The Nuts and Bolts of Running Epic on AWS

Technical Solution Brief

Table of Contents

Why Should I Read This Paper

This technical solution brief is intended primarily for technical decision-makers and influencers, including system architects and planners as well as operations and engineering staff who support and run the infrastructure for the Epic application. The purpose of this guide is to show how Epic runs on AWS and introduce key concepts that will help you better understand the issues that you will face when moving Epic from on-premises to AWS. Before we get to the meat of the paper, here are five key concepts about AWS that you need to understand.

The purpose of this guide is to show how Epic runs on AWS and introduce key concepts that will help you better understand the issues that you will face when moving Epic from on-premises to AWS.

Five Easy Pieces on AWS

This section presents five essential concepts about AWS that you as an IT professional will need to know as you plan your migration and operational strategies for Epic in the cloud.

1. Welcome to the Candy Store – The AWS Catalog

When AWS was launched in 2006, the site offered a single product, Simple Queue Service (SQS), which was soon followed by Simple Storage Service (S3) and Elastic Compute Cloud (EC2). Now AWS features a massive catalog of more than 200 products, from infrastructure technologies such as compute, storage, and databases to emerging technologies such as machine learning and artificial intelligence, data lakes and analytics, and Internet of Things (see figure 1).

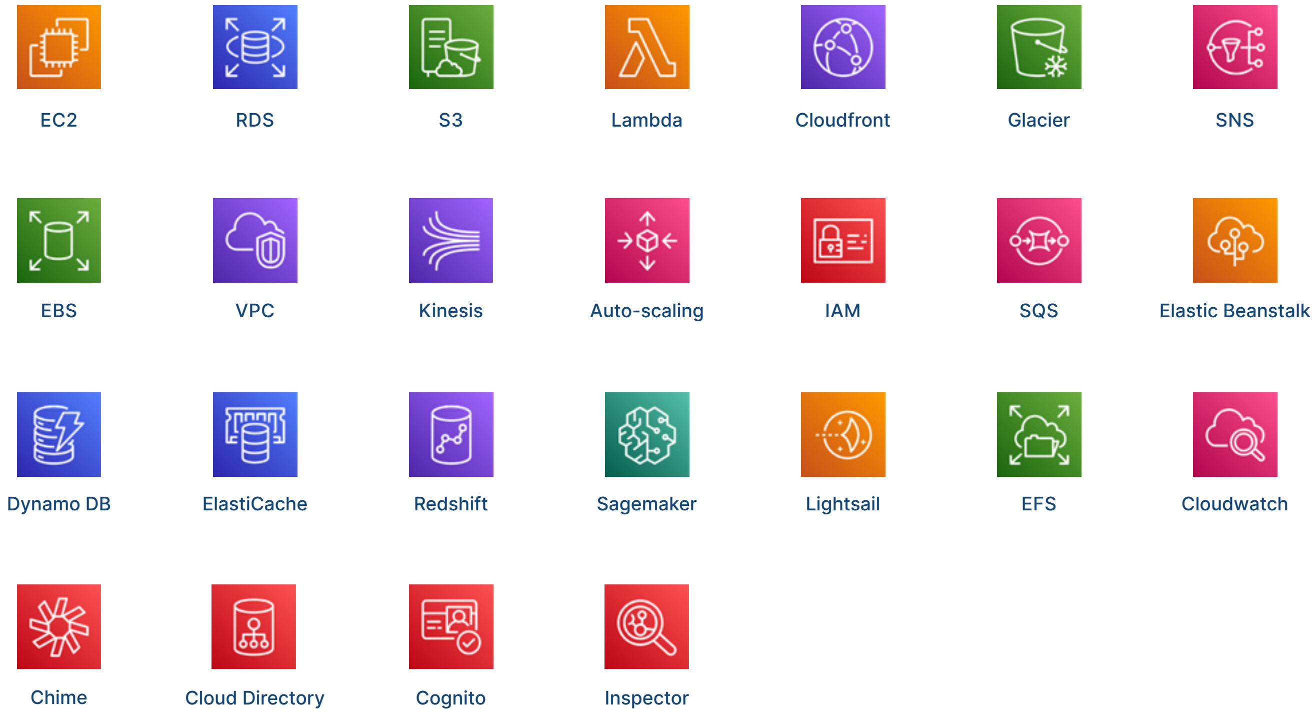

Figure 1. The 25 Most Popular AWS Services

With such an extensive array of options, there is a tendency to think of the AWS services like Lego pieces that you can just plug together to make some really cool things. The reality is somewhat different. Integrating AWS services into a working system is a complex task, requiring both substantial domain knowledge and extensive familiarity with the AWS services needed.

For example, consider a system that automates the deployment of firewalls to users (see figure 2). Procuring the individual AWS services is an easy task, but integrating, configuring, and deploying them as a working unit requires developers with extensive knowledge of both firewall security and individual AWS services such as Amazon EventBridge, AWS Lambda, and AWS Firewall Manager.

