

HOW TO PLAN & EXECUTE A SUCCESSFUL CLOUD MIGRATION

Steve Bertoldi, Solutions Director, MarkLogic



Agenda

- Cloud computing and on premise issues
- Comparison of traditional vs cloud architecture
- Review of use cloud use cases



What is cloud computing?

- Self service provisioning
- Elasticity and scale
- Pay as you use services
- Efficient and economic
- Infrastructure as a Service (IaaS)



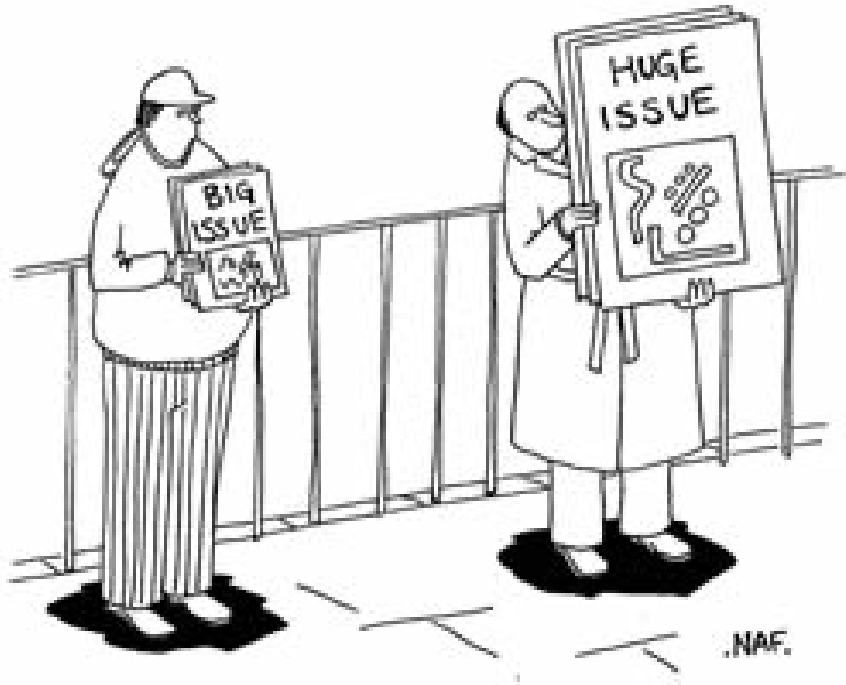
Google Cloud Platform



Cloud Providers

- Public clouds
 - Amazon Web Services (AWS)
 - Google Cloud
 - Microsoft Azure
- Private Clouds
 - GovCloud

Issues with on premise infrastructures



- Infrastructure related software bugs
 - Frequent patching causes down time
 - Patches require reboot forcing outages
 - Systems unstable after patch
- Hardware stability issues
 - Can be difficult to troubleshoot
 - Virtualized infrastructures as fixed assets

Pain points realized by customers

AAARGH!

- Customers want a stable platform
 - Tired of infrastructure headaches
- Bare metal management costs increasing
 - Procurement taking longer
- Administration issues
- Issues with some in-house directives



Don't need in-house systems

- No need to be close to data center
 - Stop hugging servers!
- The best of today will eventually be outdated



The MarkLogic AWS solution

- MarkLogic is supported on AWS Linux
- No other external dependencies
- Higher network availability
- Easier to procure and administer
- Better disaster/backup recovery
 - Different availability zones
- Simpler backups with Simple Storage Service (S3)

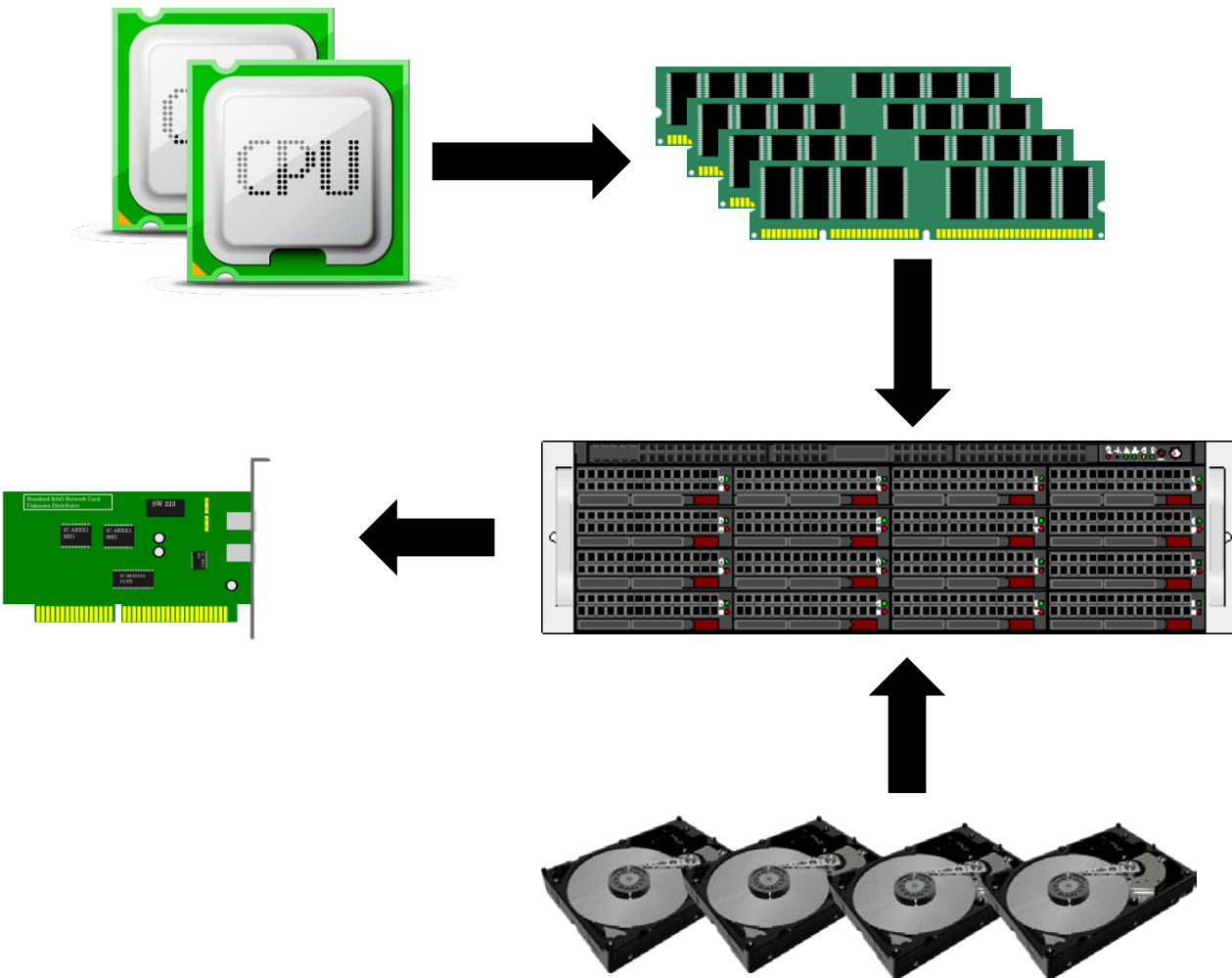
Gains of MarkLogic and cloud implementations



- Faster build and delivery times
- Scale-on-demand and elastic capabilities
- Easier tools and management
- Internal cost reduction
 - Asset recovery

