

Off the Shelf Capabilities

github.com/rkhatibi/devops2012



Ramin Khatibi

- Sys Admin, 16 yrs
- Yahoo, Netzero, several startups
- That company in the lower right
- We do cool stuff with video

SnappyTV™

Today's Premise

- Technology moves forward
- 2012 > 1996
- Expensive integration not required

SnappyTV™

Just Like Everyone Else

- Applications
- Data stores, Provisioning
- Monitoring, Config Mgmt

SnappyTV™

If You Remember One Thing

- Code reuse is good
- Service reuse is better

SnappyTV™

On Demand

- Write very little code
- About 40 hours to tie it together
- Lots of time invested into the various components
- Narrow scope

SnappyTV™

Which Cases Are Easy

- End of day processing
- Heavy resource jobs
- Planned Work

SnappyTV™

Which Cases Are Hard(er)

- Adaptive scaling
- General solutions
- For now

SnappyTV™

APIs Required

- Fix this first
- Code has to talk to something

SnappyTV™

Application Logic

- Ultimately your App
- Decides it needs resources
- and when
- and for how long

SnappyTV™

Tracking State

- RDBM, NoSQL, files on a disk
- What's the state of a resource?
- A bit of locking is nice

SnappyTV™

Workers

- At worst, cron
- Checks out a job
- Hands it back
- Ideally somewhat distributed

SnappyTV™

Provisioning

- Reliable > Speed
- How provisioned is provisioned?
- close is often good enough

SnappyTV™

Configuration

- Packages, Code, DNS, etc
- Ideally most of this happened in provisioning

SnappyTV™

Error Handling

- Defense in Depth
- Break work into discrete chunks
- Majority of new code

SnappyTV™

Acceptance

- How do we determine it worked?
- What do we do if it didn't?
- May need to live in multiple places

SnappyTV™

Monitoring

- Still need to know running state
- Stats are hard to collect after you've recycled the machines
- Easy to miss trends

SnappyTV™

Decommissioning

- Anything we brought up needs to go down
- Can be more complex
- Missing resources usually worse than unknown ones

SnappyTV™

Corner Cases

- Suck
- Architect around them if you can
- Less flexibility is usually simpler

SnappyTV™

Go Forth

And arrange the services you
already have in interesting ways

SnappyTV™