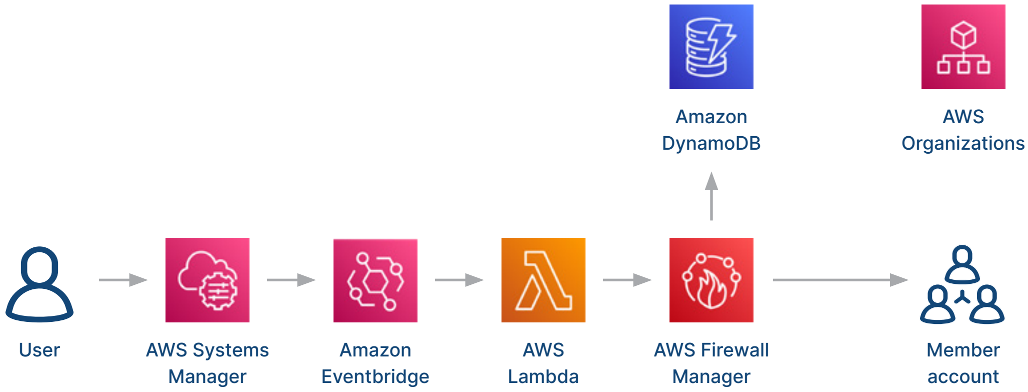

Figure 2. Architecture for an Automated System for Firewall Deployment

For these reasons, most healthcare organizations rely on a managed service provider or another third party to provide the needed expertise.

2. Share and Share Alike – Security Responsibility

The very term "shared responsibility" may give heartburn to Chief Information Security Officers (CISOs) and VPs of cybersecurity — after all, they're used to having the security system entirely under their control. However, once you understand the concept of shared responsibility, you'll see how it actually makes your information more secure and private.

In the public cloud, the responsibility for security is divided between the customer and AWS (orange and blue respectively in figure 3). AWS protects the infrastructure — the hardware, software, and supported operating systems — that runs all of the services offered in the AWS cloud. You as the customer are responsible for securing everything that sits on top of that infrastructure, such as applications and data along with any unsupported operating systems.

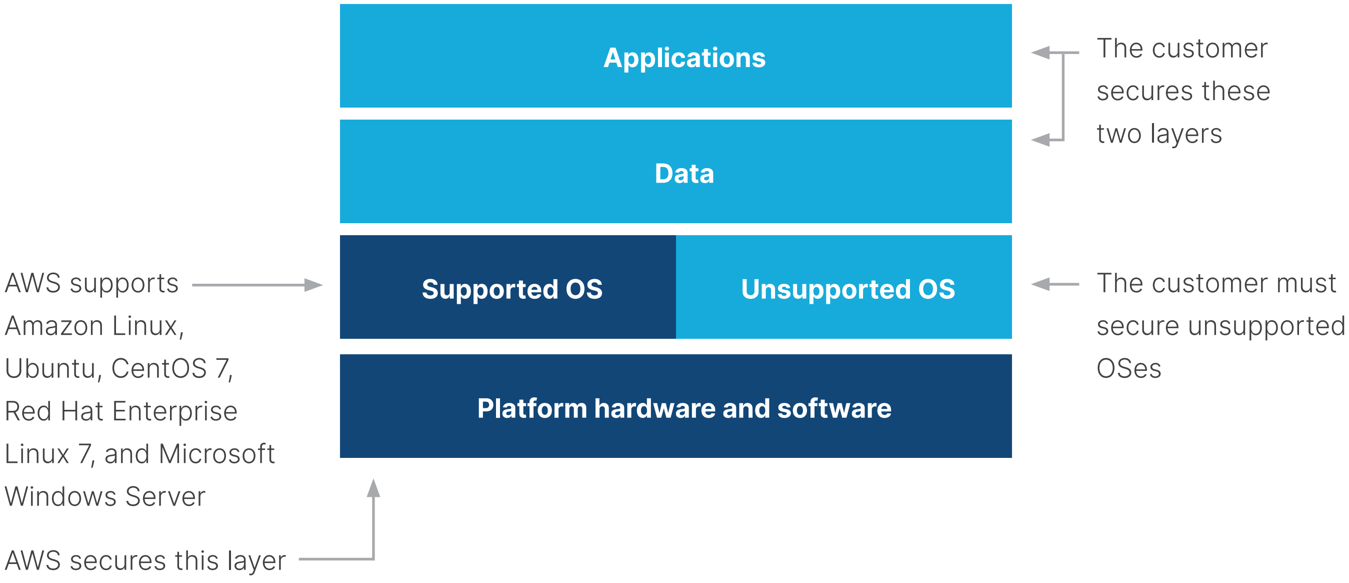
Applications
Data
Supported OS
Unsupported OS
Platform hardware and software
The customer secures these two layers
The customer must secure unsupported OSes
AWS supports Amazon Linux, Ubuntu, CentOS 7, Red Hat Enterprise Linux 7, and Microsoft Windows Server
AWS secures this layer

Figure 3. Shared Responsibility Model for Security

Why is this a good idea? Because each party is responsible for securing the parts of the stack that they know best. AWS built the infrastructure, so their security teams have intimate knowledge of the platform itself. Furthermore, AWS has invested huge sums to secure their infrastructure far better than your security team could ever do – for cloud service providers, poor security equals no business. On the other hand, your team has extensive knowledge about your applications and data and can best determine how to meet your organization's requirements. Therefore, the final decisions about application security and data privacy rest with you, the customer.

