NDGF HA dCache Delployment and Ops

2017-05-29, NeIC2017, Umeå

Mattias Wadenstein

norden

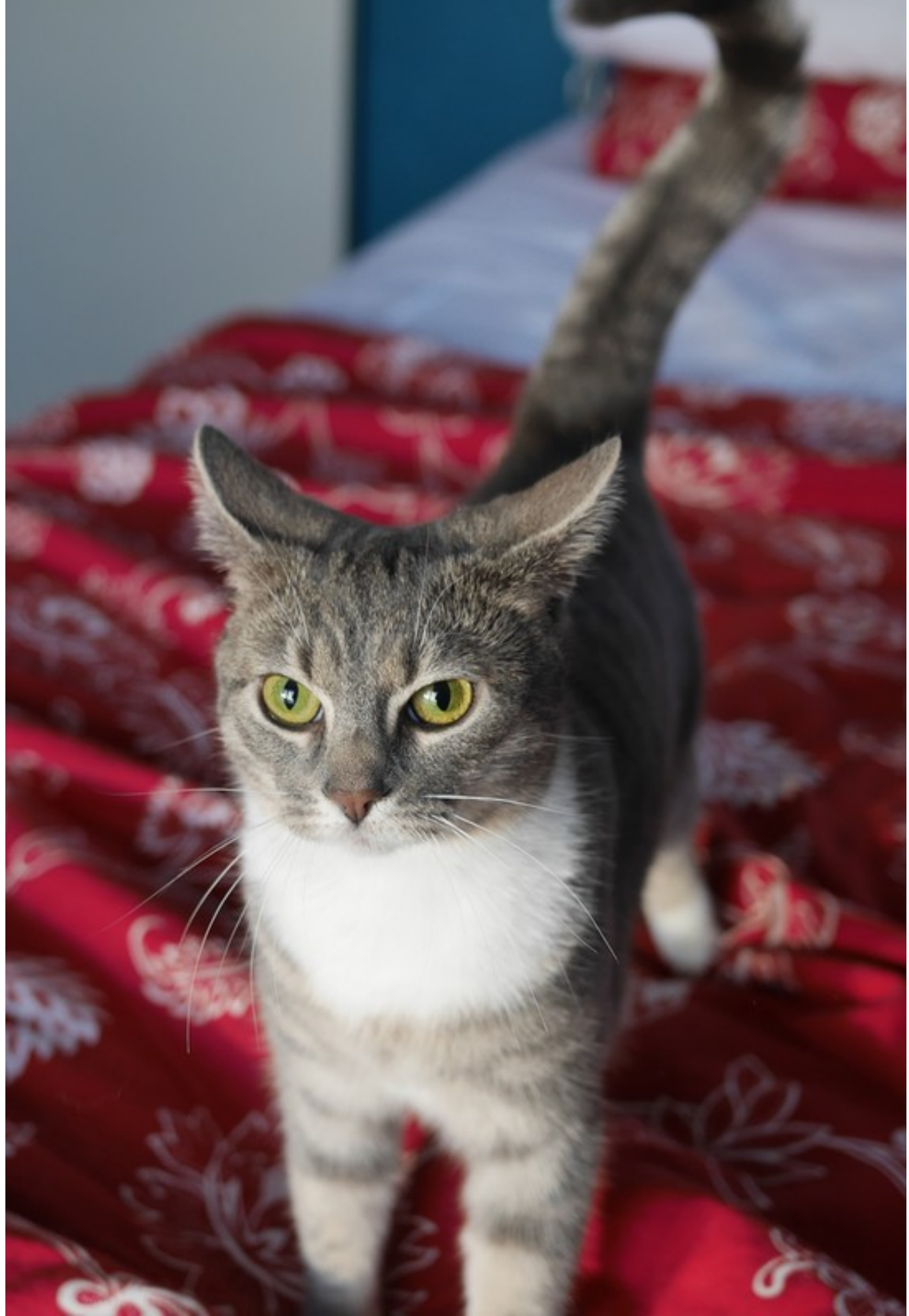
NordForsk

neic

Nordic e-Infrastructure Collaboration

Overview

- NDGF overview
- Central nodes setup
- HA dCache procedures
- Experience



NDGF Overview

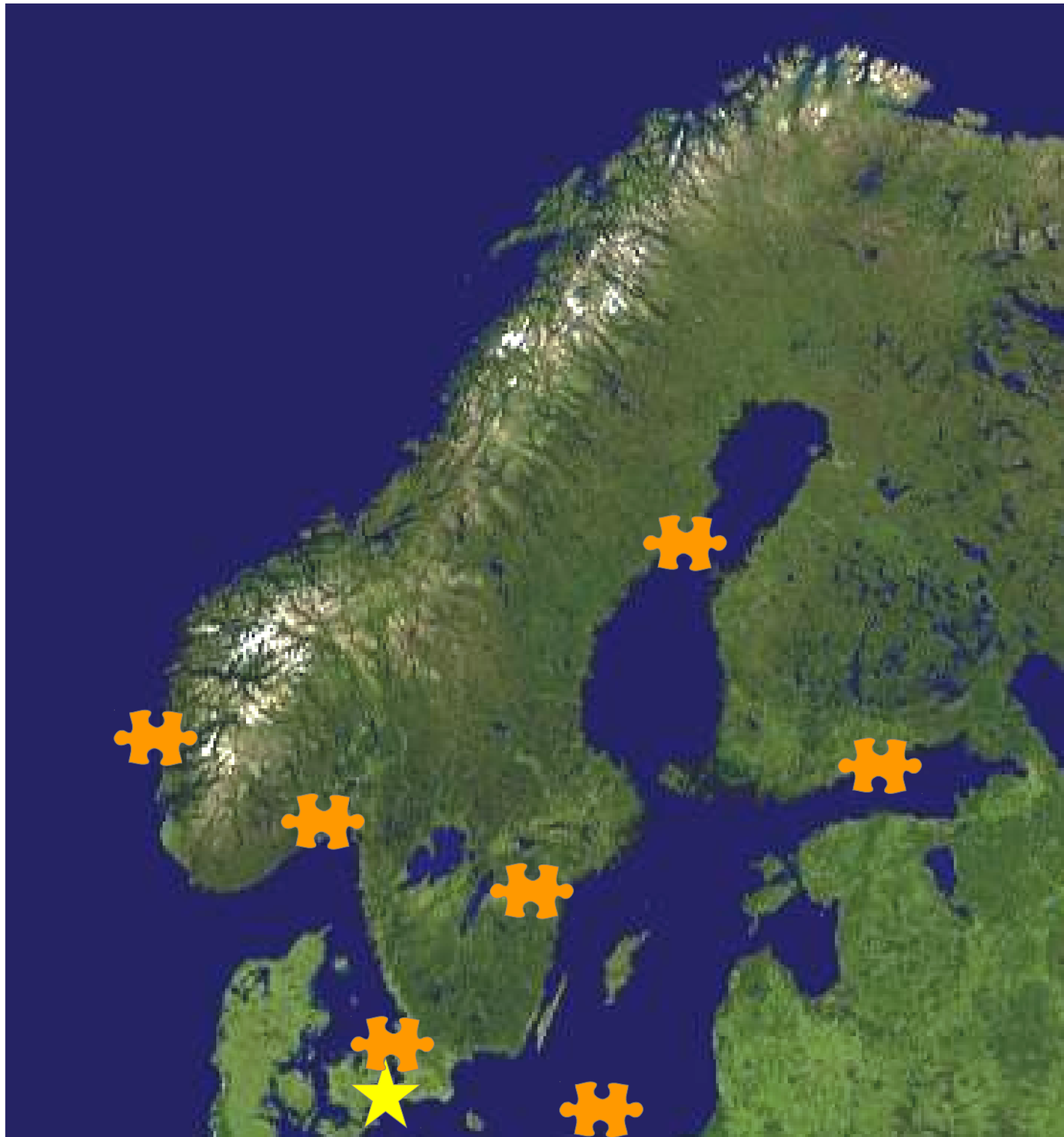
- Distributed WLCG tier-1 site
  - 6 Nordic academic HPC sites with dCache pools and ARC-CEs
  - And disk from IJSs T2 in Slovenia
- Supports ATLAS and ALICE
  - Targets: 6% of ATLAS tier-1 & 9% of ALICE tier-1 resources
- Thanks to the diskspace consolidated into the tier-1 from the tier-2s SI-SIGNET-T2 and SE-SNIC-T2 we currently have second largest ATLASDATADISK among all tier-1 sites

NordForsk

Nordic e-Infrastructure Collaboration

NDGF Overview

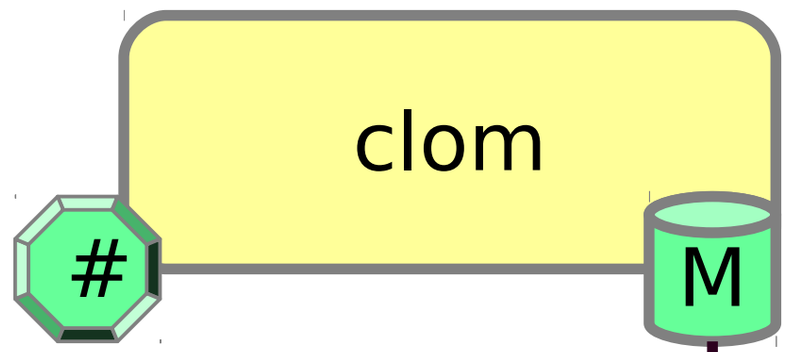
- Our locations
  - (Slovenia actually further south than on map)
- Most have all three kinds of resources
  - Disk, Tape, CPU
- Central nodes in NREN colo facility outside of Copenhagen



Central nodes setup

- We only have two servers
  - Named zanak and clom
  - 2U machines with 384G ram, 24x300G 10k rpm disks
  - Located next to the LHCOPN router
- On these we run Ganeti for hosting VMs and Postgresql
  - And recently also ucarpd&haproxy
  - Using repmgr for HA Postgresql