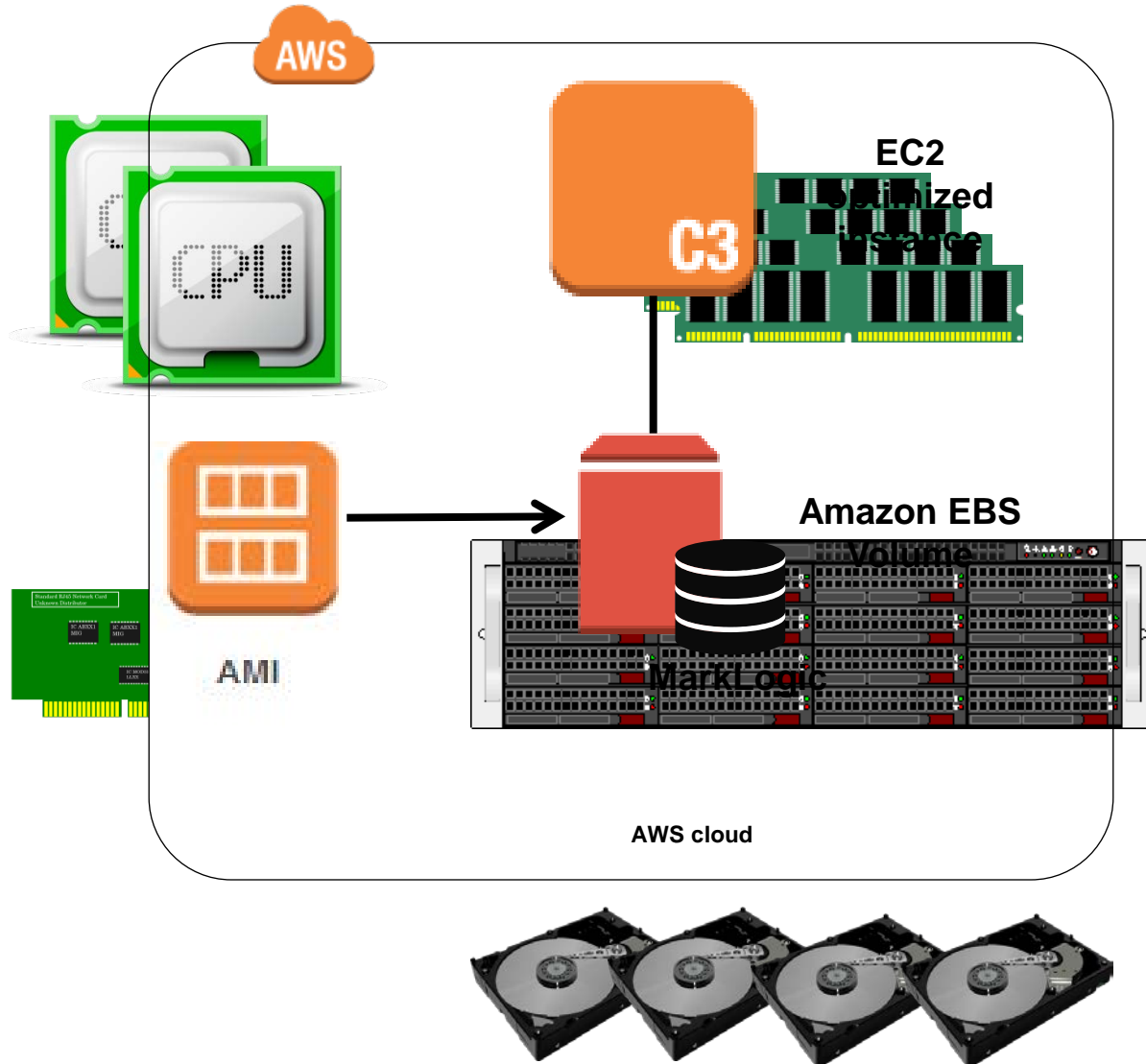
TRADITIONAL VS AWS

System builds – the old way



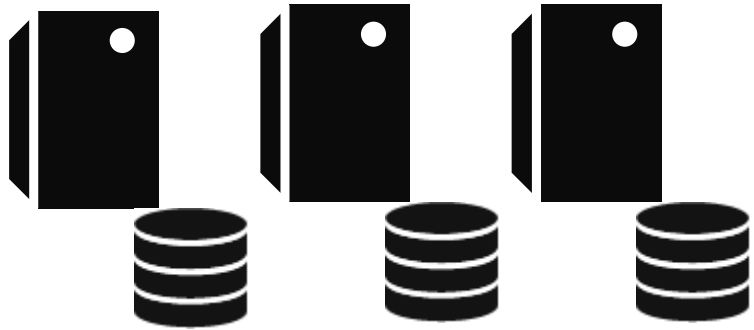
- CPU specifications
 - Cores, sockets, speed
- Memory
 - Dependent on CPU
 - Size
 - Capacity
- Motherboard
 - Chassis/capacity
- Network
 - Fabric/topology
- Storage
 - RAID?
- Send to procurement

Building the AWS way



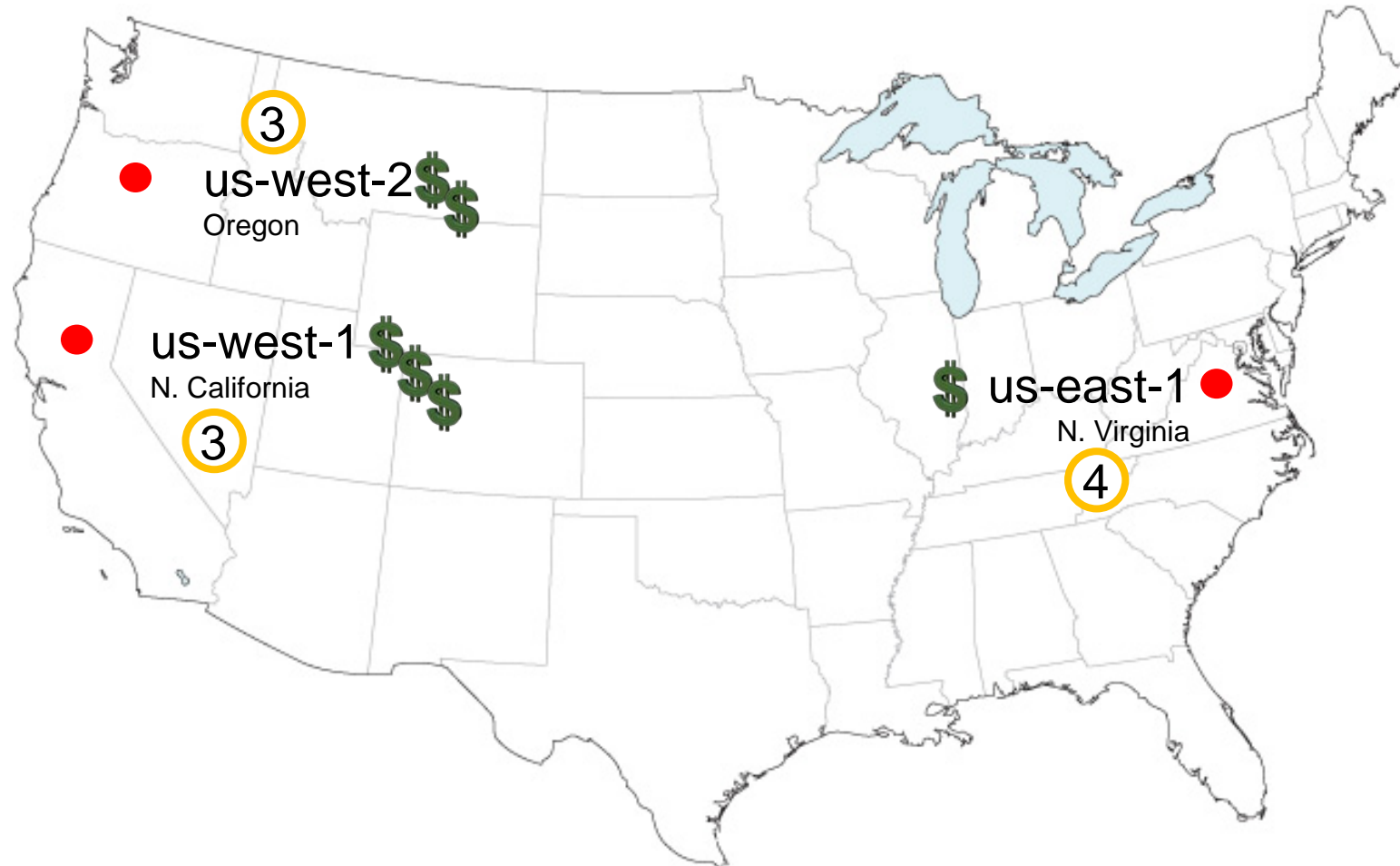
- CPU/Memory/Networking
 - Elastic Compute Cloud (EC2)
 - CPU/Memory
 - Networking
- Disk/Storage
 - Elastic Block Storage (EBS)
 - GP2
 - PIOPS
 - Magnetic
- Amazon Machine Image (AMI)
- Install MarkLogic

Migrating a typical MarkLogic Cluster




- 3 node MarkLogic cluster
 - 2 CPUs, 8 cores each
 - 32 GB RAM
 - 3 TB of local storage

Select a region





Amazon Machine Image (AMI)

**Amazon Linux** **Amazon Linux AMI 2016.03.0 (HVM), SSD Volume Type**
The Amazon Linux AMI is an EBS-backed, AWS-supported image
Free tier eligible
Root device type: ebs Virtualization type: hvm

Select the AMI

- Bootstrap image for the OS
- Provides details about the instance
- Copies image to disk