That said, you're not entirely on your own. For one thing, AWS has an extensive library of optional security, identity, and compliance tools (see figure 4). Of course, you can also choose your own tools in the security marketplace. And as noted earlier, engaging the services of a managed security services provider (MSSP) or other knowledgeable third party can be key to establishing an effective security posture.

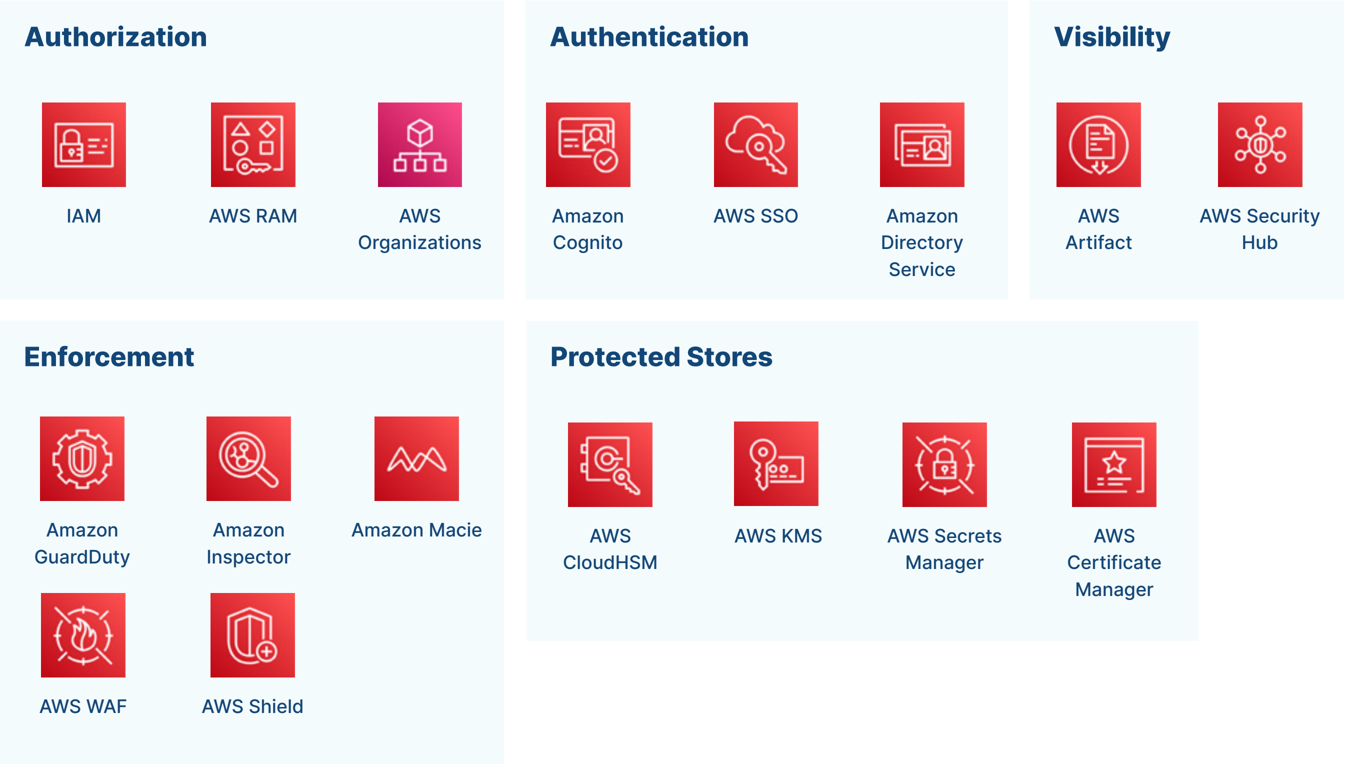

Figure 4. AWS Security, Identity, and Compliance Services (by function type)

3. Finding Your Comfort Zone – AWS Regions and Availability Zones

To meet the highest levels of security, compliance, and data protection, AWS partitions the world into regions. Different regions offer different service qualities in terms of availability of AWS services resources, latency, and cost. Each region is in turn logically partitioned into three to six availability zones (AZs), which are collections of geographically separated data centers. Each AZ has independent power, cooling, and physical security and is connected to the other AZs in the region via redundant, ultra-low-latency networks. AZs are the entities that are visible to customers – you cannot see any individual data centers.

