HA dCache operations

NeIC NT1 Manager
Mattias Wadenstein
<maswan@ndgf.org>

2025-05-20
dCache workshop
Lyon, France

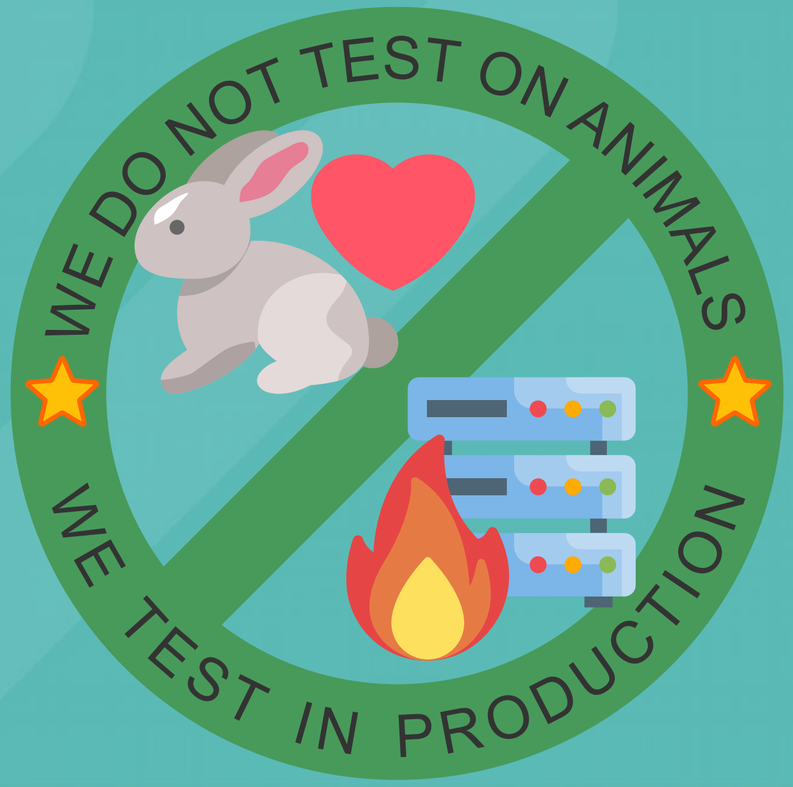
NORDIC E-INFRASTRUCTURE COLLABORATION

Overview

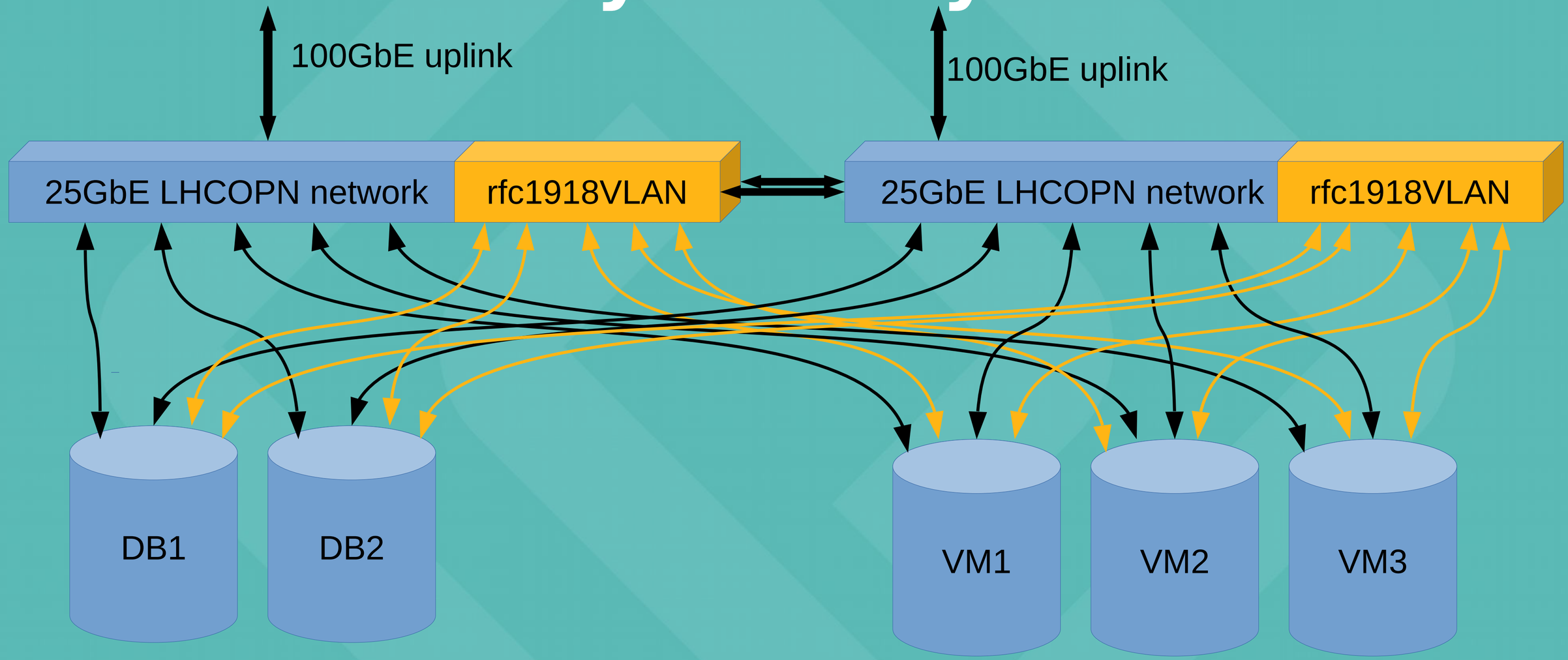
- Hardware
- Postgresql
- Virtual machines
- Headnodes
- haproxy

Goals

- Any single piece of hardware failing is either:
  - Automatically failed over from and handled by redundancy
  - Fixable remotely by the operator on call being woken up by 24/7 alarm
- Update to new security patches
  - Promptly, including reboots, no downtime
- Testable in preprod
  - Preprod running the same ansible as prod



Physical layer



5 x Dell r640 with 2x5222 (2x4 core 3.8GHz) and 192GB RAM, 2TB SSD
4x25GbE for redundant (LACP) internal and external network
Network 2xS5212F-ON (12x25+3x100) with multi-chassi link aggregation