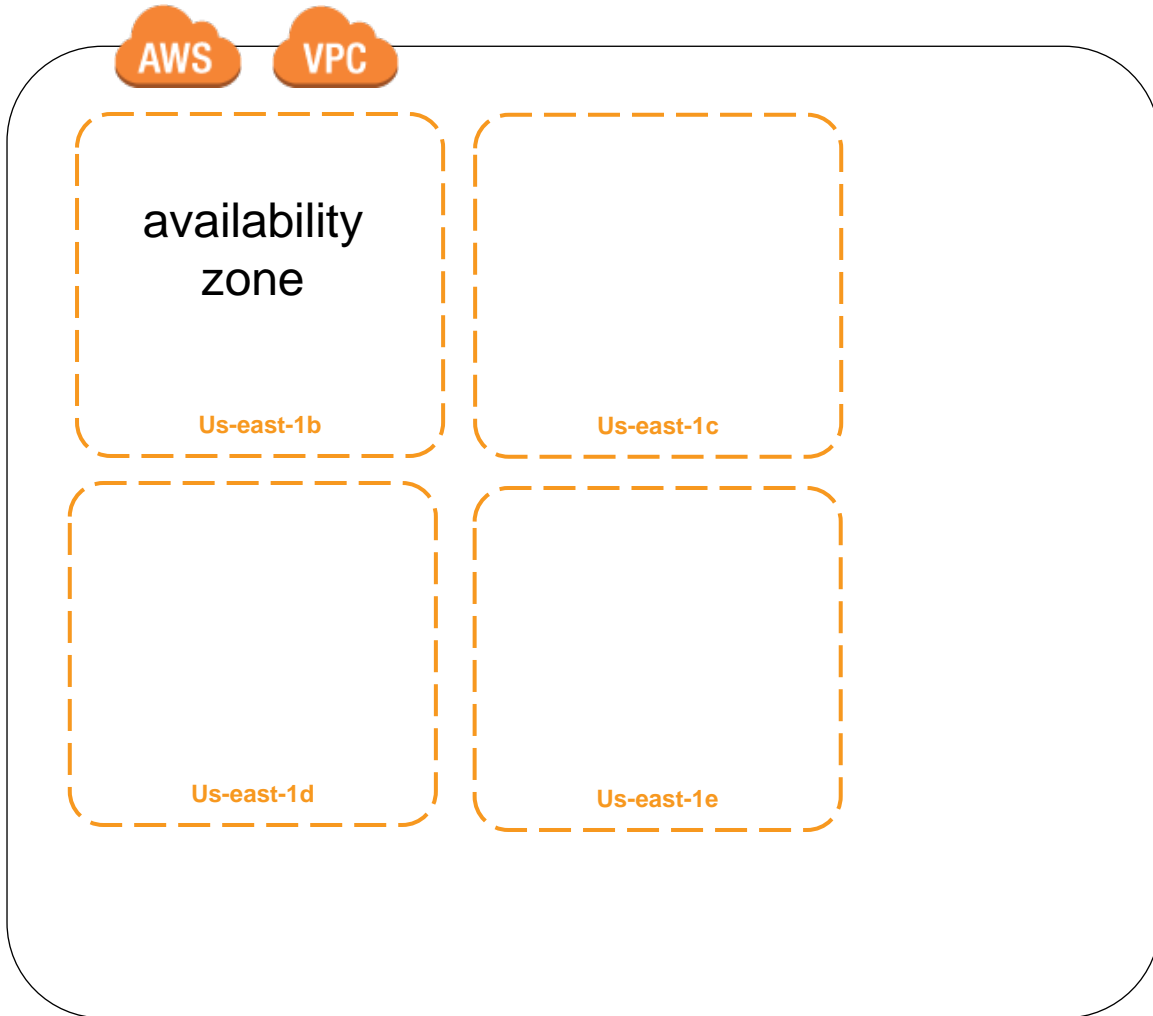
EC2 instance
small



EC2 instance
large

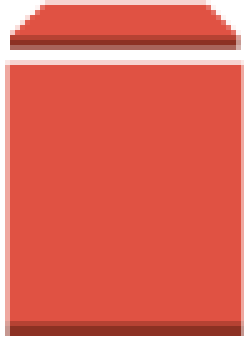
Choose an instance type

- Provides compute power
- Choose based upon the system needs
- c4.8xlarge
 - 32 vCPUs
 - 60 GiB RAM
 - Enhanced networking



Define instance details

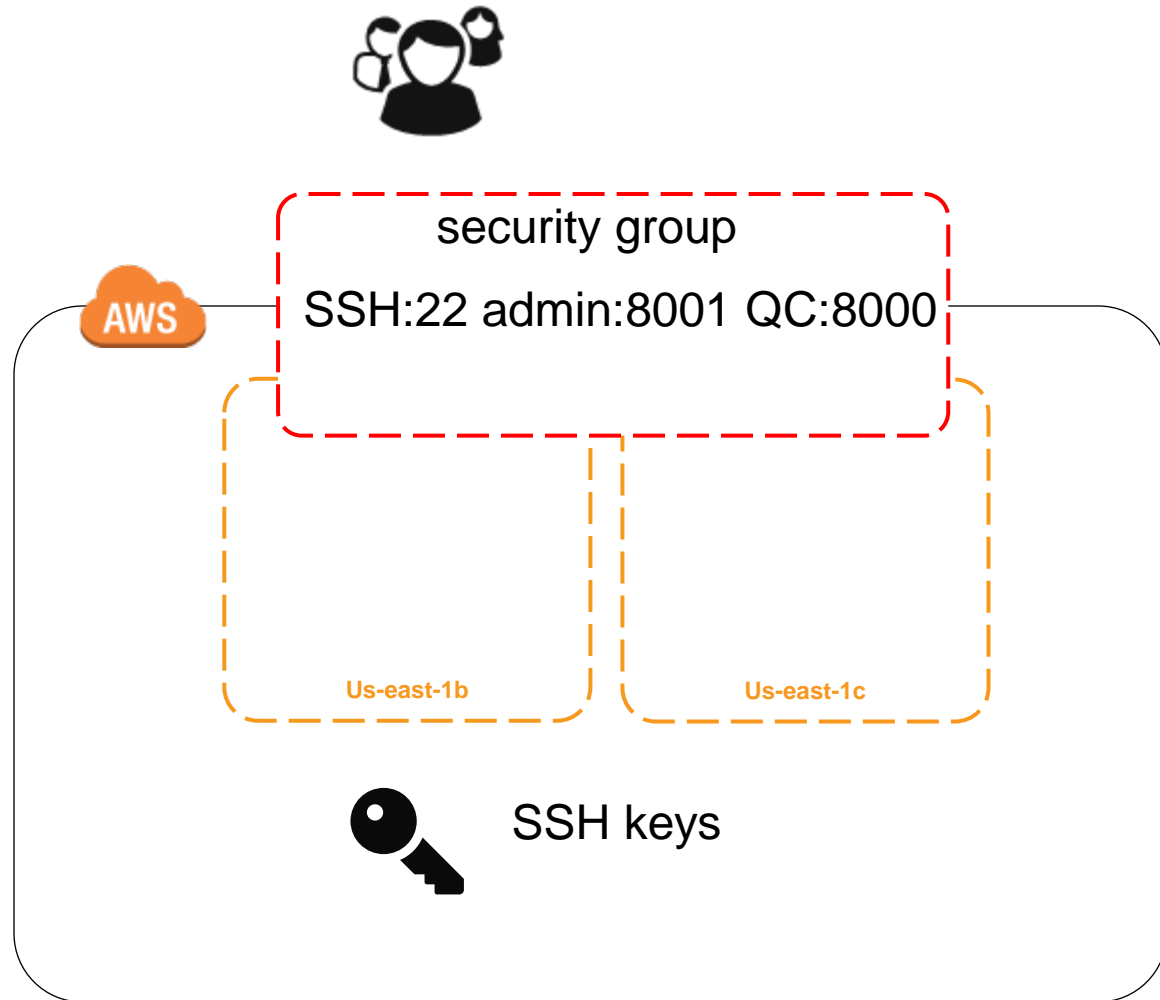
- Select availability zone (AZ)
- Do you need a Virtual Private Cloud (VPC)
- Networking options/subnet
- Many more options



