Pool Party!

Autoscaling on EC2
EC2 for the criminally lazy
autoElastic cloud computing on EC2

Ari Lerner – CitrusByte

Amazon's EC2

What

- Clustered, web-service
- Easy scaled computing
- Completely customizable
- Pay for only what you use
- Highly elastic
- Incredibly flexible

Issues

- No auto-scaling
- No load-balancing
- Non-persistent instance data storage
- Volatile with no failover solution

Pool Party

- Self-healing computing clusters
- Auto-scales cloud based on user-configured parameters
- Inherent load-balancing
- Persistent data storage on instances using s3
- Failover safety
- CloudSpeak DSL – easily communicate with your cloud

How?

Load Balancing

Haproxy

Fail safety

Heartbeat

How?

Persistent data storage

S3F(use)

Auto-scaling

PoolParty

*nix

HAProxy

This application-agnostic, hot-swappable load-balancer built from the ground up for high availability clusters is speedy because of it's closeness to the os.

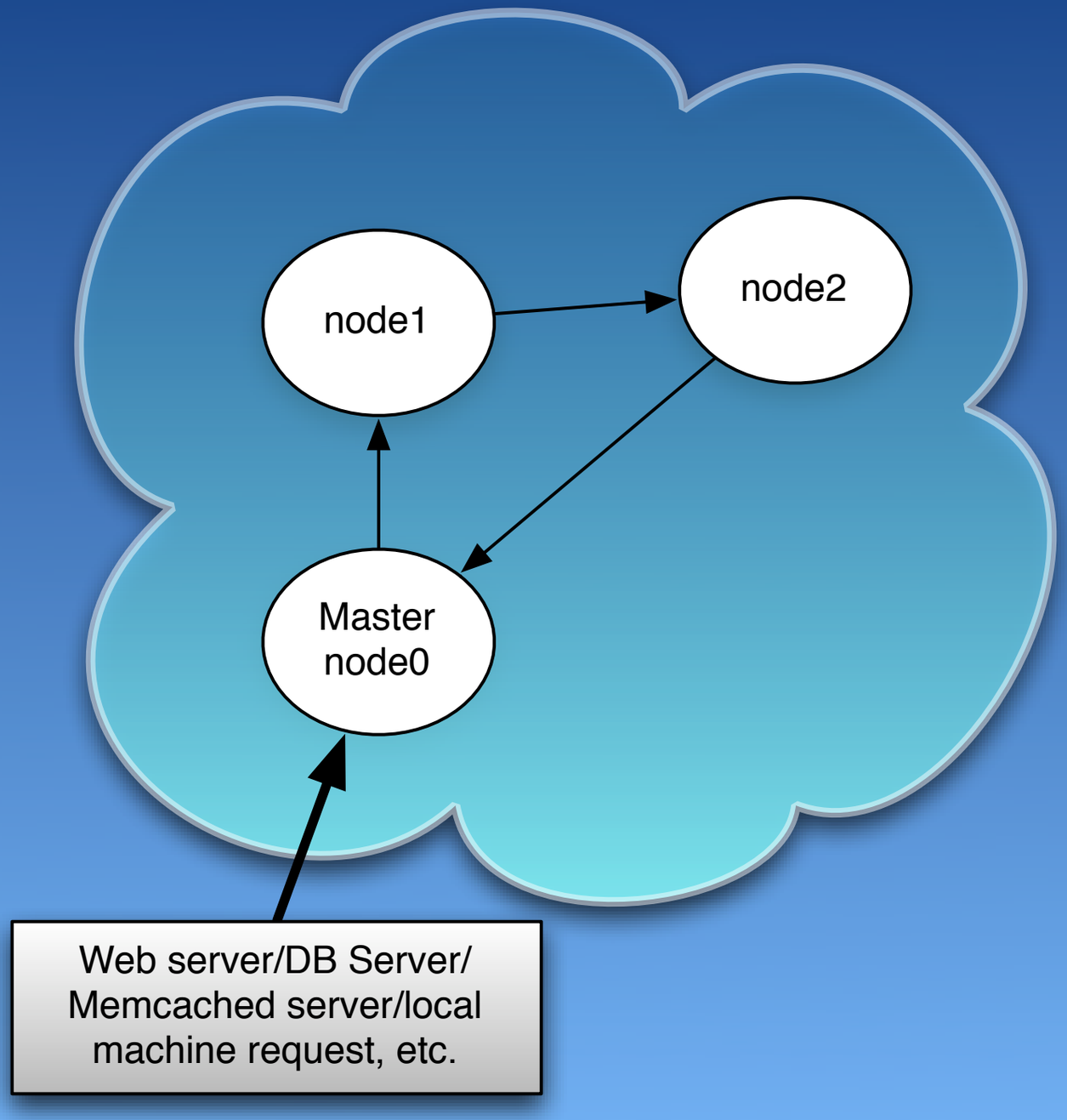
Heartbeat

Robust, enterprise-standard services monitor that provides pool party failover capabilities.

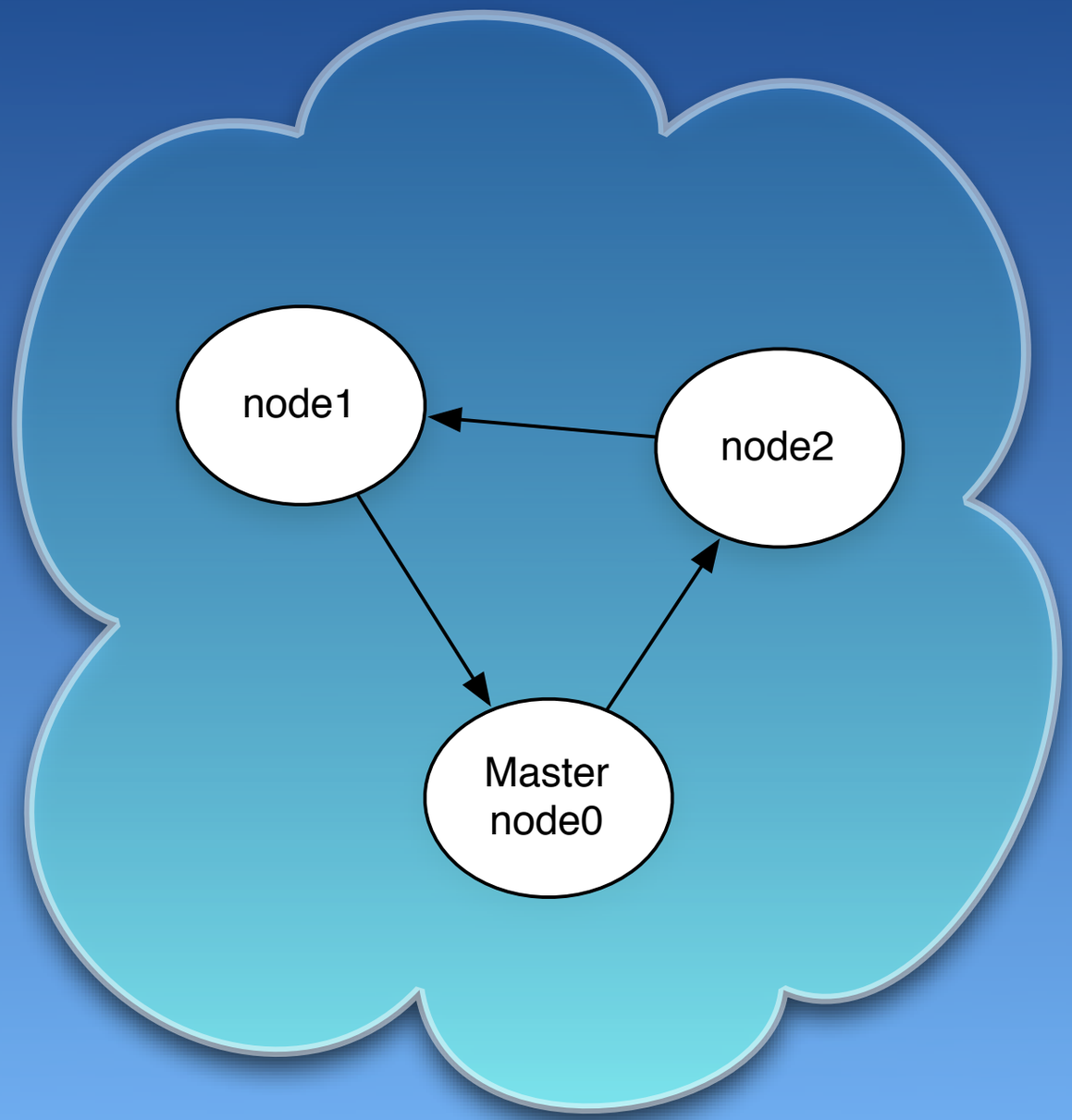
S3Fs

S3 backed filesystem in userspace (FUSE) uses little-to-no caching to reduce race conditions. Pool party uses the unlimited s3 data-store to provide instances access to the same data across the instances while providing an easy deployable.