Central nodes setup



Remote pools

piggy

zoo1

zoo2

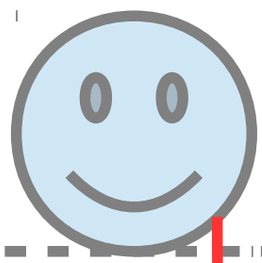
kermit

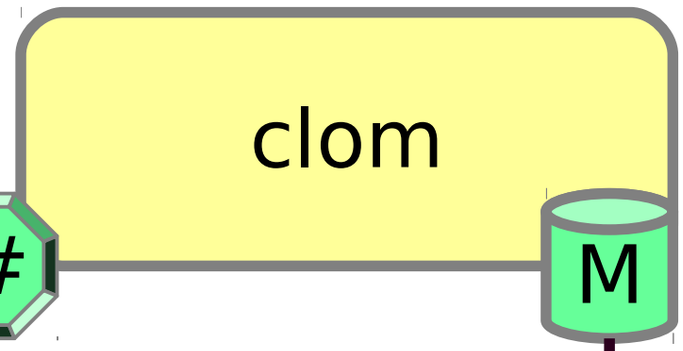
zoo3

ns

Central nodes setup

User makes a request to srm.ndgf.org, hits the floating IP





Remote pools

piggy

zoo1

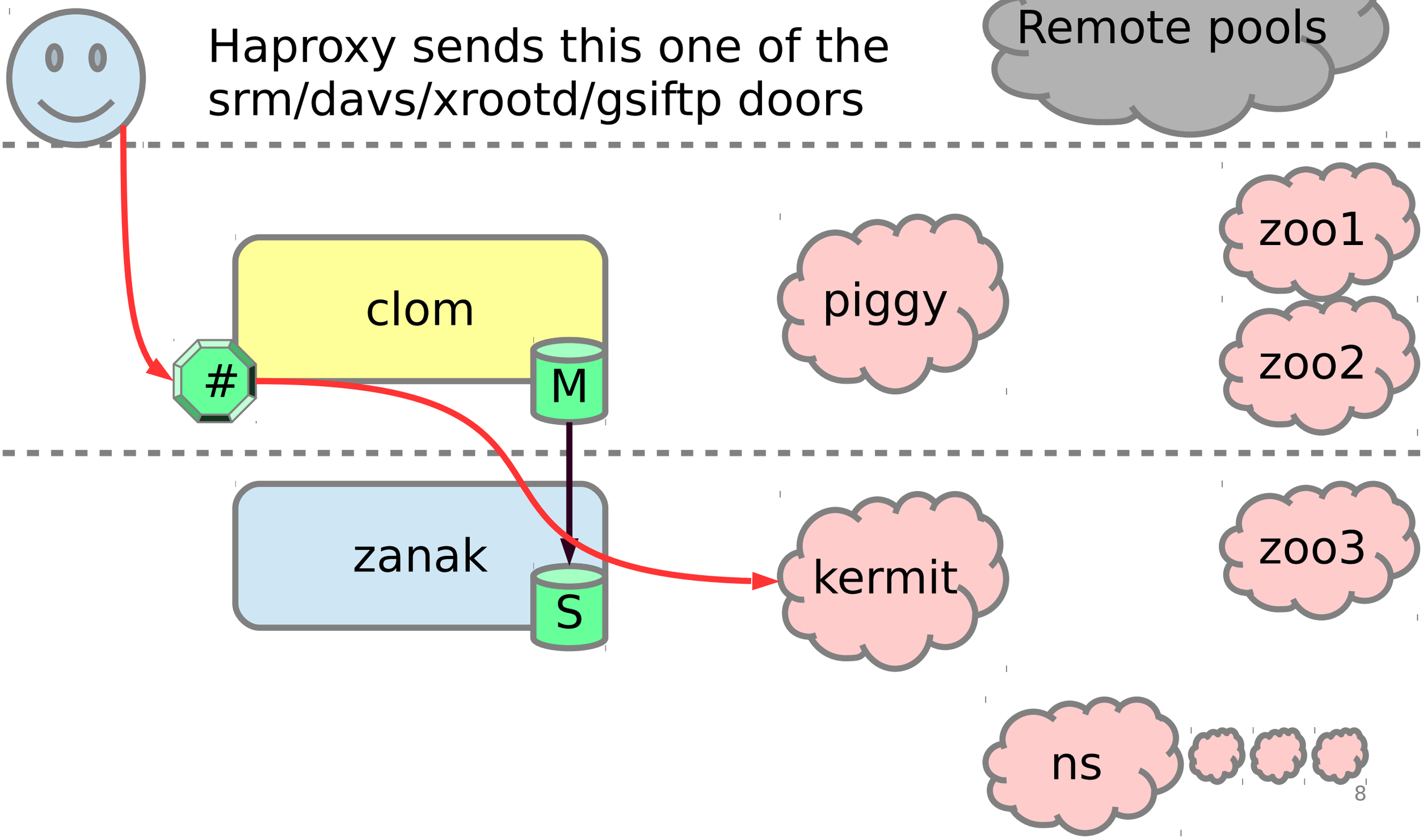
zoo2

kermit

zoo3

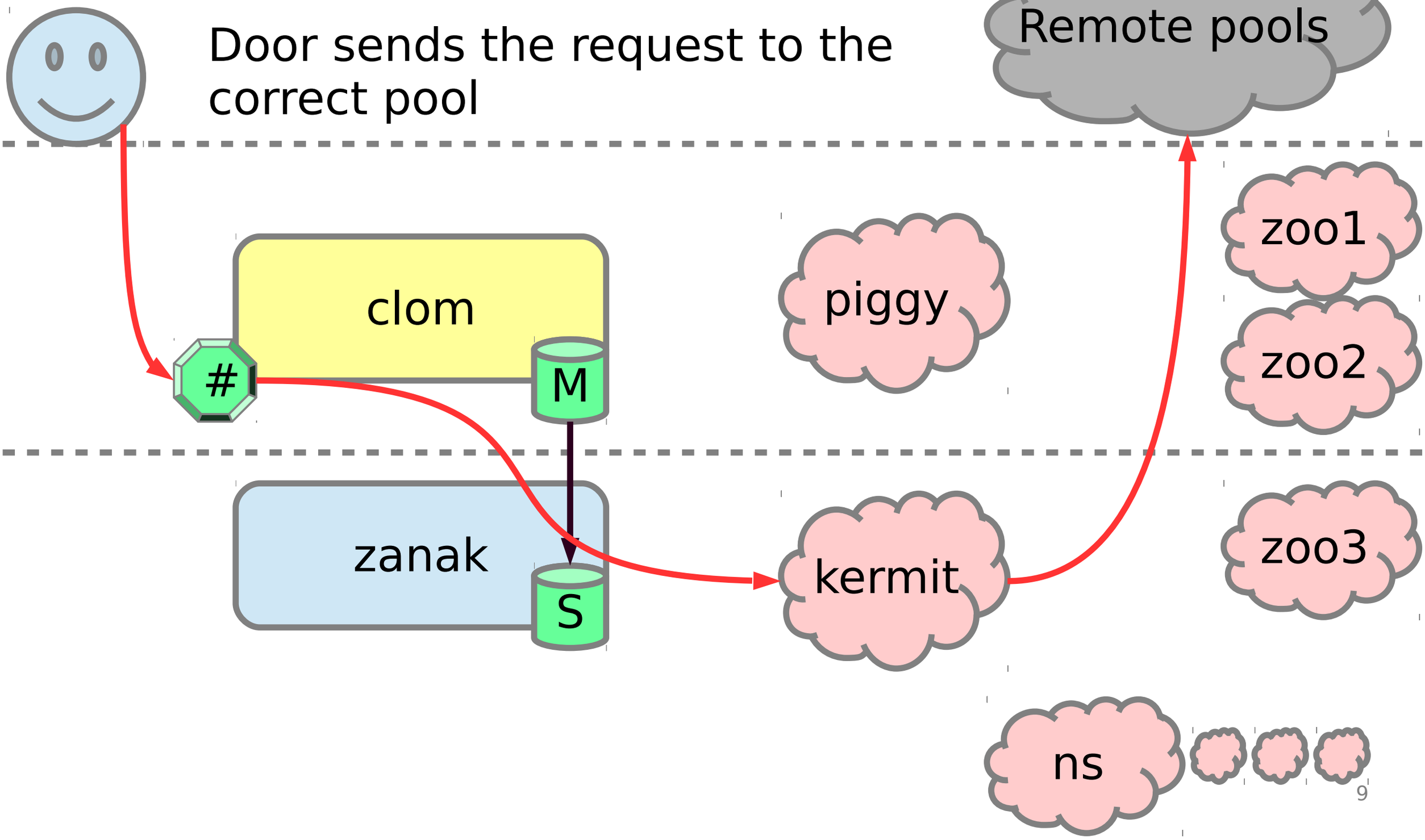
ns

Central nodes setup



Central nodes setup

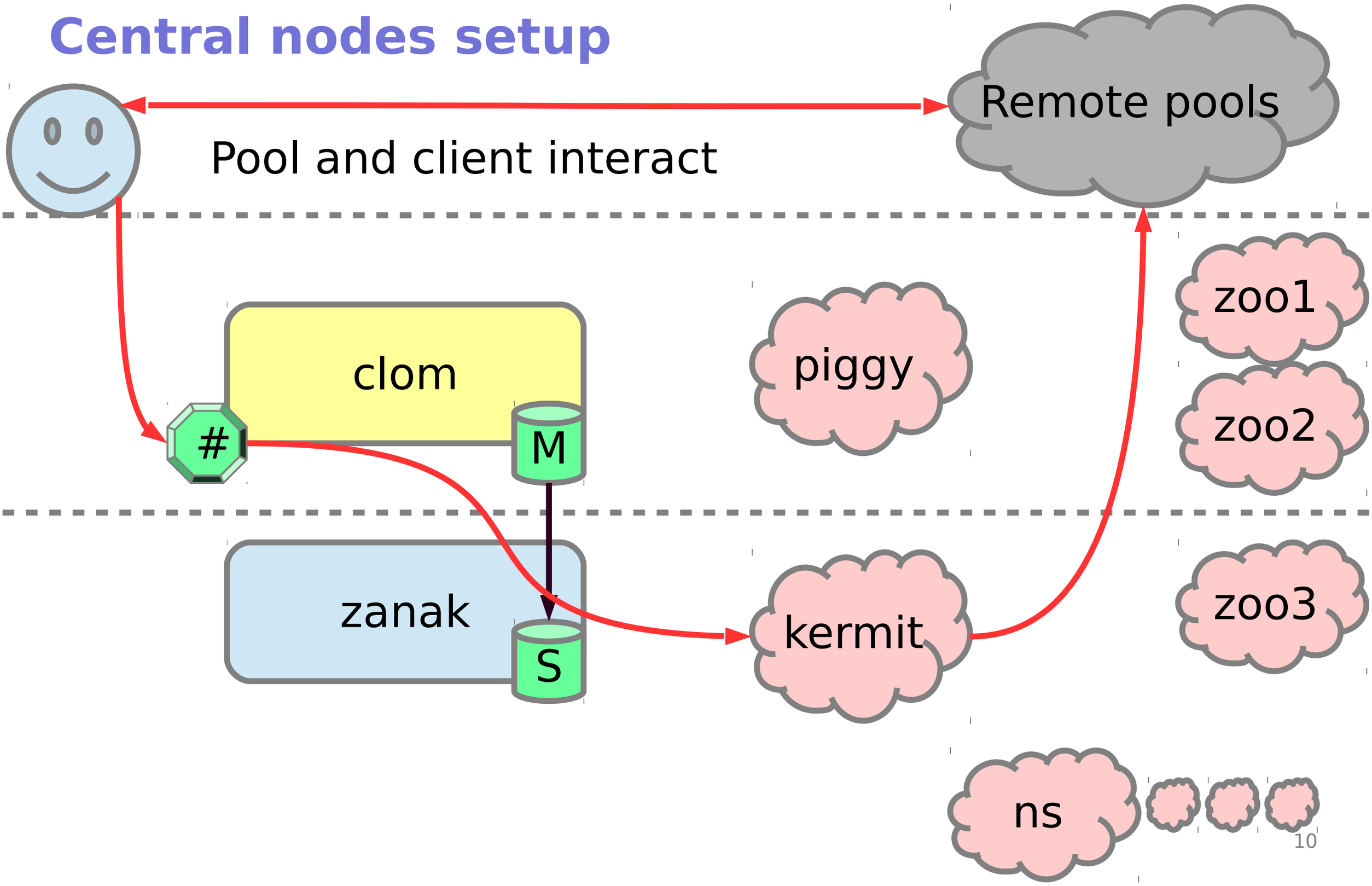
Door sends the request to the correct pool