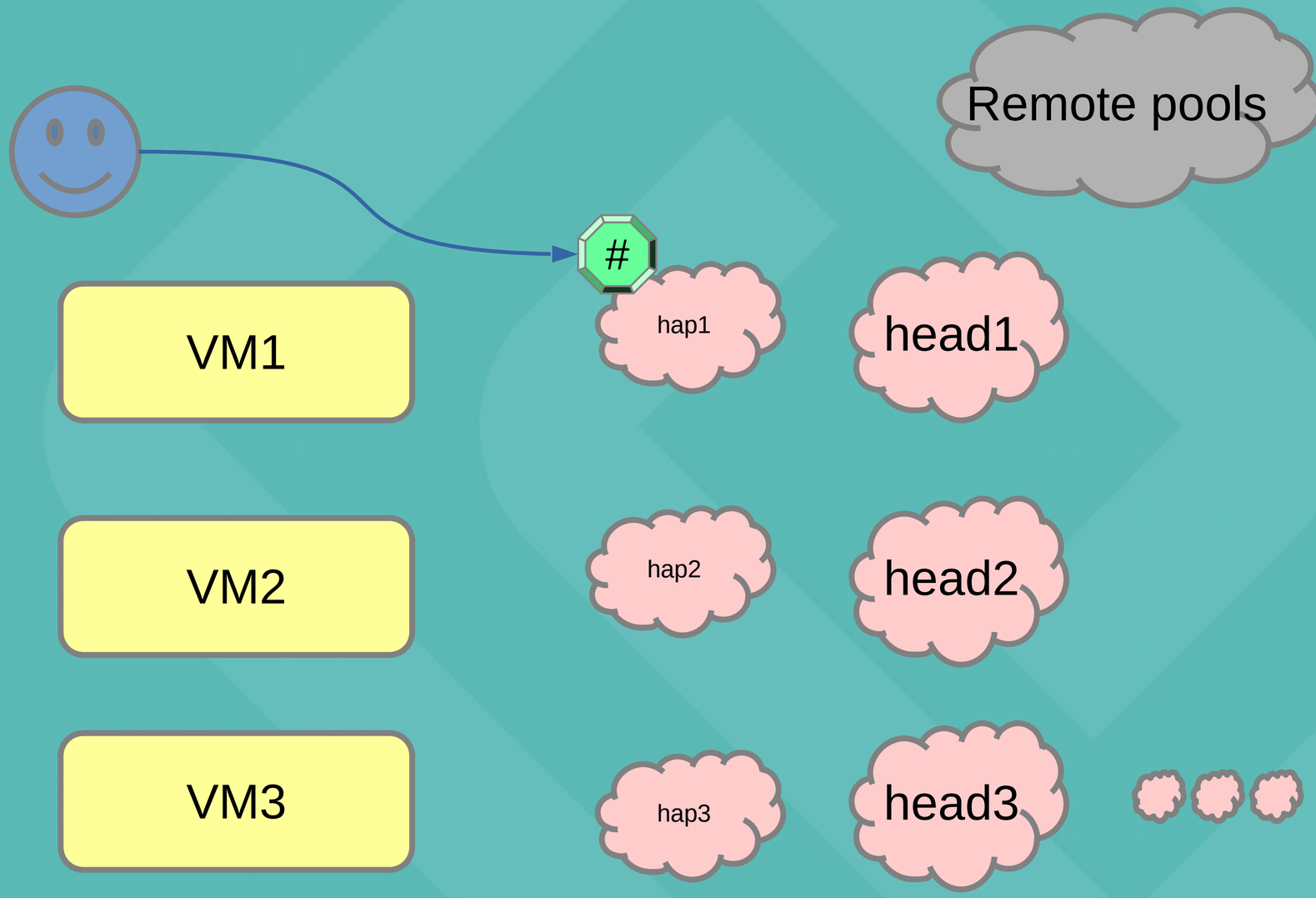
Postgresql

- Two node primary/standby cluster managed by repmgr
- Procedure:
  - If posgres update: Unhold the postgres packages
  - Upgrade and reboot the standby
  - Failover the primary
    - If this fails, probably due to incompatible repmgr upgrades, do this the complicated way
  - Upgrade and reboot the new standby
  - Failover the primary again
    - We have some non-critical applications running against the same DB that don’t believe in repmgr
  - apt hold the postgres packages again

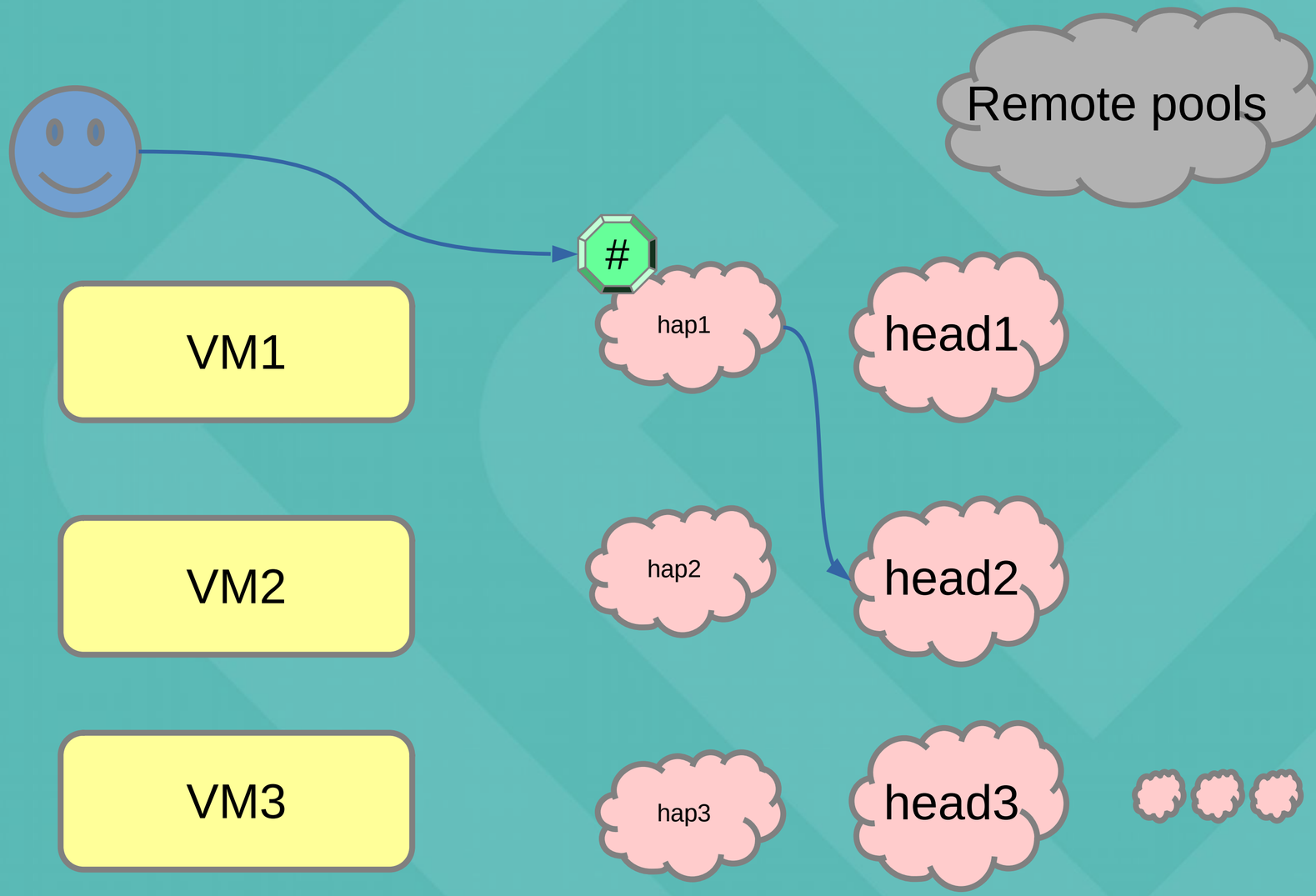
haproxy

- In front of dCache we have haproxy to direct user connections to our headnodes
- Load balancing between responding doors
  - Resiliency and performance
- One of these has a floating IP that’s dav.ndgf.org etc
- On upgrade of the haproxy VMs we’ll failover the floating IP
  - but that interrupts all active transfers, so we don’t upgrade them as frequently as we’d like
  - That said, clients should just reconnect, so it should be fine

