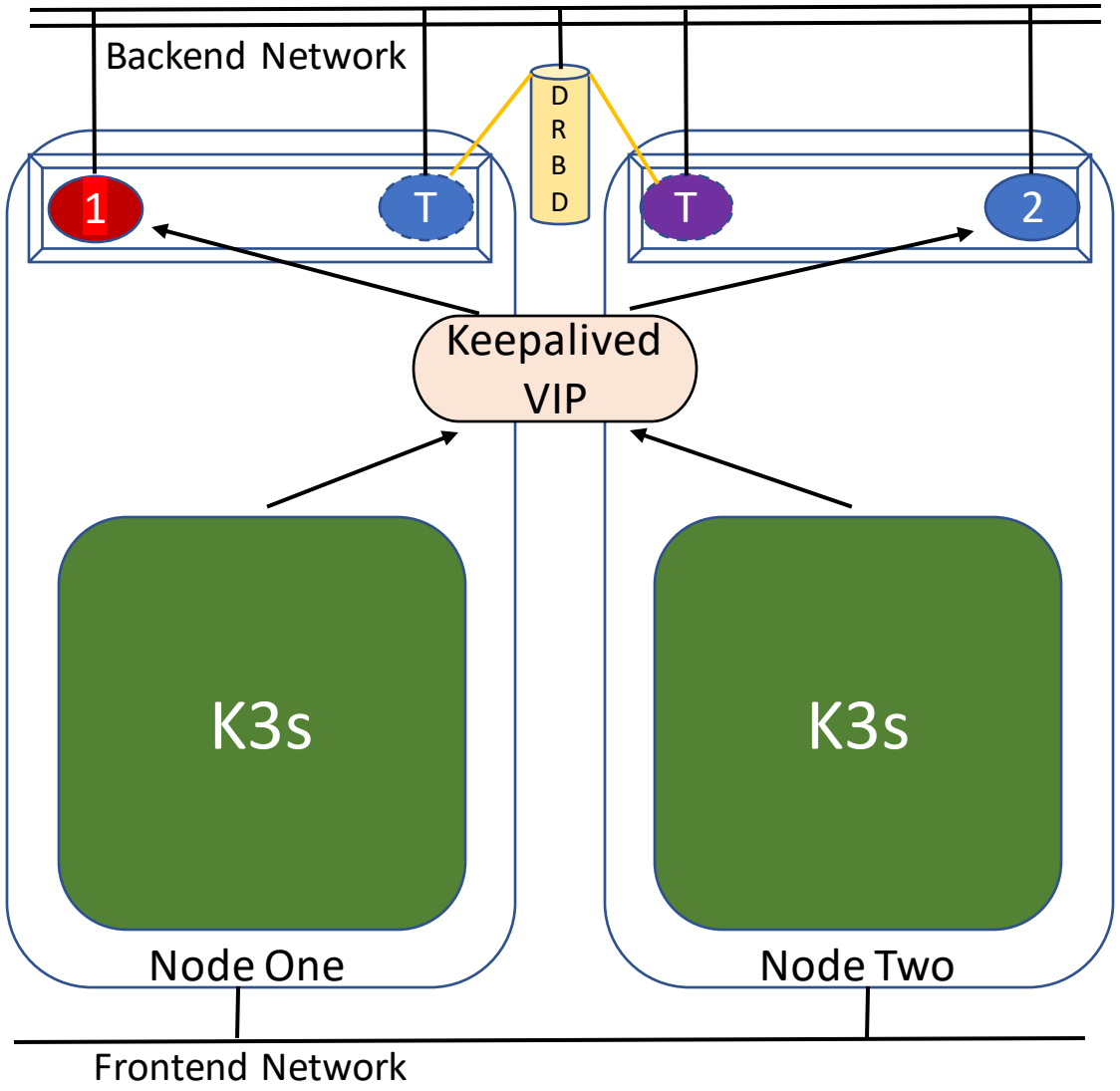
etcd for a two-node HA K3s cluster



- Priority Voting Instance has the shortest election-timeout setting so it will always win leader elections when Node One is online
  - Will likely need to run `etcdctl move-leader` as part of this instance's startup
  - Setting Node One's instance as the default leader supports more consistent behavior and quicker recovery in most scenarios
- Transient instance exists on DRBD. It runs on Node One while normal communications are observed between etcd instances 1 and 2
- If Node One loses communication to Node Two, either split-brain exists or Node Two has failed:
  - Split-brain or Node Two has failed = DO NOTHING
    - Node One has 2/3 quorum so services continue uninterrupted
- If Node Two loses communication to Node One, it needs to establish split-brain or node failure:
  - Split-brain = DO NOTHING, Node One has 2/3 quorum
  - Node One has failed = Start Transient Instance
    - Node Two now has 2/3 quorum
- If the external "Witness Service" does not respond when communication is lost, etcd will be unavailable if Node One is down
  - This is considered a failure of more than one cluster resource and is not covered under the standard definition of High Availability

2 = etcd Regular Instance 1 = etcd Priority Instance T = etcd Transient Instance = Podman

Using web server as external means to verify split-brain 1/2

- Each node monitors the arrival of packets (most will be etcd heartbeat/data packets) on the backend network interface
- Upon loss of communication with the sibling node:
  - A script will be invoked on both nodes that will discern between node failure and split-brain:
    - Each node wait assigned amount of time before continuing:
      - Node One waits zero seconds
      - Node Two waits a much longer timeout (e.g. 15 seconds)
    - Node One writes an alive file to the webserver:
      - `curl -X PUT MY-K3S-CLUSTER-FAILURE-FENCING/my-k3s-cluster-node-1-is-alive http://172.16.240.2/MY-K3S-CLUSTER-FAILURE-FENCING/my-k3s-cluster-node-1-is-alive`
      - NOTE: Need to explore the need to have Node One enter a failed state if the write exits non-zero
    - Node Two checks to see if the alive file exists on the webserver:
      - `curl -X GET http://172.16.240.2/MY-K3S-CLUSTER-FAILURE-FENCING/my-k3s-cluster-node-1-is-alive 2>&1 /dev/null | grep "404 Not Found"`
      - Exit 1 (from the grep command) = file exists, exit 0 = file does not exist

Using web server as external means to verify split-brain 2/2

- If Node Two discovers that Node One's alive file exists:
  - Split brain exists
  - Node Two will do nothing since Node One is alive and has 2/3 quorum
  - etcd will catch up Instance 2 after communication is restored
- If Node Two discovers that Node One's alive file DOES NOT EXIST:
  - Node One has failed
  - Node Two starts its Transient Instance
  - Node Two now has 2/3 quorum, elects a leader and continues
- Node Two never stops monitoring for new packets from Node One as well as the existence of Node One's alive file
  - If either packets from Node One, or Node One's alive file are detected, Node Two immediately stops its Transient Instance
  - NOTE: This is an area of important timing. When Node One boots up, it should not start its Transient Instance quicker than the interval of Node Two checking for the alive file plus the time to normally shut down its Transient Instance