NordForsk

Nordic e-Infrastructure Collaboration

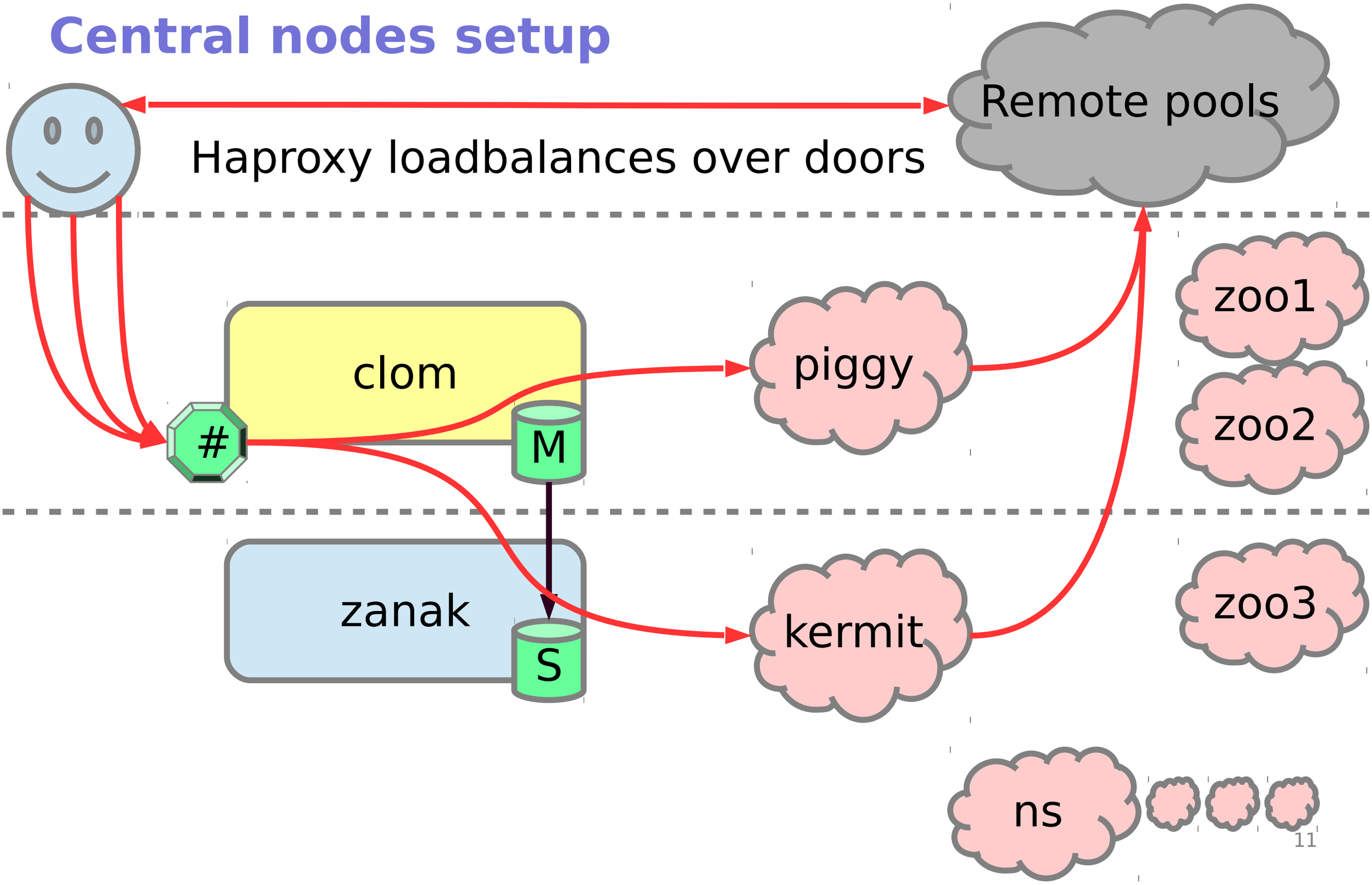
Central nodes setup



NordForsk

Nordic e-Infrastructure Collaboration

Central nodes setup

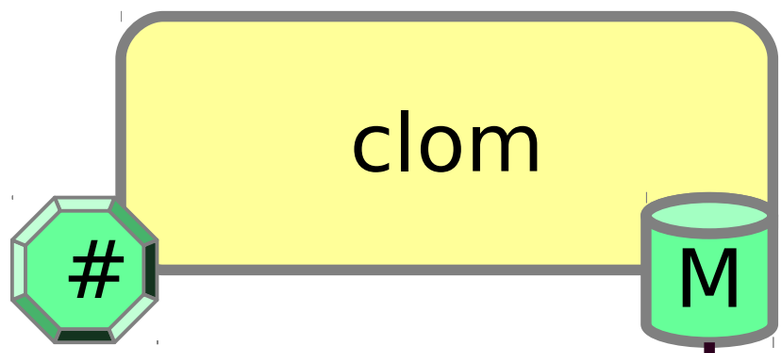


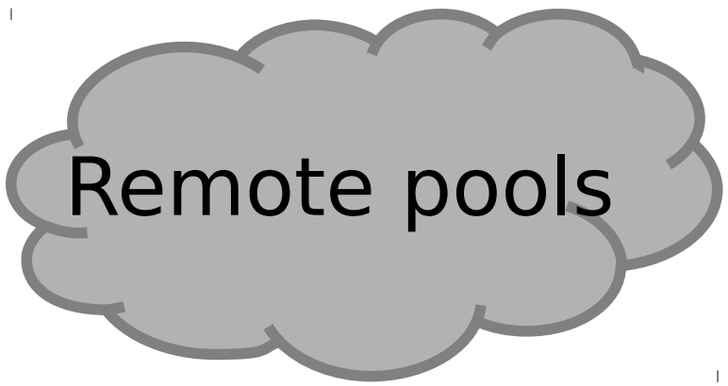
HA dCache upgrades

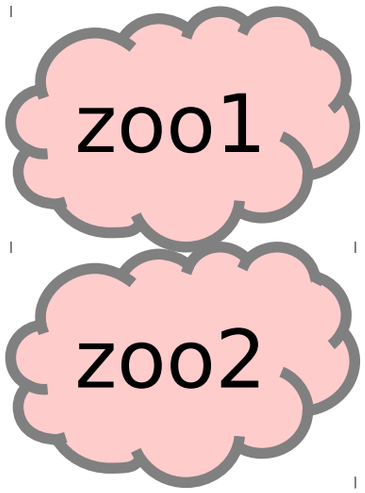
- With the new HA features in dCache we can do system updates including reboot into new kernels with no downtime
- Can typically be done in a day, but takes a bit of watching to make sure we don't interrupt any client accesses
- Can also do dCache upgrades of headnodes without anyone noticing, over a couple of days
  - Unless something goes wrong, of course
- Hardware and headnode upgrades on different days
  - Headnode upgrades depends on haproxy draining state – this is reset by a reboot of the hardware that runs haproxy

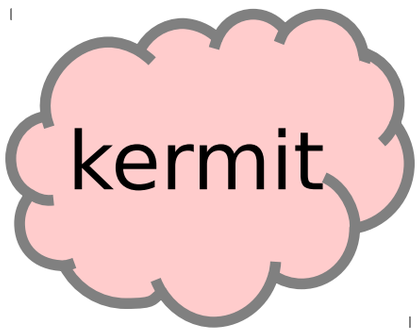
Kernel update on HW nodes

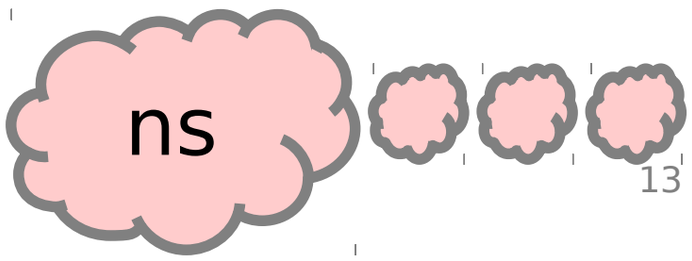
Ubuntu releases a security update for the linux kernel!





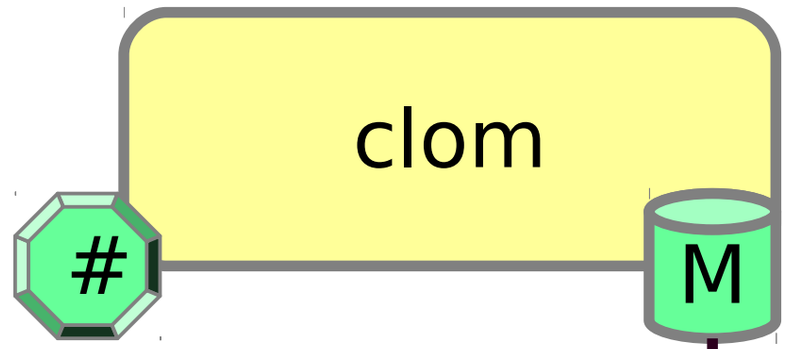


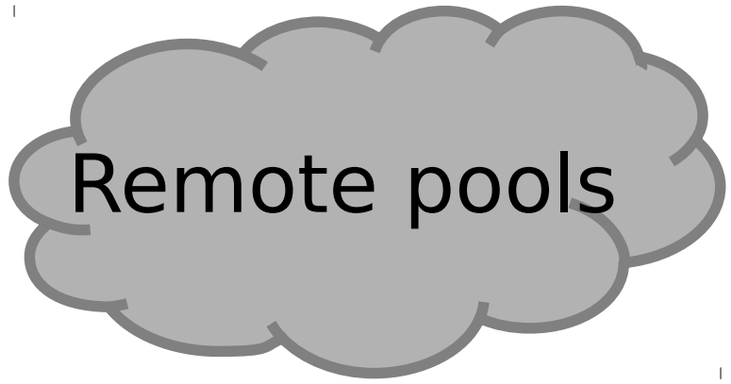


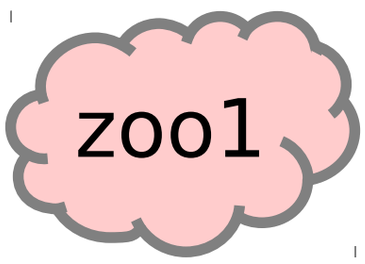


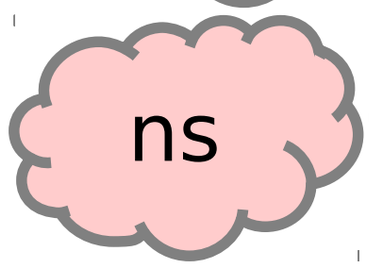
Kernel update on HW nodes

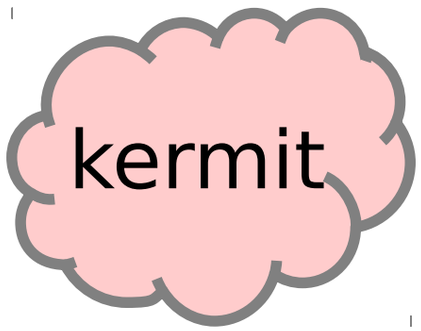
Live migrate all VMs to zanak
z# gnt-node migrate clom





