Your Own Cloud for $500?
Who is Darren?

Husband
Dad
Computer Programmer
Home SysAdmin
Linux User for Over 20 Years
Actually worked on Trusted OS
Will provide any links at end of Slides
Why?

- See if I could build a cloud for <$500
- Learn cloud tech better by running one
- Had some uses for my own cloud
- Seemed like fun
- Turned out to quite useful
Why Apache CloudStack?

- Open Source
- Simpler setup than other options
- Matched up with work proposal
- Wanted to work with a new cloud
- Already worked with Eucalyptus and AWS
Requirements

- Cheap
- 64-bit CPU w/ VM support
- Minimum of two nodes
- NFS Storage
- Single network
- 3-5 instances
- 16GBs for compute node
What Did I Skip?

- Network switch.
  - Enough Ports Available (two total)
  - Had recently upgrade network gear
- Cases
  - Can look ugly
  - Can just build something with scrap wood
  - Inspired by minimalistic crypto mining rigs
Process

1. Pick Cheapest 64-bit CPU w/ VM support
2. Find cheapest motherboard to support CPU
3. Pick cheapest mem w/o too many DOAs in reviews
4. Find Cheapest power supply
5. Reasonable?
   a. No, start again with next cheapest CPU
   b. Yes, pick HDs to stay within budget

The Hardware Ordered

Newegg.com - Fall 2017

2 AMD A6-6400K + $100.79

2 PSUs, 2 ASRock A68M-ITX Motherboards, 3 8GB DDR3 Memory + $346.23

2 250GB HD + $40.50

Total Shipped: $487.52

-$30 in rebates.....
Upgraded/Repaired Instead

2 250GB SSDs + $75.98 (cheap HDs died)  
Replacement Management server

1 Raspberry Pi 4 with 4GB + $55

1 Raspberry Pi case + $9.47

1 power supply + $7.95

Shipping + $21.7

Total with Shipping = $170.10
Where to Save Money

- Use All Raspberry Pies
- Most of us have spare HD or SSDs laying around unused
- Pick a refurbished CPU
  - may required purchasing CPU cooler
  - CPU Socket requires figuring out
  - Motherboard for Socket not available
- Pick refurbished motherboard
- Repurpose old Motherboard
- Use Old CPU from last server upgrade
- Spare memory? Pick CPU and motherboard for what you have
- Pick memory first, and then CPU and Motherboard to fit
- Used rackmount can be really cheap
  - Often includes storage, Memory, and CPU
  - Can be loud
  - Requires rack (real cost money, but can be made from lumber or Ikea Lack table)
Installation

- Started with Ubuntu on two nodes: 1x compute & 1x management/storage
- Transitioned to: 2x compute & 1x management/storage on raspberry pi
- Manual setup of compute node networking
- Custom ansible roles (https://gitlab.com/coledarr/cloudstack-roles)
- Pulled all the rolls together with a playbook and static inventory
- Automated purging of nfs setup, compute, and management
  - Big time saver
  - Quicker for testing
  - Easier to reset to known good state
- Automated source installation and building
- Manual setup of management server webinterface
Gotchas

- Systemvm loading No longer needed
- Networking issues
- Custom template creation
Networking Issues

- Did not set vlan tagging for bridges
- All networking on 10.0.0.0/8
- Final put everything in their own network
  - 10.0.0.1/16 - pod
  - 10.0.1.1/24 - management network
  - 10.0.2.1/24 - Guest “public ip”
Custom Template Creation

- Use newest cloud-init in my distro
- Partition for “/” Did Not Grow
  - cloudstack doesn't grow filesystem
  - Cloud-init grows filesystem
  - Cloud-init only grows last partition if it is a filesystem, don’t put swap there
- Rocky needs cloud-init-utils
Uses of my Cloud
Uses

- Ansible test and development to destroy and recreate as needed
- Testing different distros, or even full setups like TruNAS Core
- Testing new services like Plex, Wireguard, QMK building, etc.
- Devnode that can be reinstalled as needed
Learn from what I did

- CPU mattered more than memory
- Memory not as critical as expected
- Save templates outside the cloud to speed up re-installs
- Keep templates size minimal
- Storage is at a premium, but most people probably have bigger HD lying around.
- More networking knowledge would definitely help
Probably Less than AWS

- $50-$70 per month for similar instance, storage, and AMIs in AWS
  - 10 month return on investment (@$50)
  - No need to worry about shutdown or termination
  - No extra costs for custom templates (AMIs)
  - Some testing would be a non-starter (Plex server)
Final Thoughts

- Implement linux package caching
- Keep templates, ISOs, and anything else saved outside the cloud
- Add extra switch for storage network
- Setup up vlan tagging and trunking
- Try some of the SDN options
- SSL certs for Cloudstack
- New managed Kubernetes cluster
Promised links

- dcole@aseg.com or darren.cole@gmail.com
- This Presentation: https://docs.google.com/presentation/d/18IIdHMCXiX0FXq6o7VD-0f7RyLGX23haYarRfl2S9Jw/edit?usp=sharing
- Cloudstack roles and example plays: https://gitlab.com/coledarr/cloudstack-roles
- Ikea Lack rack page: https://wiki.eth0.nl/index.php/LackRack