AZs and regions provide a great deal of flexibility and reliability in designing a cloud architecture. Here are two examples:

Using Multiple AZs Improve Application Availability

To improve availability, many network architects use multi-AZ redundancy in which applications are deployed in two AZs. If an entire AZ goes down – a rare occurrence – AWS can fail over your application to another AZ in the same region (see figure 5).

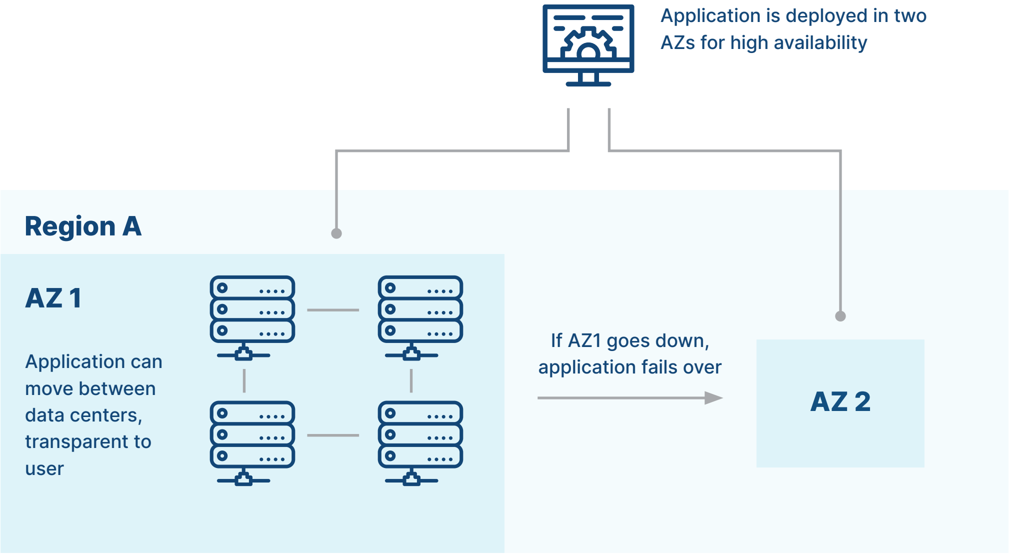
Application is deployed in two AZs for high availability
Region A
AZ 1
Application can move between data centers, transparent to user
If AZ1 goes down, application fails over
AZ 2

Figure 5. Multi-AZ Architecture for Application Availability

Implementing Data Bunkers for Disaster Recovery

Remember the old model for disaster recovery (DR) in which companies stored figurative mountains of information on tapes that were then literally stored inside a real mountain? The tape analogy in the cloud is a data bunker, a secure account that holds critical data in a different region. Information in the data bunker serves as the ultimate recovery point in the event of a major disaster affecting the primary AWS region – an extremely rare occurrence – but one with potentially crippling implications for the business.

Architecturing a data bunker in AWS is straight-forward, requiring just three AWS services: Amazon S3, AWS Organizations, and AWS Cloudtrail (see figure 6).

Data bunkering is gaining in popularity as a defense against ransomware attacks. In this approach, the organization regularly updates a copy of all critical information, including Epic data, in a different region from the DR site. Access to this copy is strictly controlled and the bunkered data is never used unless there is a need to quickly restore key operational information following a successful ransomware attack.

