Module 1: Cloud Migration Strategies

Mohanraj Shanmugam

Overview

- Many businesses are looking for better ways to migrate their existing applications to a cloud-based infrastructure so that they can enjoy the same advantages seen with greenfield application development in cloud.
- Most cloud gives businesses the freedom of choice to choose the programming models, languages, operating systems and databases they are already using or familiar with. As a result, many organizations are moving existing applications to the cloud today.
- It is true that some applications (“IT assets”) currently deployed in company data centers or co-located facilities might not make technical or business sense to move to the cloud.
- There are several assets within an organization that can be moved to the cloud today with minimal effort.

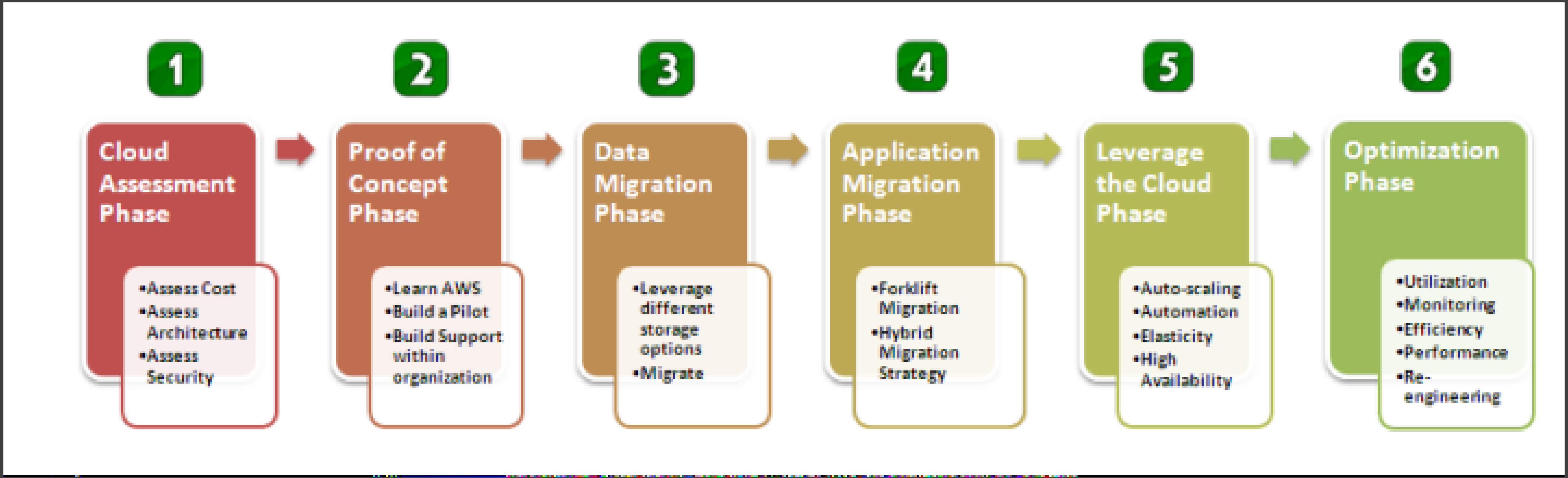
Overview

- The step by step, phase-driven approach, will help you identify ideal projects for migration, build the necessary support within the organization and migrate applications with confidence
- Many organizations are taking incremental approach to cloud migration.
- It is very important to understand that with any migration, whether related to the cloud or not, there are one-time costs involved as well as resistance to change among the staff members (cultural and socio-political impedance).
- Focus on long-term ROI as well as tangible and intangible factors of moving to the cloud and be aware of the latest developments in the cloud so that you can take full advantage of the cloud benefits.

Overview

- A successful migration largely depends on three things:
  - The complexity of the application architecture
  - how loosely coupled your application is
  - how much effort you are willing to put into migration