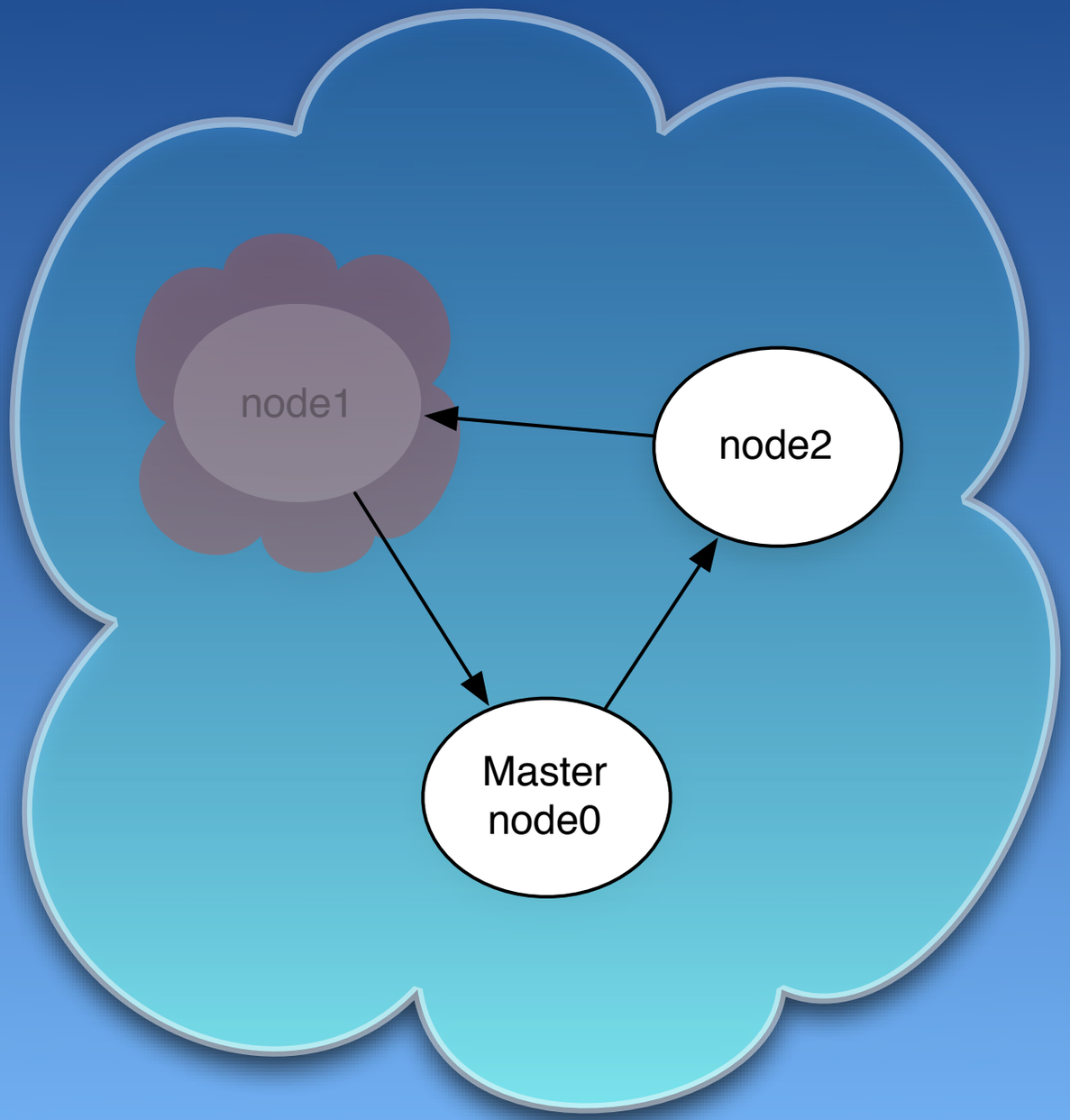
Load Balancing



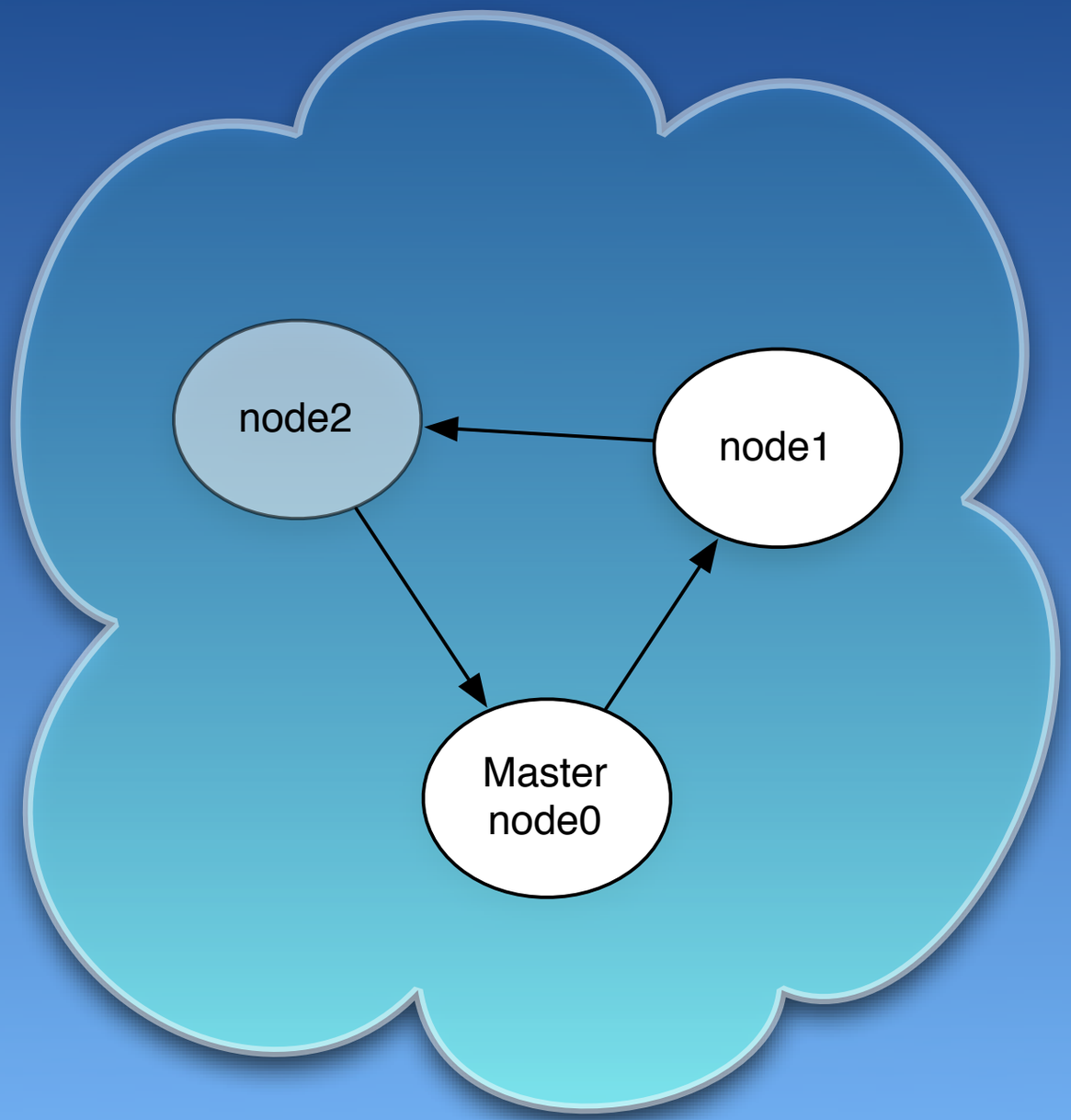
Failover (instance)