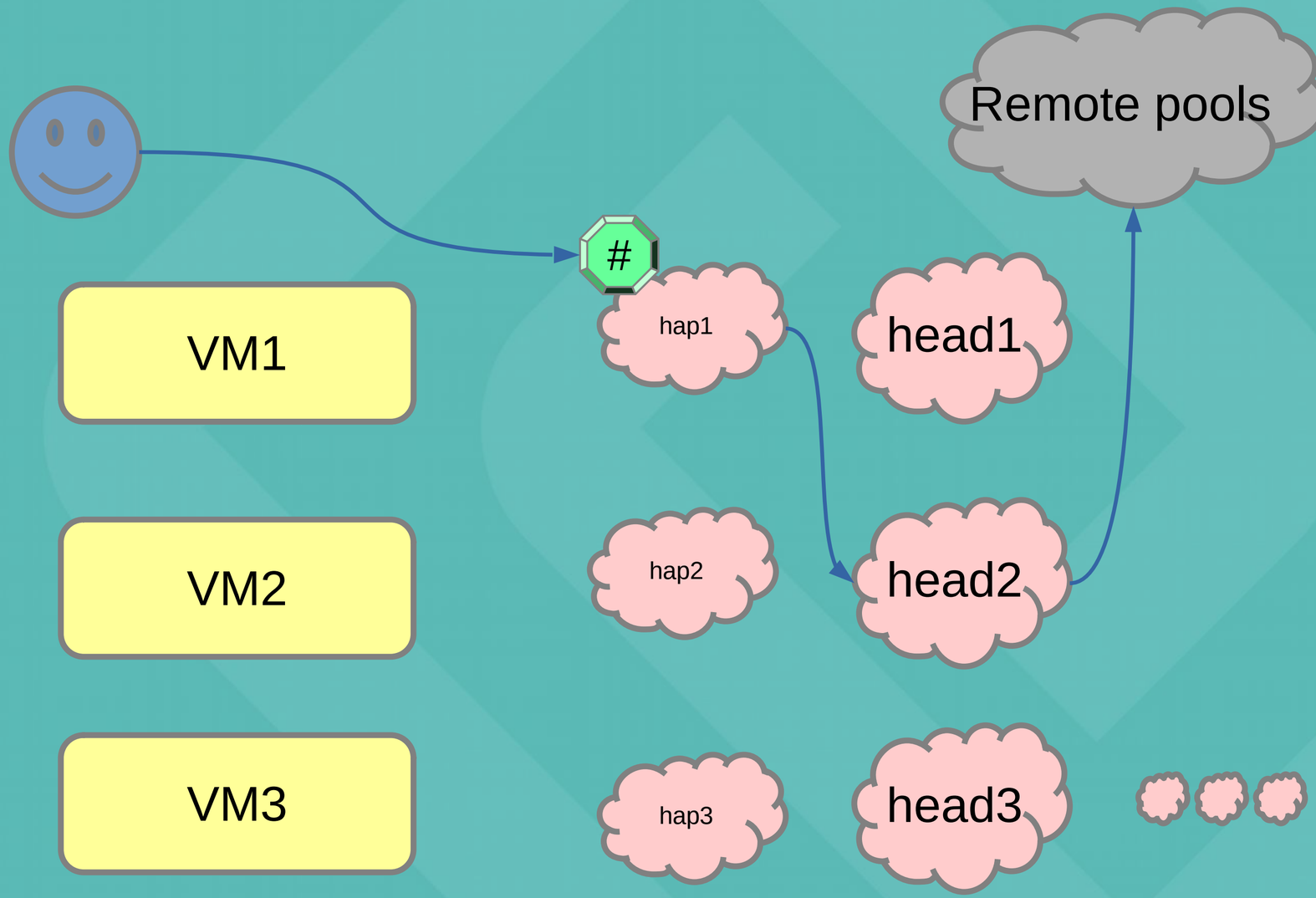
User request to floating IP



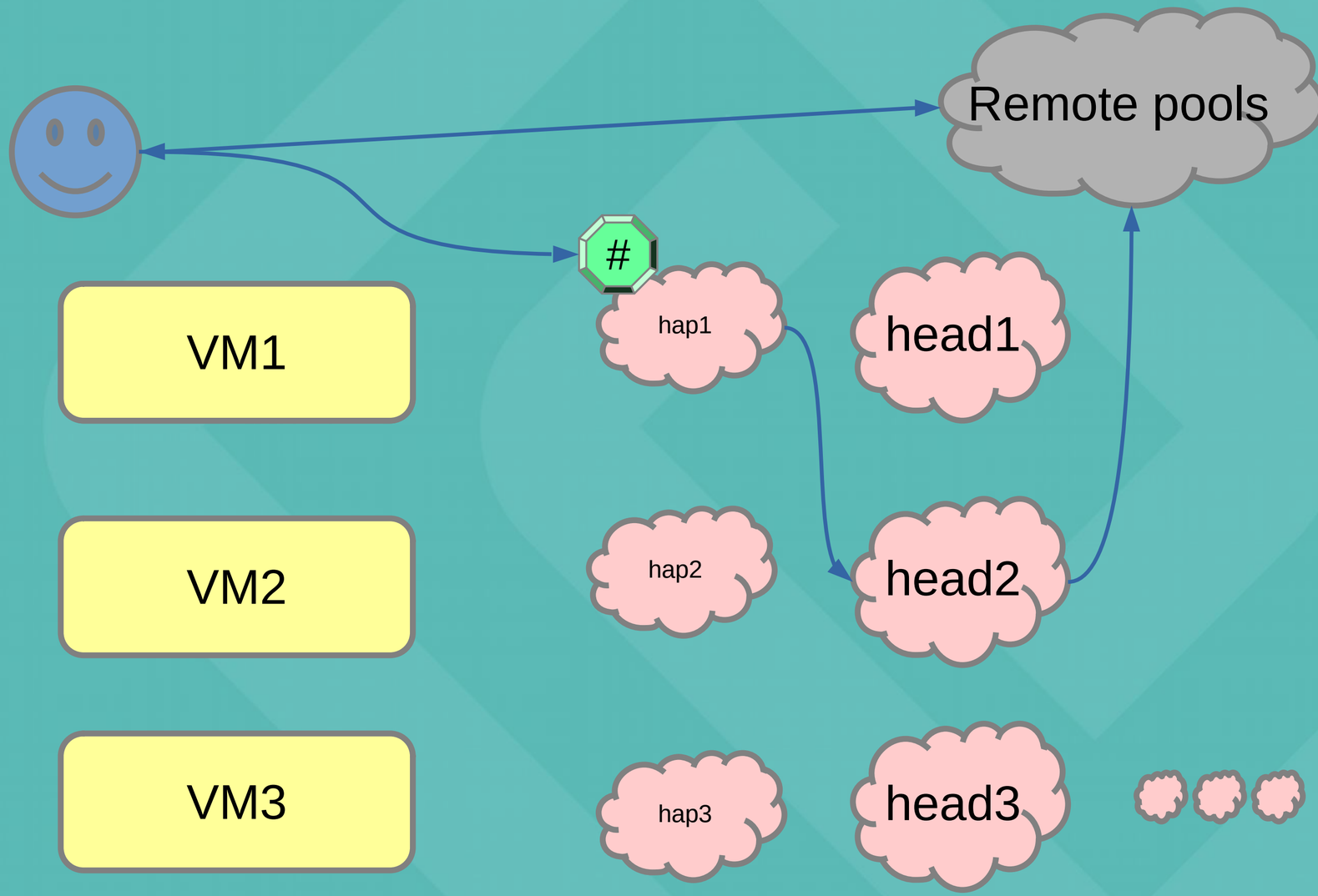
Forwarded to an active headnode



Redirect to pool



Data transfer

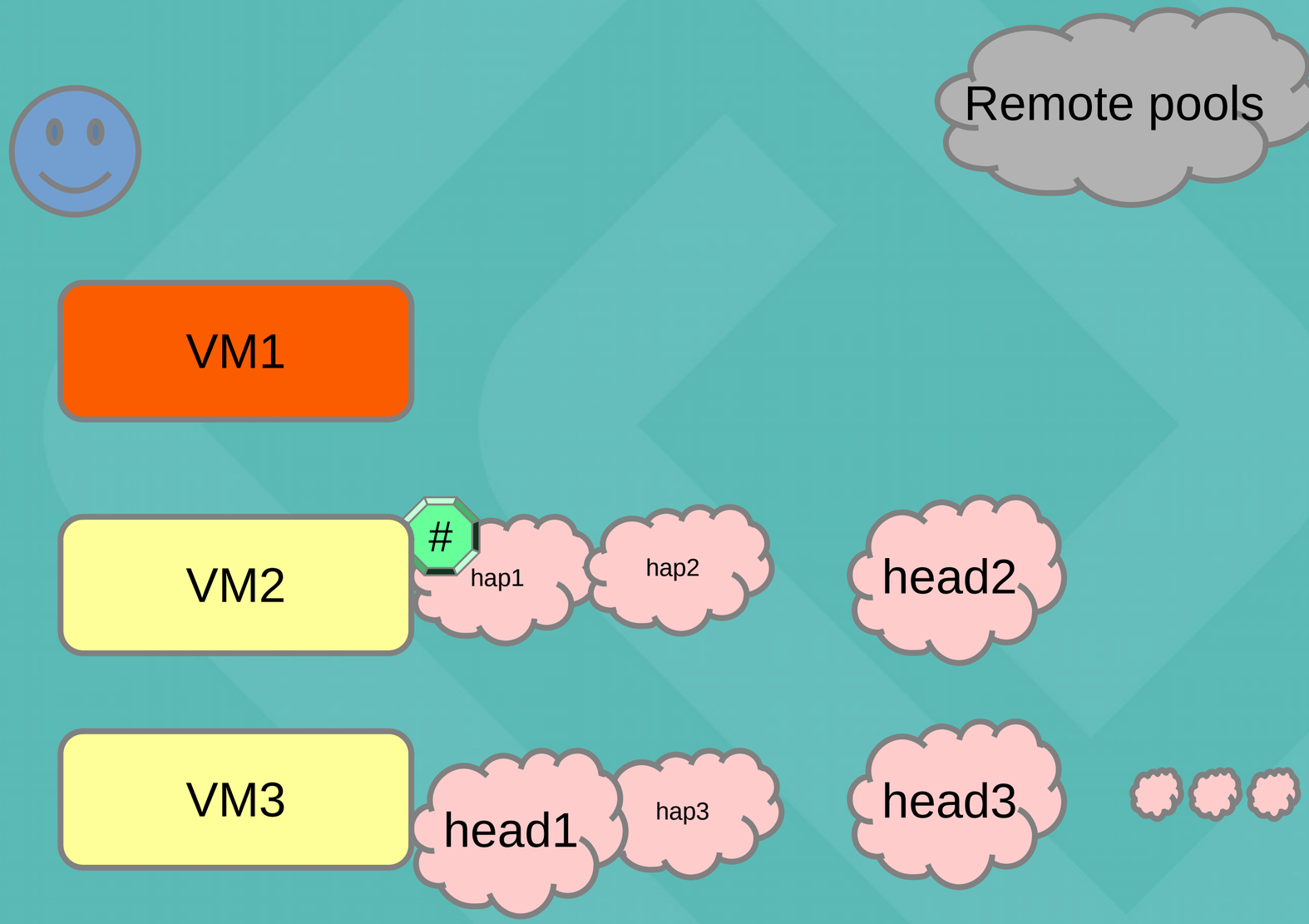


VM physical nodes

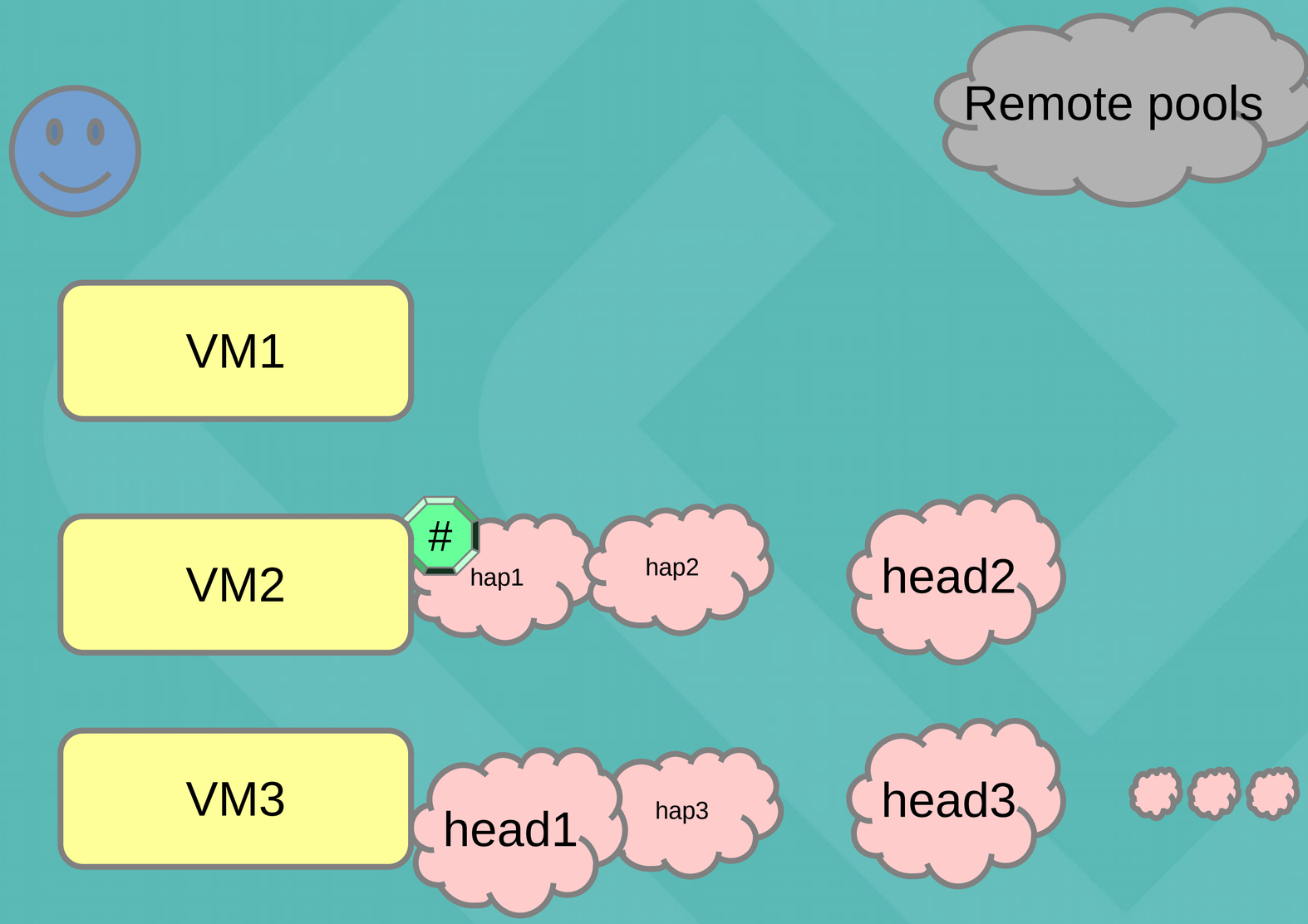
- Foreach node:
  - Live migrating all the VMs to their secondaries
  - Upgrade and reboot
  - The one with the floating admin IP gets a master-failover as well
- In case of major interventions:
  - Set the node to drain
  - Rebalance the cluster so that the disk images also get migrated out
  - Intervene
  - Undrain
- End with a cluster rebalance