A Phased Strategy for Migration



1

Cloud Assessment Phase

- Assess Cost
- Assess Architecture
- Assess Security

2

Proof of Concept Phase

- Learn AWS
- Build a Pilot
- Build Support within organization

3

Data Migration Phase

- Leverage different storage options
- Migrate

4

Application Migration Phase

- Forklift Migration
- Hybrid Migration Strategy

5

Leverage the Cloud Phase

- Auto-scaling
- Automation
- Elasticity
- High Availability

6

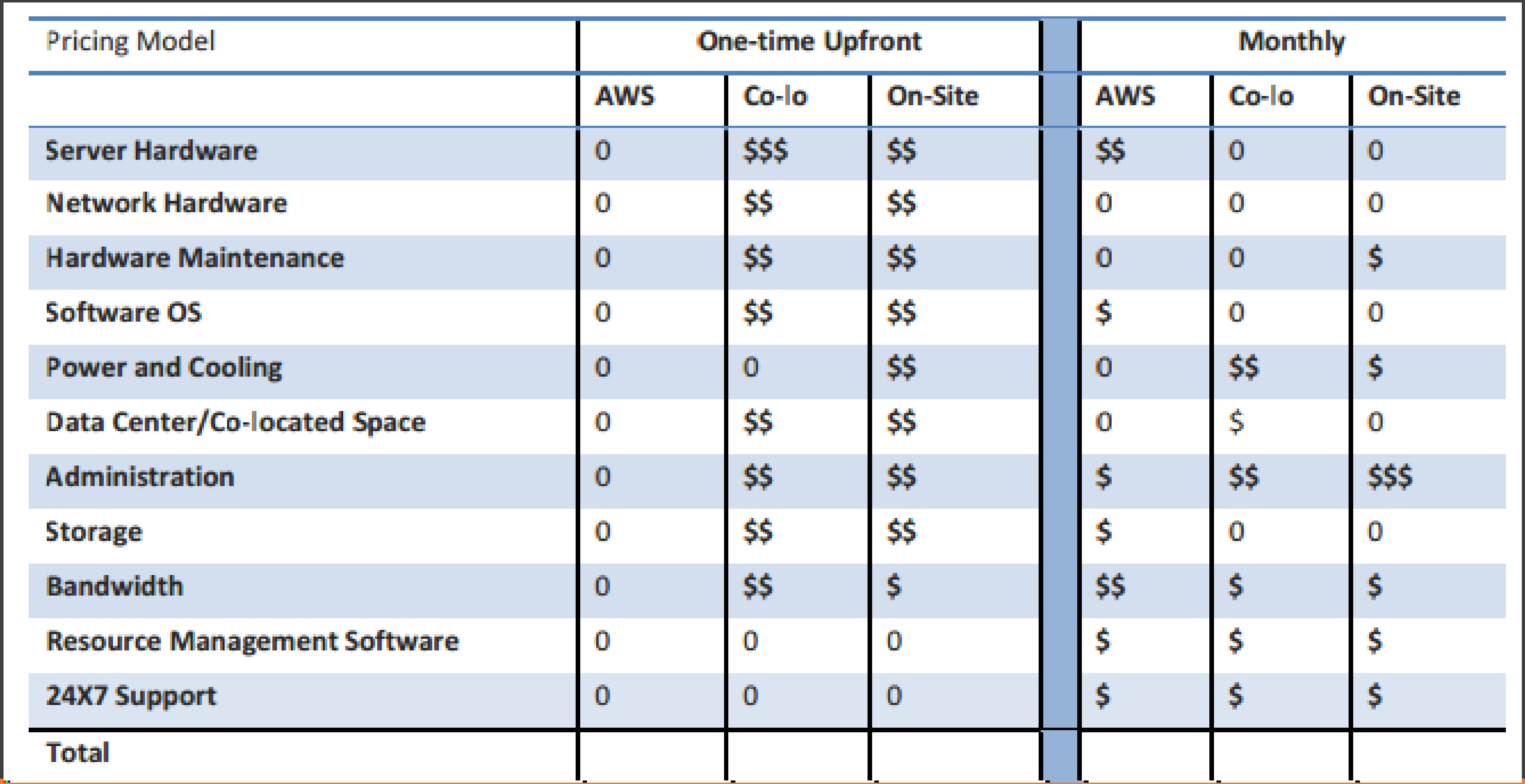
Optimization Phase

- Utilization
- Monitoring
- Efficiency
- Performance
- Re-engineering

Cloud Assessment

- Financial Assessment (TCO calculation)
- Security and Compliance Assessment
- Technical Assessment (Classify application types)
- Identify the tools that can be reused and the tools that need to be built
- Migrate licensed products
- Create a plan and measure success

Financial Assessment



Pricing Model	One-time Upfront			Monthly		
	AWS	Co-lo	On-Site	AWS	Co-lo	On-Site
Server Hardware	0	$$$	$$	$$	0	0
Network Hardware	0	$$	$$	0	0	0
Hardware Maintenance	0	$$	$$	0	0	$
Software OS	0	$$	$$	$	0	0
Power and Cooling	0	0	$$	0	$$	$
Data Center/Co-located Space	0	$$	$$	0	$	0
Administration	0	$$	$$	$	$$	$$$
Storage	0	$$	$$	$	0	0
Bandwidth	0	$$	$	$$	$	$
Resource Management Software	0	0	0	$	$	$
24X7 Support	0	0	0	$	$	$
Total						

Financial Assessment

- http://media.amazonwebservices.com/The_Economics_of_the_AWS_Cloud_vs_Owned_IT_Infrastructure.pdf
- This analysis compares only the direct costs of the IT infrastructure.
- Indirect economic benefits of cloud computing, including high availability, reliability, scalability, flexibility, reduced time-to-market, and many other cloud-oriented benefits.
- Decision makers are encouraged to conduct a separate analysis to quantify the economic value of these features.
- Amazon Monthly Cost Calculator:
  - http://aws.amazon.com/calculator

Security and Compliance Assessment

- If organization has specific IT security policies and compliance requirements, recommend that you involve your security advisers and auditors early in the process.
- At this stage, you can ask the following questions:
  - What is my overall risk tolerance? Are there various classifications of my data that result in higher or lower tolerance to exposure?
  - What are my main concerns around confidentiality, integrity, availability, and durability of my data?
  - What are my regulatory or contractual obligations to store data in specific jurisdictions?
  - What are my security threats? What is a likelihood of those threats materializing into actual attacks?

Security and Compliance Assessment

- Am I concerned about intellectual property protection and legal issues of my application and data?
- What are my options if I decide that I need to retrieve all of my data back from the cloud?
- Are there internal organizational issues to address to increase our comfort level with using shared infrastructure services?