Amazon EBS

Add storage

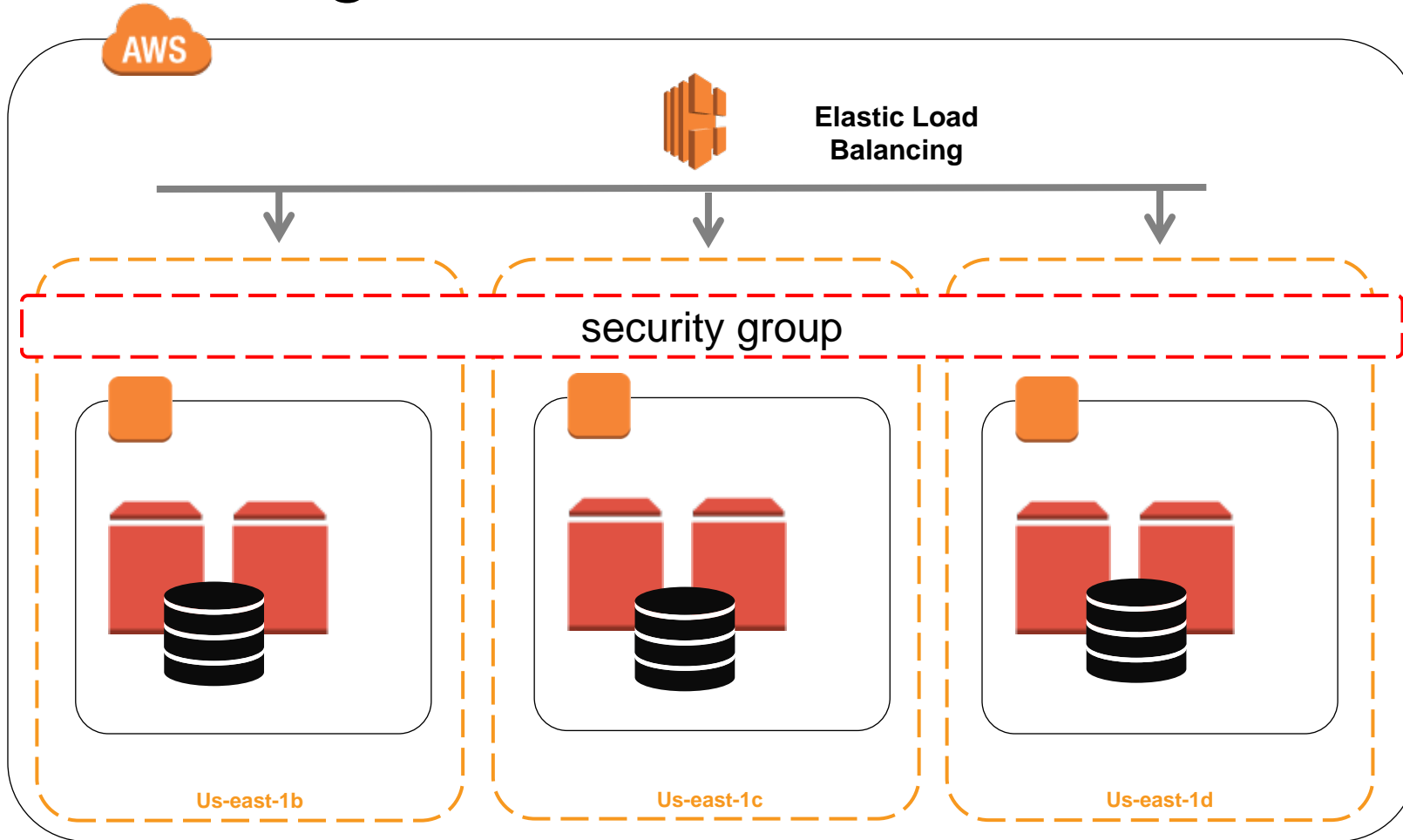
- Need to have a root volume
 - Size to your needs
- Data volume types
 - GP2
 - PIOPS
 - Magnetic
- 3TB GP2



Setup Security

- Configure Security Group
 - Firewall rules to control traffic
 - Spans availability zones
 - SSH keys for private/public key authentication
- Launch!

Building out the cluster



- First instance in the cluster
- Repeat for remaining nodes
- Install MarkLogic
- Elastic Load Balancer (ELB)

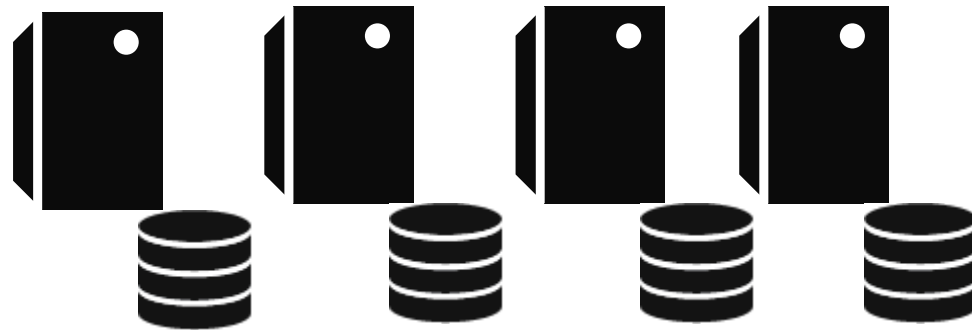
Building MarkLogic clusters in AWS

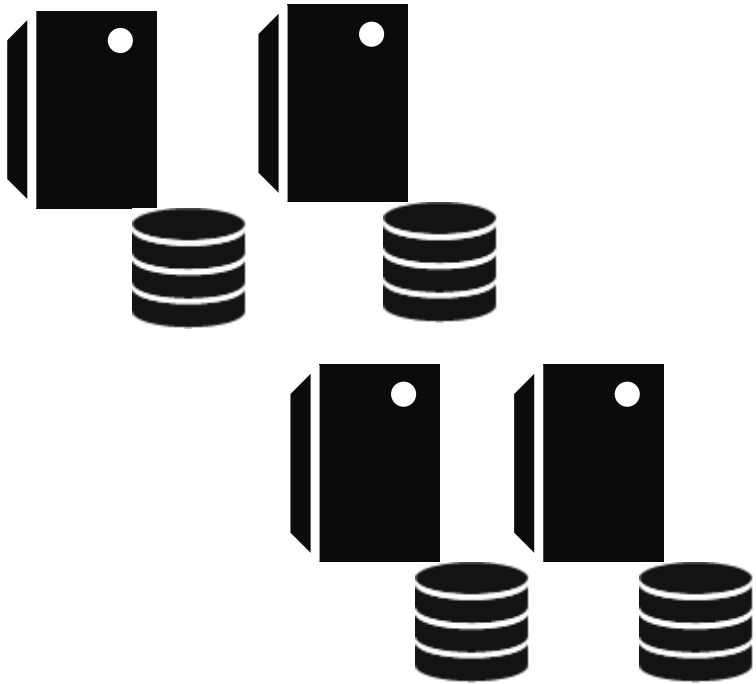
- Unmanaged/manual install
- Launching with a MarkLogic AMI directly
 - AWS Marketplace
 - Bring your own license (BYOL)
 - Metered
- MarkLogic Cloud Formation templates
 - Managed clusters

USE CASES

Use case: AWS for testing

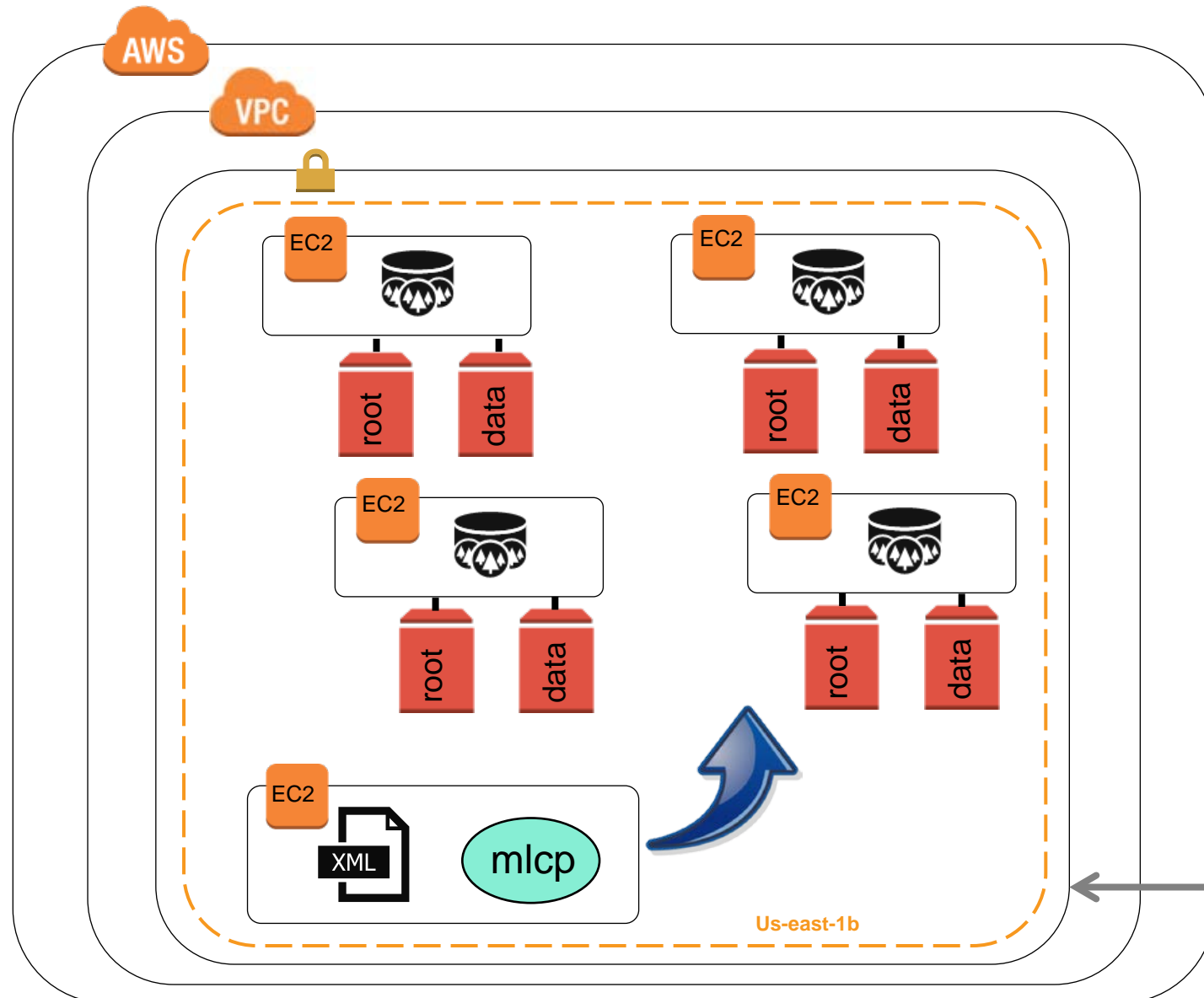
- New system part of a larger infrastructure
- Target SLAs defined
- 4 node cluster
- Customer hardware in procurement – cannot change
- Testing on internal virtualized systems