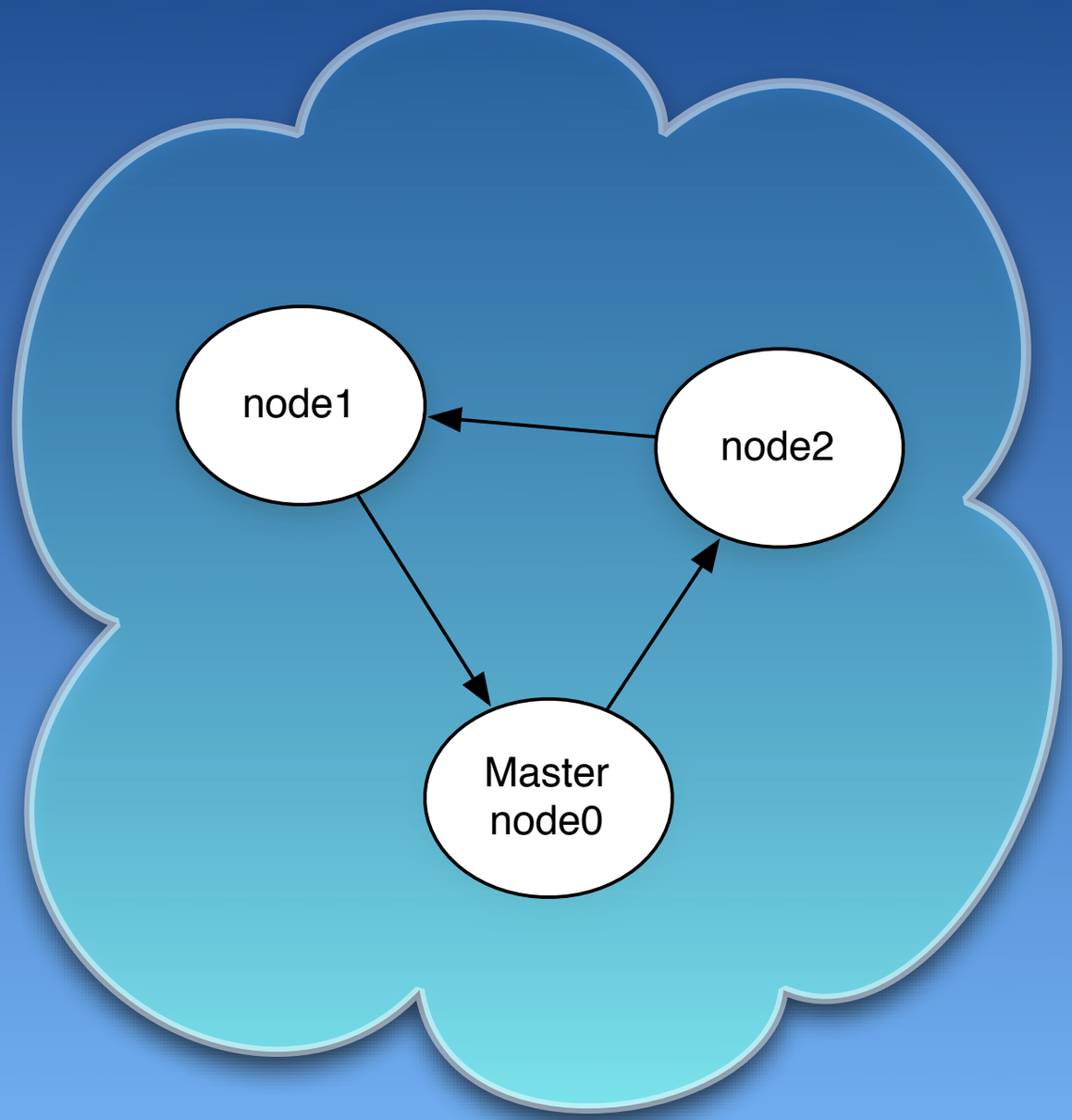
Failover (instance)



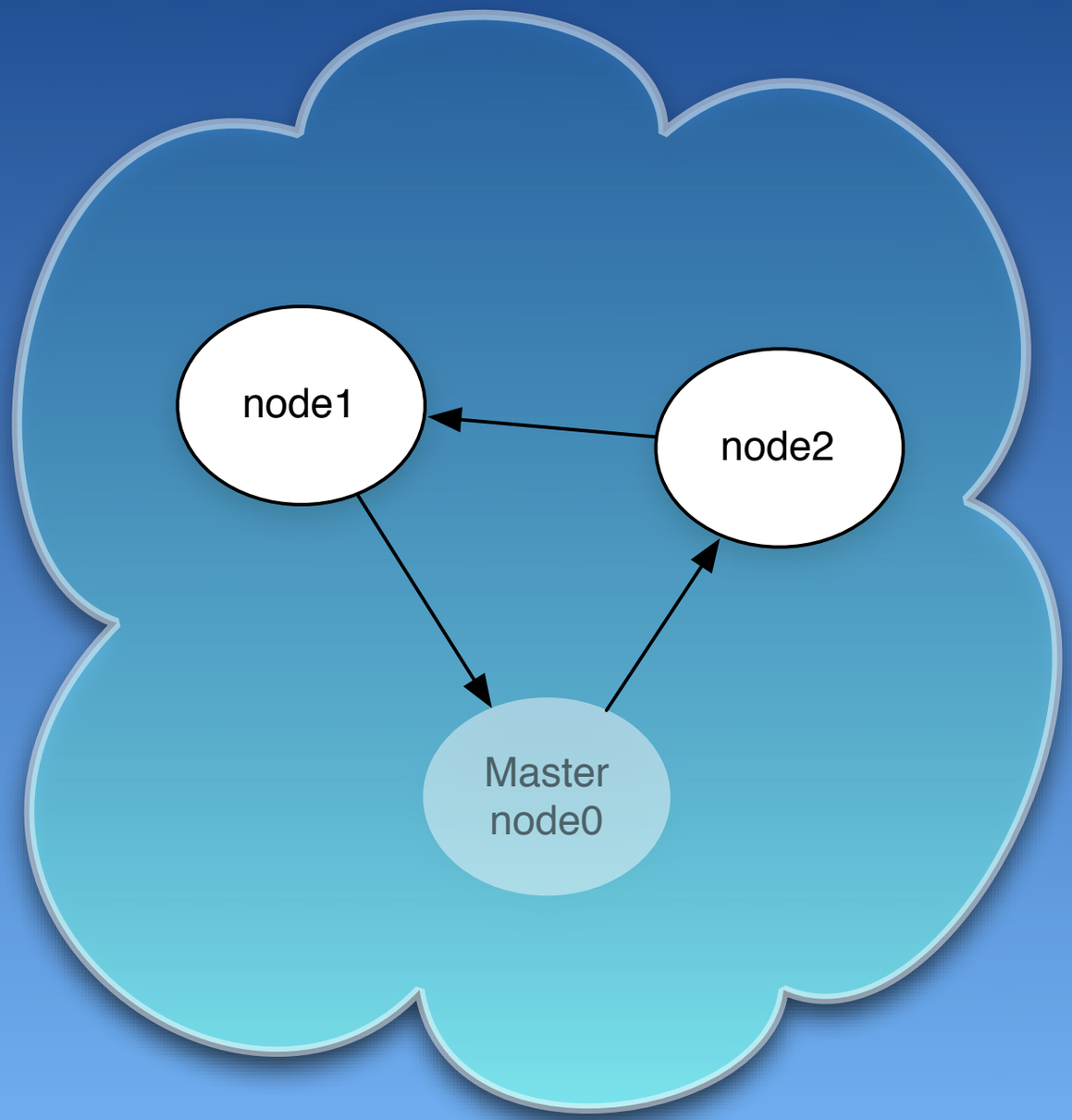
Failover (instance)