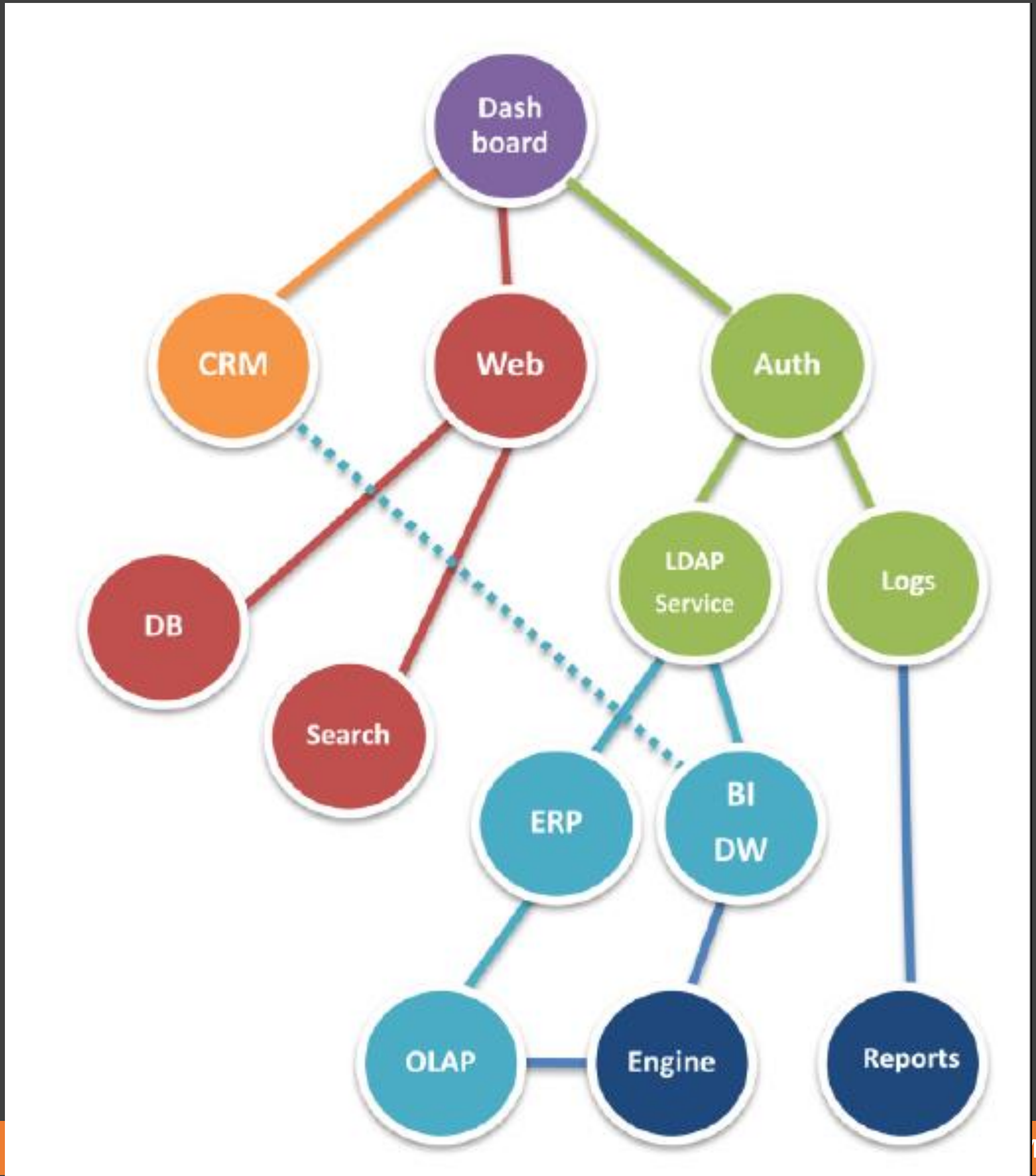
AWS Security Considerations

- You own the data, not AWS.
- You choose which geographic location to store the data. It doesn't move unless you decide to move it.
- You can download or delete your data whenever you like.
- You should consider the sensitivity of your data, and decide if and how you will encrypt your data while it is in transit and while it is at rest.
- You can set highly granular permissions to manage access of a user within your organization to specific service operations, data, and resources in the cloud for greater security control.

Technical and Functional Assessment

- A technical assessment is required to understand which applications are more suited to the cloud architecturally and strategically. At
- enterprise architects should ask the following questions:
  - Which business applications should move to the cloud first?
  - Does the cloud provide all of the infrastructure building blocks we require?
  - Can we reuse our existing resource management and configuration tools?
  - How can we get rid of support contracts for hardware, software and network?

Technical and Functional Assessment



- **Create a Dependency Tree and a Classification Chart** based on their dependencies, risks, and security and compliance requirements
- ***Classify your applications*** *into different categories:*
  - Applications with Top Secret, Secret, or Public data sets
  - *Applications with low, medium and high compliance requirements*
  - *Applications that are internal-only, partner-only or customer-facing*
  - *Applications with low, medium and high coupling*
  - Applications with strict, relaxed licensing

Identifying the Right “Candidate” for the Cloud

- For a Web-based application or Software as a Service (SaaS) application, the dependency tree will consist of logical components (features) of the website such as database, search and indexer, login and authentication service, billing or payments, and so on.
- For backend processing pipeline, there will be different interconnected processes like workflow systems, logging and reporting systems and ERP or CRM systems.
- **The best candidates for the cloud are the services or components that have minimum upward and downward dependencies.**
- **Some examples are backup systems, batch processing applications, log processing systems, development, testing and build systems, web-front (marketing) applications, queuing systems, content management systems, or training and pre-sales demo systems.**

Good Candidates for Cloud

- Applications with under-utilized assets;
- Applications that have an immediate business need to scale and are running out of capacity;
- Applications that have architectural flexibility;
- Applications that utilize traditional tape drives to backup data; applications that require global scale (for example, customer-facing marketing and advertising apps)
- Applications that are used by partners.
- **Deprioritize applications that require specialized hardware to function (for example, mainframe or specialized encryption hardware).**

Prioritize your list of applications

- Once you have the list of ideal candidates, prioritize your list of applications so that it helps you :
  - Maximize the exposure in all aspects of the cloud (compute, storage, network, database)
  - Build support and awareness within your organization and creates highest impact and visibility among the key stakeholders.

Technical and Functional Assessment Checklist

- Are you able to map the current architecture of the candidate application to cloud architecture? If not, how much effort would refactoring require?
- Can your application be packaged into a virtual machine (VM) instance and run on cloud infrastructure or does it need specialized hardware and/or special access to hardware that the AWS cloud cannot provide?
- Is your company licensed to move your third-party software used in the candidate application into the cloud?
- How much effort (in terms of building new or modifying existing tools) is required to move the application?
- Which component must be local (on-premise) and which can move to the cloud?
- What are the latency and bandwidth requirements?
- Does the cloud support the identity and authentication mechanism you require?

Identify the Tools That You Can Reuse

- Identify the tools that you can reuse in the cloud without any modification
- estimate how much effort (in terms of new development and deployment effort) will be required to add "AWS support" to them.
- You might be able to reuse most of the system tools and/or add AWS support very easily.
- Example
  - **Resource Management Tools**: In the cloud, you deal with abstract resources (AMIs, Amazon EC2 instances, Amazon S3 buckets, Amazon EBS volumes and so on). You are likely to need tools to manage these resources.
  - **Resource Configuration Tools**: Take a look at open source tools like Chef, Puppet, and CFEngine, etc.
  - **System Management Tools**: After you deploy your services, you might need to modify your existing system management tools (NOC) so that you can effectively monitor, deploy and "watch" the applications in the cloud. To manage Amazon Virtual Private Cloud resources, you can use the same security policies and use the same system management tools you are using now to manage your own local resources.
  - **Integration Tools:** You will need to identify the framework/library/SDK that works best for you to integrate with AWS services.

Migrating Licensed Products

- **Bring Your Own License (BYOL)**
  - Amazon has teamed with variety of ISVs who have permitted the use of their product on Amazon EC2.
  - This EC2-based license is the most friction-free path to move your software into the cloud.
  - You purchase the license the traditional way or use your existing license and apply it to the product which is available as a pre-configured Amazon Machine Image.
  - This EC2-based license is the most friction-free path to move your software into the cloud. You purchase the license the traditional way or use your existing license and apply it to the product which is available as a pre-configured Amazon Machine Image.
  - If you don’t find the software that you are looking for in the AWS cloud, talk to your software vendor about making their software available in the cloud.

