

# Pros and Cons of using Cloudstack with local storage

Swen Brueseke

Twitter: @SwenBrueseke

Blog: [swen.io](http://swen.io)

# WHO WE ARE AND WHAT WE DO

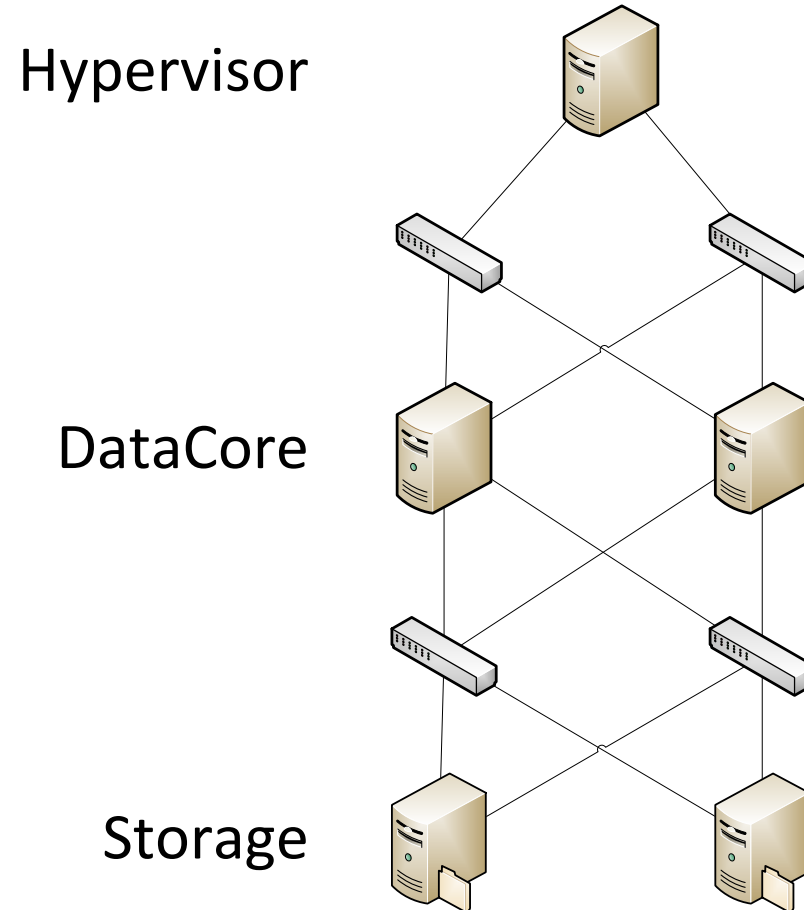
- RemoteHands
- German cloud provider (megona.de)
- Managed Services



# WHAT YOU NEED TO ASK FIRST

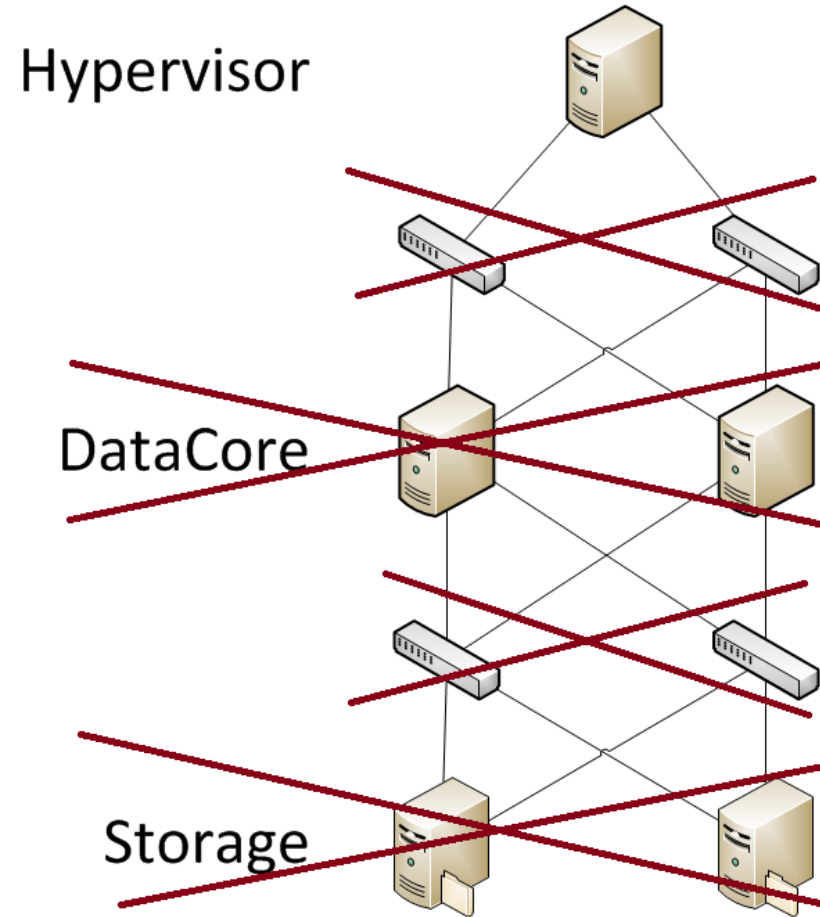
- System VMs on shared or local storage?
- Think about your SLA
- What do your customers really need?
- Management overhead
- Hypervisor (KVM not supported, CLOUDSTACK-7982)
- Compute Offerings
- Disk Offerings
- Keep it simple

