Containers: Not just for the cloud?

David North
dnorth@apache.org
About me

• Head of Engineering for CoreFiling, a software SME with development teams based in Oxford, UK
• We develop primarily in Java
• We sell to financial regulators, banks, insurers and governments
We all love Docker

• ... or rkt
Show of hands...

• Who has just one deployment of their containerized application and sells it as Saas?
• ... and who does on-premises?
• How do you orchestrate your containers?
Common situation

• No public internet access in production, not now and not ever
• RHEL is the most likely Linux
Issues to solve

• Getting the images there
• Databases: inside the cluster or out?
• Upgrades and security updates
• Management by on-site IT
• Logging
• Backup and recovery
Getting the images there

• Demo: docker save and docker load
Inside the cluster or out?

• Corporate DBA often tightly regulates databases and won't want them buried inside a black box
• Sometimes the I/O intensive bits might be best deployed on physical hardware directly
• Firewalling/networking issues
Upgrades

• Ship a new image bundle?
  – Sizing concerns
Security updates

• The usual containers answer: “just” rebuild and redeploy them regularly
• Our approach: centralized audit of base images + mailing list monitoring for security alerts
Management by on-site IT

• May not have encountered containers before
• Demo: K8s dashboard
Logging

• ELK works
  – Demo: talking on-site IT through dumping logs for a support request
• But can also arrange Docker to go to syslog or other supported locations
Backup and recovery

• If you're just running it yourself in AWS, image snapshots + RDS automatic backups
• Need a documented (tested!!) process for the client to run on-site
Architectural choices

• Can't tightly integrate cloud-only APIs
  – Though where significant extra value is only available this way, it's a handy way to drag customers towards multi-tenant or an installation hosted and managed by us...
Using customer public cloud

- The client won't buy it in AWS from us, but would like us to install it into their AWS/Azure account
- Usually easier, our second favourite option
- Give us some API keys and revoke when we hand it over
Things left to the customer

• SSL – it's their certificate
• Similarly HSTS et al
What about Windows?

• Microsoft, Docker and K8s developers are working very hard to solve that for us

• For now we have a compromise where a Linux VM runs stateless K8s services and the data stores/IO intensive bits run directly on Windows
That's all, folks

• Questions?

• https://people.apache.org/~dnorth/
  • dnorth@apache.org