Use a Utility Pricing Model with a Support Package

- Amazon has teamed with elite ISVs and they are offering their software as a Paid AMI
- This is a Pay-As-You-Go license in which you do not incur any upfront licensing cost and only pay for the resources you consume.
- ISVs charge a small premium over and above the standard Amazon EC2 cost which gives you an opportunity to run any number of instances in the cloud for the duration you control.
- For example*, RedHat, Novell, IBM, Wowza offer pay-as-you-go licenses.* ISVs, typically, also offer a support package that goes with pay-as-you-go license.

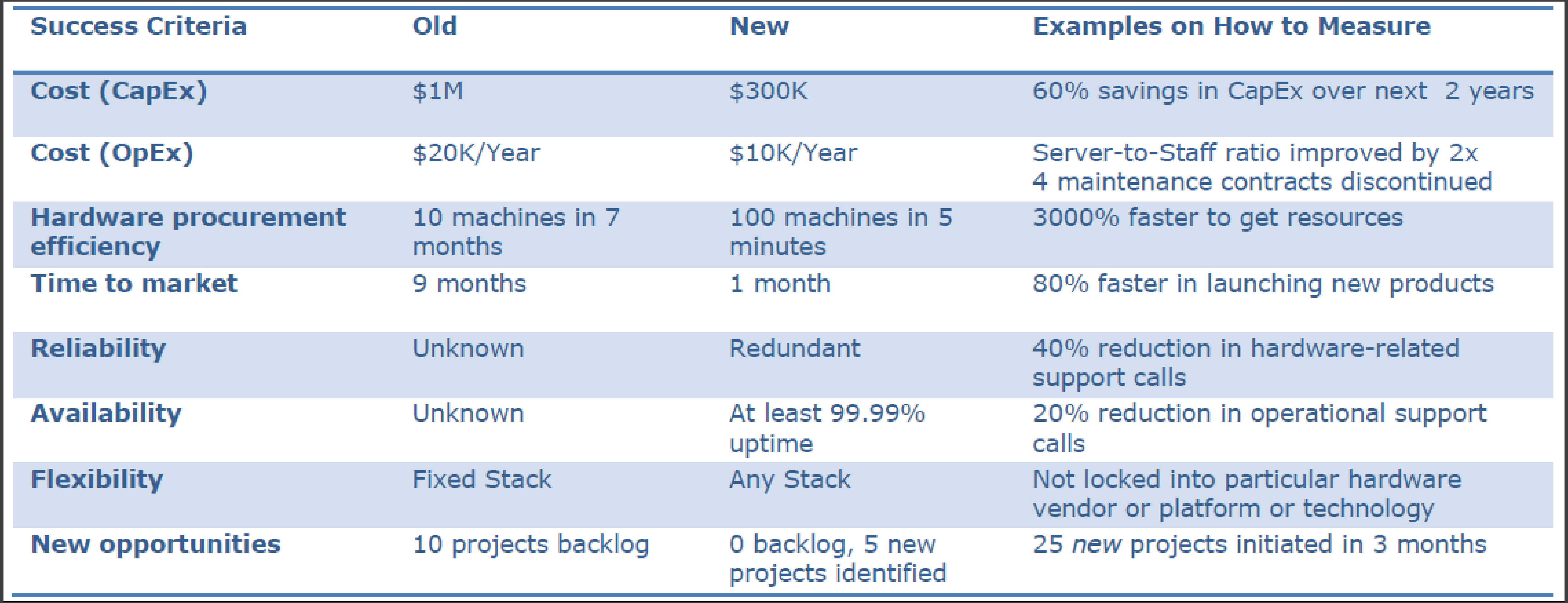
Use an ISV SaaS-based Cloud Service

- Some of the ISVs have offered their software as a service and charge a monthly subscription fee.
- They offer standard APIs and web-based interfaces and are fairly quick to implement.
- This offering is either fully or partially managed inside the AWS cloud.
- This option is often the easiest and fastest way to migrate your existing on-premise installation to a hosted on-demand offering by the same vendor or an equivalent offering by a different vendor. In most cases, ISVs or independent third-party enterprise cloud services integrators offer migration tools that can help you move your data.
- For example, *Mathematica, Quantivo, Pervasive and Cast Iron provide a SaaS offering based on AWS.*

On-Premise and Cloud Integration

- If your enterprise applications are tightly coupled with complex third-party enterprise software systems that have not yet been migrated to the AWS cloud or if you have already invested in multi-year on-premise licensing contracts with the vendor, you should consider refactoring your enterprise applications into functional building blocks.
- Run what you can in the cloud and connect to the licensed software systems that still run on-premise.
- Amazon VPC may be used to create an IPSec VPN or Direct connect tunnel that will allow resources running on AWS to communicate securely with resources at the other end of the tunnel in your existing data center.

Define Your Success Criteria



Success Criteria	Old	New	Examples on How to Measure
Cost (CapEx)	$1M	$300K	60% savings in CapEx over next 2 years
Cost (OpEx)	$20K/Year	$10K/Year	Server-to-Staff ratio improved by 2x 4 maintenance contracts discontinued
Hardware procurement efficiency	10 machines in 7 months	100 machines in 5 minutes	3000% faster to get resources
Time to market	9 months	1 month	80% faster in launching new products
Reliability	Unknown	Redundant	40% reduction in hardware-related support calls
Availability	Unknown	At least 99.99% uptime	20% reduction in operational support calls
Flexibility	Fixed Stack	Any Stack	Not locked into particular hardware vendor or platform or technology
New opportunities	10 projects backlog	0 backlog, 5 new projects identified	25 *new* projects initiated in 3 months

Create a Roadmap and a Plan

- By documenting the
  - dependencies,
  - creating a dependency tree, and
  - identifying the tools that you need to build or customize,
- you will get an idea of how to prioritize applications for migration, estimate the effort required to migrate them, understand the one-time costs involved and assess the timeline.
- You can construct a cloud migration roadmap.