Start with one

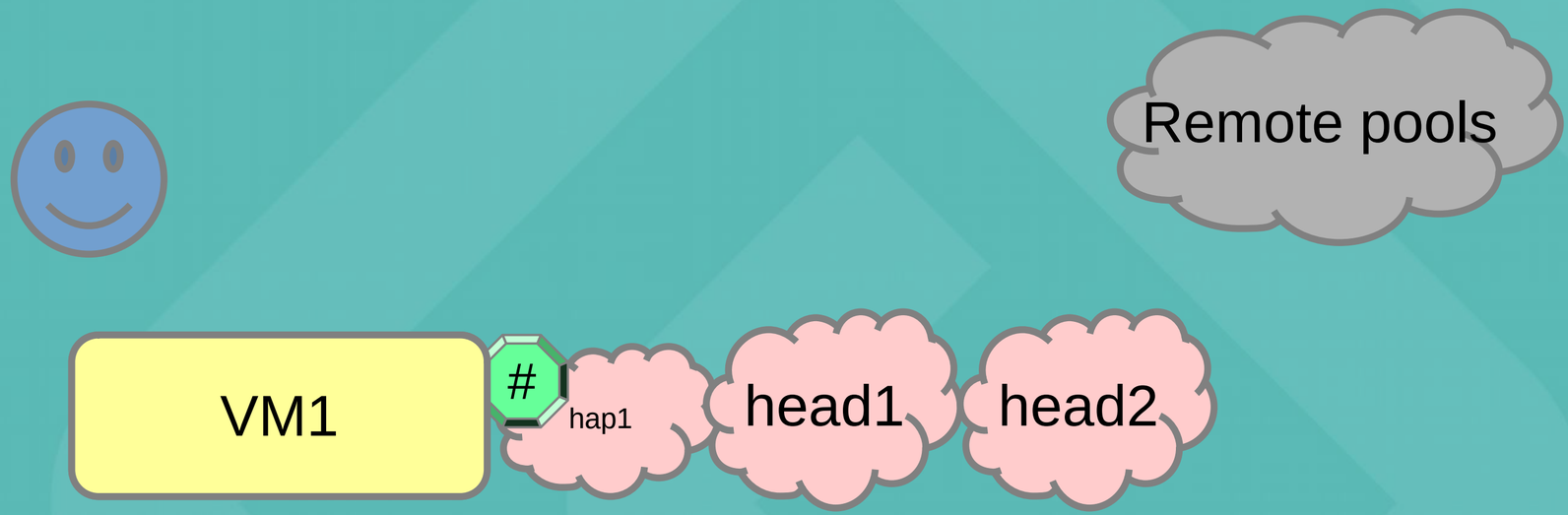
Evacuate guests



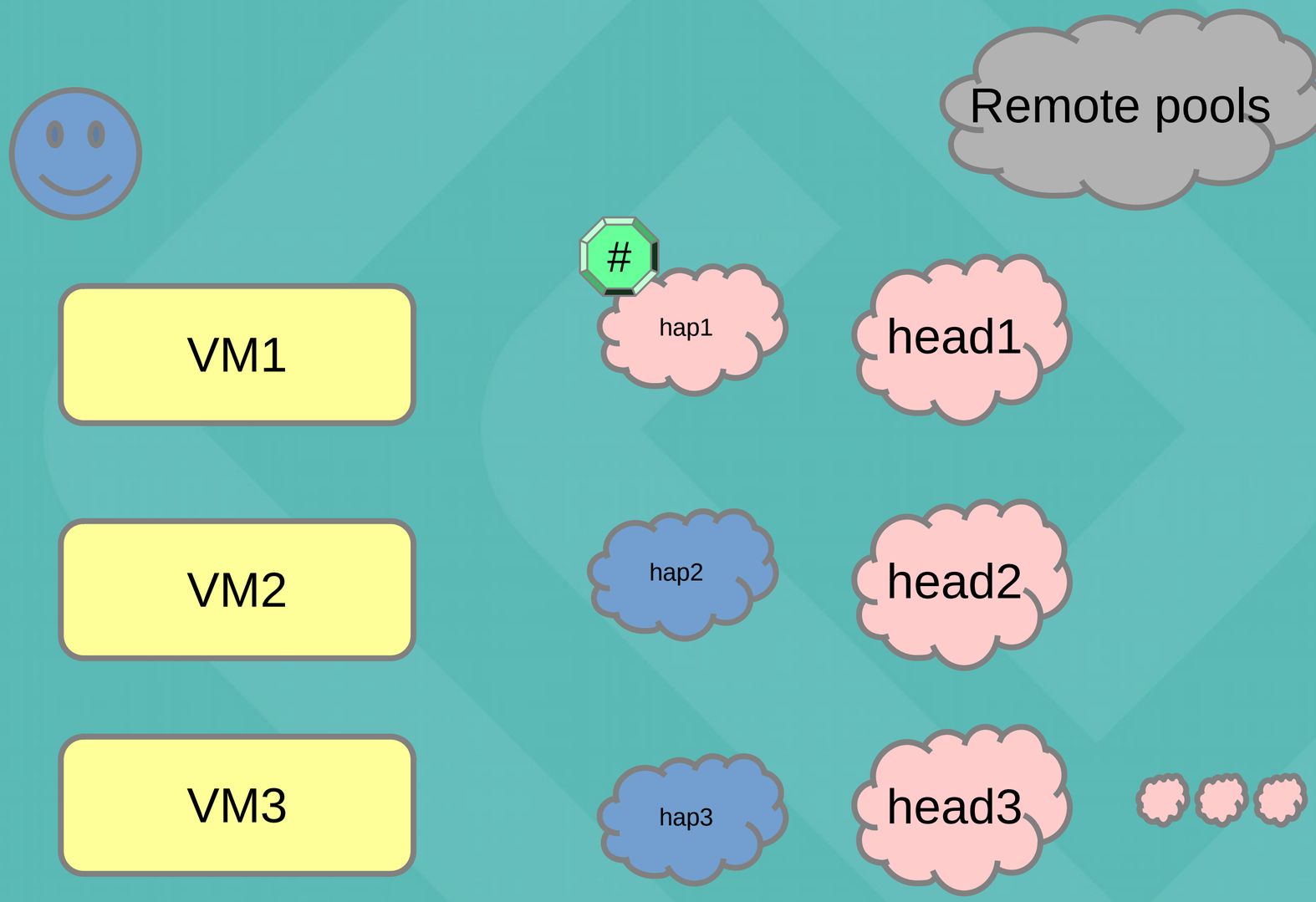
Reboot



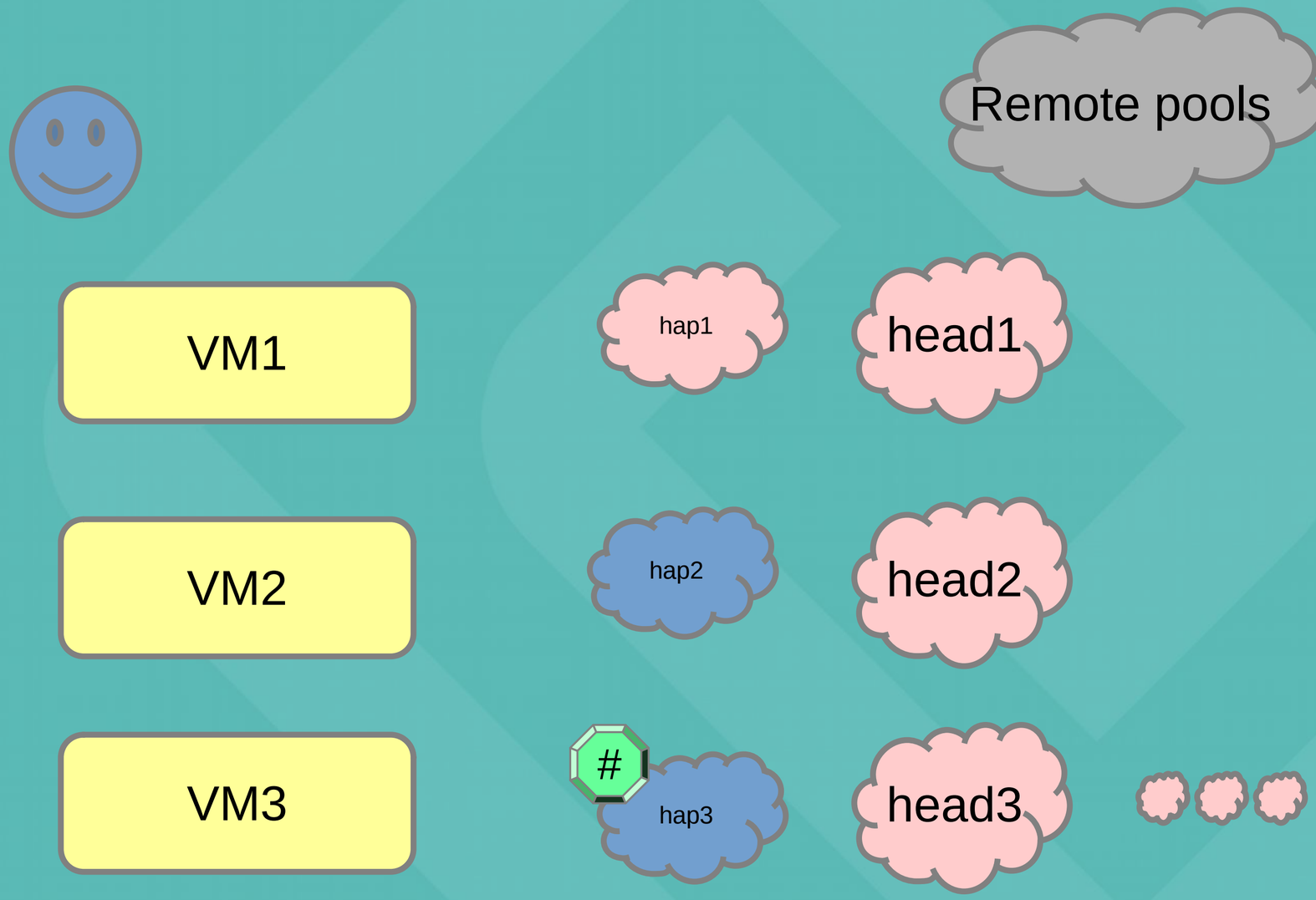
Repeat



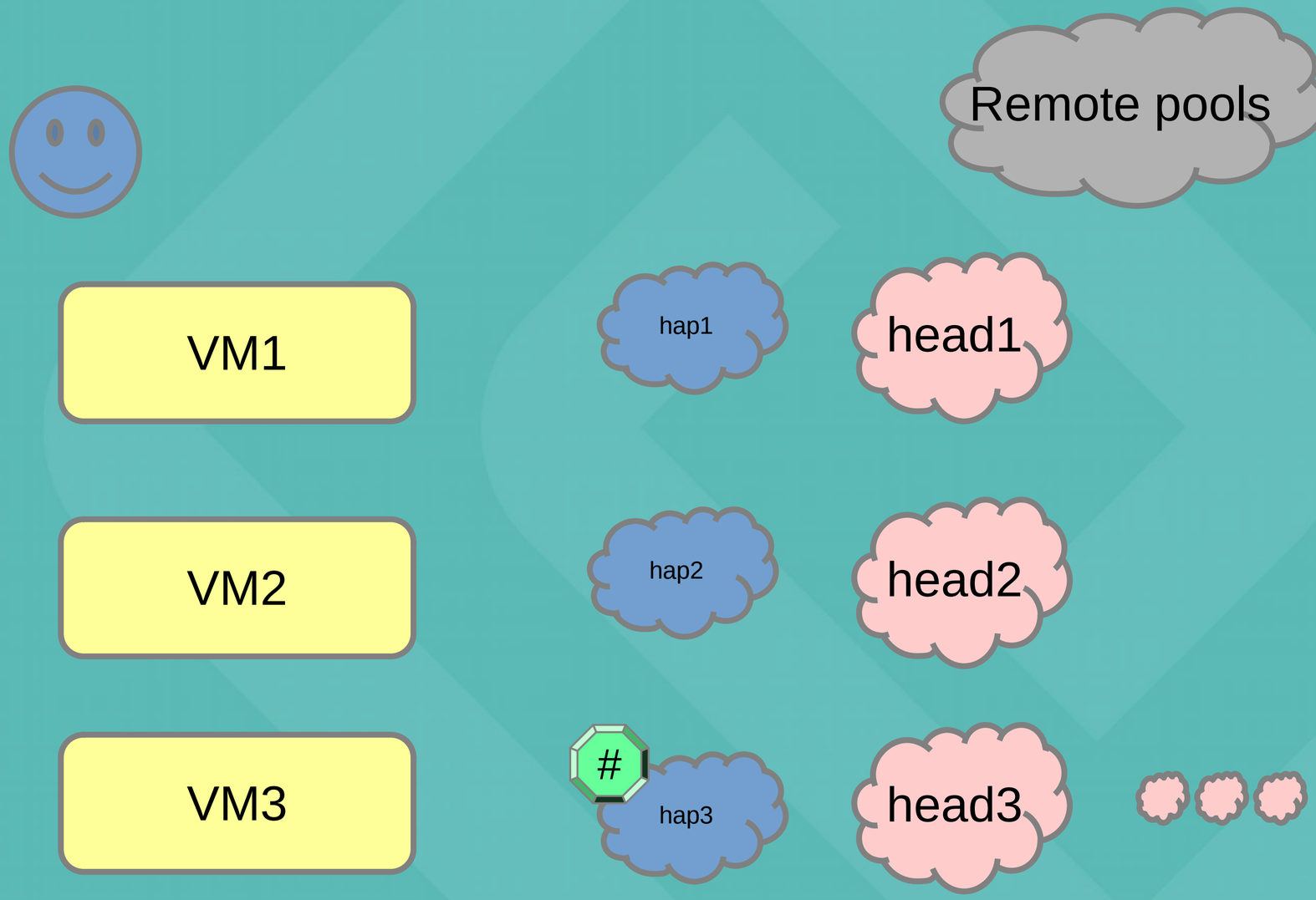
HAproxy: Update secondaries



Failover floating IP



Upgrade last node



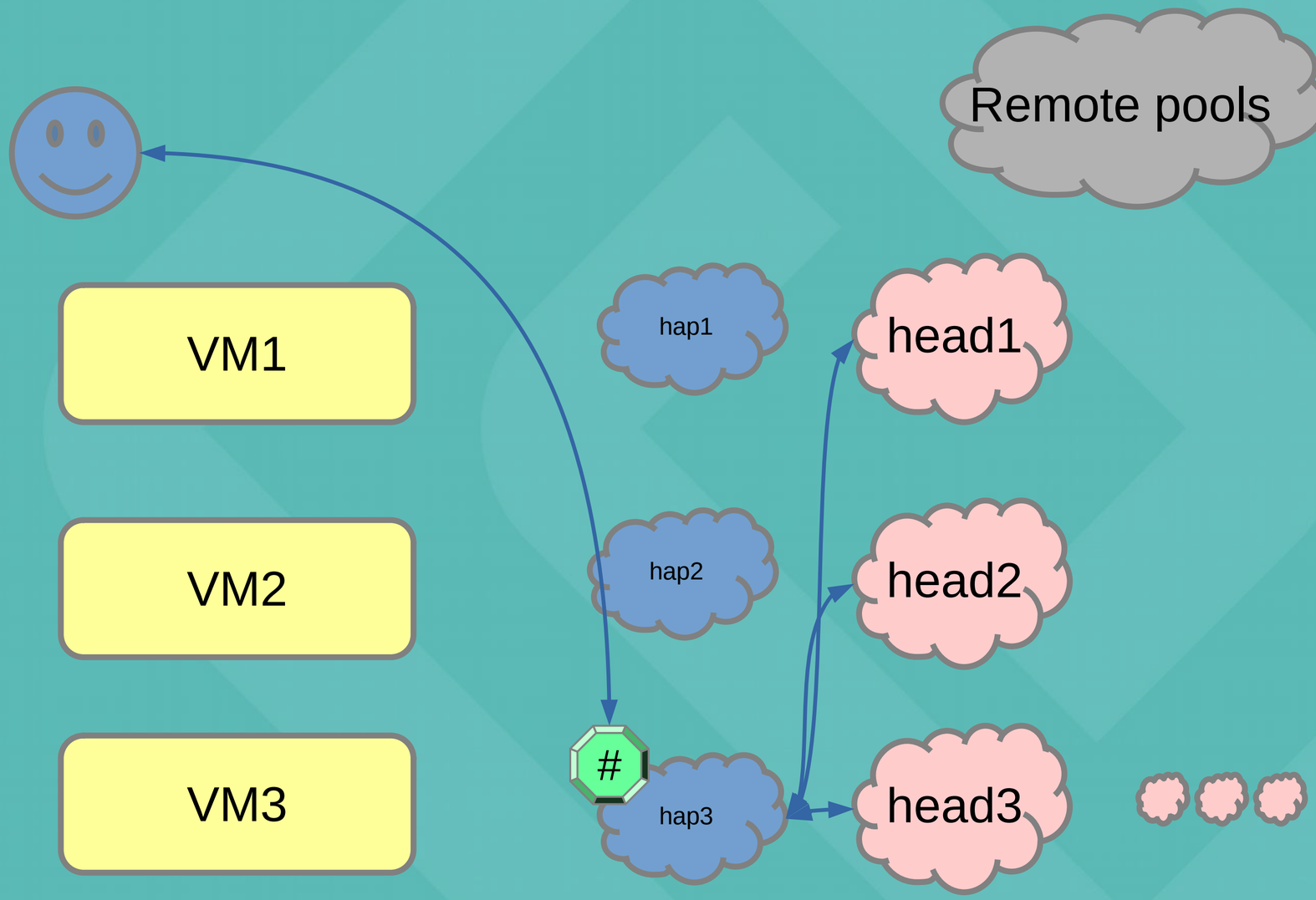
Headnode upgrade

- Run drain playbook on one headnode
  - Sets haproxy status to drain
  - Disables in loginbroker
  - Repoints a legacy domain name
    - We should try dropping it again, was used for protocols that couldn’t handle proxy
- Wait forever (12-24h)