# OUR SETUP WE STARTED WITH



# WHAT WE SAVED

- DataCore licenses
- DataCore hardware
- DataCore network infrastructure
- Storage switches
- A lot of storage hardware



# PROBLEMS WE RUN INTO

- With volumes (migration)
- With data volumes (only works when attached to VMs)
- With snapshots (cleanup, leftovers)
- With templates (cleanup, leftovers)
- With local storage overprovisioning (value ignored in earlier versions)
- With XenServer (monitor local storage, thin provisioning, upgrades)

# WHAT WE USE

---

- Samsung SSD PM863 / SM863
- Dell PowerEdge R620/R720/R730
- Intel X520 Dual Port 10G NIC
- Juniper EX4550

# THE FUTURE

---

- New in XenServer 7.1: Live Patching (Enterprise Edition)
- Storage Live Migration (Support for KVM)
- Data disk migration



# PROS AND CONS

## PROS

- Better IOPS
- Save network infrastructure
- Save storage hardware
- Keep it simple

## CONS

- No live migration on KVM
- Data disk migration only when attached
- Unavailable data by host outage
- No HA for VMs
- Management overhead

# Thank You

Swen Brueseke

Twitter: @SwenBrueseke

Blog: [swen.io](http://swen.io)

# MIGRATION FROM SHARED TO LOCAL

1. Create new service offerings (compute and disk)
2. Write down internal name of instance
3. Shutdown instance
4. Migrate primary storage of instance via UI to local storage
5. Get id of new service offerings out of db  
`mysql> select * from disk_offering where name='<name of offering>' \G`  
Note down id of disk\_offering where state is 'Active'
6. Get id of volumes attached to instance  
`mysql> select * from volumes where instance_id='<id>' \G`  
Note down id  
The instance\_id is the last number of internal name of instance. (i-2-220-VM)
7. Change service\_offering for instance  
`mysql> update vm_instance set service_offering_id='<id of 5>' where id='<instance_id>';`
8. Change disk\_offering of volumes  
`mysql> update volumes set disk_offering_id='<id of 5>' where id='<id of 6>';`
9. Double-check values in UI
10. Start instance

# VOL-SNAPSHOTS WORKFLOW

1. CS is telling XenServer to create a snapshot (first)
2. XenServer is creating a snapshot on local storage
3. XenServer is mounting secondary storage and copies snapshot
4. XenServer is unmounting secondary storage
5. Workflow end
6. CS is telling XenServer to create a second snapshot
7. XenServer is creating a delta snapshot (chain) on local storage
8. XenServer is mounting secondary storage and copies delta snapshot
9. XenServer is unmounting secondary storage
10. XenServer is using coalesce to merge snapshot chain on local storage
11. Workflow end
12. And so on...

So the last snapshot of a volume will always stay on local storage. In earlier version there was no cleanup job when the vm was deleted. So your local storage will fill up with unneeded snapshots.