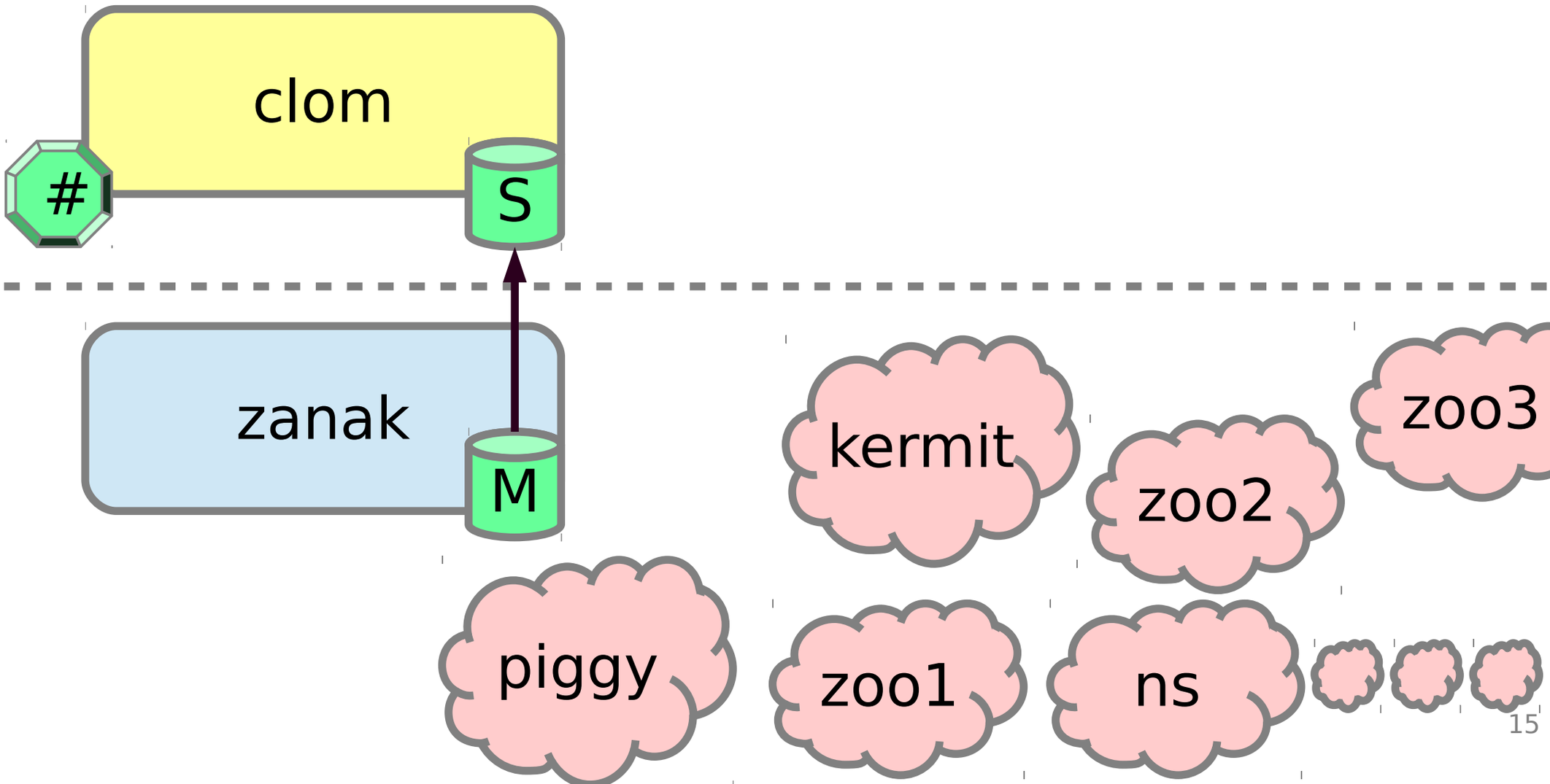






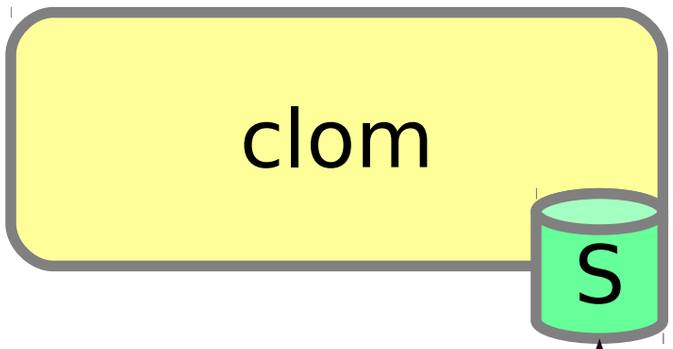
Kernel update on HW nodes

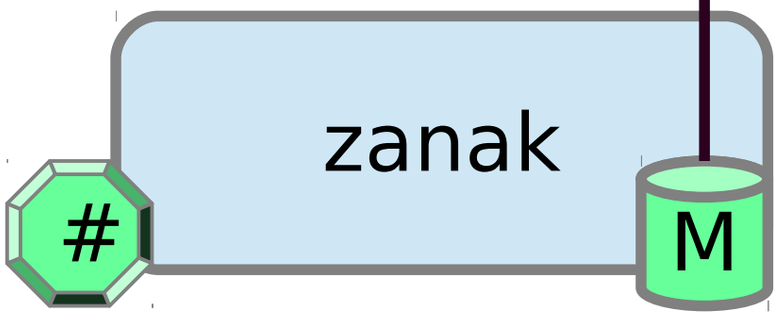
Switch postgresql master
z# repmgr standby switchover