Cloud Assessment Benefits

- Business case for migration
  - Lower TCO,
  - faster time to market,
  - higher flexibility & agility,
  - scalability + elasticity
- Identify gaps between your current traditional legacy architecture and next -generation cloud architecture

Phase 2: Proof of Concept

- Get your feet wet with AWS
- Build a pilot and validate the technology
- Test existing software in the cloud

- Benefits
  - Build confidence with various AWS services
  - Mitigate risk by validating critical pieces of your proposed architecture

Proof of Concept

- Build a proof-of-concept that represents a microcosm of your application, or which tests critical functionality of your application in the cloud environment.
- Start with a small database (or a dataset); don’t be afraid of launching and terminating instances, or stress-testing the system
- For example, if you are thinking of migrating a web application, you can start by deploying miniature models of all the pieces of your architecture (database, web application, load balancer) with minimal data.
- In the process, learn how to build a Web Server AMI, how to set the security group so that only the web server can talk to the app server, how to store all the static files on Amazon S3 and mount an EBS volume to the Amazon EC2 instance, how to manage/monitor your application using Amazon CloudWatch and how to use IAM to restrict access to only the services and resources required for your application to function

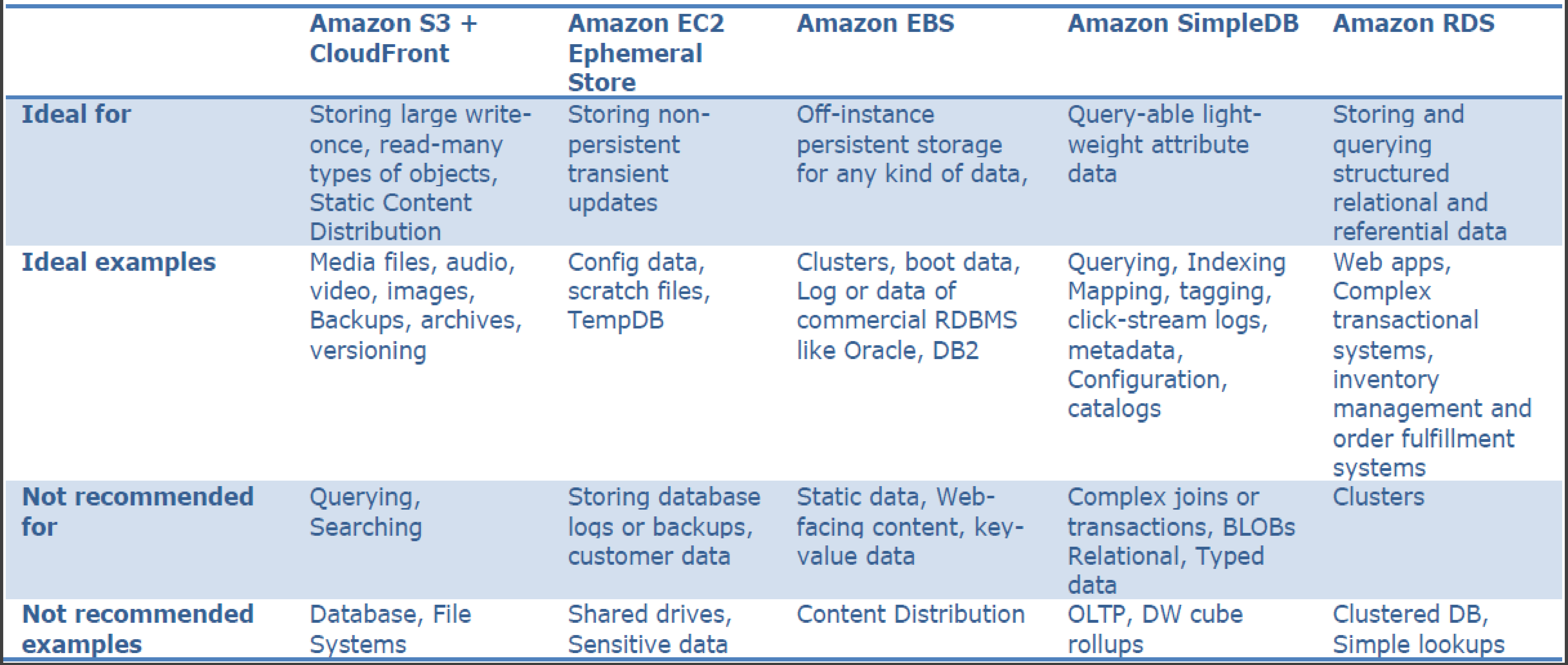
POC Checklist

- Did I learn the basic AWS terminology (instances, AMIs, volumes, snapshots, distributions, domains and so on)?
- Did I learn about many different aspects of the AWS cloud (compute, storage, network, database, security) by building this proof of concept?
- Will this proof of concept support and create awareness of the power of the AWS cloud within the organization?
- What is the best way to capture all the lessons that I learned? A whitepaper or internal presentation?
- How much effort is required to roll this proof-of-concept out to production?
- Which applications can I immediately move after this proof of concept?

Phase 3: Data Migration Phase

- Architects should ask following questions:
    - What are the different storage options available in the cloud today?
    - What are the different RDBMS (commercial and open source) options available in the cloud today?
    - What is my data segmentation strategy? What trade-offs do I have to make?
    - How much effort (in terms new development, one-off scripts) is required to migrate all my data to the cloud?