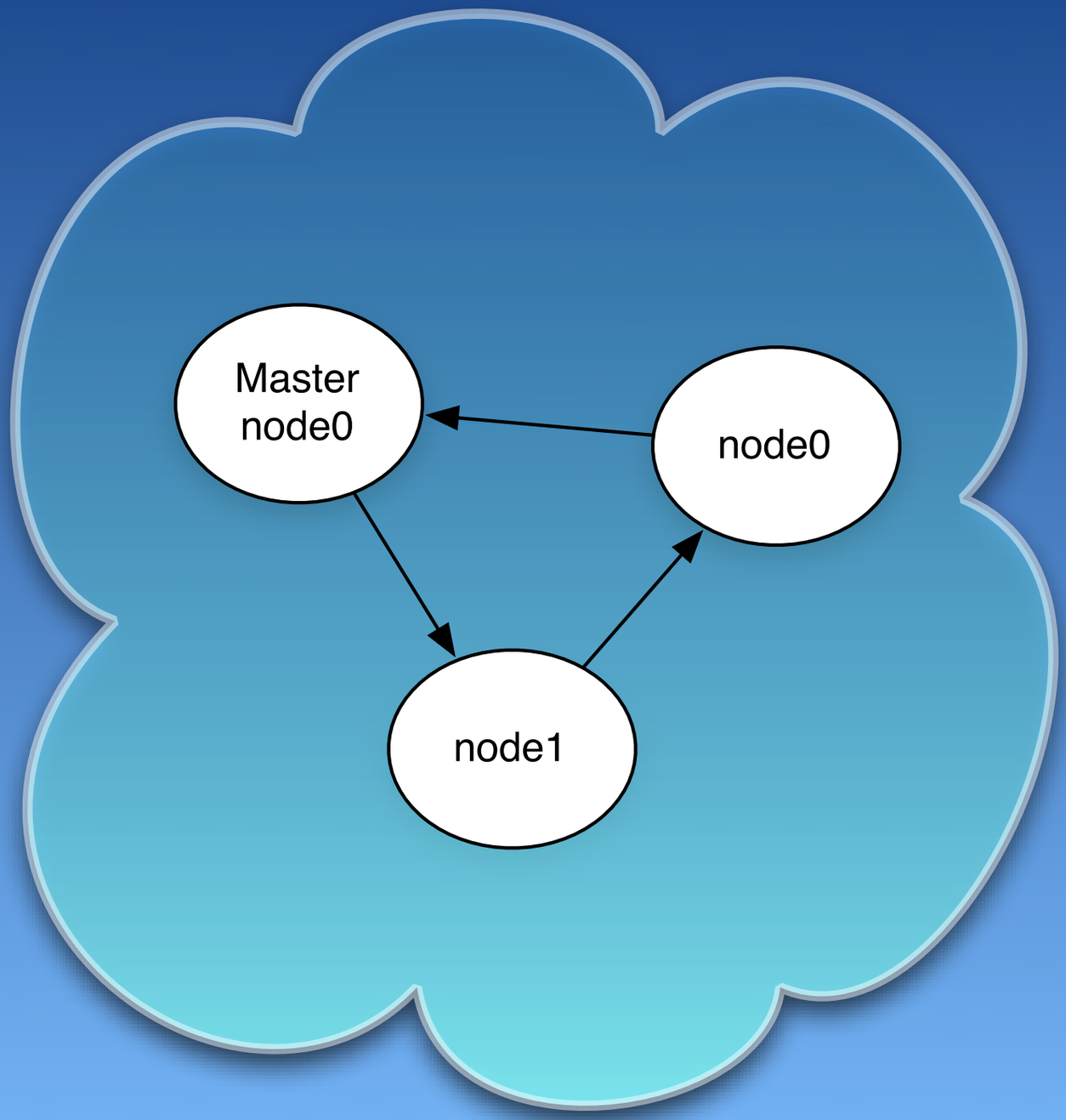
Failover (master)



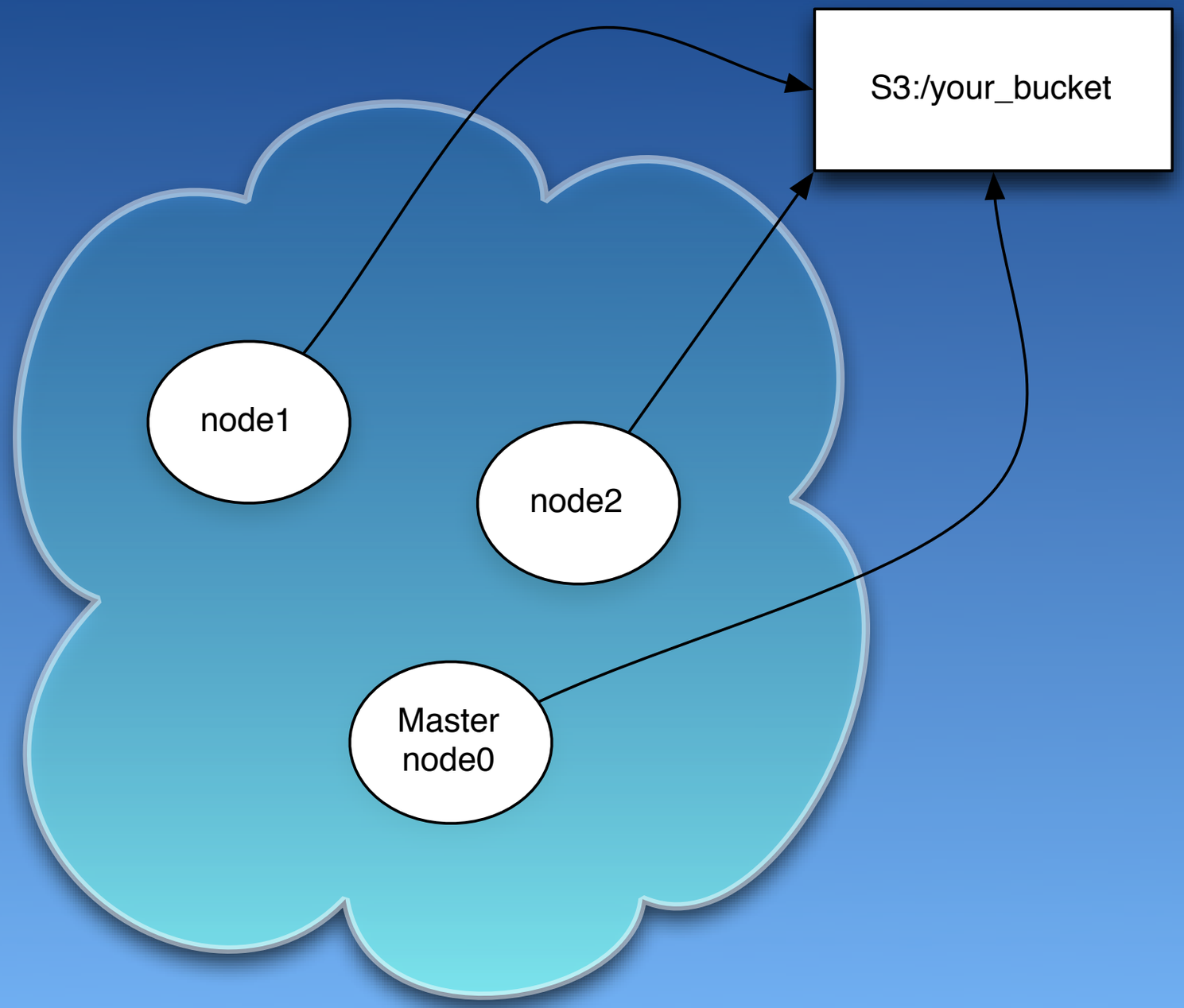
Failover (master)



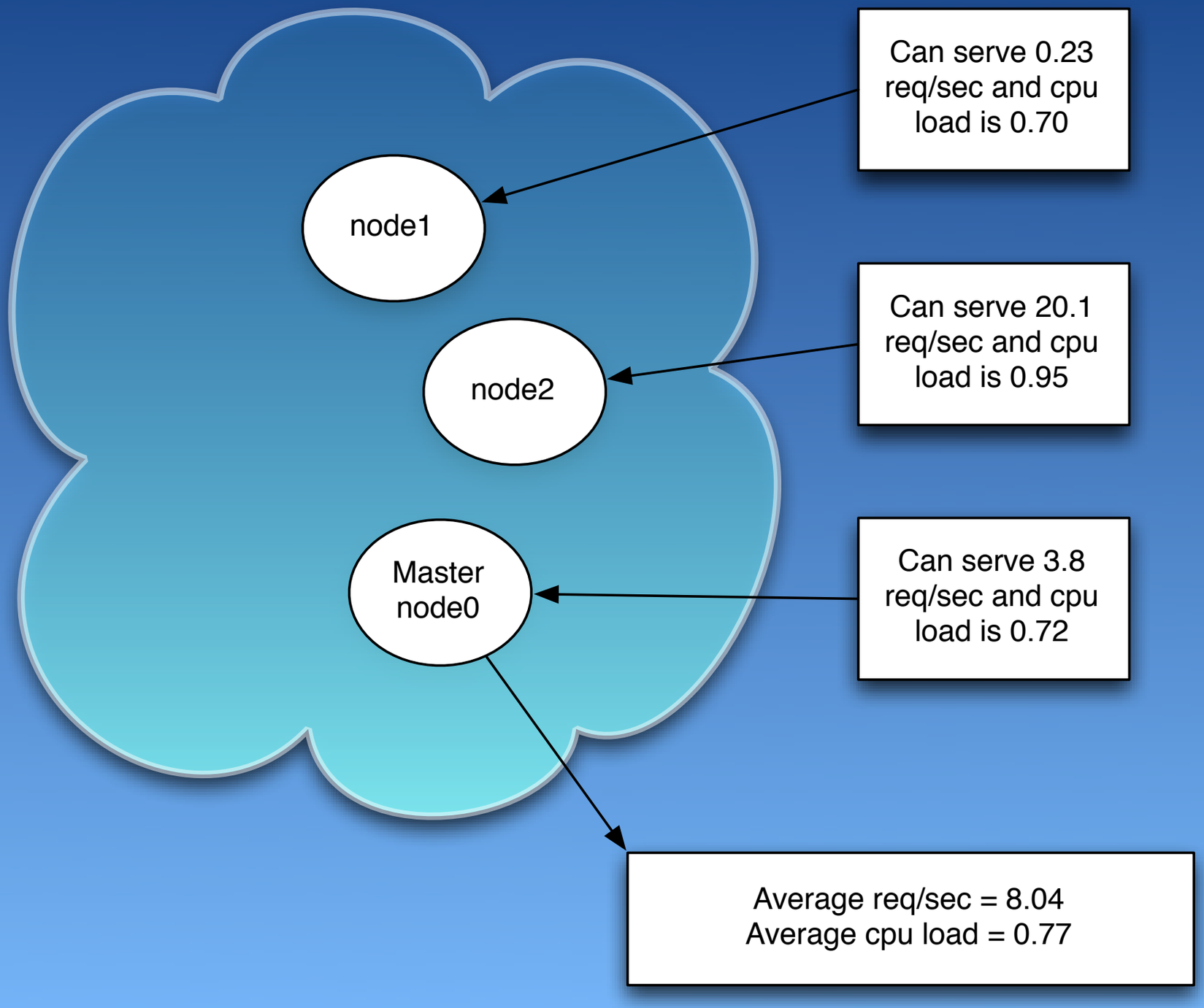
Failover (master)