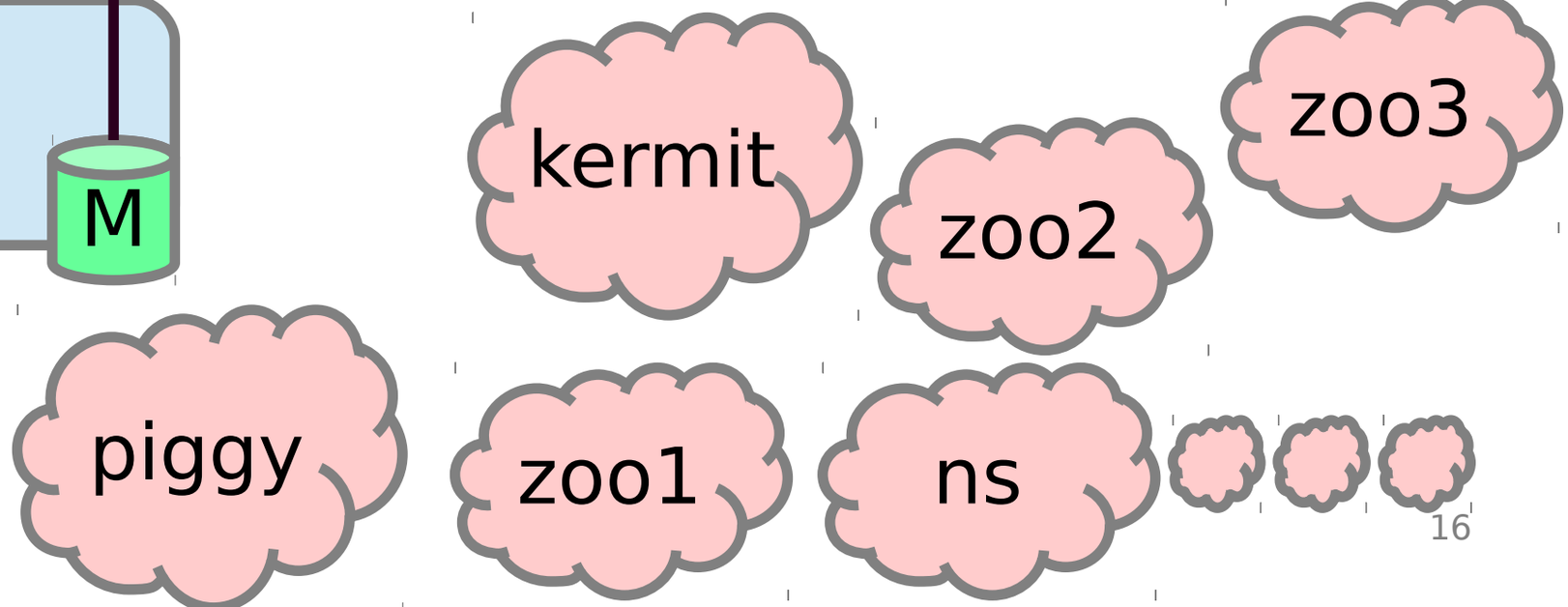


Kernel update on HW nodes

Move the floating IP
c# pkill ucarp

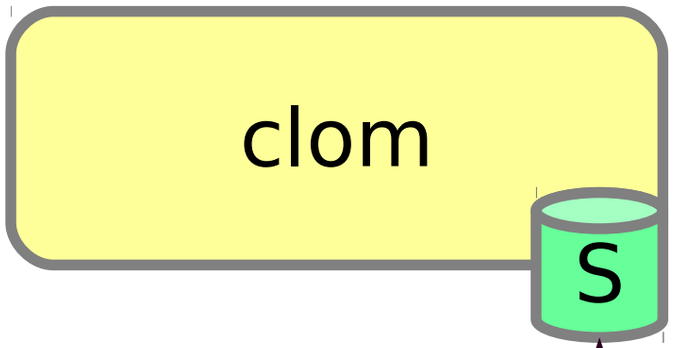


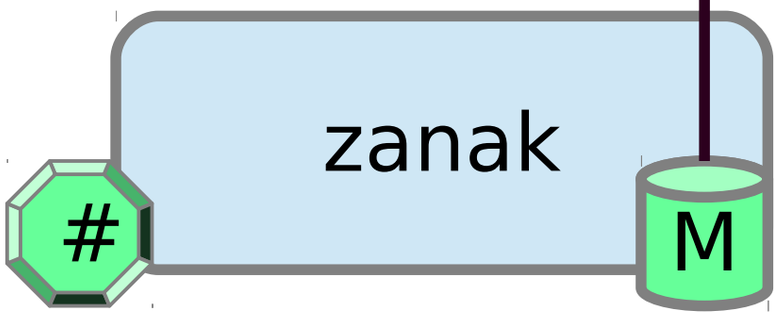


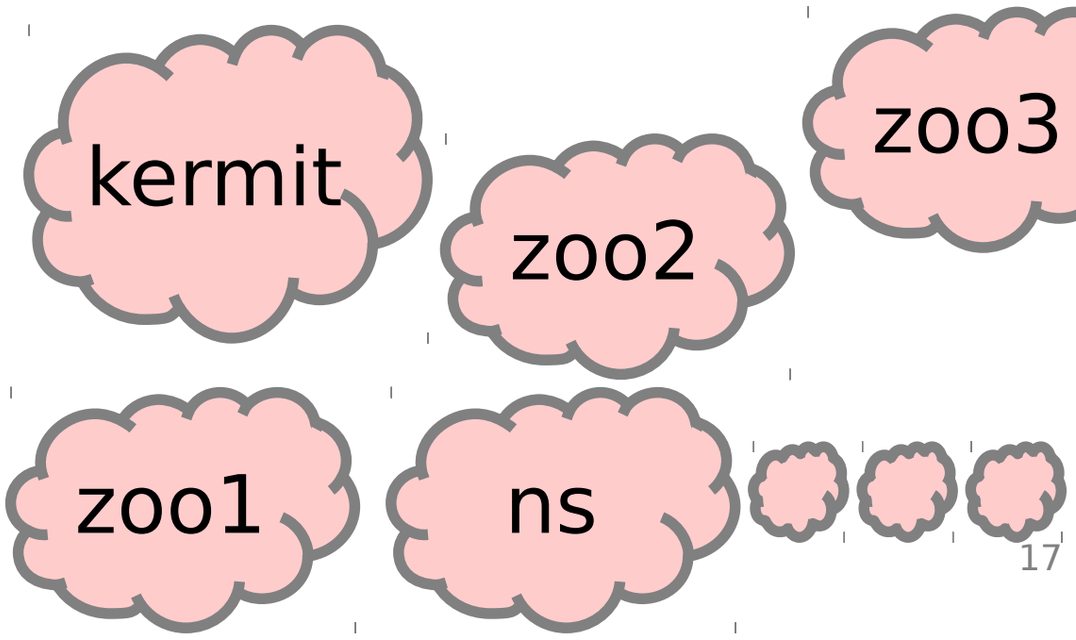


Kernel update on HW nodes

Reboot clom
c# reboot

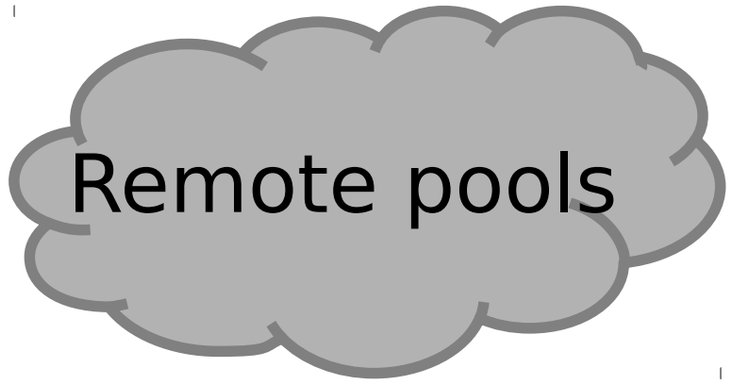


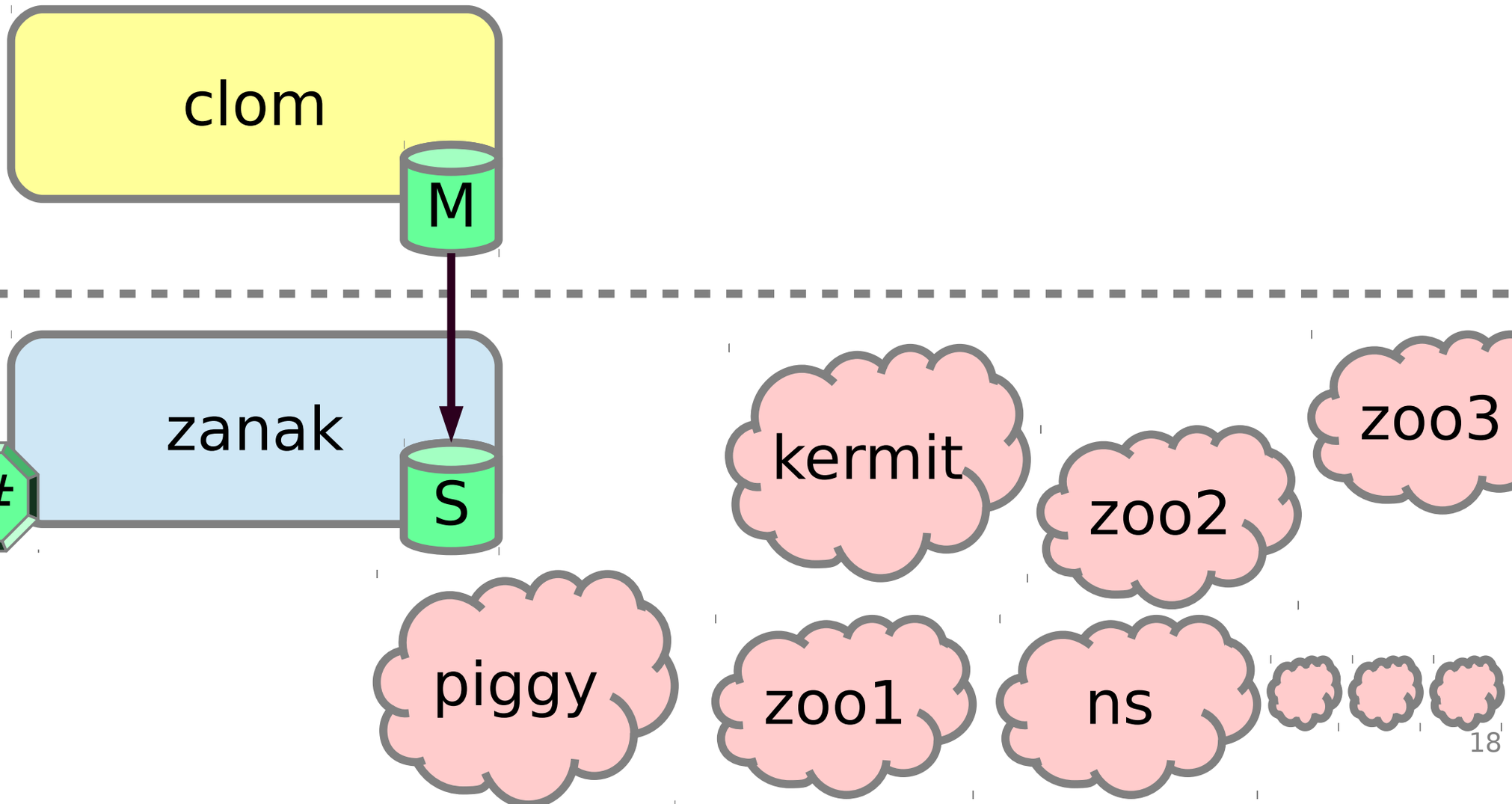




Kernel update on HW nodes

Switch postgres back
c# repmgr standby switchover





Kernel update on HW nodes

Switch IP back
z# pkill ucarp

Remote pools

clom

#

M

zanak

S

kermit

zoo2

zoo3

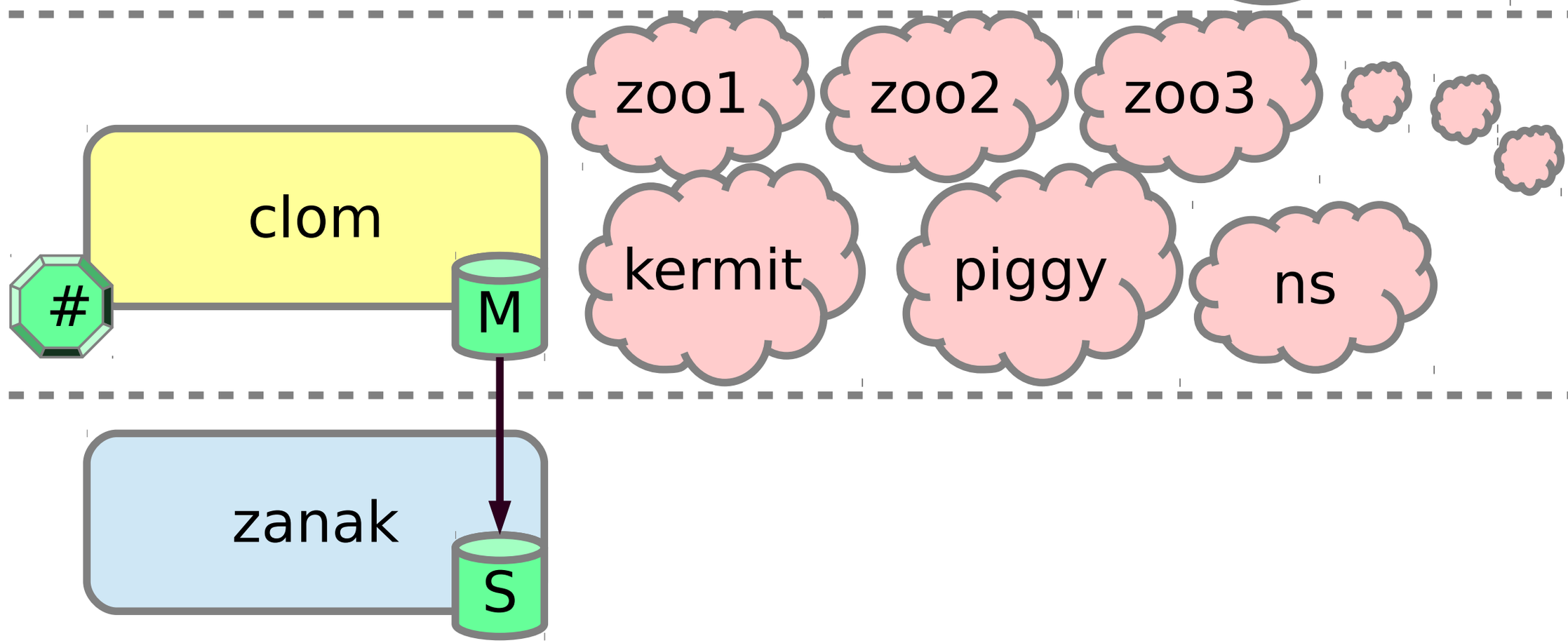
piggy

zoo1

ns

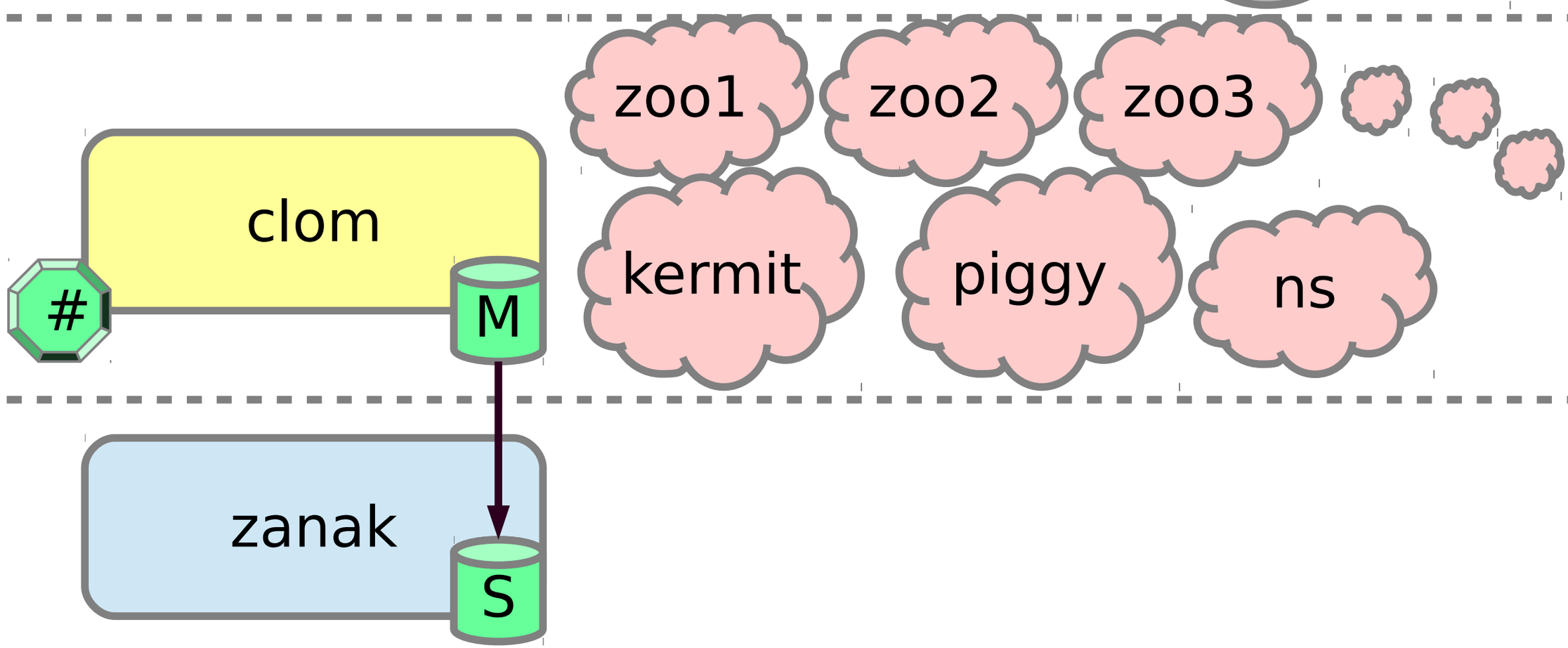
