Using the Public Cloud for Software Engineering Education

Armando Fox
fox@cs.berkeley.edu
Software Engineering for Software-as-a-Service

High productivity, cloud deployment ⇔ Projects work & have immediate, global visibility

“Frequent release” culture ⇔ weekly progress during semester

“Incremental” testing discipline & great testing tools ⇔ testing as root of process, not after-chore

Rails

SaaS + Cloud

Agile

http://tinyurl.com/about-saas
Learn By Doing: Tools > Methodologies

- Software arch., design patterns, coding practices
- Test-first development, unit testing
- Behavior-driven design, integration testing
- Agile, iteration-based project management
- Version management & collaboration skills
- SaaS technologies, deployment & operations

- Ruby & Rails
- RSpec
- Cucumber
- Pivotal Tracker
- Git & GitHub
- EC2, Heroku, SauceLabs CIT
Uses of Public Computing Infrastructure

- Courseware distribution
- Project deployment
- Evaluation (vs. “runs on my computer”)
- Quantitative checking of code metrics (coverage, code/test ratio, quality)
- Project management
- Version control w/instructors observing
- Experiments about horizontal scaling
- (Fall 2012) cloud-based continuous integration testing
Cloud Computing as a Supporting Technology

- Elasticity is great for courses!
  - Lab deadlines, final project demos
  - "Grant" from AWS; ~$100/student
  - See database fall over: ~200 servers needed

- VM image simplifies courseware distribution
  - TAs prepare image ahead of time
  - Students can install weird cutting-edge SW

- Students get better hardware
  - run VM on cloud vs. locally
  - potentially better tech support
Success stories
SaaS Changes Demands on Instructional Computing?

- Runs on your laptop or class Un*x account
- Obscure course project
- Project scrapped when course ends
- Intra-class teams
- Courseware: tarball or custom installs
- Code never leaves UCB
- Per-student/per-course account

- Runs in cloud, remote management
- Your friends can use it
- Gain customers ⇒ app outlives course
- Teams cross class & UCB boundaries
- Courseware: VM image
- Code released open source, résumé builder
- General, collaboration-enabling tools & facilities
Lessons/Takeaways

New opportunity to give students:
- More realistic assignments
- Build, deploy something "real" (outlives class)
- Use same tools they’d use in real life
- Smoother courseware experience

• Challenges
  - Instructor administration not quite there yet
  - Can't install/tweak own kernel
  - Lots of demand; donations won't go on forever

• “Cheaper but faster” than local IT