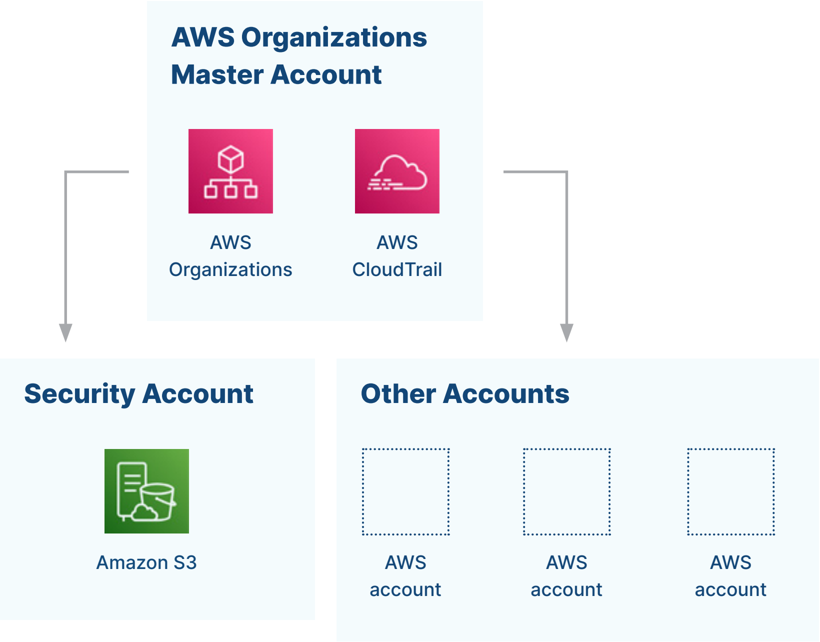

Figure 6. Data Bunker in AWS

4. Joined at the HIPAA – Data Privacy

Data privacy goes hand in hand with security — you need to be confident that your protected health information (PHI) will not be compromised in any way. In the same vein, you also need to demonstrate compliance to a wide range of regulations such as the US Health Insurance Portability and Accountability and US Health Information Technology for Economic and Clinical Health Acts (HIPAA/HITECH), the Federal Risk and Authorization Management Program (FedRAMP), the European Union's General Data Protection Regulation (GDPR), and the Payment Card Industry Data Security Standard (PCI).

Fortunately, AWS has strong controls around data privacy. You alone control access to your information – AWS will not access or use your content for any purpose without your agreement. AWS does provide tools for access management, including AWS Identity and Access Management, AWS Organizations, and AWS CloudTrail, but you are free to use your own.

In accordance with the shared responsibility model discussed earlier, you are solely responsible for securing your own information, both at rest and in transit. That said, AWS offers you encryption features to help streamline the task. More than 100 AWS services include data encryption capabilities that enable you to encrypt, delete, and monitor the processing of your data. AWS also gives you the option to manage your own encryption keys using AWS Key Management Service (KMS).

Data localization is now a big part of doing business globally – more than 100 countries now require their citizen data to be stored in servers physically located within their borders.1 The AWS architecture offers a mechanism for meeting these requirements by allowing you to choose the regions in which you store and back up your information. AWS will not move information to another region without your permission unless mandated to do so by law.

[1] https://www.brinknews.com/data-localization-is-now-a-big-part-of-doing-business-globally/

5. There's No Free Launch – AWS Pricing

The AWS pricing model is simple in concept. There are three general pricing categories: pay as you go, commit to a minimum level of usage, and volume discounts (see figure 7).

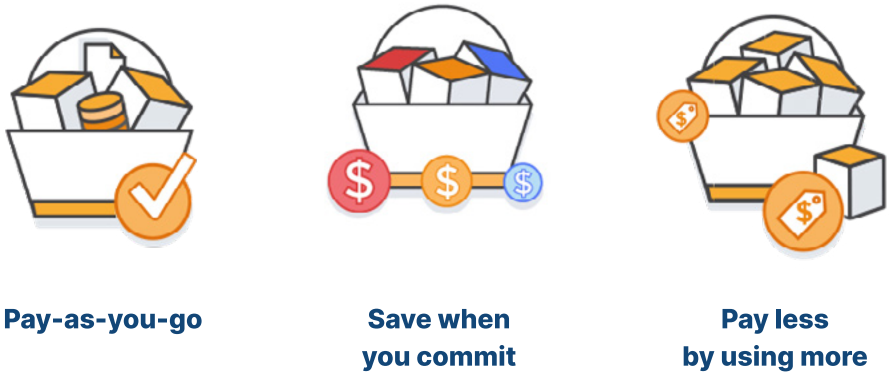

Figure 7. Three Options in the AWS Pricing Model

Pay As You Go – AWS On-Demand Pricing

The original rollout of AWS featured a pay-as-you-go pricing approach for cloud services that's similar to the way you pay for utilities such as water and electricity — and it's still popular today. With this option, you pay only for the individual services you need, for as long as you use them, and without requiring long-term contracts or complex licensing. You only pay for the services you consume, and once you stop using them, there are no additional costs or termination fees.

One caveat concerning On-Demand pricing: It's very easy to run up a large bill for the first month or two if you aren't familiar with how to configure and optimize individual AWS services and control internal access to the self-service portal. This is an area where an experienced MSP can be invaluable.

Make a Commitment – AWS Savings Plans

If you have good visibility into your usage requirements, you can take advantage of AWS Savings Plans. Savings Plans is a flexible pricing model offering lower prices compared to On-Demand pricing, in exchange for a specific usage commitment (measured in $/hour) for a one or three-year period. Many organizations start with on-demand pricing as a way to get up to speed quickly on AWS EC2 services and then move to savings plans to optimize their AWS spend.

Spend and Save – AWS Volume-Based Discounts

AWS offers volume-based discounts that allow you to realize significant savings as your usage increases. For services such as S3, pricing is tiered, meaning the more you use, the less you pay per GB. This approach has the obvious benefit of avoiding a long-term commitment but requires proper configuration to ensure that you get the volume-based pricing and not the pay-as-you-go pricing.

Manage Your AWS Spend Carefully!