Kernel update on HW nodes

Move all the VMs to clom
z# gnt-node migrate zanak



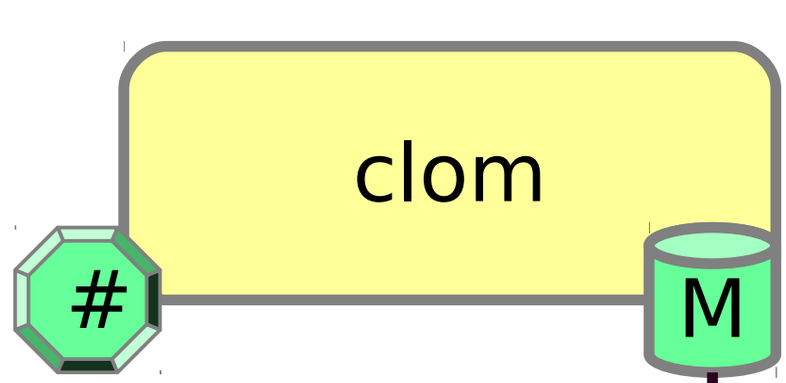
Kernel update on HW nodes

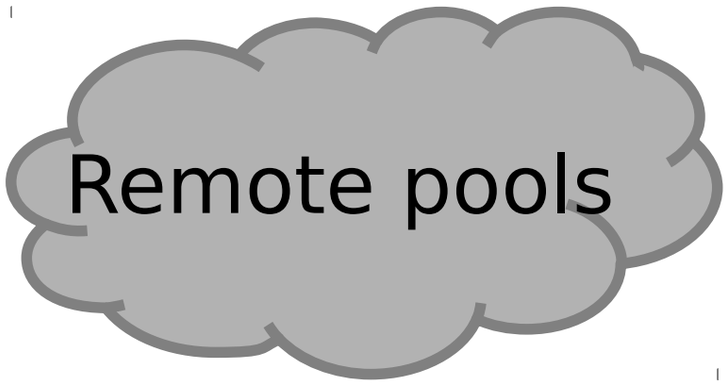
Reboot zanak
z# reboot

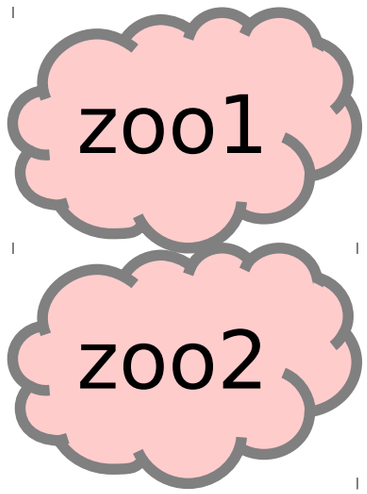


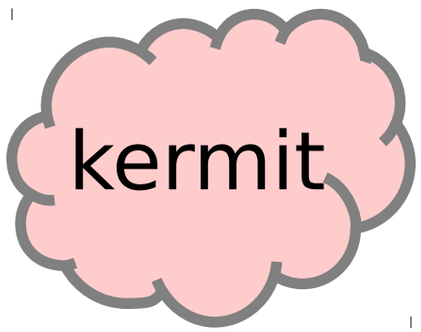
Kernel update on HW nodes

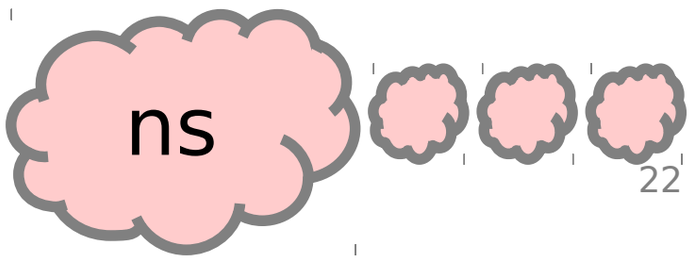
Migrate the VMs back to normal
(where zanak can be lost)





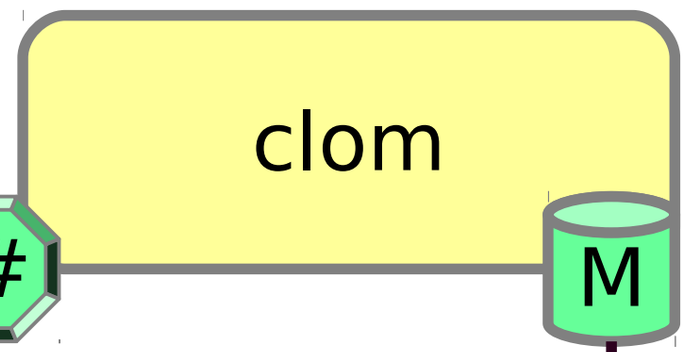


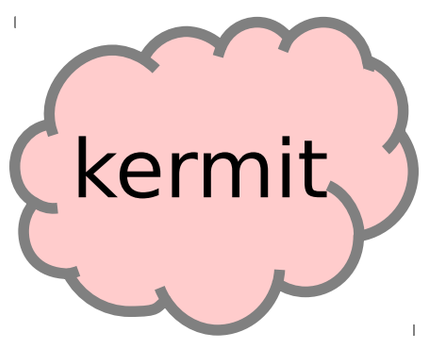


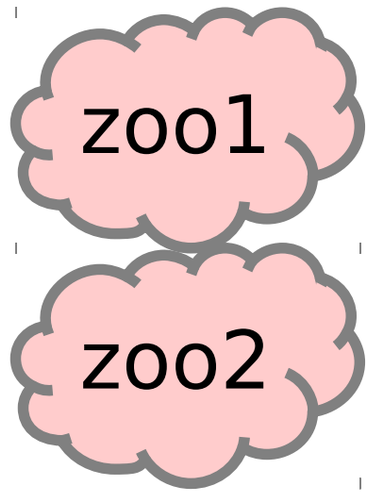


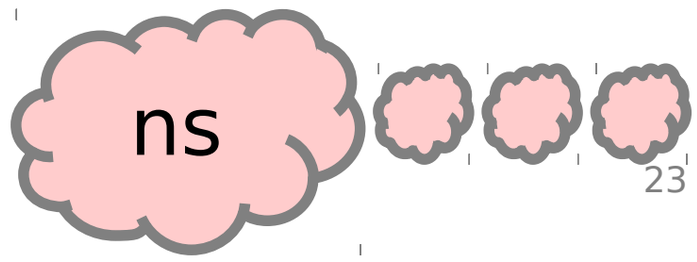
Kernel update on VM nodes

Maybe we need to reboot the virtual machines as well?