Phase 3: Data Migration Phase

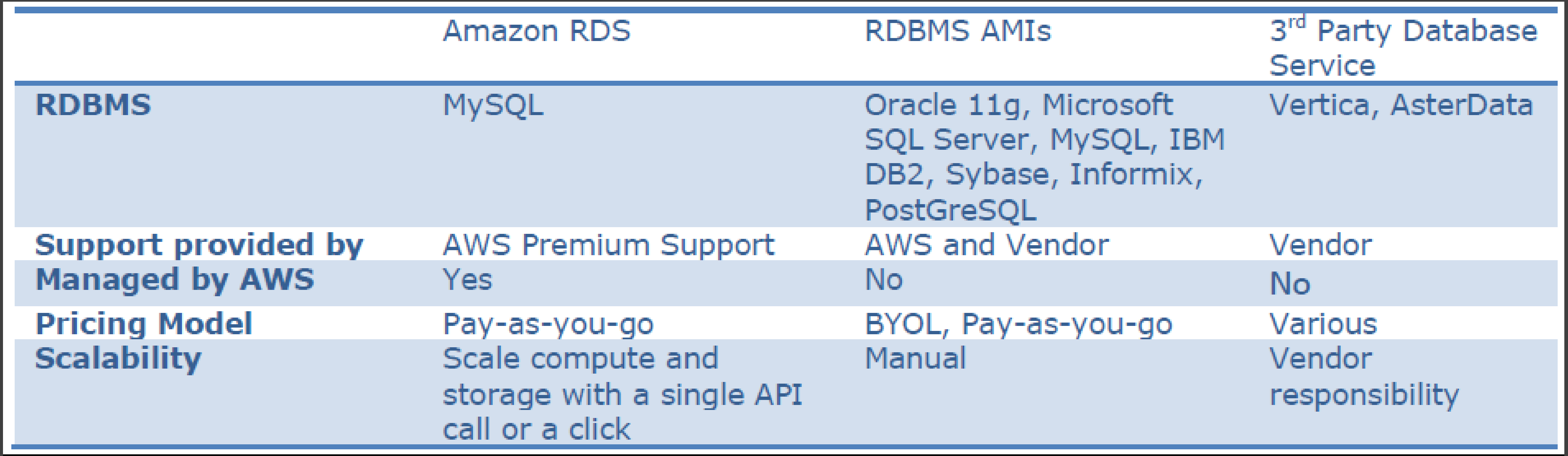


	Amazon S3 + CloudFront	Amazon EC2 Ephemeral Store	Amazon EBS	Amazon SimpleDB	Amazon RDS
Ideal for	Storing large write-once, read-many types of objects, Static Content Distribution	Storing non-persistent transient updates	Off-instance persistent storage for any kind of data,	Query-able light-weight attribute data	Storing and querying structured relational and referential data
Ideal examples	Media files, audio, video, images, Backups, archives, versioning	Config data, scratch files, TempDB	Clusters, boot data, Log or data of commercial RDBMS like Oracle, DB2	Querying, Indexing Mapping, tagging, click-stream logs, metadata, Configuration, catalogs	Web apps, Complex transactional systems, inventory management and order fulfillment systems
Not recommended for	Querying, Searching	Storing database logs or backups, customer data	Static data, Web-facing content, key-value data	Complex joins or transactions, BLOBs Relational, Typed data	Clusters
Not recommended examples	Database, File Systems	Shared drives, Sensitive data	Content Distribution	OLTP, DW cube rollups	Clustered DB, Simple lookups

Migrate your Fileserver systems, Backups and Tape Drives to Amazon S3

- If your existing infrastructure consists of Fileservers, Log servers, Storage Area Networks (SANs) and systems that are backing up the data using tape drives on a periodic basis, you should consider storing this data in Amazon S3.
- Existing applications can utilize Amazon S3 without major change. If your system is generating data every day, the recommended migration flow is to point your “pipe” to Amazon S3 so that new data is stored in the cloud right away.
- Then, you can have an independent batch process to move old data to Amazon S3.
- Most enterprises take advantage of their existing encryption tools (256-bit AES for data at-rest, 128-bit SSL for data in-transit) to encrypt the data before storing it on Amazon S3.

Understand various RDBMS options in the AWS cloud



	Amazon RDS	RDBMS AMIs	3rd Party Database Service
RDBMS	MySQL	Oracle 11g, Microsoft SQL Server, MySQL, IBM DB2, Sybase, Informix, PostGreSQL	Vertica, AsterData
Support provided by	AWS Premium Support	AWS and Vendor	Vendor
Managed by AWS	Yes	No	No
Pricing Model	Pay-as-you-go	BYOL, Pay-as-you-go	Various
Scalability	Scale compute and storage with a single API call or a click	Manual	Vendor responsibility

Migrate your databases to Amazon RDS

- If you use a standard deployment of MySQL, Maria DB or Postgre SQL moving to Amazon RDS will be a trivial task.
- Using standard tools like mysql dump, Oracle Migration tools you will be able to move and restore all the data into an Amazon RDS DB instance.
- After you move the data to a DB instance, make sure you are monitoring all the metrics you need.
- It is also highly recommended that you set your retention period so AWS can automatically create periodic backups

Move Large Amounts of Data using Amazon Import/Export Service

- When transferring data across the Internet becomes cost or time prohibitive
- **AWS Import/Export Snowball** With AWS Import/Export Service, you load your data on snowball devices and ship them via a carrier to AWS.
- AWS then uploads the data into your designated buckets in Amazon S3.
- Copy Data from S3 to EBS Volume
- Upload the Data to RDS
- The full backup is restored by the import service and your incremental backups are transferred over the Internet and applied to the DB Instance in the cloud.
- Once the last incremental backup is applied, you can begin using the new database server.

Phase 4: Application Migration Phase

- Two main application migration strategies:
  - Forklift Migration Strategy and
  - Hybrid Migration Strategy.

Forklift Migration Strategy

- Rather than moving pieces of the system over time, forklift or “pick it all up at once” and move it to the cloud.
  - Stateless applications
  - tightly coupled applications
  - self-contained applications might be better served by using the forklift approach.
  - Components of a 3-tier web application that require extremely-low latency connectivity between them to function and cannot afford internet latency might be best suited to this approach if the entire application including the web, app and database servers, is moved to the cloud all at once.

Forklift Migration Strategy

- Copying your application binaries
- Creating and configuring Amazon Machine Images
- Create the VPC Tiered network
- Setting up security groups and elastic IP addresses
- DNS
- Switching to Amazon RDS databases.
- Deploy Backup, Monitoring and Management tools
- Having a backup strategy, a rollback strategy and performing end-to-end testing is a must when using this strategy