Using the pricing model is not as simple as it might appear to be. Some practices that seem innocuous can run up a large bill. For example, under the pay-as-you-go model, you incur costs every hour for every EC2 instance, even if the instance is idle or underutilized. Savings plans such as reserved instances and spot instances can significantly reduce your AWS spend. Upgrading EC2 instances to the latest technology can also provide savings.

In addition, cloud computing in general challenges organizational financial controls, allowing individual engineers to spin up instances and thereby incur substantial costs without going through the traditional approval process. These challenges point to the need to ensure that anyone using AWS resources understands clearly how the AWS pricing model works. Many organizations turn to a knowledgeable managed service provider (MSP) to help optimize their AWS architecture and thereby minimize expenses.

Epic Architecture Overview

This section explains three key concepts of the Epic architecture: the stack, primary and secondary environments, and the disaster recovery environment.

Stack Architecture

The Epic architecture can be visualized as a stack consisting of four layers (see figure 8).

1. The hyperspace presentation layer is the customer-facing part of Epic where clinicians, business users, and managers interact with the Epic clinical information software (CIS) via a Citrix or VMware Horizon virtual desktop.
2. Virtual servers host web services, file system, and printer functions
3. The reporting environment gives analysts tools such as Epic Cogito and Epic Clarity to extract insights for managers and executives,
4. The operational database holds the critical patient EHR information in Chronicles, a real-time database that enables all care and support activities within the Epic CIS layer.

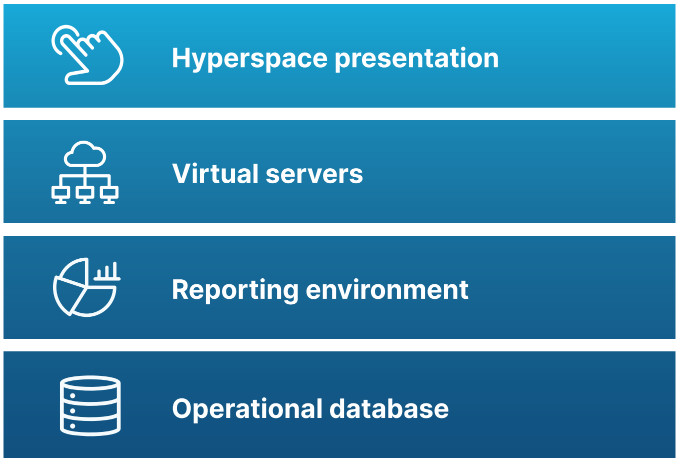

Figure 8. Epic Stack Architecture

Architecture Best Practices

The recommended architecture for Epic on AWS consists of three environments: a production environment running the Epic software; a non-production environment used for development, testing, training, and alternate production/DR; and a data bunker for ransomware recovery (see figure 9).

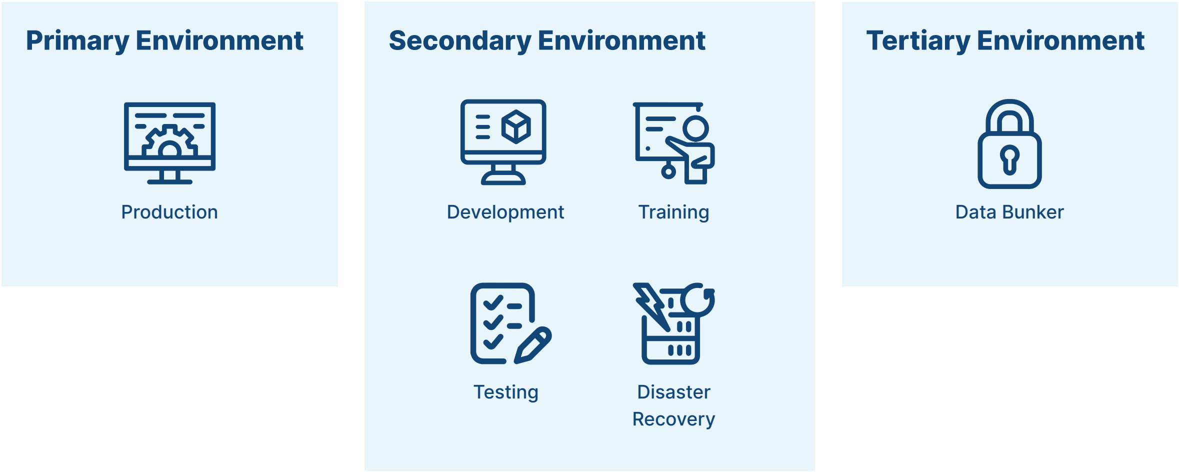

Figure 9. Epic Environments

Alternate Production (Disaster Recovery)

The Epic CIS is the key enabler of the healthcare organization's clinical operations, so unplanned downtime is unacceptable at any level. The disaster recovery hardware and software are always separated geographically from the production environment to ensure business continuity in the event of a natural disaster or human-made disruption to operations (see figure 10).