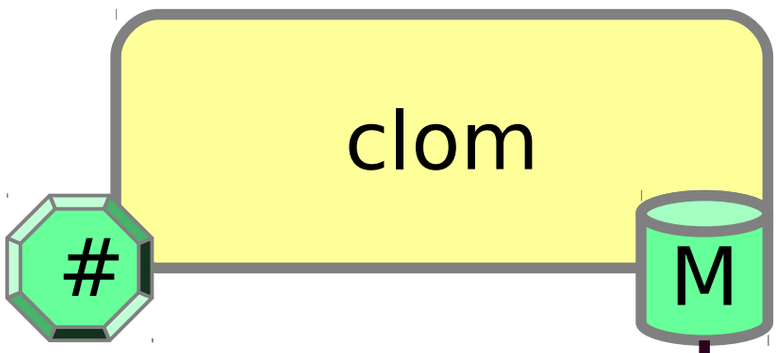


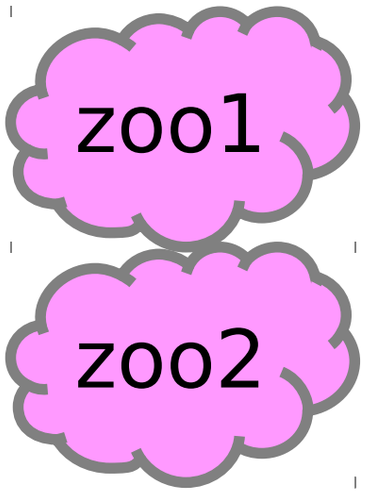


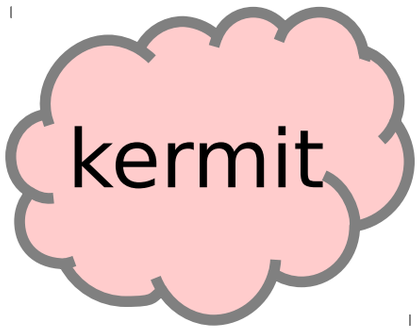


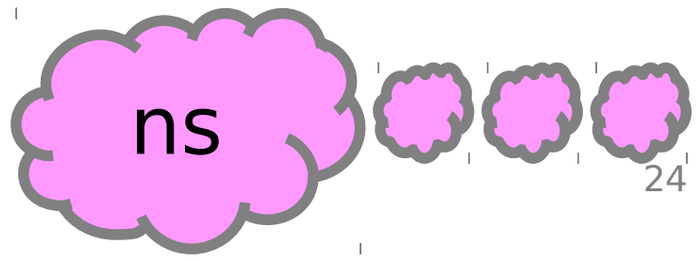
Kernel update on VM nodes

ZooKeeper nodes one by one,
others just whenever





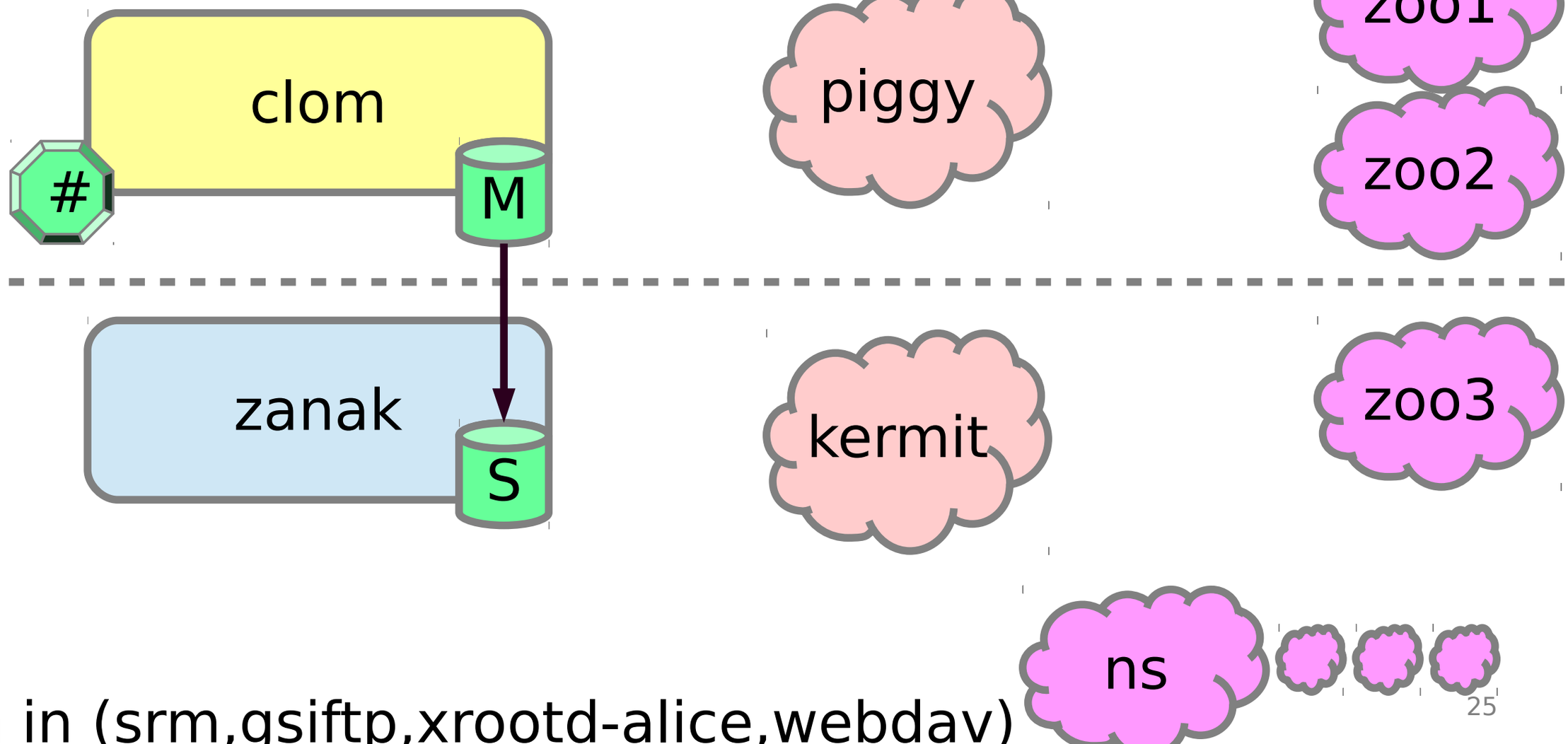




Kernel update on VM nodes

Disable piggy in haproxy:
disable nn/piggy | socat ha.sk

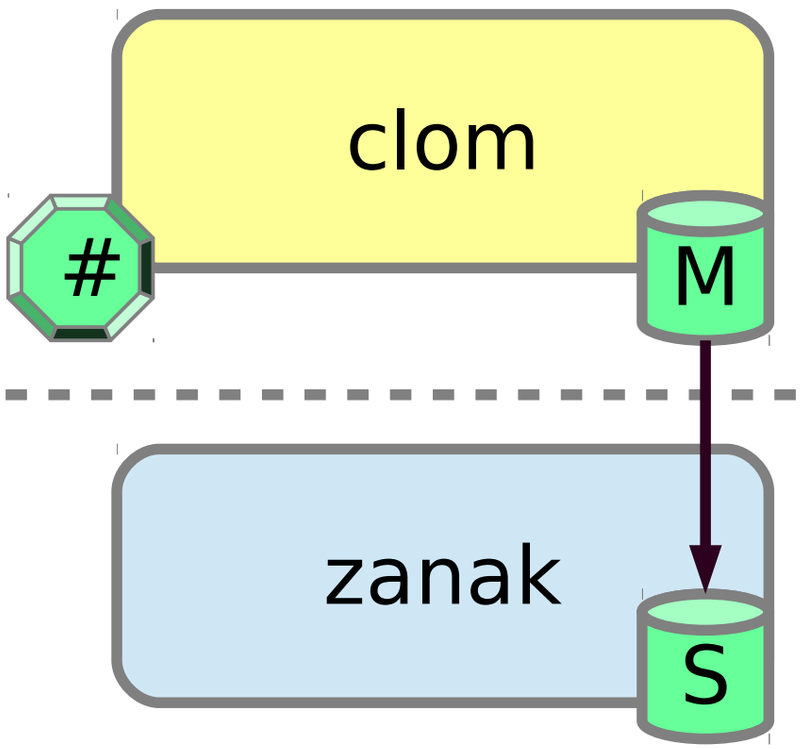
Remote pools

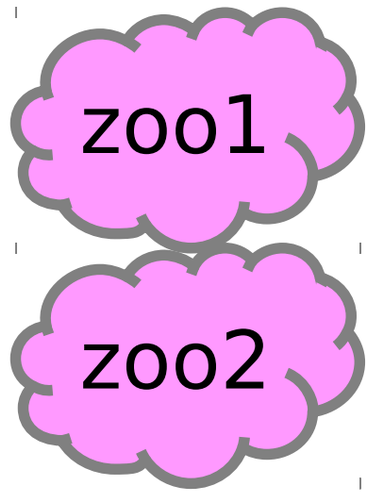


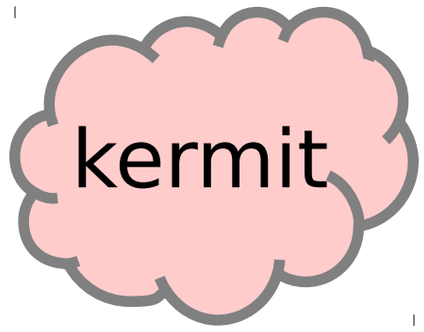
for nn in (srm,gsiftp,xrootd-alice,webdav)

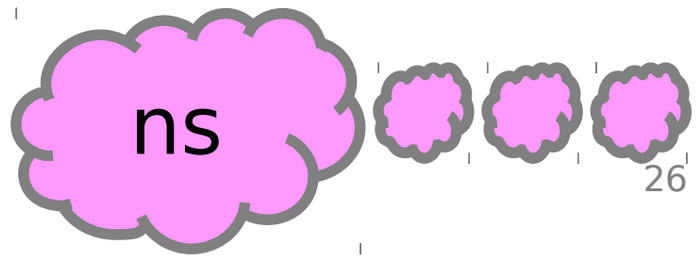
Kernel update on VM nodes

Disable piggy doors in dCache:
\s nn*piggy lb disable







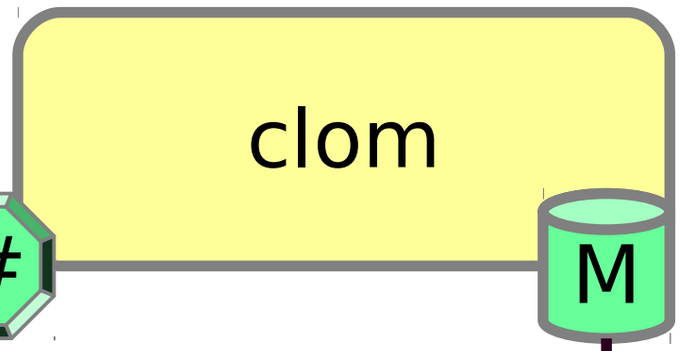


for nn in (gsiftp,webdav,xrootd)