Hybrid Migration Strategy

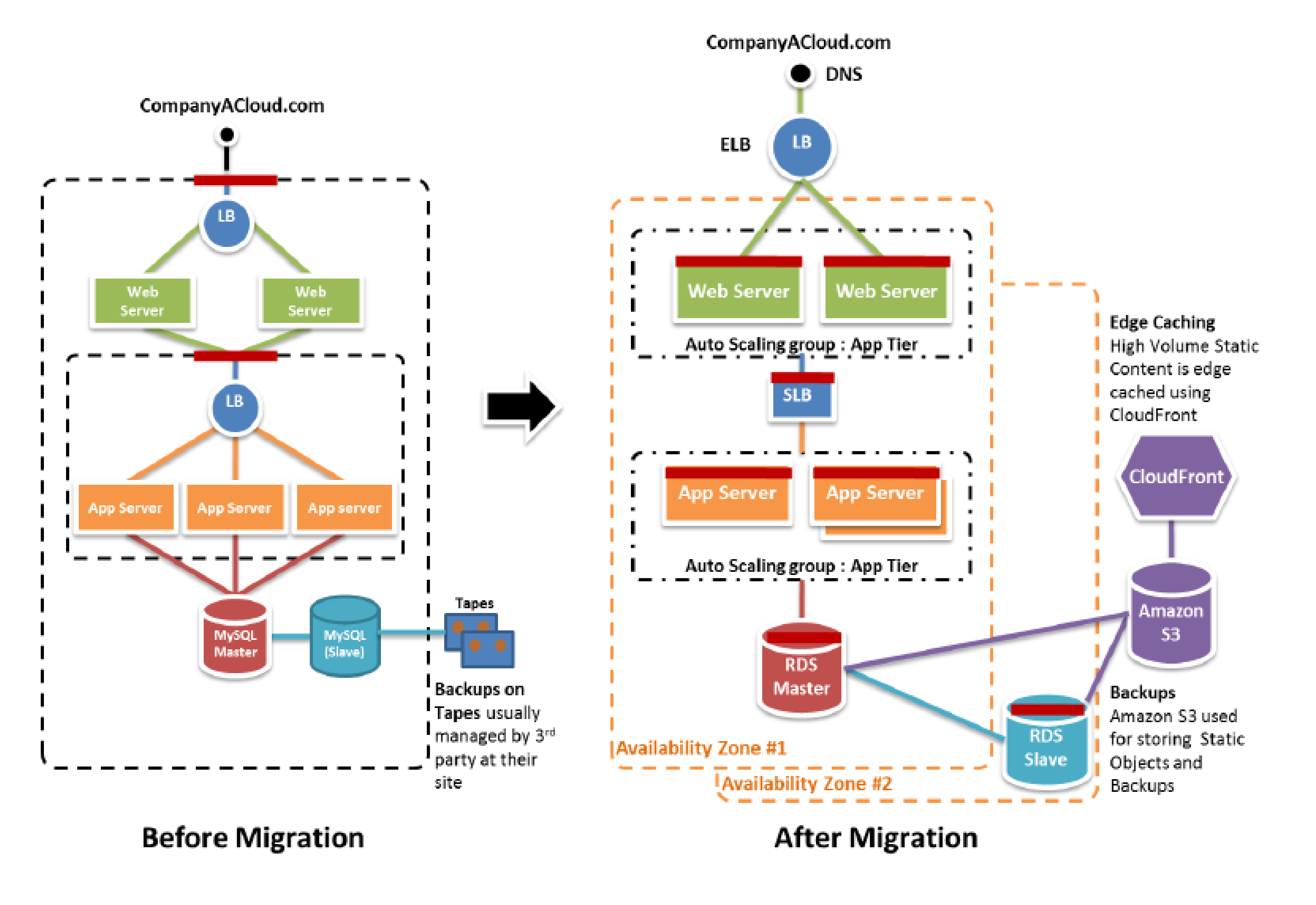
- A hybrid migration consists of taking some parts of an application and moving them to the cloud while leaving other parts of the application in place.
- The hybrid migration strategy can be a low-risk approach to migration of applications to the cloud.
- Rather than moving the entire application at once, parts can be moved and optimized one at a time.
- For example, if you have a website and several batch processing components (such as indexing and search) that power the website, you can consider using this approach. The batch processing system can be migrated to the cloud first while the website continues to stay in the traditional data center.
- In this strategy, you might have to design, architect and build temporary “wrappers” to enable communication between parts residing in your traditional datacenter and those that will reside in the cloud. These wrappers can be made “cloud-aware” and asynchronous (using Amazon SQS queues, wherever applicable) so that they are resilient to changing internet latencies

Phase 5: Leverage the Cloud

- Now that I have migrated existing applications, what else can I do in order to leverage the elasticity and scalability benefits that the cloud promises? What do I need to do differently in order to implement elasticity in my applications?
- How can I take advantage of some of the other advanced AWS features and services?
- How can I automate processes so it is easier to maintain and manage my applications in the cloud?
- What do I need to do specifically in my cloud application so that it can restore itself back to original state in an event of failure (hardware or software)?