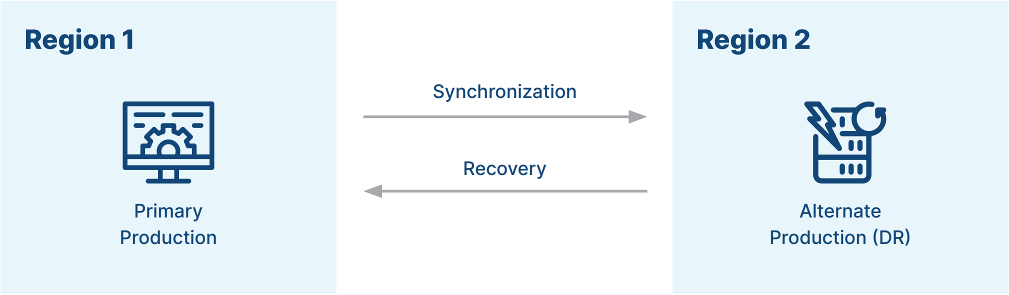

Figure 10. Epic Production Architecture

Deployment Scenarios

Here are three specific ways that real-world customers can realize tangible business value by moving specific parts of their Epic environment to AWS.

Minimize Idle Resources in Disaster Recovery System

In this case, the customer wanted to migrate from a DR system that was on-premise at the DR location to a cloud-based DR. In the original deployment, the number of VMs in the hyperspace and server layers are equal so that, in the case of a failover, the DR site has the necessary computing resources to ensure business continuity. The hyperspace VMs in the DR site are hardly ever used – most of the time, they just sit idle even though the customer is paying for their use.

To make the DR function more efficient without sacrificing utility, we redesigned the hyperspace architecture. The new implementation features just 12 hyperspace VMs that can adsorb the initial load in case of a failover, giving the system time to allocate the additional VMs needed for full functionality. The net result is that the number of hyperspace VMs in steady state drops from 105 to 12, an 89 percent reduction (see table 1). Importantly, these savings come with no loss of efficacy – properly designed, the cloud-based DR system should offer the same recovery time objective (RTO) and recovery point objective (RPO) as the on-premises implementation.

LAYER	PRODUCTION SITE	DR SITE (ON-PREM)	DR SITE (CLOUD)	DELTA	REDUCTION
Hyperspace	105	105	12	93	89%
Server	57	57	40	17	30%
Reporting	11	0	0	0	–
Database	4	1	1	0	–
TOTALS	177	163	53	110	67%

Table 1. Comparison of Number of VMs Needed for DR, On-prem Versus Cloud

Allocate Training Resources Dynamically

Training traditionally has a low overall utilization rate of resources for several reasons. On-premises training facilities are provisioned for the largest possible class size, so unused capacity is a virtual certainty during training sessions. In addition, training groups can rarely schedule their classrooms five days a week, 52 weeks a year, so idle periods are built into the process. CFOs can be forgiven for grimacing when they walk by empty training rooms with a computer on every desk.

The inherent flexibility of cloud computing is a natural fit for the vagaries of training curricula. In a cloud-based training system, on Sunday night the instructor can easily and quickly provision the exact number of training instances needed for the Monday morning session using a purpose-built script. When the class ends on Thursday afternoon, the instructor runs another script that tears down all the student instances, effectively turning off the cost tap (see figure 11).

This approach has a number of advantages. First, training no longer puts strains on the capital budget because operating costs map closely to the delivered services–in other words, you only pay for the computing cycles and storage bytes that you use. In addition, the training staff spends relatively little time managing the training environment each time because setup and teardown are automated via scripts. Finally, the training department no longer has to settle for hand-me-down hardware obsoleted from other departments — now trainers have access to the same cloud-based computing technology as development, testing, and operational groups. As a result, training activities can be made more realistic and therefore more effective.

