

Prerequisites for the Cloud

David Nalley

David.Nalley@cloud.com

david@gnsa.us

david@cloudstack.org

[@ke4qqq](#) on Twitter/[identi.ca](#)

whoami

- Recovering sysadmin
- Involved in a few open source projects
- Community Manager for a F/LOSS IaaS project

Agenda

- Tools
- Break
- The rest of the stuff you need

The promise of the cloud



Observations from the Clouds

- Everyone wants it
- Few know what it is
- Even PHBs are deciding that it's a must have
- Few realize the level of change that it requires
- Failure is everywhere
- Success is possible, if a bit elusive, but when success occurs it's phenomenal.

The reality

- Cloud computing is one tool
- When used properly it can help enable awesomeness
- When used in isolation it is full of fail.

Before we dig too deep: What is 'cloud'

- On-Demand Self-Service
- Broad Network Access
- Resource Pooling
- Rapid Elasticity
- Measured Service

Better cloud definition:

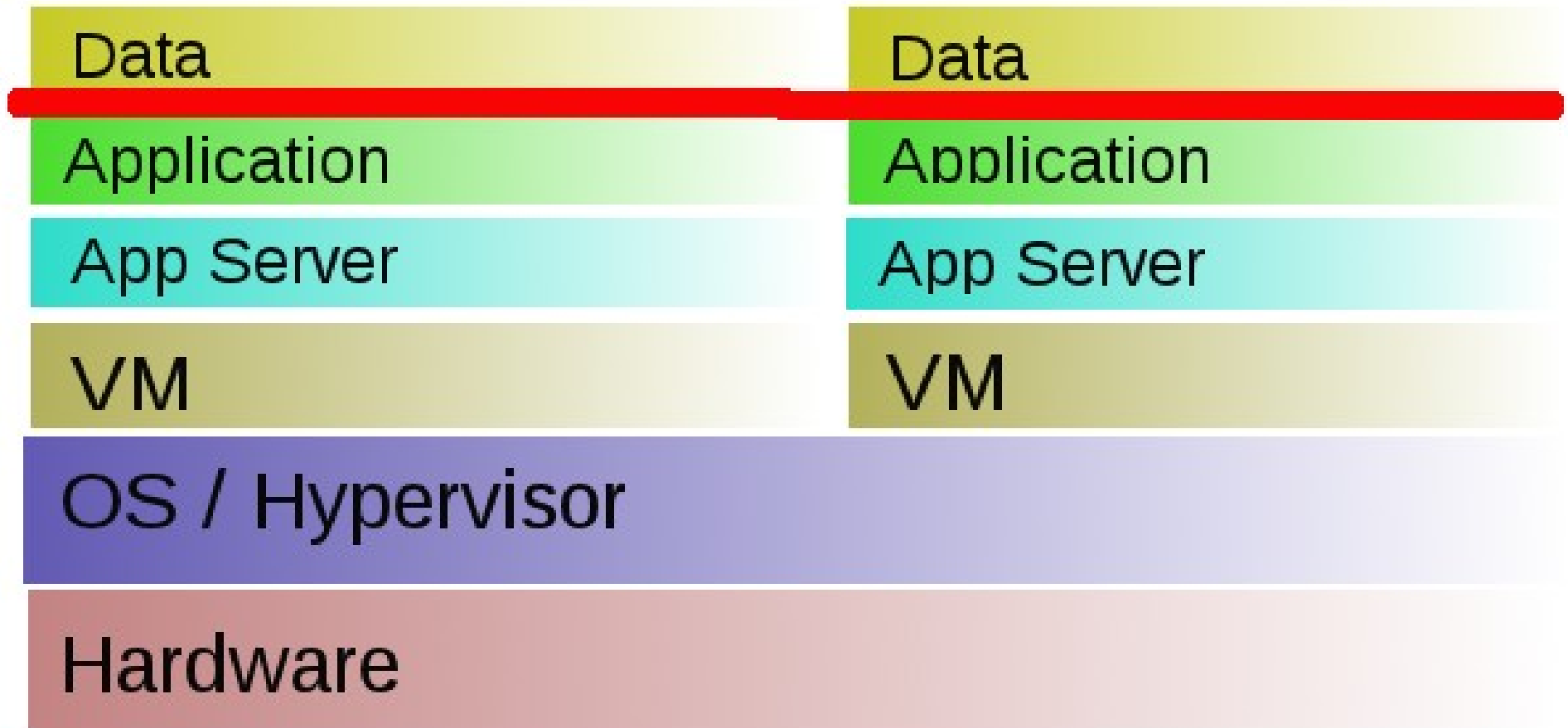
The cloud is AWESOME:

- O – On-demand
- S – Self-Service
- S – Scalable
- M – Measurable

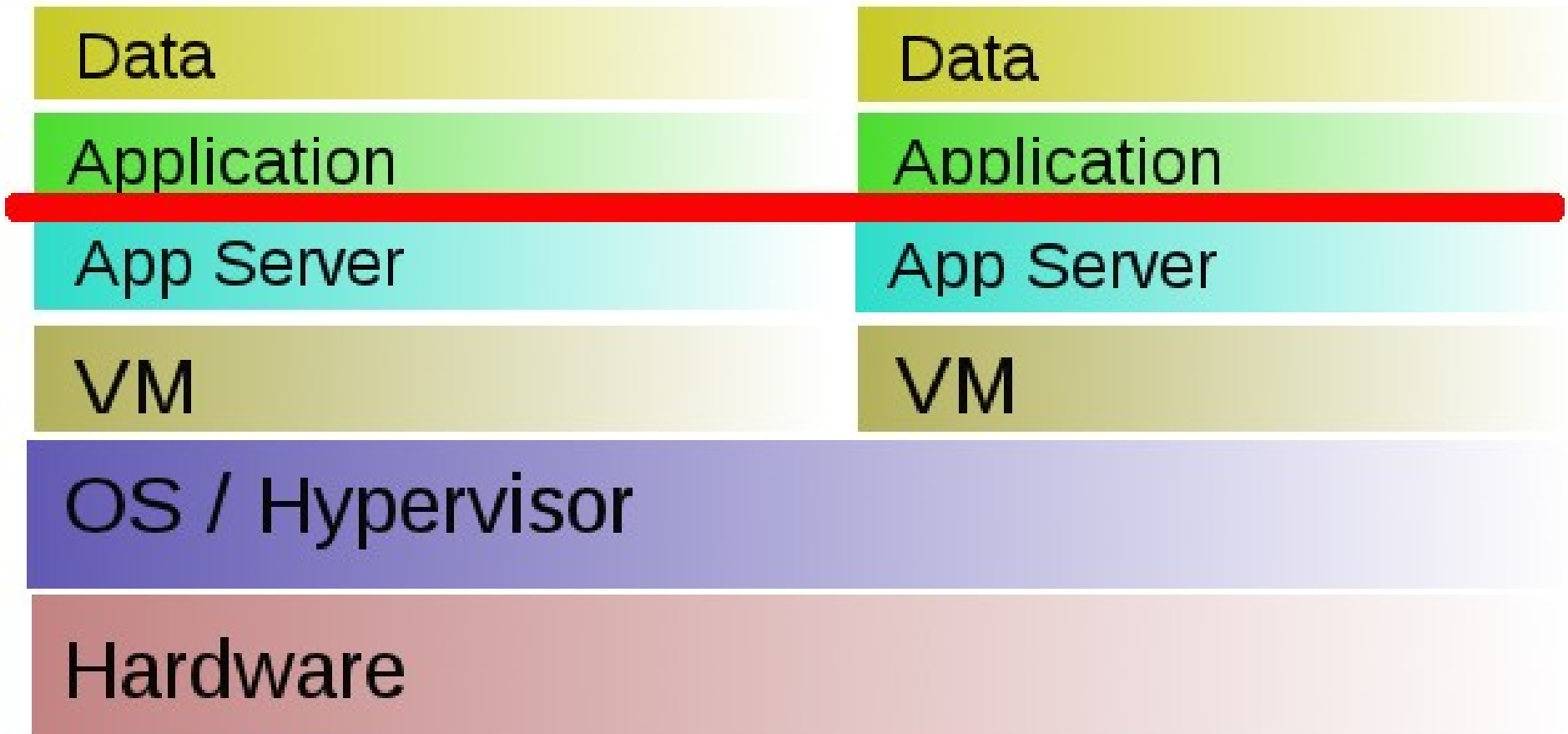
Types of Clouds:

- Software as a Service
- Platform as a Service
- Infrastructure as a Service
 - Storage
 - Networking
 - Compute

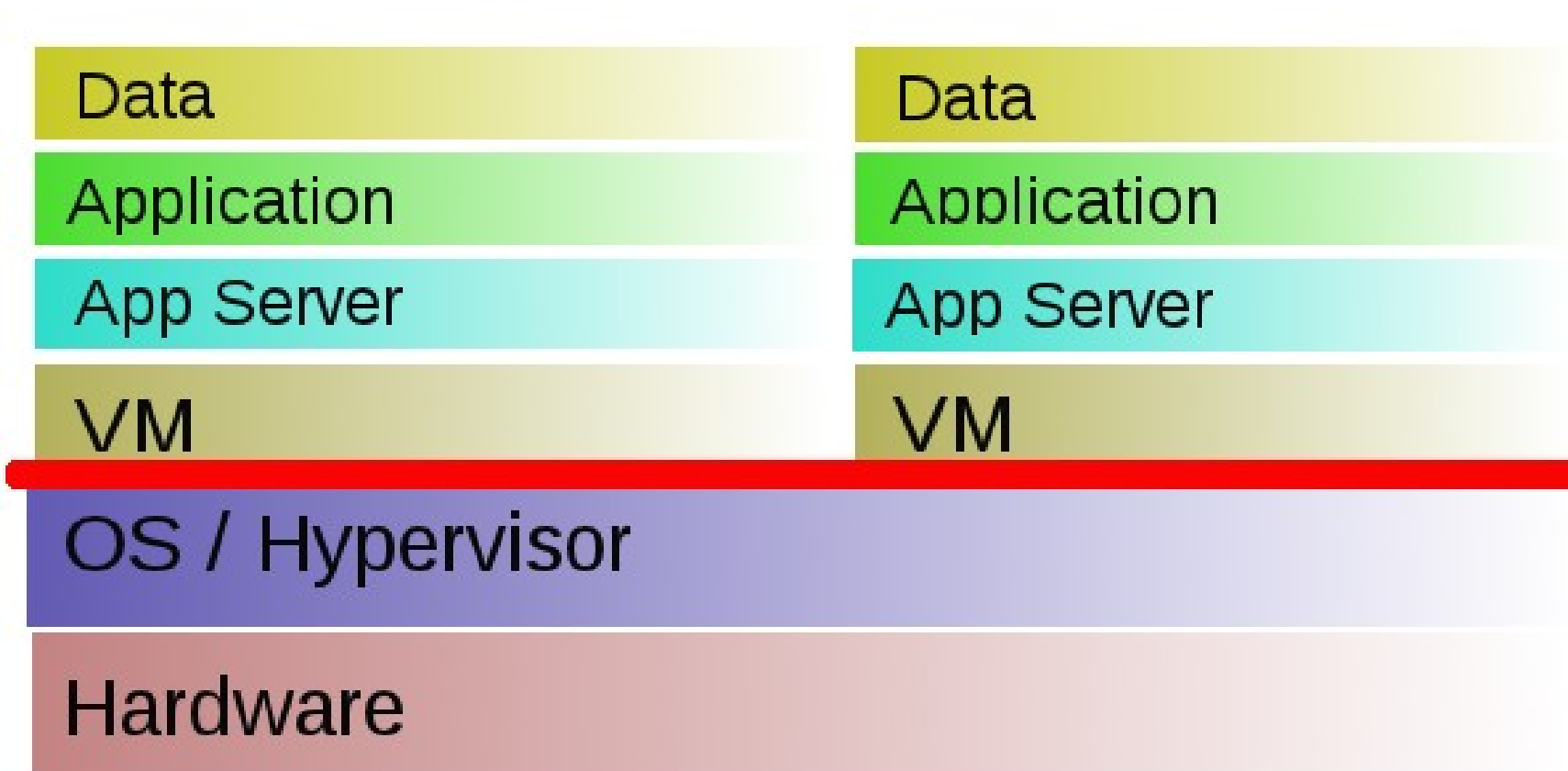
SaaS



PaaS



IaaS



Any questions on what a 'cloud' is?