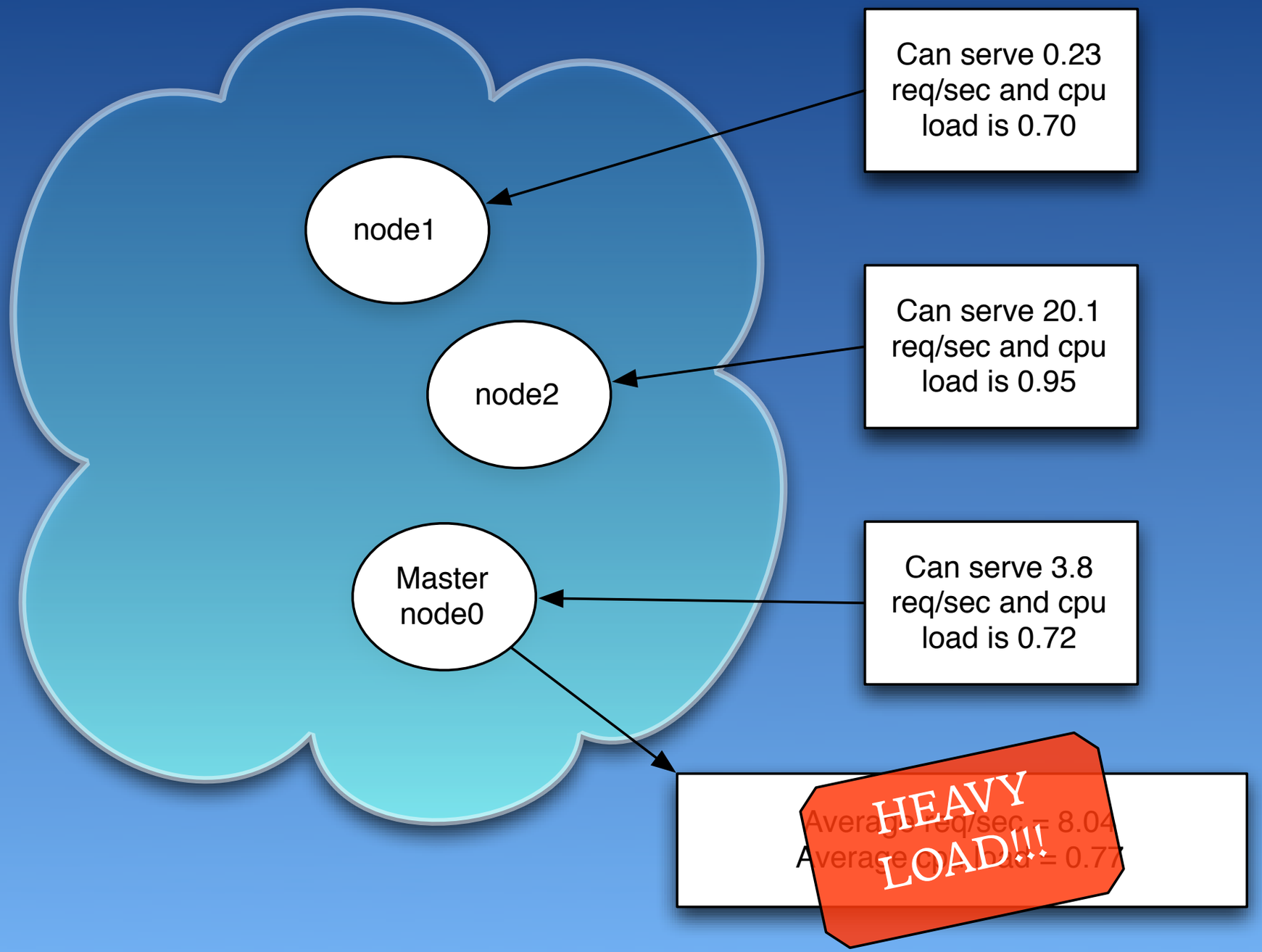
Persistent data store



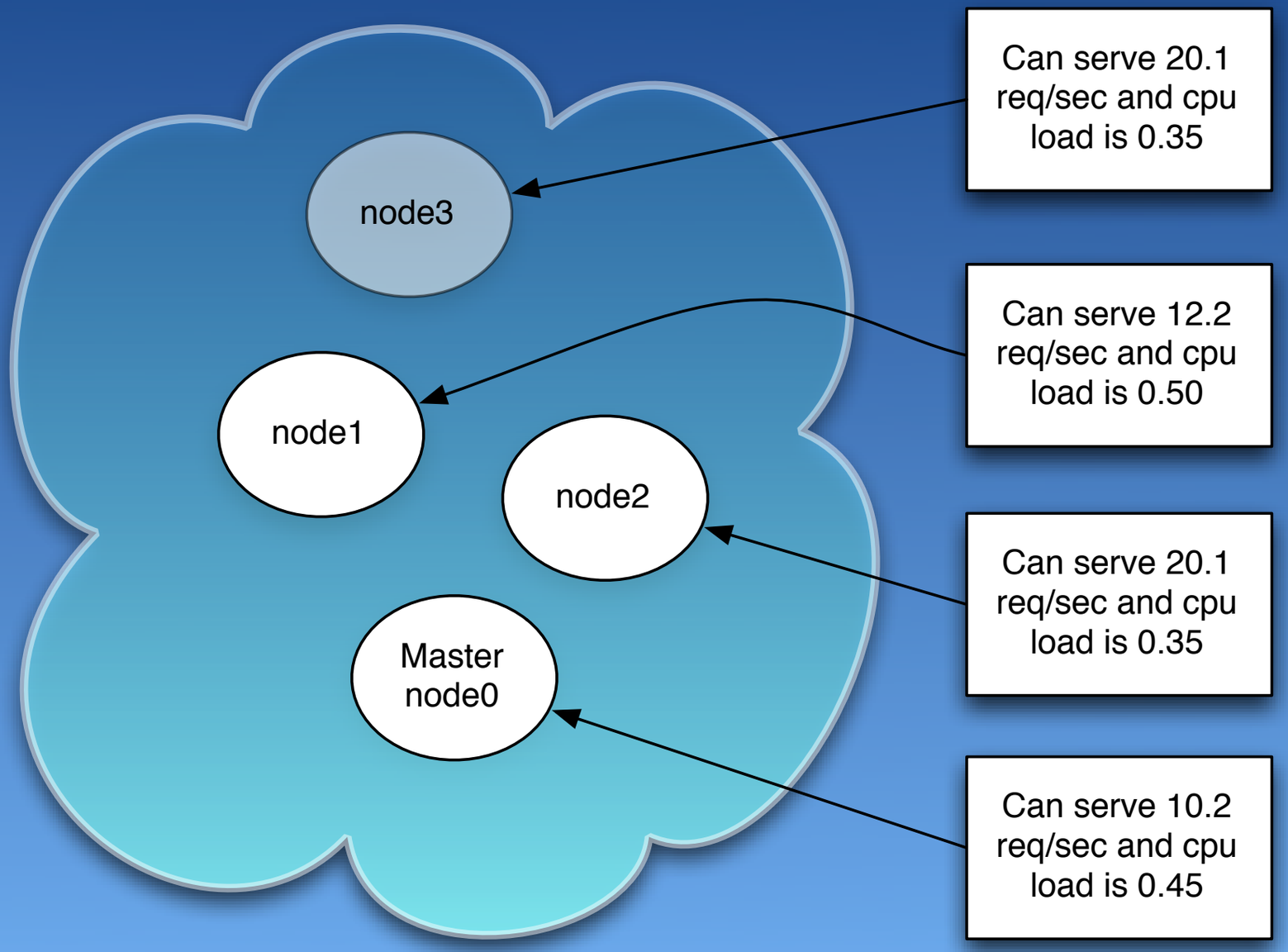
Auto-scaling



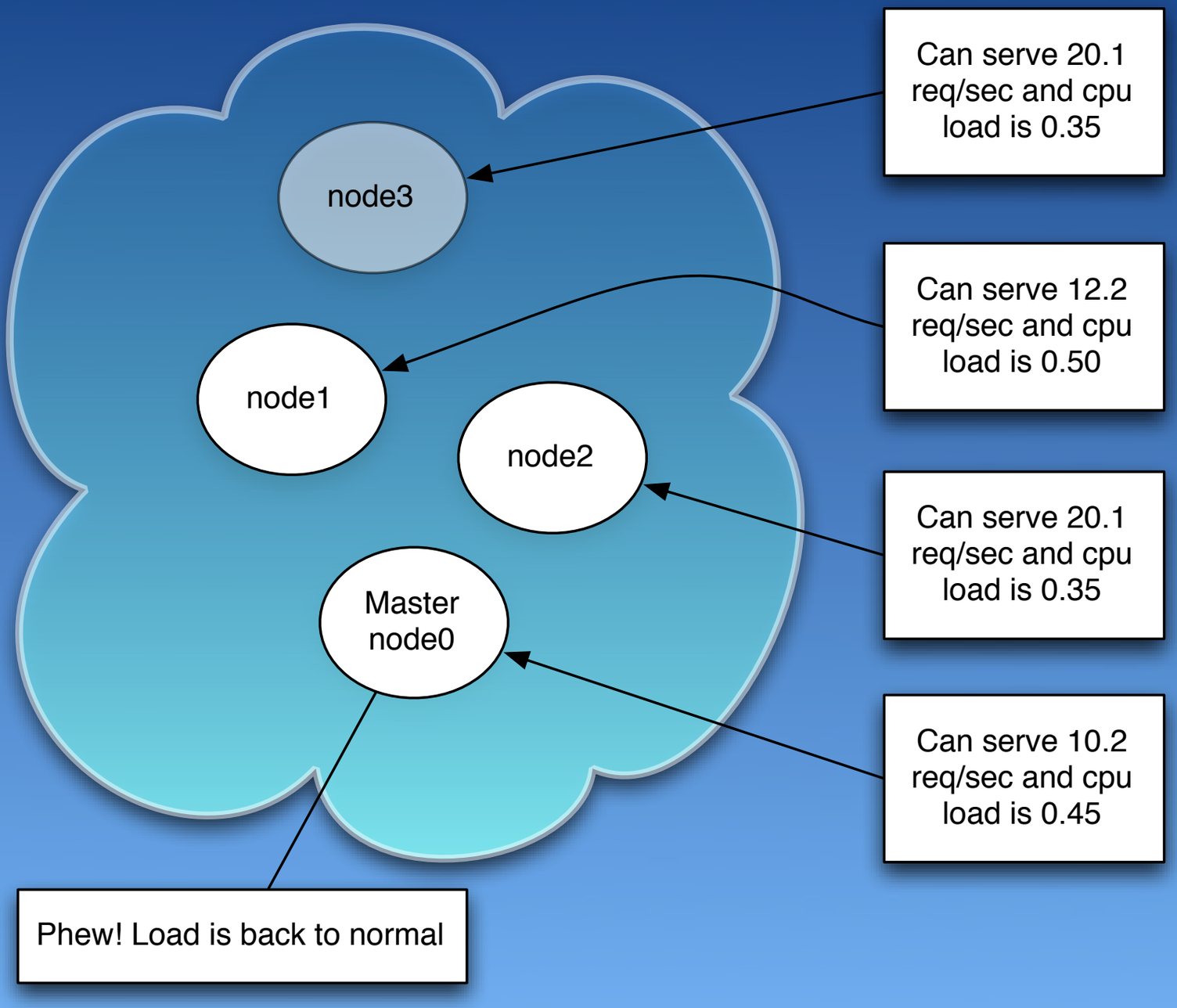
Auto-scaling



Auto-scaling