Operations and problems

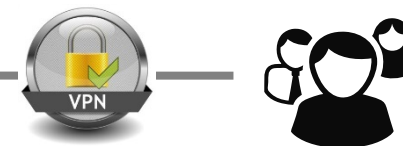
- Transformation hub
- Large amount of record loading
- Not meeting SLAs
- Bad performance on local VMs

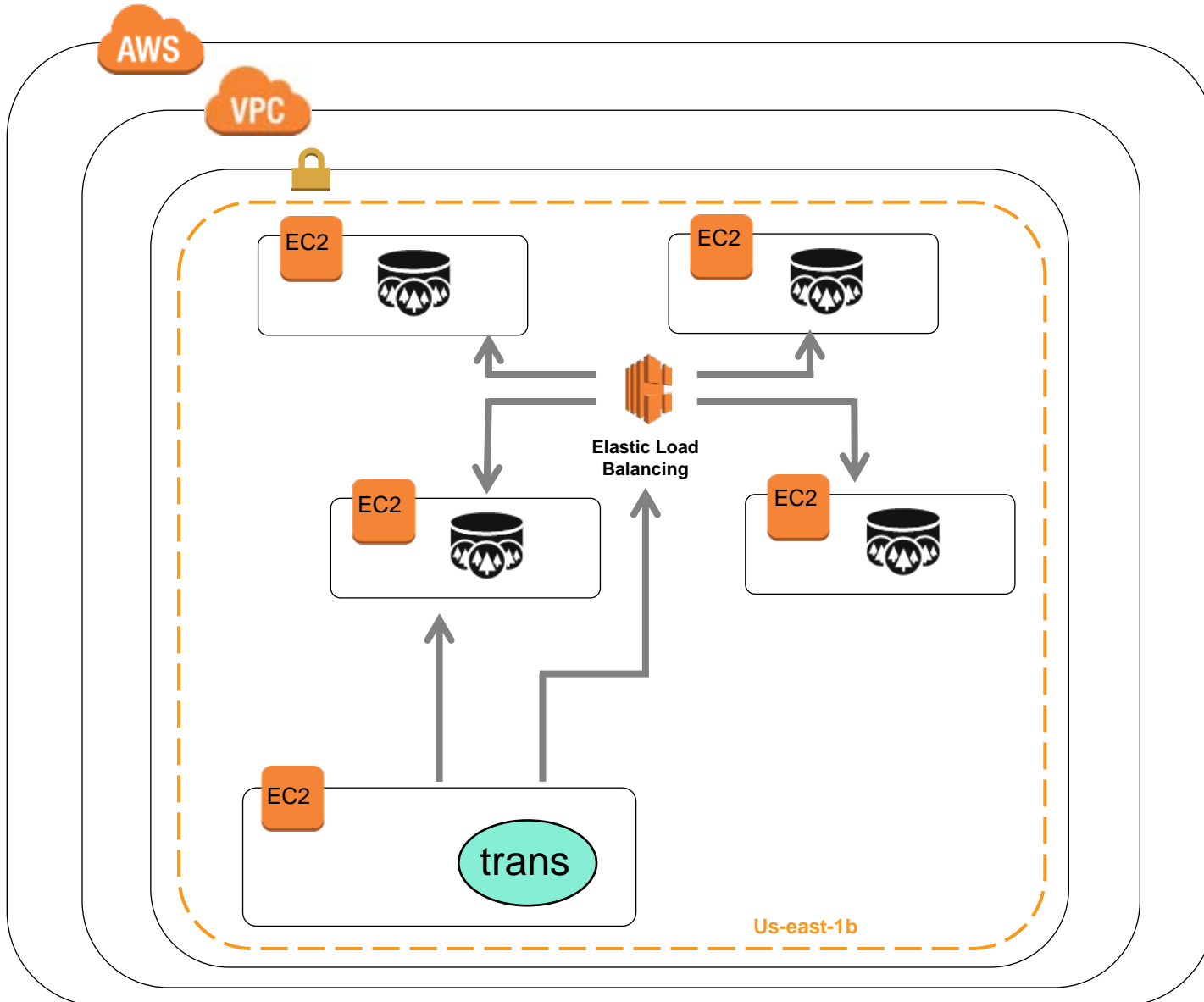


Testing infrastructure

Setup Virtual Private Cloud

- Users connect through VPN
- 5 m4.4xlarge instances
- Build MarkLogic cluster
- Run loading tests
- Run times < SLA



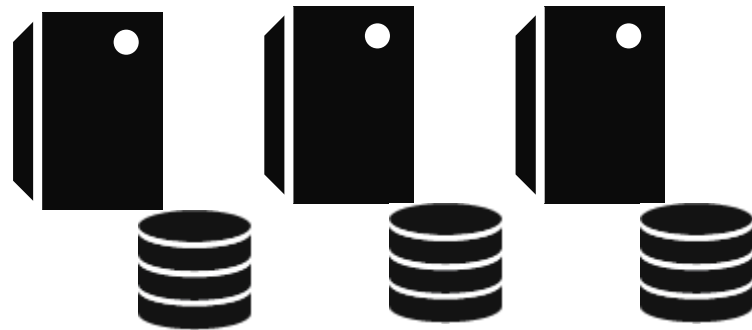


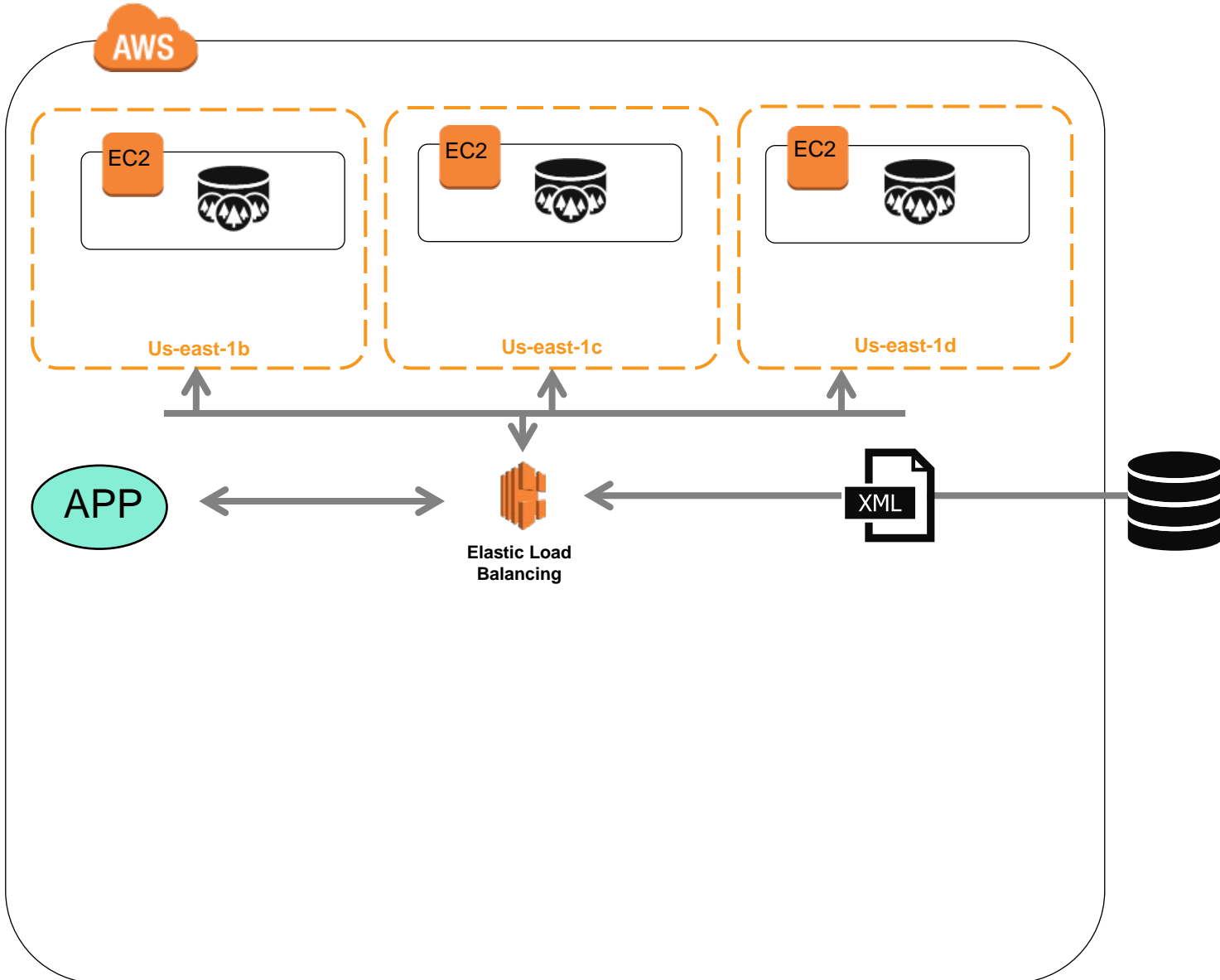
Transform job

- Transformation job
 - Runs on a single node
- Not meeting SLA
- Added elastic load balancer
 - Routed traffic to all nodes
- Performance < SLA