Cloud platforms are good at:

- Self Service
- Measuring
- Scaling

So what's needed for their care and feeding

- Automated Provisioning
- Configuration Management
- Monitoring
- Orchestration

Great, we've got all (or most) of those

Congratulations you are better off than most

Sadly the demands that most clouds impose means that using tools the way most use them just isn't going to work 'in the cloud'

This isn't advocating switching tools necessarily, but it is talking about making sure your tools can demand with on-demand, self-service, rapidly scaling features.

Automated Provisioning

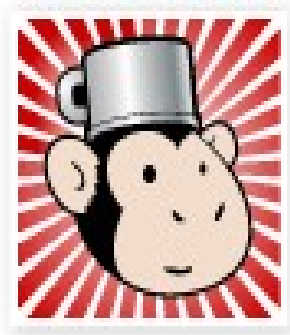
- Everyone knows you should have Automated Provisioning, right?
- Everyone has, or should have, some variant of PXE booting (either custom written, FAI, Cobbler, etc.)
- Many cloud platforms don't have facilities for PXE booting.
- There are different disk image types. (you weren't putting all of your eggs in one basket were you?)

Automated provisioning

- The cloud variant of this problem is a bit different and in some ways more complicated:
 - Multiple disk formats
 - Multiple potential destinations
 - Multiple potential cloud platforms
- Too many providers and platforms want you to use 'templates' or master images.

Automated Provisioning

- Subliminal advertising:



BoxGrinder

Don't forget to demo boxgrinder

Configuration Management

- Problem is only slightly more complicated:
 - How do you add a node?
 - How do you ensure that ONLY your nodes are getting access to config management?
 - How do you clean up all of that data?
 - Lots of folks already using ENC – but are they cleaning the data?

Configuration Management

- Purely my opinion – but if you are serious about approaching a cloud project, you need puppet or chef. The other config management tools are less suited for cloud deployment IMO. I'll go further and say, I don't personally know of any non-service provider, large scale cloud deployment that has been successful without chef or puppet.

Monitoring

- Monitoring is one of the areas that significantly lags behind in a cloud environment.
- The problem:
 - Machines come and go, sometimes hourly.
 - How does your monitoring system deal with machines that go away?
 - What happens if the type of machine changes – you were monitoring a database server one day – tomorrow it's a web server?
 - How do you keep that historical data?

Monitoring

- How do you figure out what to aggregate??
Especially when it changes?
- Logging is its own nightmare – what do you keep after the machine goes away – how do you ensure that you are seeing all of the webserver logs – even ENCs fail a bit here – how a machine is classified today may be different than tomorrow.

Orchestration

- You have all of this 'power' what are you going to do now?
- Now that you can spin up 500 machines in 5 minutes; the idea of taking months to do an application deploy seems pointless, and a waste of time.
- Subliminal message: DevOps is the future

Orchestration

- Orchestration enables you to programmatically reach out and make machines work.
- Tools:
 - Func
 - Juju
 - Mcollective
 - Run Deck
 - Control Tier

Orchestration problems

- How do you know what machines to run against? DNS? ENC?
- How are newly added or newly destroyed machines handled?

We've talked about the tools

The way of thinking

- Don't get locked into the way of currently doing things
 - VLANs versus Security Groups
 - Failure becomes the expectation, and acceptable – 'Failure is assured'
 - Smaller changes faster
 - Lean/Agile versus ITIL
- Subliminal message: DevOps is the future

Now comes the really difficult part

You have the tools, and people say they want 'cloud' and don't realize it requires cultural change

Cloud Culture

A culture where individual contributors are empowered to get things done, the tools around self-service, and on-demand scalability mean that many feel a loss of control – A culture that goes from 8 weeks with various levels of permissions to provision a machine to 8 minutes all decided by a single individual is quite a difference.

Cloud Culture

- Means moving to faster application deployments, more rigorous and automated testing (and rollbacks)
- Going to the cloud doesn't mean going sloppy or cowboy – if done right it should be done with even more control, just not with the same illusion of control that exists today.

Self-service

- This is one of the first things that people run into. When folks that aren't in 'operations' have the ability to provision hosts, set up load balancers, open up firewall ports – it can be quite scary.

Questions

Contact

- Twitter/identi.ca @ke4qqq
- Email: david@cloudstack.org
- IRC: ke4qqq on freenode