- Poll haproxy stats to see no active connections
- Upgrade and reboot
- Undrain

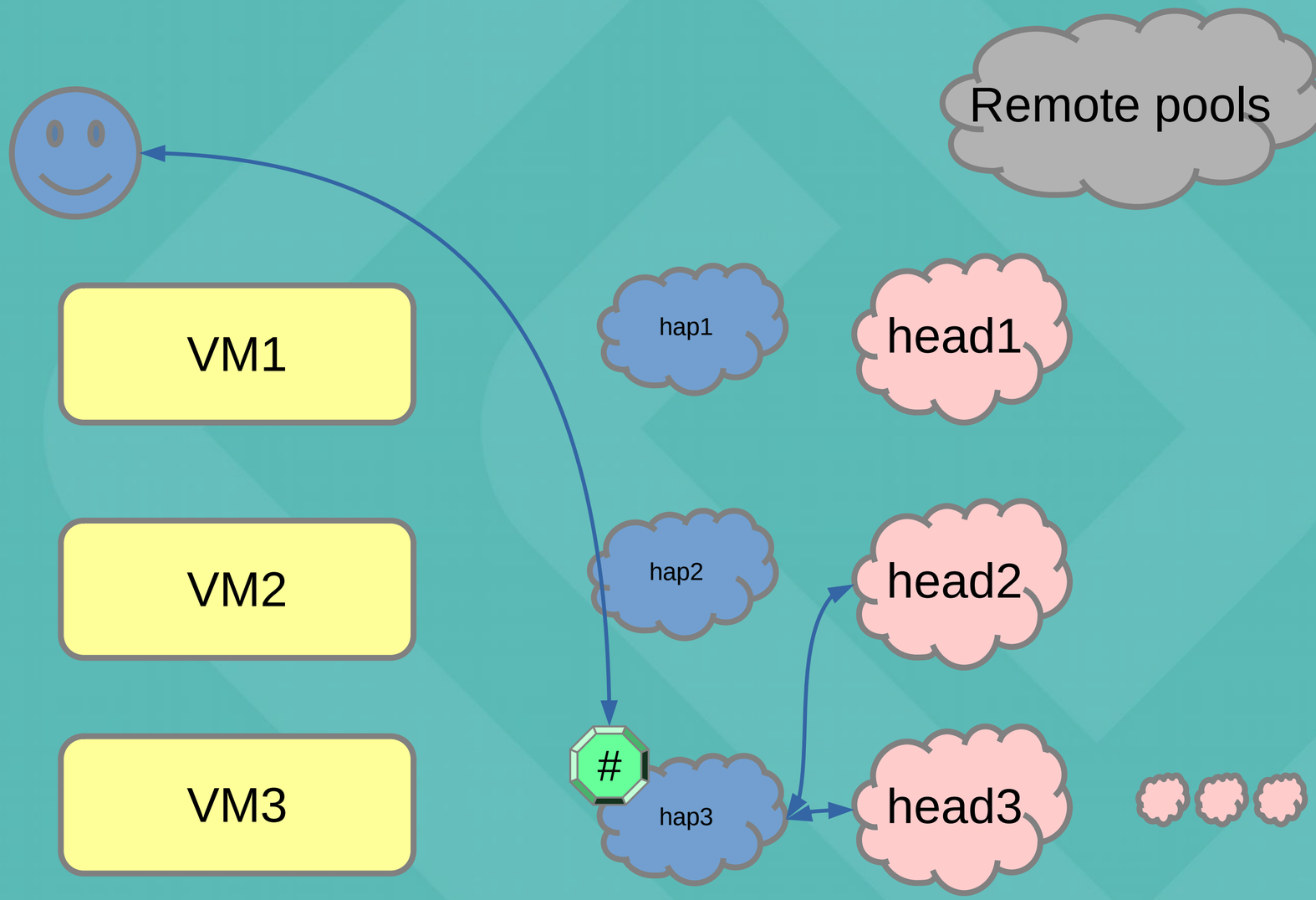
Headnode upgrade

- One headnode also has billing logs and bulk
  - Bulk issn’t HA
  - Billing logs only goes on the headnode with fat disks, but are buffered on the other headnodes while the service is down
  - We try to keep that one down for as short as possible
- Our draining scripts have logic for not draining the last working instance of services
  - Needed workaround for bulk this year
- Don’t reboot/restart the haproxy while draining headnodes! Loss of state → redo from start