Auto-scaling

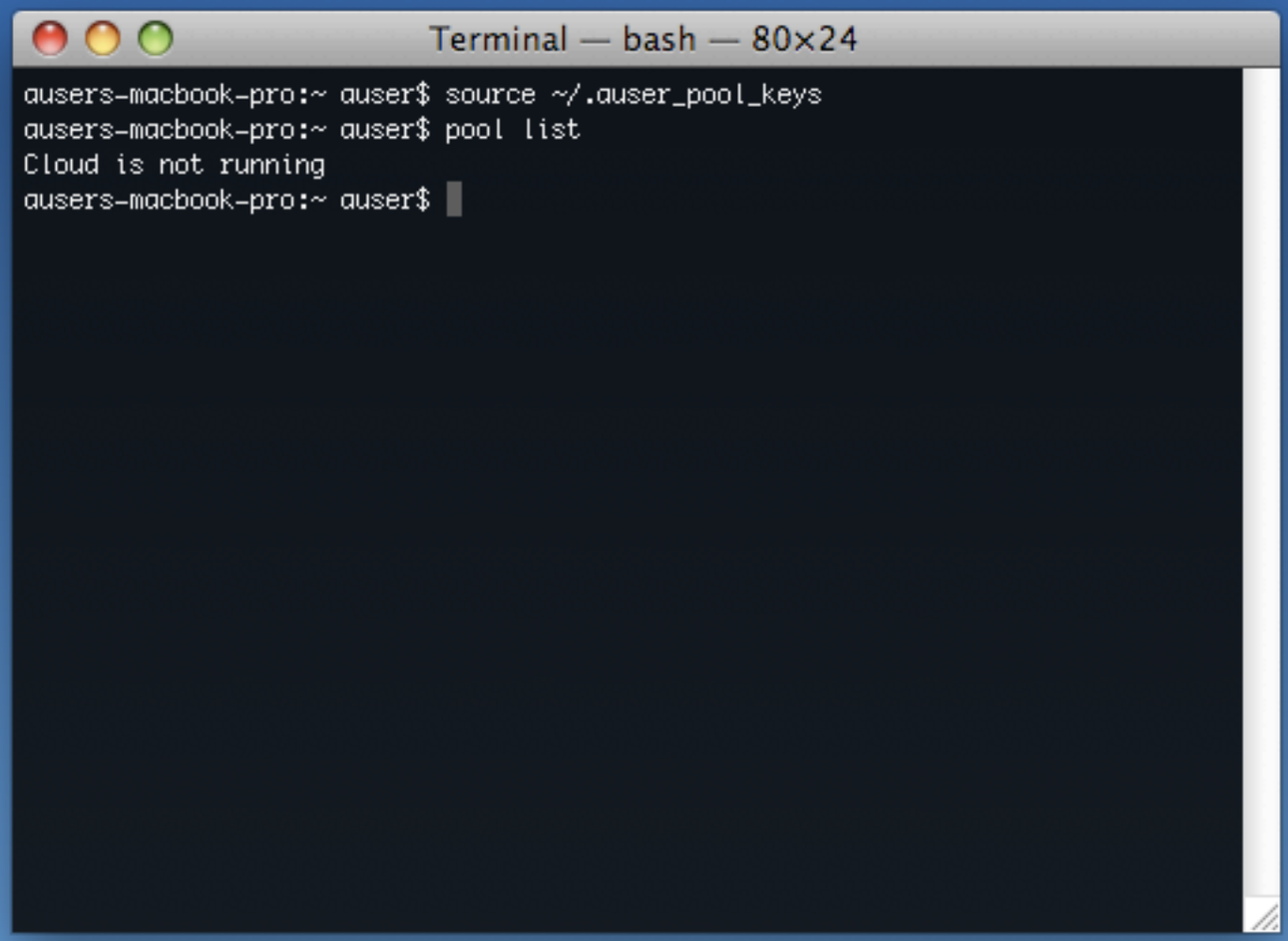


CloudSpeak

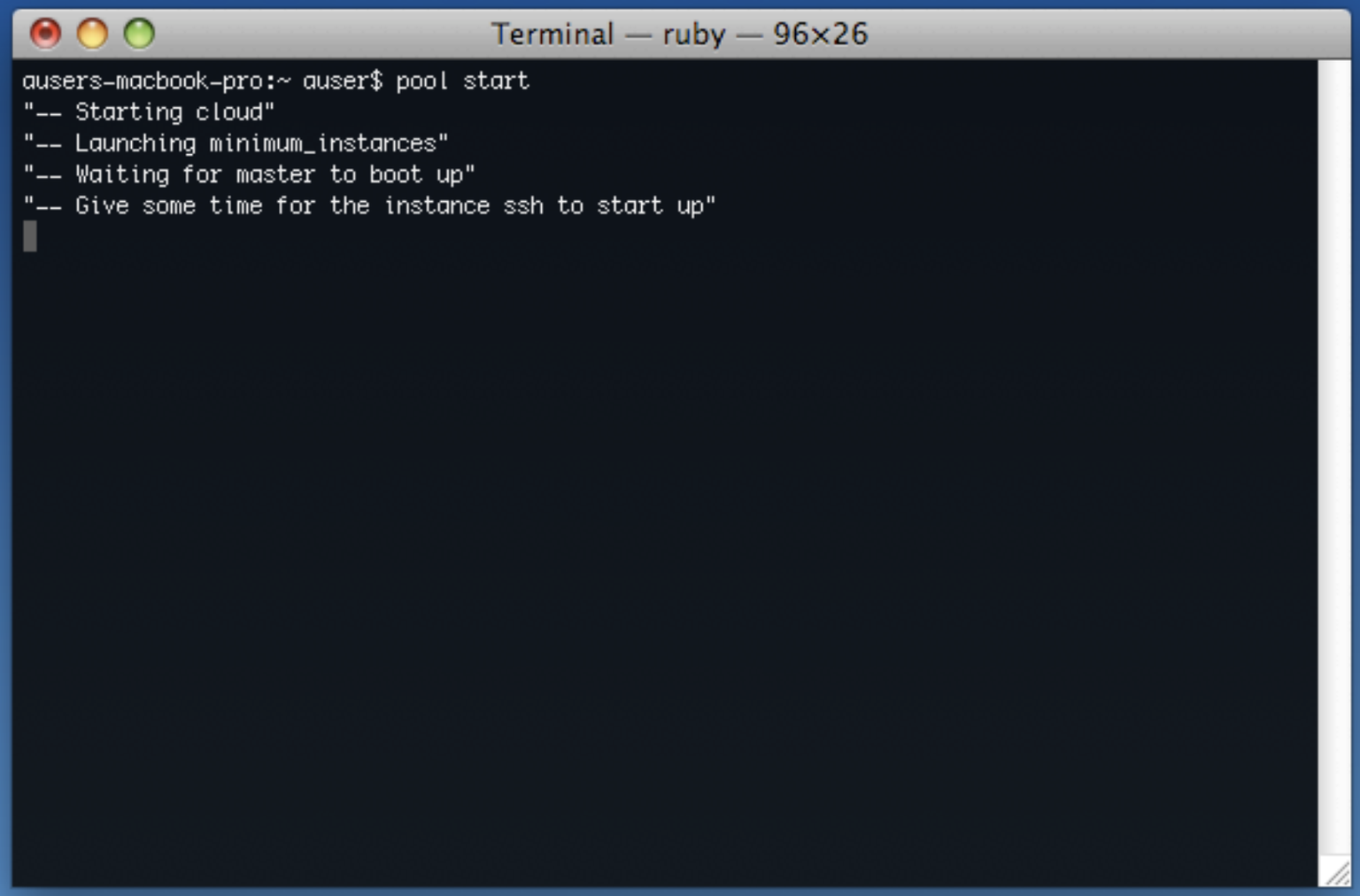
- pool
  - start | stop | restart | list | maintain
- instance