Use case: Utilizing scaling features

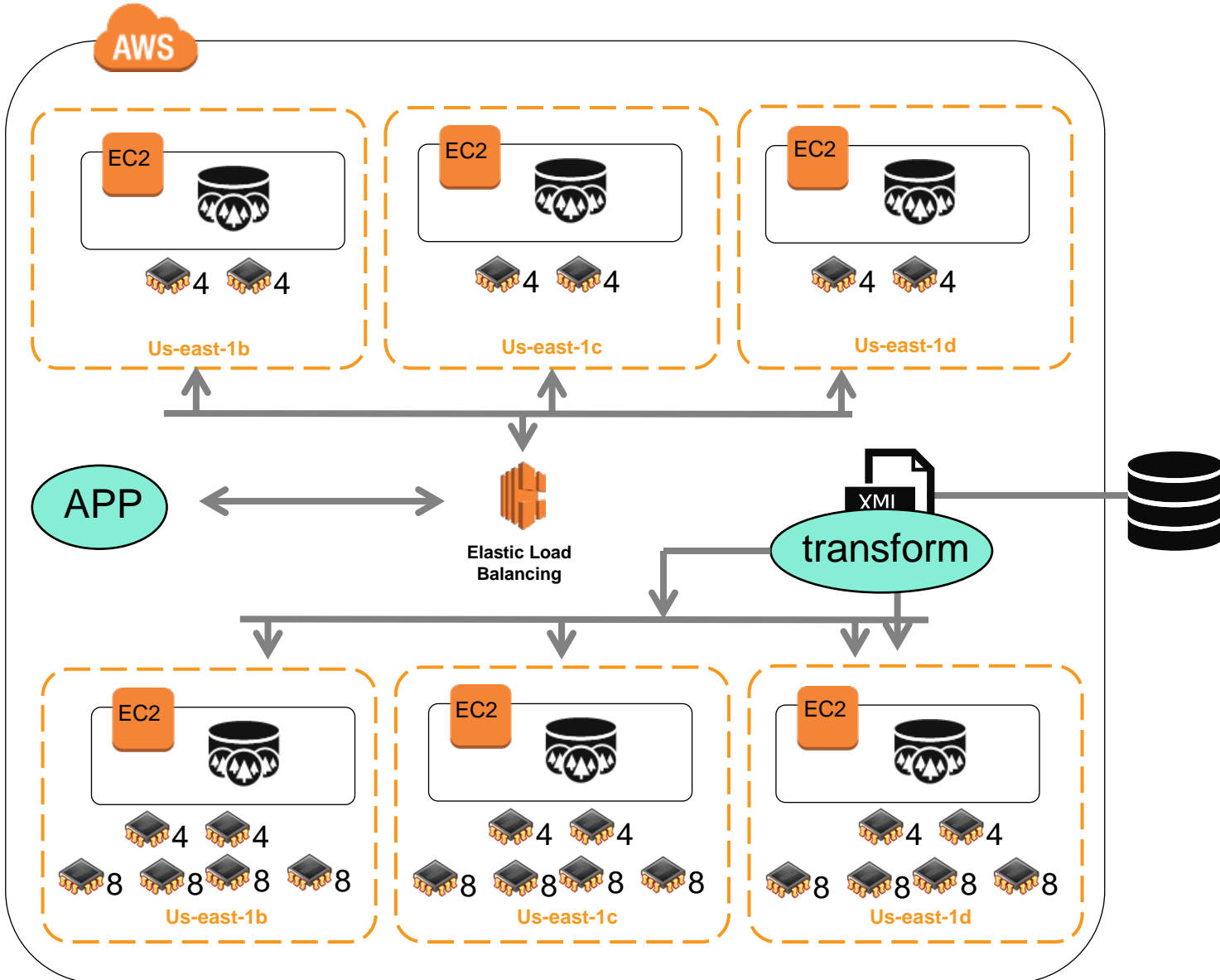
- Search and lookup during the day
- Millions of documents from legacy system
- Transform batch job
- Cannot meet SLAs





Initial infrastructure

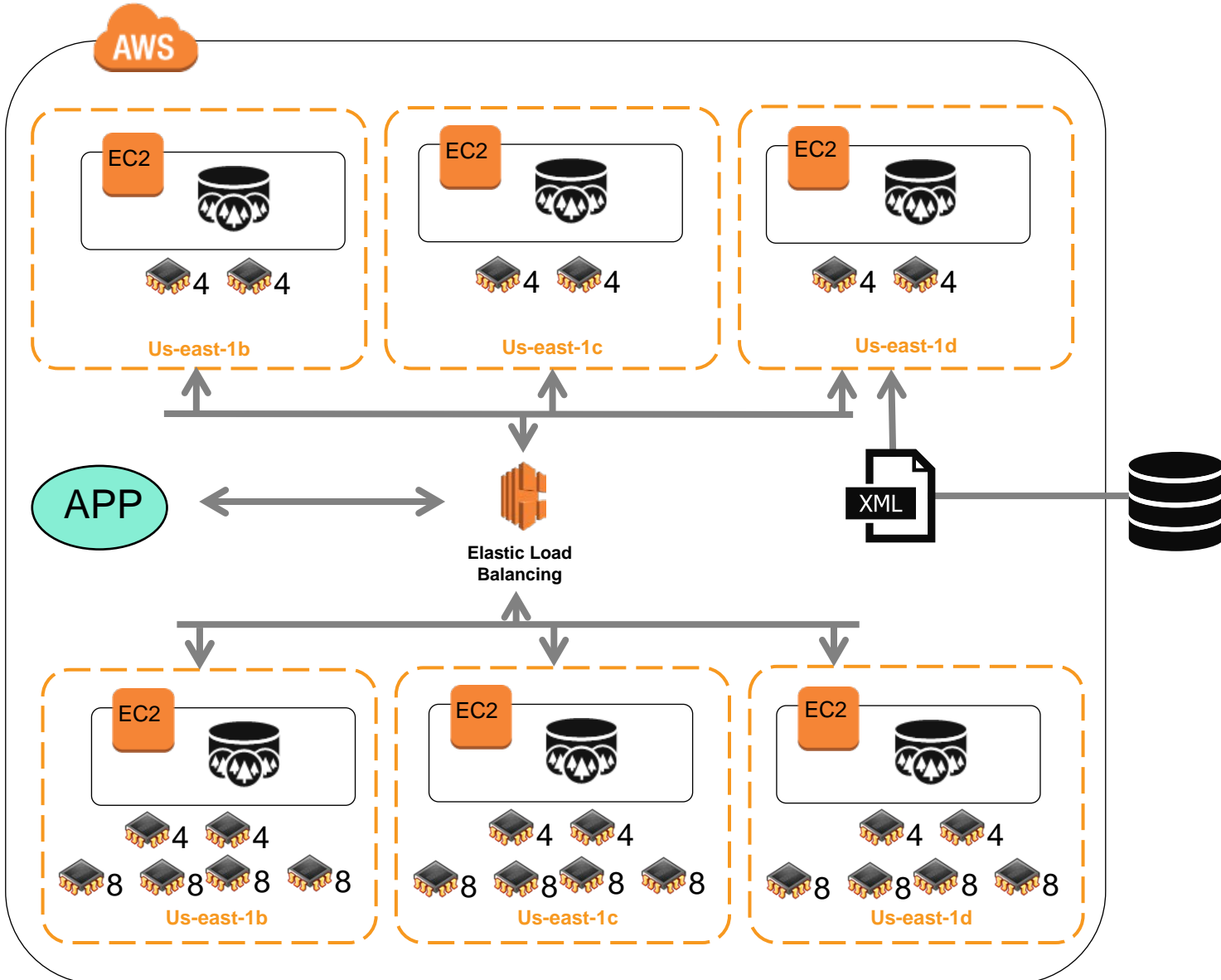
- 3 node architecture
 - Medium sized instances
- Loading occurs all day
- SLA in place



Scaling infrastructure

- Primary made smaller
- Second cluster added
 - 8 vCPU each
 - Load to secondary