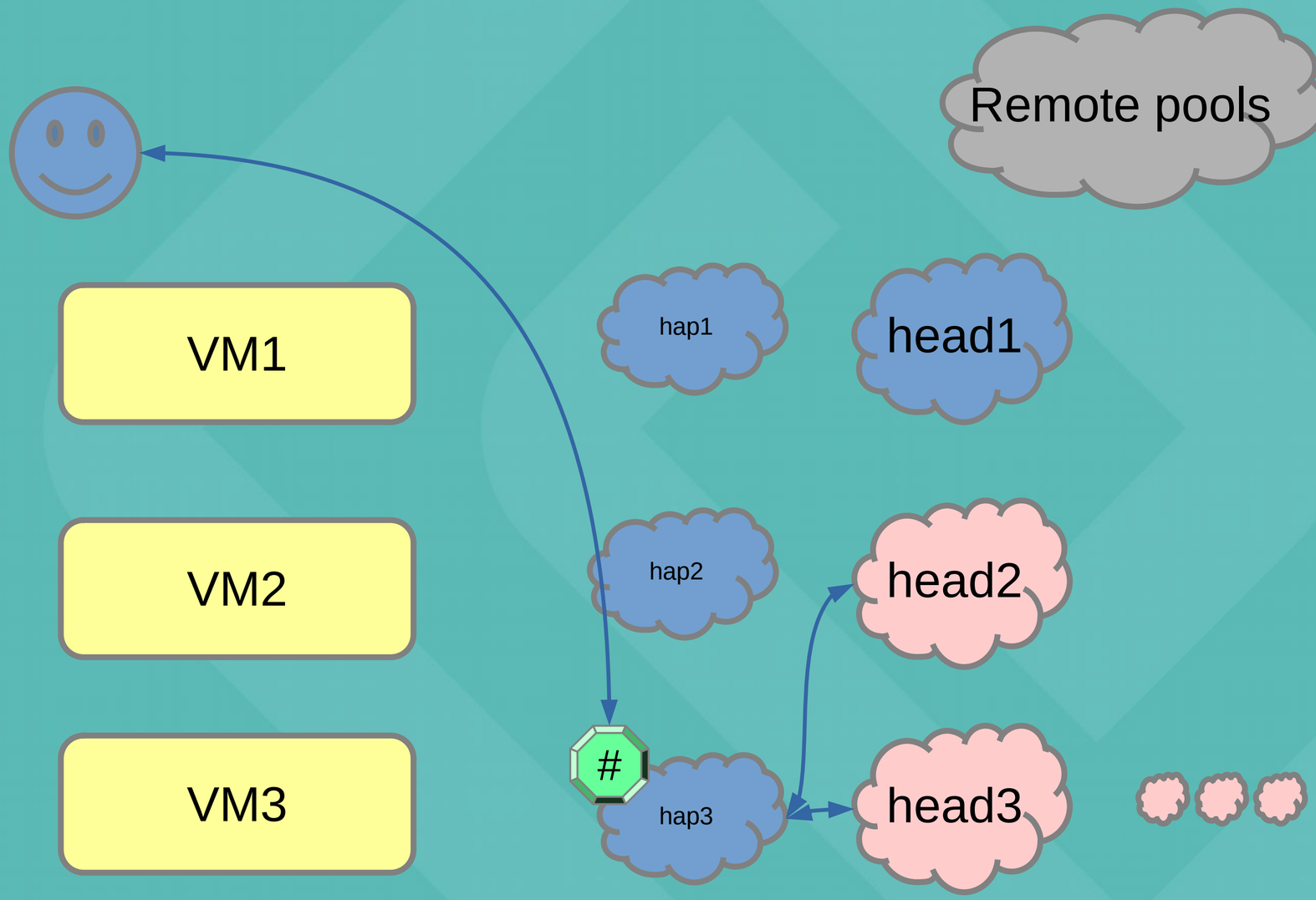
Headnode upgrade



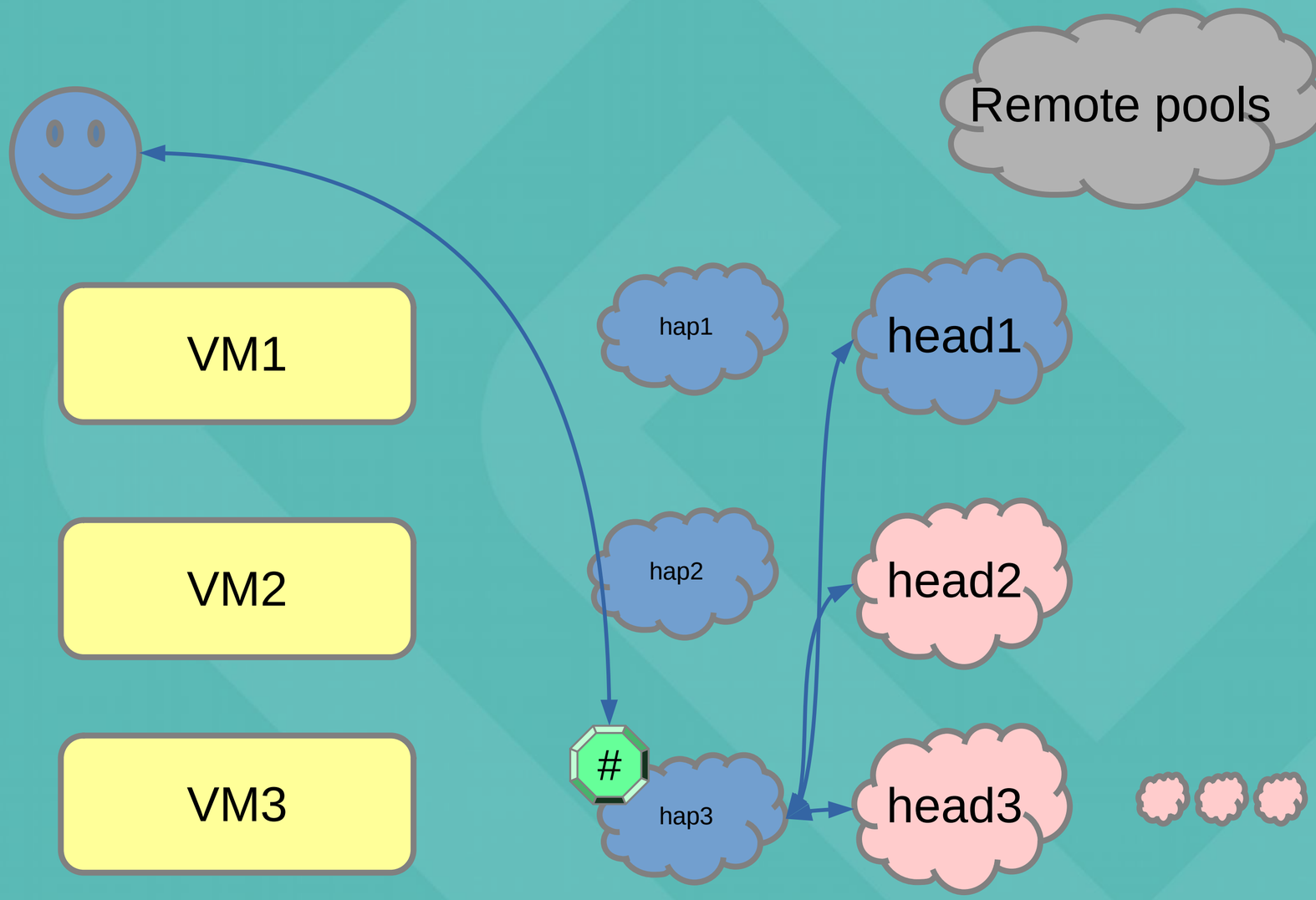
Drain



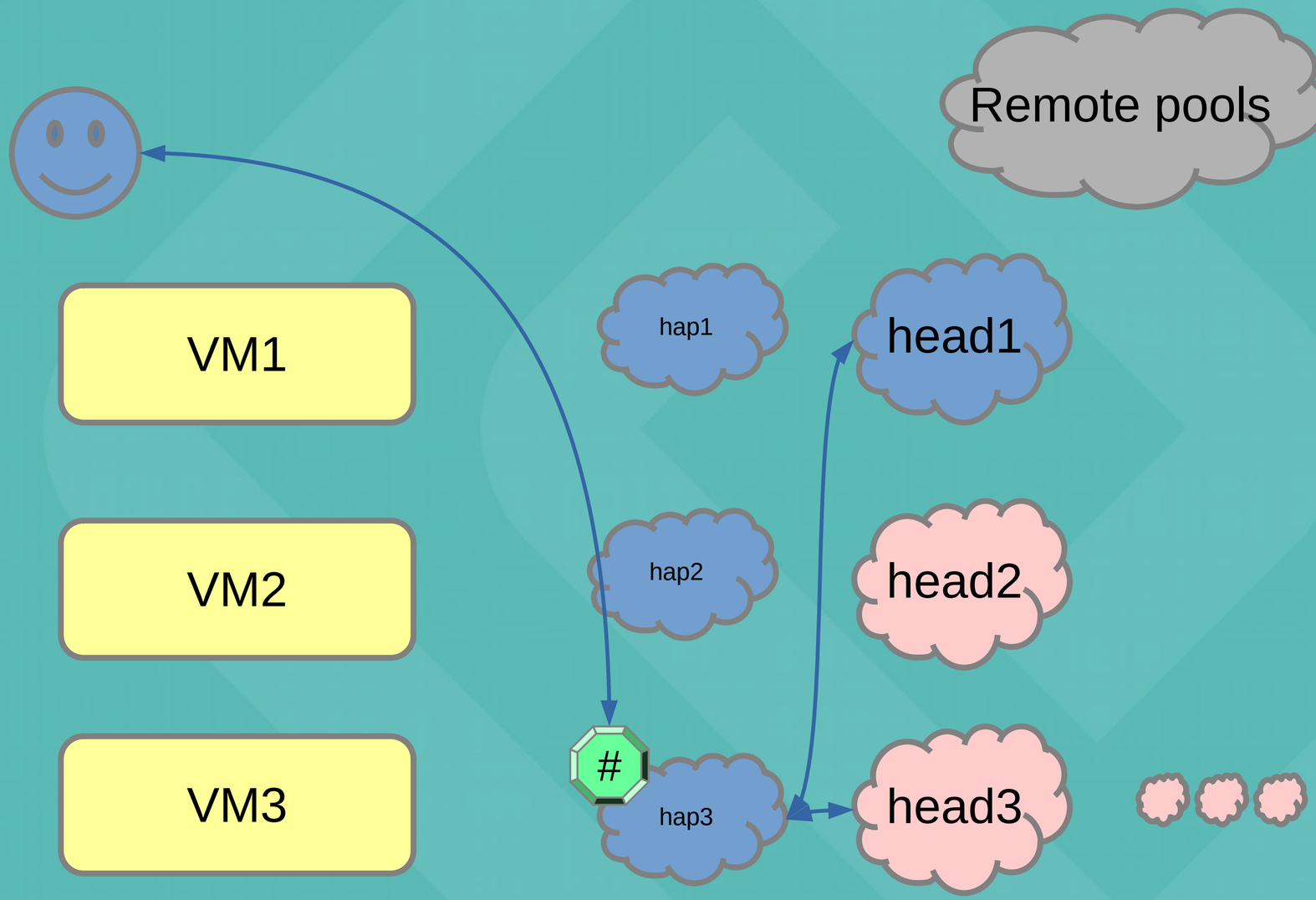
Upgrade



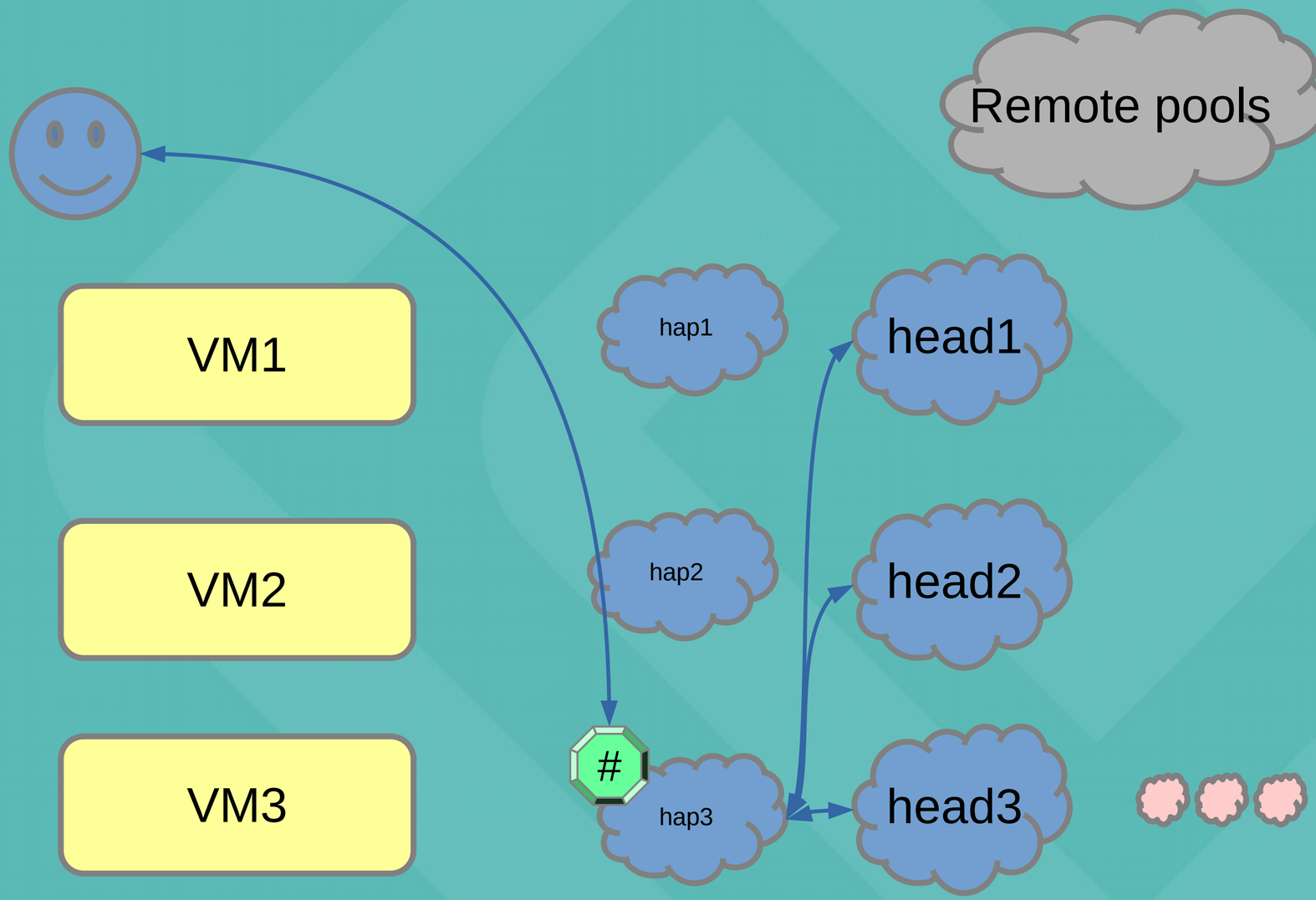
Undrain



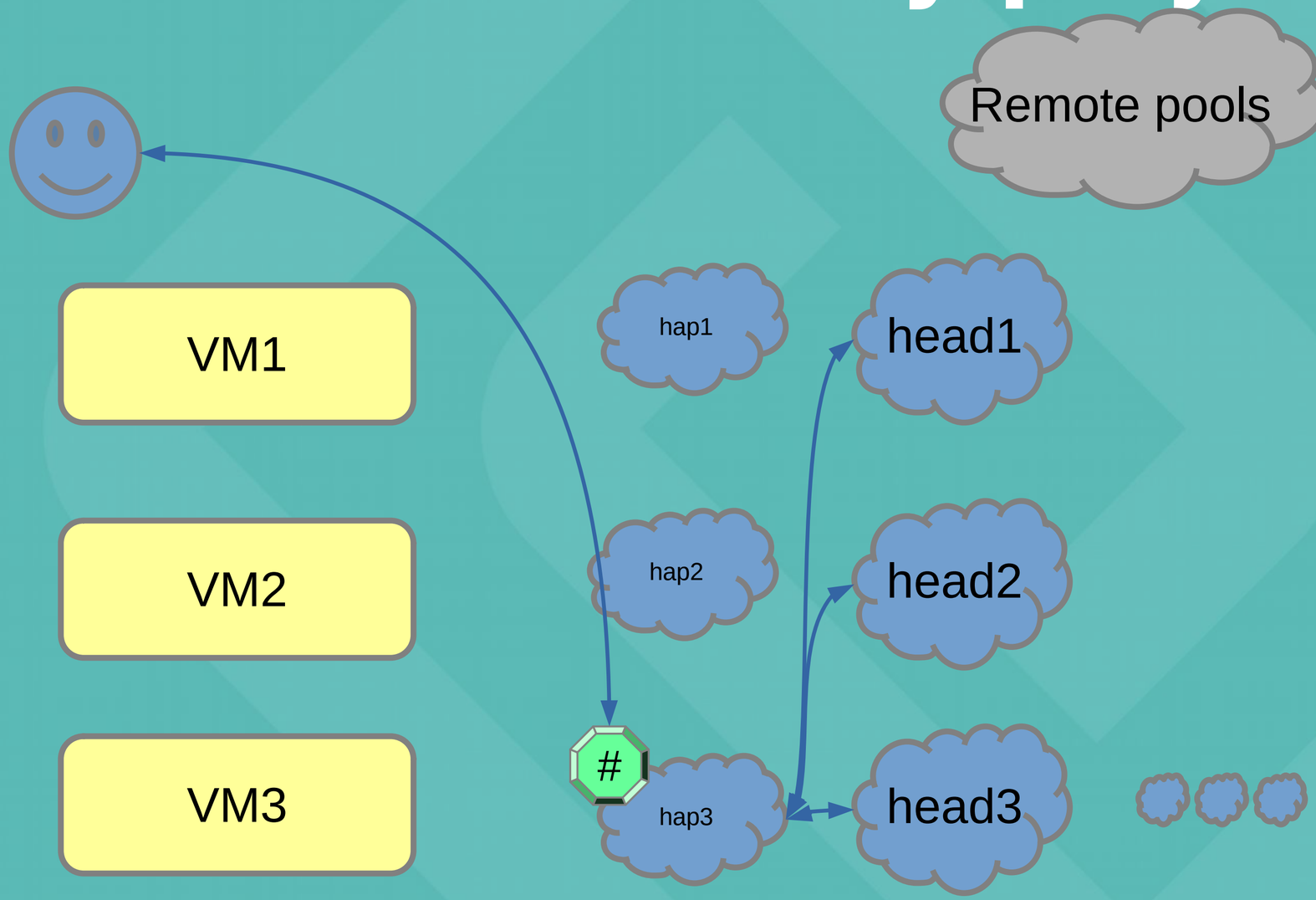
Repeat



Done



Run weekly playbook



Weekly playbook

- Reboots rebootable-at-will
  - Kafka, victoria metrics, nagios, etc, etc
- Reboots zookeepers one at a time
- Other housekeeping

Questions?

NORDIC E-INFRASTRUCTURE COLLABORATION