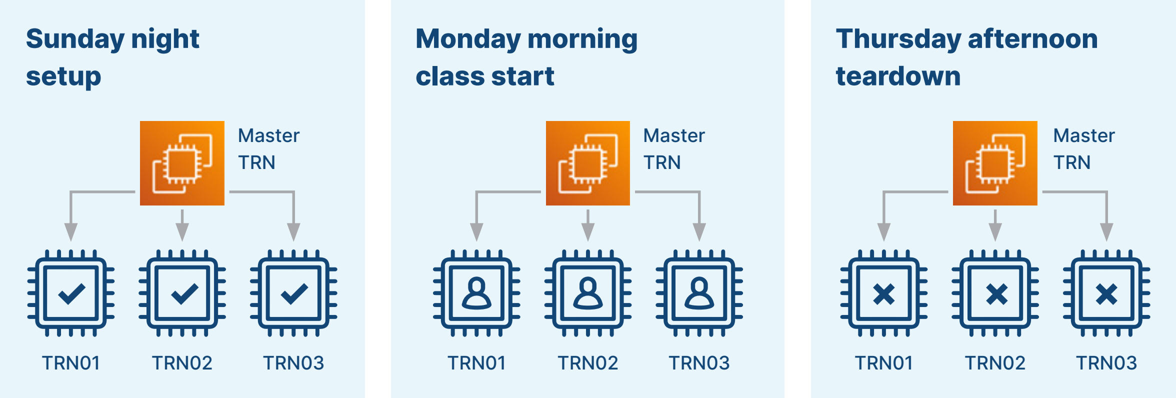

Figure 11. Training Instances Automatically Set Up and Torn Down as Needed

Scale Production with Minimal Disruption

Scaling in an on-premises environment is slow and costly. Adding additional computing or storage capacity requires a lengthy cycle of procurement, installation, and configuration, not to mention a hefty capital investment – two to three weeks is not an uncommon timeframe. In contrast, scaling in AWS is uncomplicated and fast – literally minutes – and requires no capital outlays at all.

Here's what scaling looks like on AWS. In this case, the hospital's Epic production operations are hosted on a single AWS m5.12xlarge instance, which consists of 192 GiB of Memory and 48 vCPUs. (See the full list of AWS instances here.) As the hospital experiences more and more growth, at some point the IT group will need to scale up the number of computing cores and memory to support that growth. Fortunately, that process is relatively straightforward and only takes about 10 minutes (see figure 11):

1. Shut off the production instance.
2. Create an Amazon Machine Image (AMI) of the existing instance.
3. Use the AMI to create a new instance with more compute and storage capacity.
4. Reattach the existing storage to the new instance.

Scaling can also be an opportunity to migrate to the latest technology to gain processing power and reduce overall cost (see figure 12).

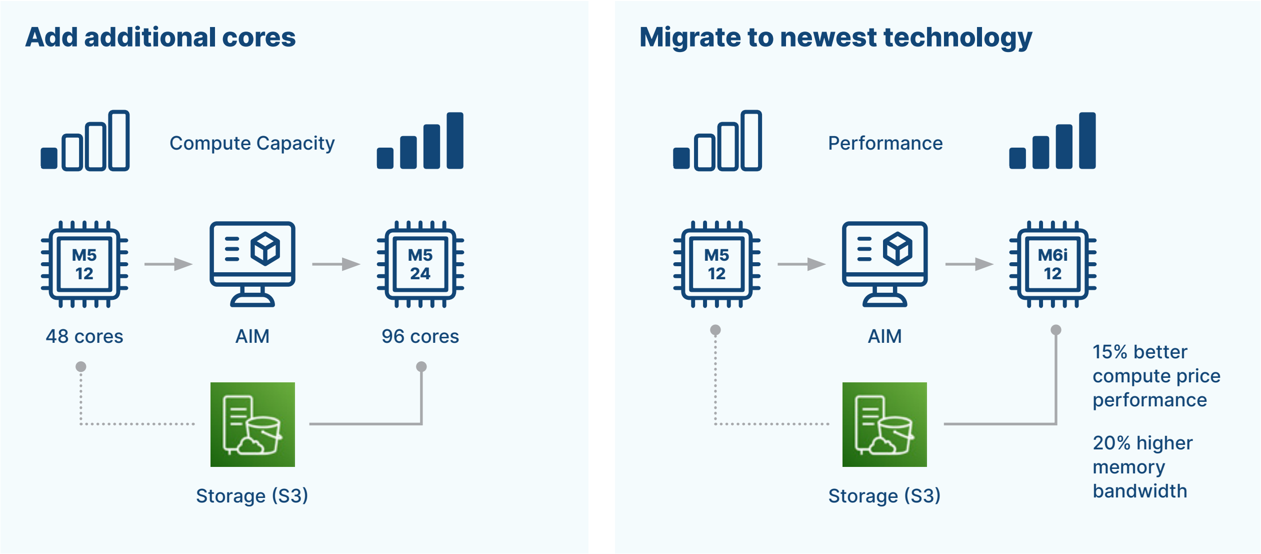

Figure 12. Scaling Options

Next Steps

Ideally, this solution brief has given you a better feel for some of the technical issues that you will likely encounter if you decide to migrate your Epic CIS to AWS (the considerations are similar for other cloud service providers). Here are some alternatives for next steps based on what needs to be done to move the process forward.

- To make the general case for migrating to the cloud: Read the Cloudticity white paper "Why You Should Migrate Your Epic EHR to the Cloud."
- To better understand the practical challenges involved: Read the Cloudticity white paper "Migrating Epic EHR to AWS – Typical Challenges and Solutions."
- To move forward now: Establish a business relationship with a third-party expert who can guide you in the planning process. We strongly urge you to consider engaging an MSP who has the necessary expertise and experience to help you through this critical stage of your Epic migration. Schedule a free consultation to learn how Cloudticity can help.

Connect with a Cloudticity expert

cloudticity

Start Your Epic Migration Today

SCHEDULE A FREE CONSULTATION