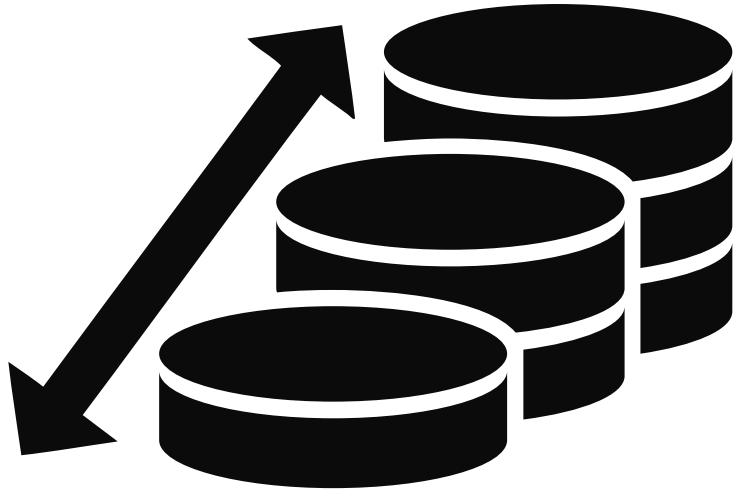
- Transform scale-up
 - Loading disabled
 - Scale up instances
 - 32 vCPU each
- Transform on all 3 nodes
- Met SLAs



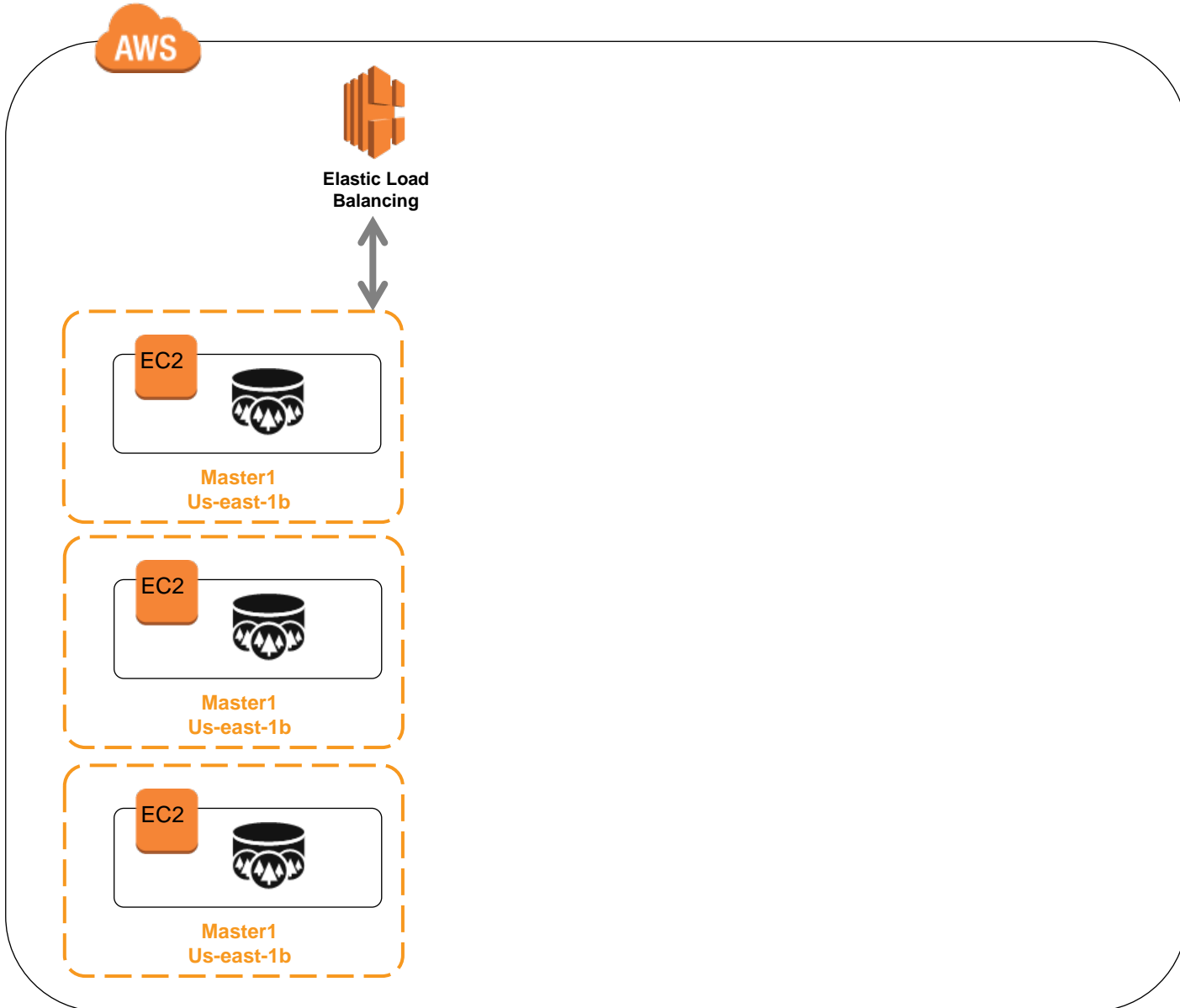
Scale down and switch

- Transform complete
 - 2 clusters exist
- Scale 2nd cluster to 8 vCPUs
- Swap cluster at load balancer
- Clear primary
 - Attach loader to primary

Use case: Elastic auto-scaling architecture

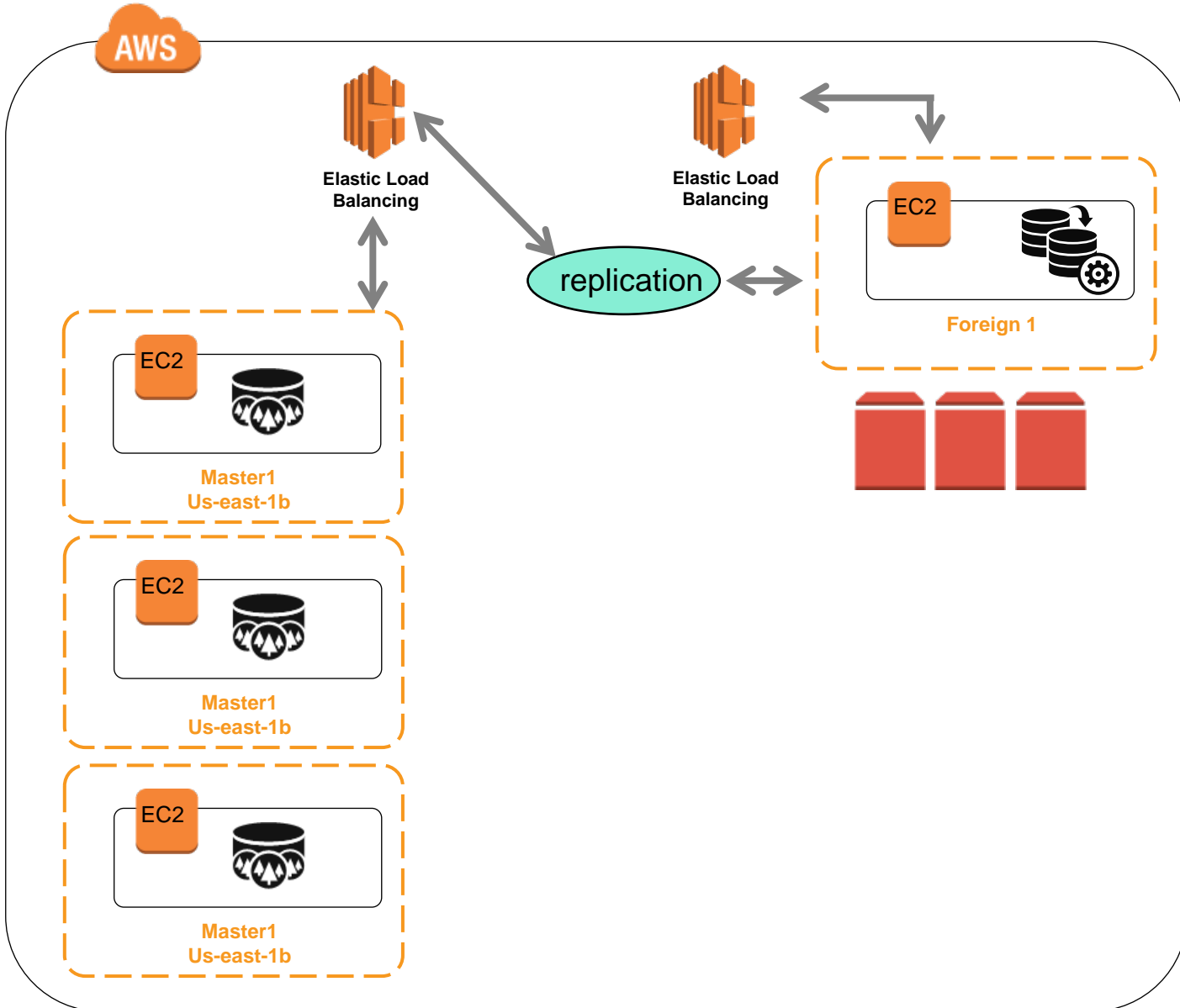


- Document based system with searching
- Provides alerting and highlighting
- 3 node primary system
- Need to meet peak demands