  - start | stop | reconfigure | install | ssh | scp | cmd | start_maintain | stop_maintain

Usage example

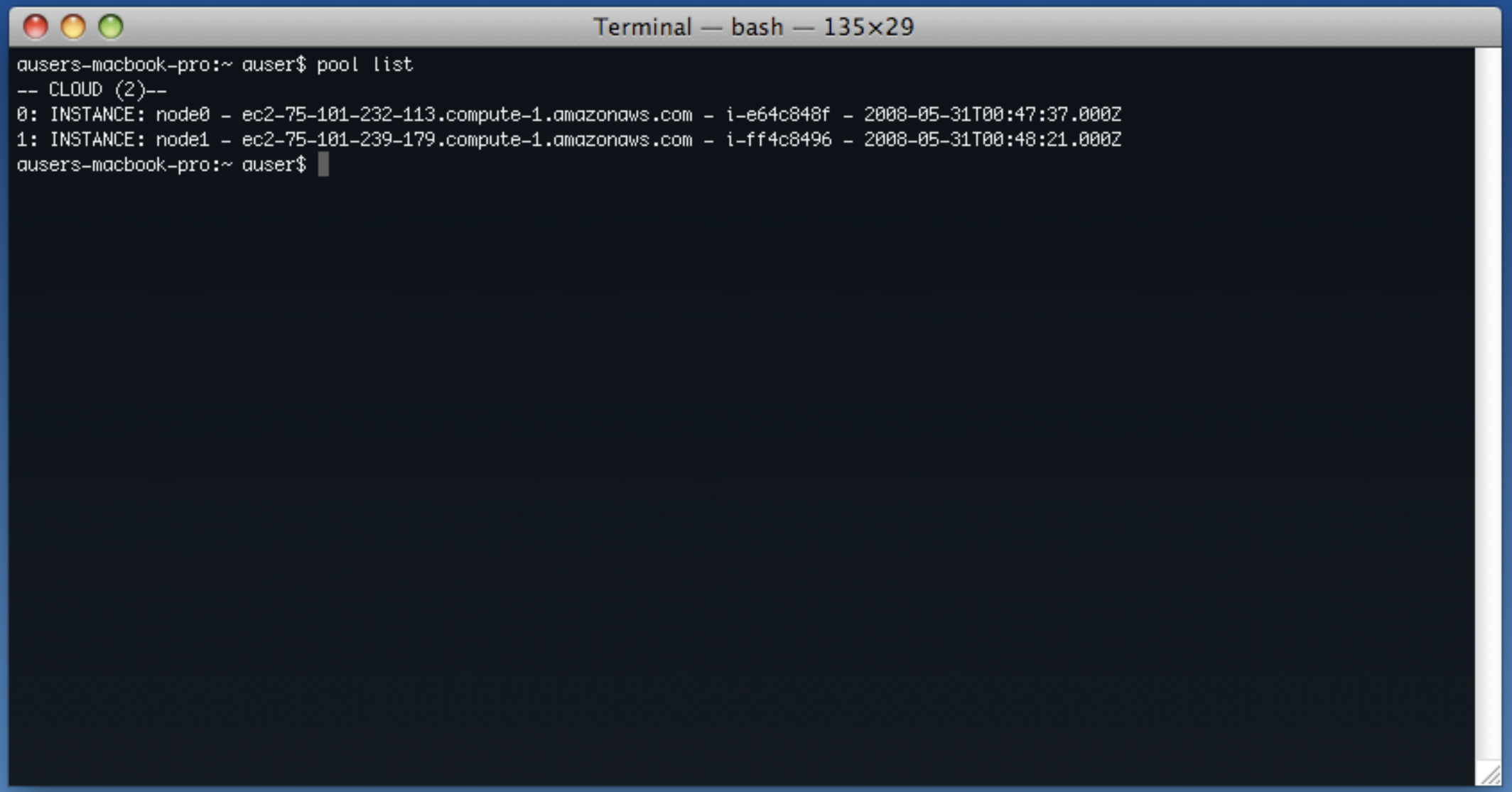
Start with
rake dev:setup
source ~/.key_pool_keys



