

The evolution of Paas

paas : plateform as a service

what is cloud ?

- what you care
- self-managed
- self provisioned
- metered and paid per use
- you don't want to care about vm's, component versions

what do i care about (as a dev)

- my code running on a virtual app server (not vm)
 - not just code
 - queues
 - databases
 - logs
 - portals
 - not just runtime
 - CI
 - VCS
 - governance of pf
 - development enviroment

cloud native

- distributed/Dynamically wired
- elastic
 - scales up and down as needed
- Multi-tenant
 - works with underlying laaS
 - cost when you use it
- self-service
 - de-centralized creation and mgmt of tenants
 - automated governance across tenants
- incrementally deployed and tested
- availability

open paas

- key concern
 - lock-in is a huge concern
- should mean
 - to run in different places
 - to run on different laaS infrastructure
 - to run different types of applications
 - not just java or ruby
 - open source ?
 - to new services
 - not just queues...

dealing with multi-tenancy

- stratos
 - opensource
 - apache license
 - http://stratoslive.wso2.com
 - based on osgi
- separate
 - hardware
 - VM's
 - containers
 - zones
- diff techniques for diff systems
 - java : thread isolation, shared classloaders
 - linux : LXC, chroot
 - mysql : create a DB / tenant
 - cassandra : keypace isolation

layers

- software aas
- pf aas
- infrastructure aas

paths between with

- dev/ops
- customer pf mgmt

types

- public
 - force.com
 - heroku
- private / public paas
- open private / public paas

economy

- based on the Central Limit Theorem
- private cloud : does not work well
 - mst be multi tenancy

different kinds of paas

WSO2

Paul Fremantle

- application
 - Google
 - Heroku
- integration
 - stratos AS
 - Mule Ion
- process
 - Stratos ESB
 - ArisOnline
- messaging
 - stratos BPS
 - StormMQ
- development
 - Stratos MB
 - CloudBees
- complete paas
 - I dont want to know

not the end

dev has not mad its industrial revolution