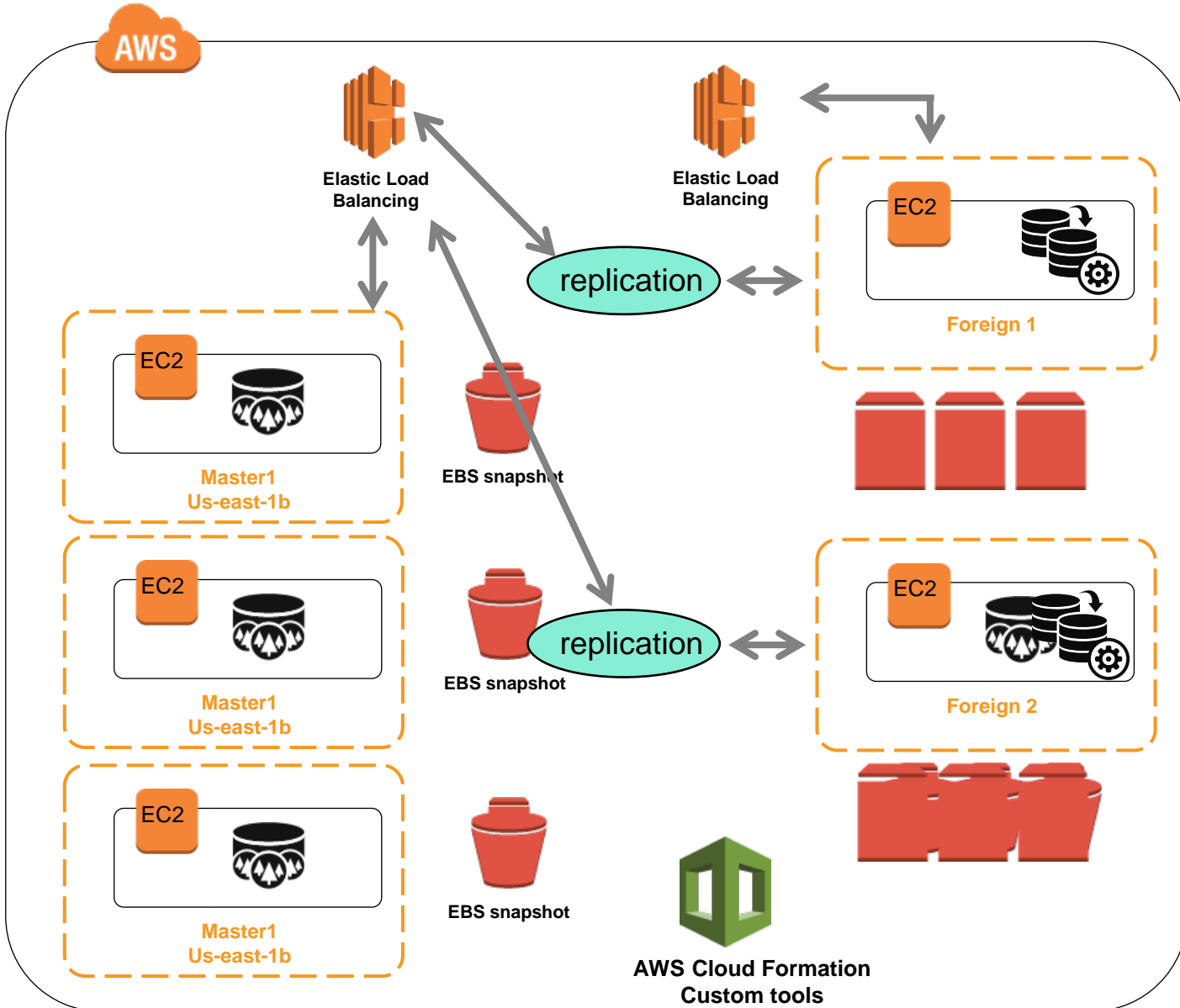
Master cluster

- 3 node cluster
 - Separate availability zone
- Used for highlighting
- C3.2xlarge
 - 8vCPU
 - 15GB RAM



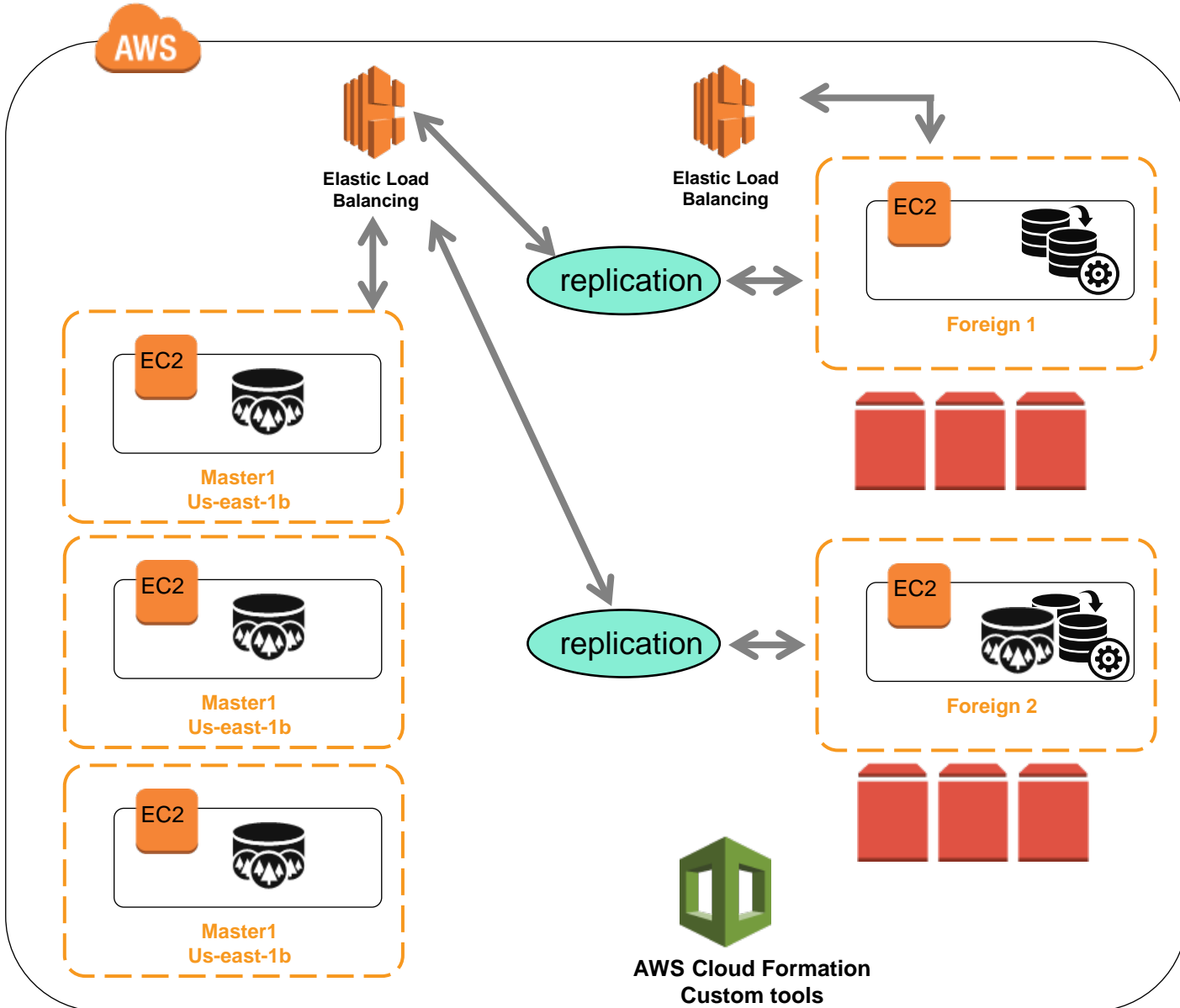
Foreign node

- Single node
- Uses database replication
- Search capability
- Scheduled auto-scaling
 - Base set for normal loads
 - Increase for peak loads
 - Reduce for weekend loads



Scale up

- Snapshot master
- Create new foreign node
- Attach snapshots
 - Create volumes
- Setup replication
 - Attach foreign cluster
 - Attach replicas



Scale down

- Recover resources
- Disable replication
 - Remove replicas
 - Detach foreign node
- Cleanup instance
 - Stop instance
 - Remove instance
 - Remove volumes

WRAPPING UP

Takeaways

- It works!
- Produce results faster and cheaper
- Flexibility, scalability and elasticity
- Easy to test out several configurations
 - No restrictions from fixed infrastructures
- New enhancements and growing feature sets
- Cost declining

Q&A