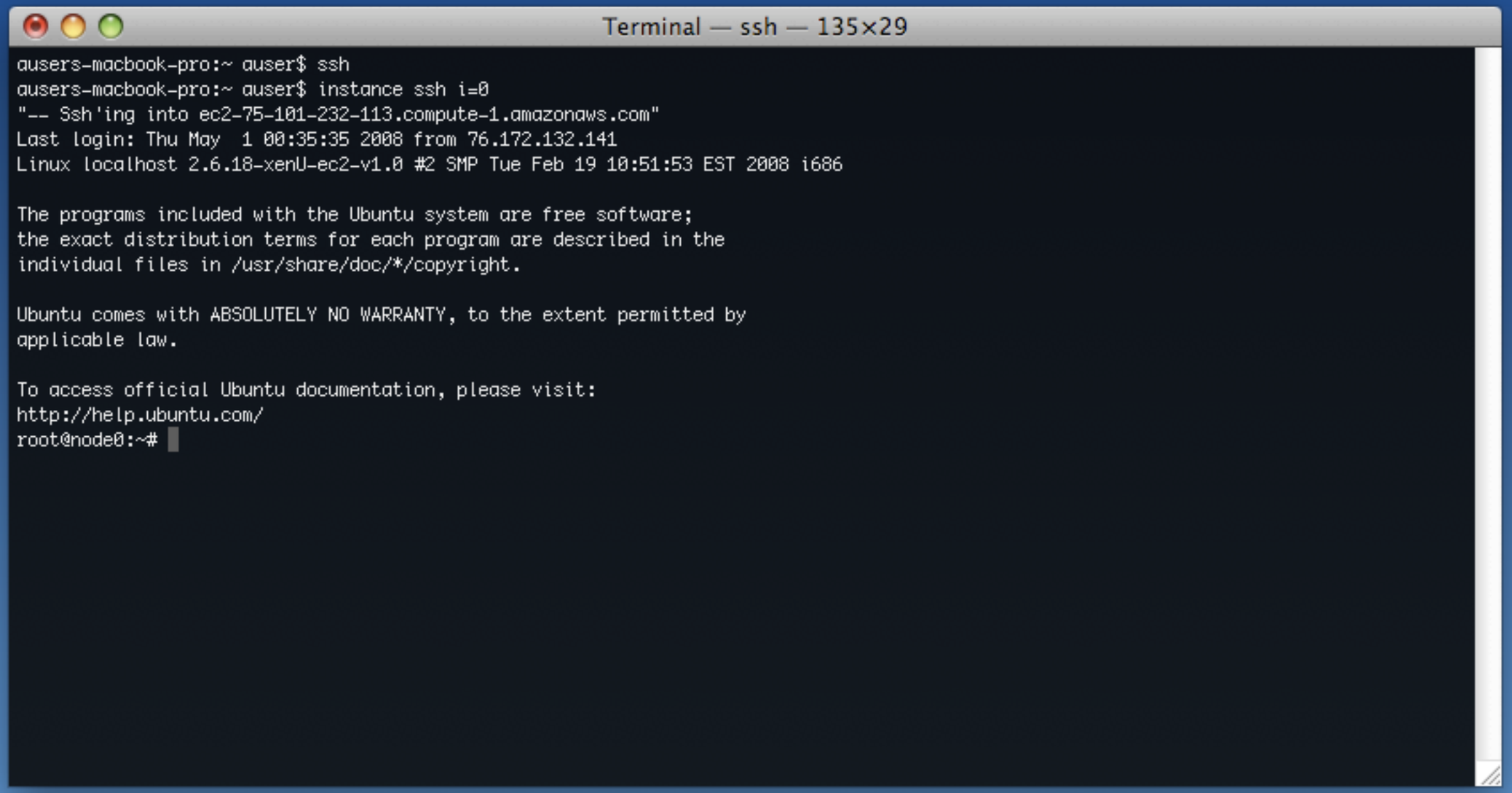
pool start


Terminal — ruby — 96×26
ausers-macbook-pro:~ auser$ pool start
"-- Starting cloud"
"-- Launching minimum_instances"
"-- Waiting for master to boot up"
"-- Give some time for the instance ssh to start up"

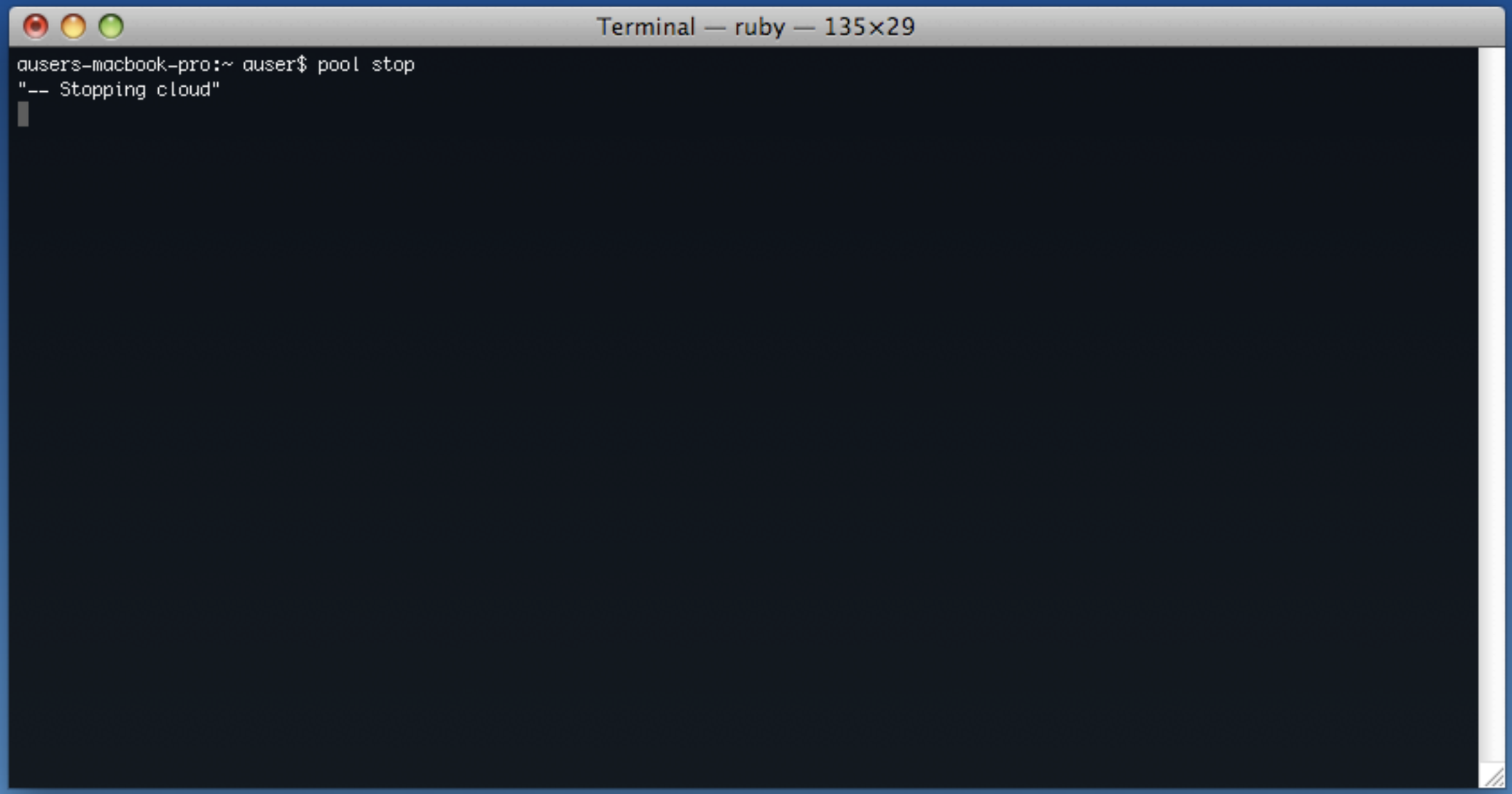
pool list


Terminal — bash — 135×29
ausers-macbook-pro:~ auser$ pool list
-- CLOUD (2)--
0: INSTANCE: node0 - ec2-75-101-232-113.compute-1.amazonaws.com - i-e64c848f - 2008-05-31T00:47:37.000Z
1: INSTANCE: node1 - ec2-75-101-239-179.compute-1.amazonaws.com - i-ff4c8496 - 2008-05-31T00:48:21.000Z
ausers-macbook-pro:~ auser$

instance ssh



pool stop


Terminal — ruby — 135×29
ausers-macbook-pro:~ auser$ pool stop
"-- Stopping cloud"

Upcoming features

in development

- Speeding up configuration-time
- Plugin and extensible support
- Front-end client
- Monitor logging
- Capistrano deployment

Questions?

Resources

- http://poolpartyrb.com
- http://blog.citrusbyte.com
- http://aws.amazon.com/
- mailto: ari.lerner@citrusbyte.com

Thanks