Kernel update on VM nodes

Wait roughly forever (24h default SRM life) until empty

Remote pools



piggy

zoo1

zoo2

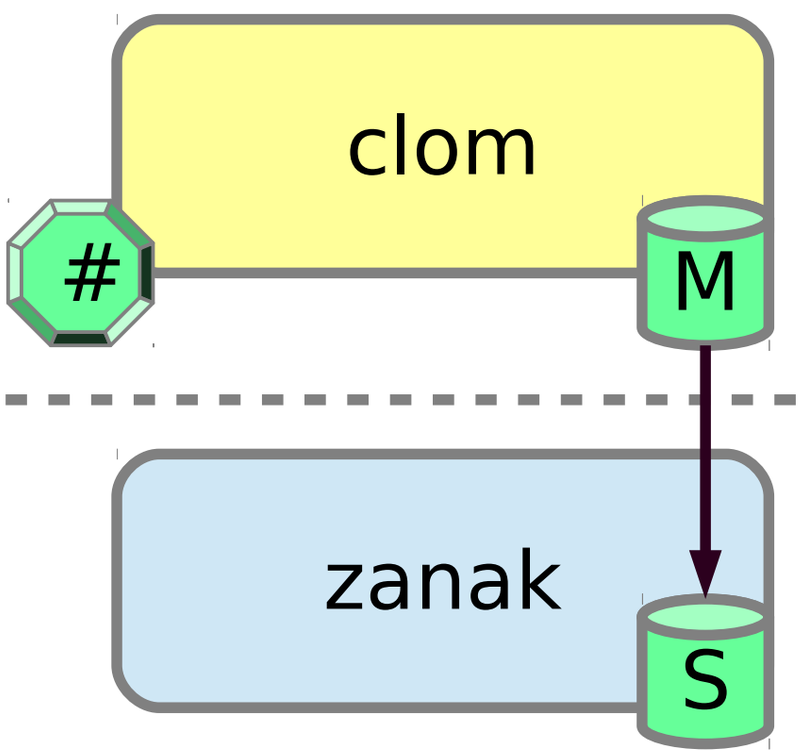
kermit

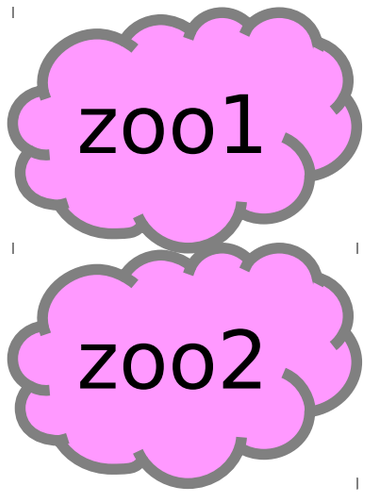
zoo3

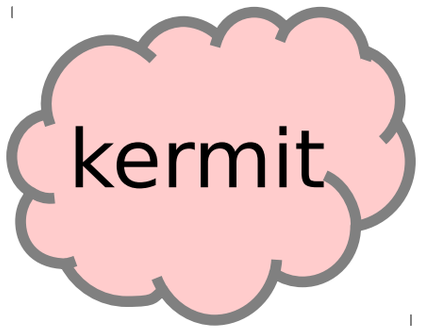
ns

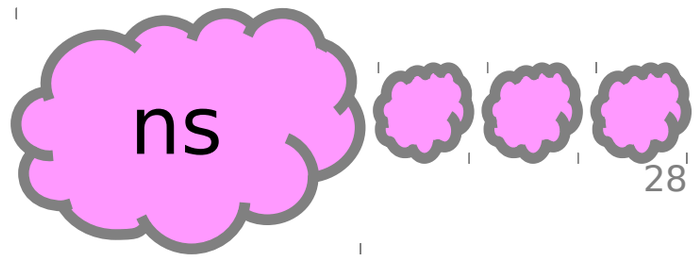
Kernel update on VM nodes

When haproxy and SRM show 0
piggy# reboot





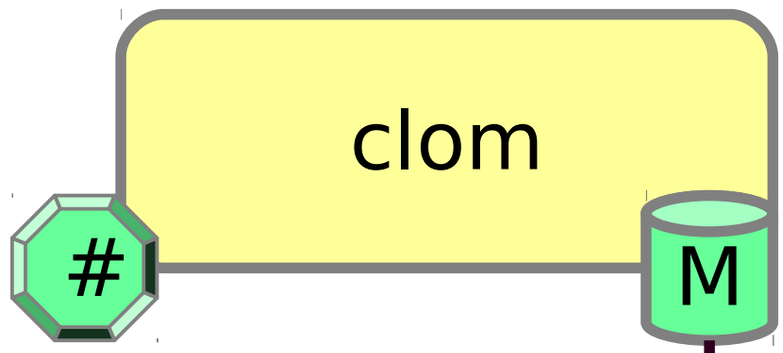




Kernel update on VM nodes

Re-enable haproxy for piggy
The lb state gets reset on reboot

Remote pools



piggy

zoo1

zoo2

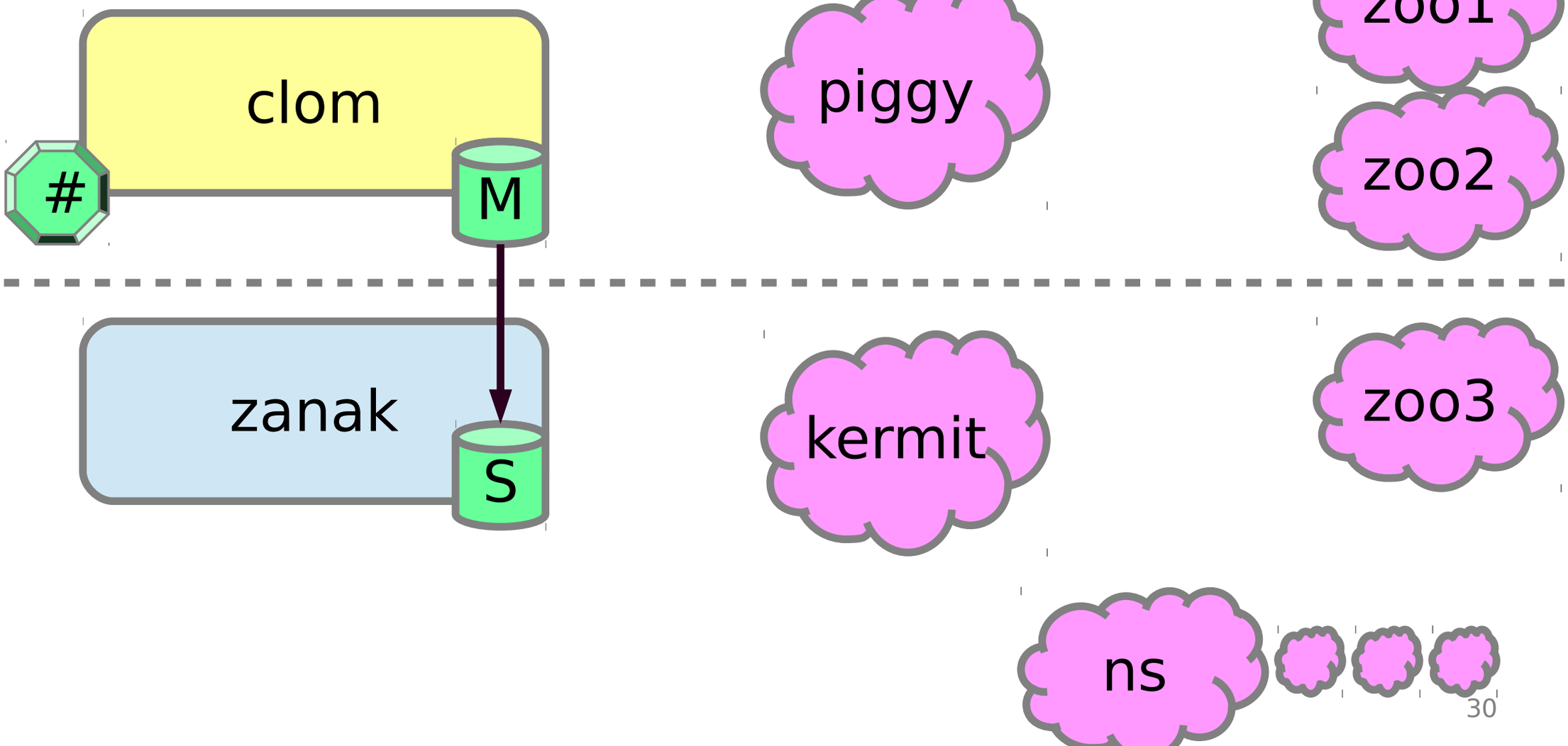
kermit

zoo3

ns

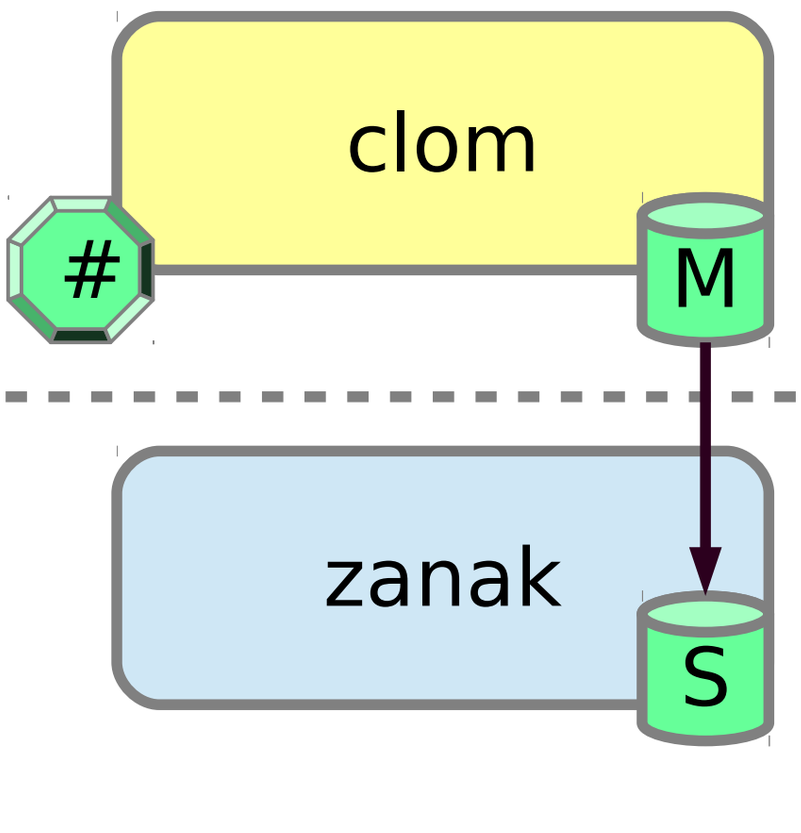
Kernel update on VM nodes

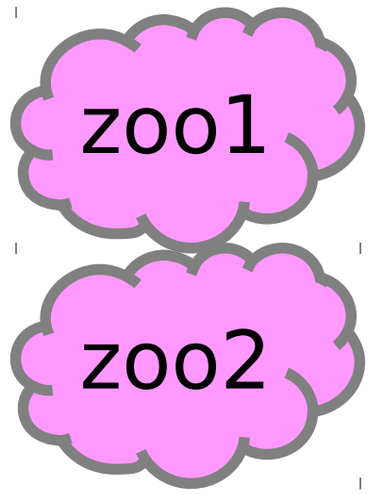
Repeat for kermit ...
24h later: done

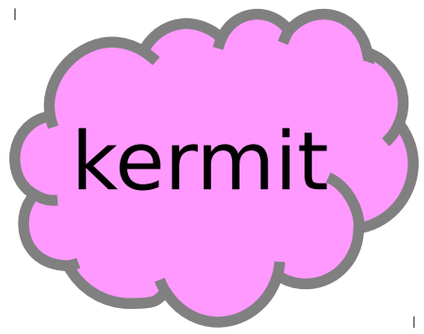


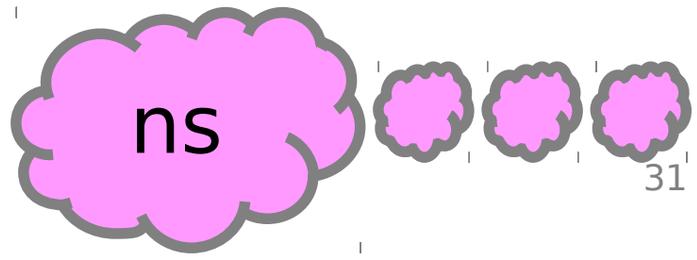
dCache upgrade on VM nodes

Same procedure for dCache upgrades!









Experience

- This is how we have done upgrades of both dCache and OS (including kernel updates with reboots) for the last 6 months
- And we’re running on Ubuntu that releases kernel patches early and often
  - A couple of reboots per month, on average
- No user inconvenience
  - Except possibly for gsi-xrootd users
- No need for planning or scheduling downtime
- 3 headnodes might be nice (quorum for autofailover!)

Experience

- The thing about gsi-xrootd
  - Excluded from the neat diagrams above
  - We use a simple ftp1.ndgf.org name for gsi-xrootd only
  - Repointing DNS (600s TTL) instead of ucarp/haproxy
  - Because clients up until recently(?) used old Globus-style “security” in host certificate verification:
    - Reverse of IP has to match DN, instead of SAN compared to what the user requested to connect to
  - Reportedly fixed in modern xrootd
  - Other workaround would be to share one hostcert all over

Performance regression on HPE raid controller

- When updating the dCache tape pools to a newer Ubuntu version, we discovered a severe performance regression
  - 1.8GB/s -> 0.6GB/s sequential IO
  - https://bugs.launchpad.net/ubuntu/+source/linux/+bug/1668557
- This applies to anything with a change that came with mainline kernel 3.18.22
  - Linux now respects what the driver/HW says for max_sectors_kb
  - But the hpsa cards we have access to say 4096, but perform way worse with max_sectors_kb > ~1k
  - Workaround udev script that sets it to 512 linked in above bug
  - For those of you with RHEL-derivatives, 3.10.0-327.36.3.el7.x86_64 is reported to be the last fast version in CentOS7

Questions?