Sample Migration Examples

Web App Migration



Backups on Tapes usually managed by 3rd party at their site

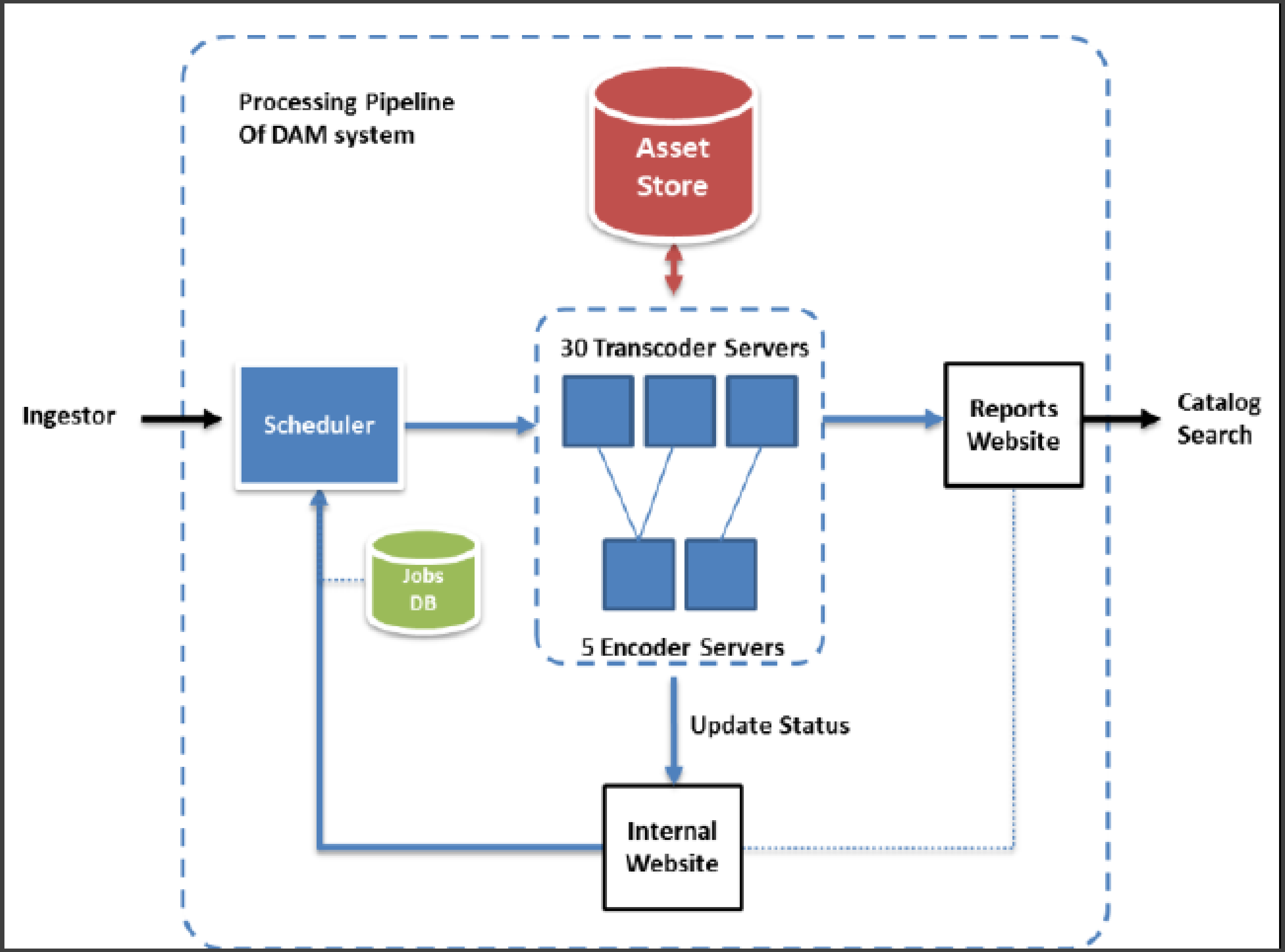
Edge Caching
High Volume Static Content is edge cached using CloudFront

Backups
Amazon S3 used for storing Static Objects and Backups

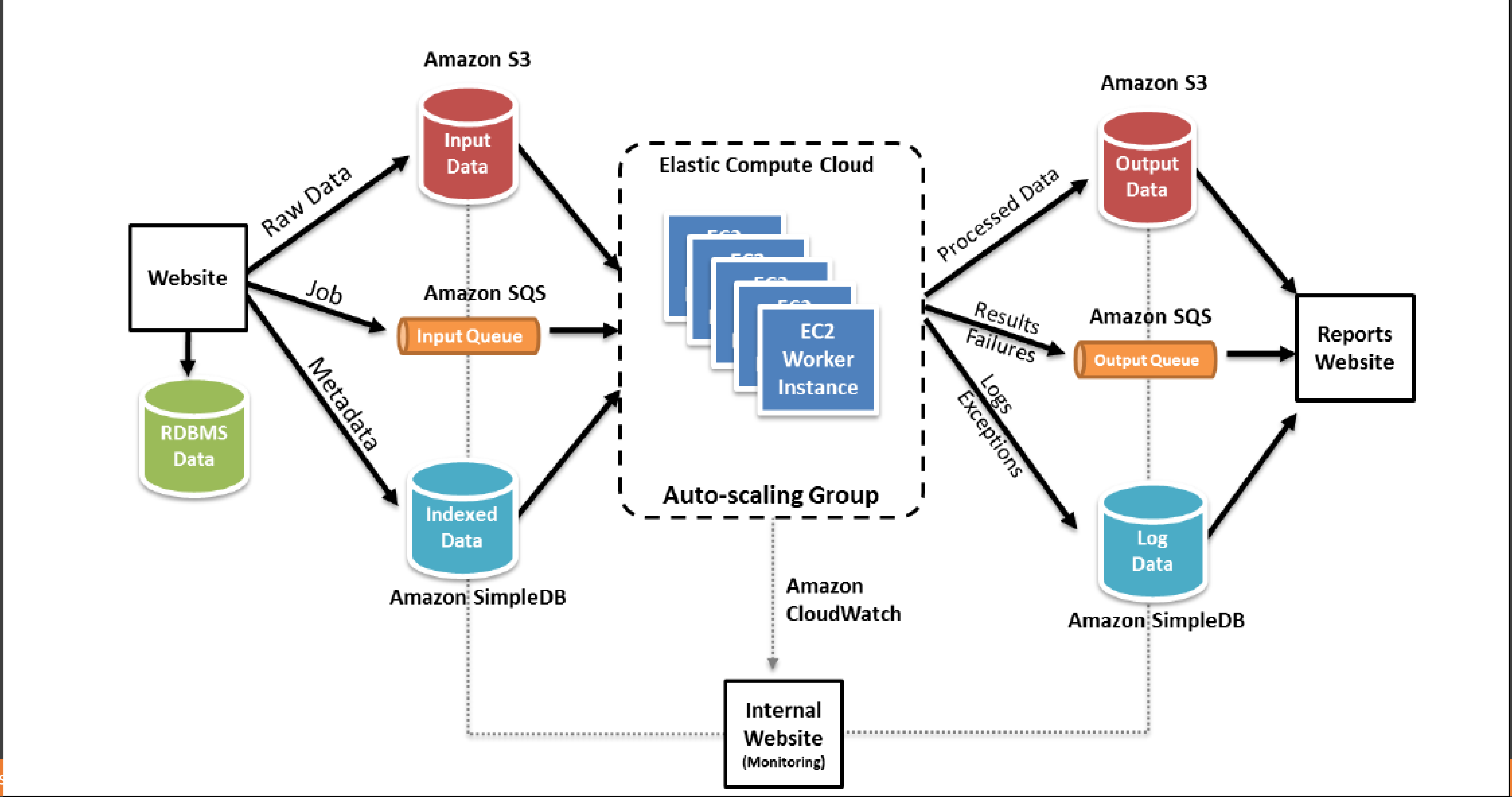
Before Migration

After Migration

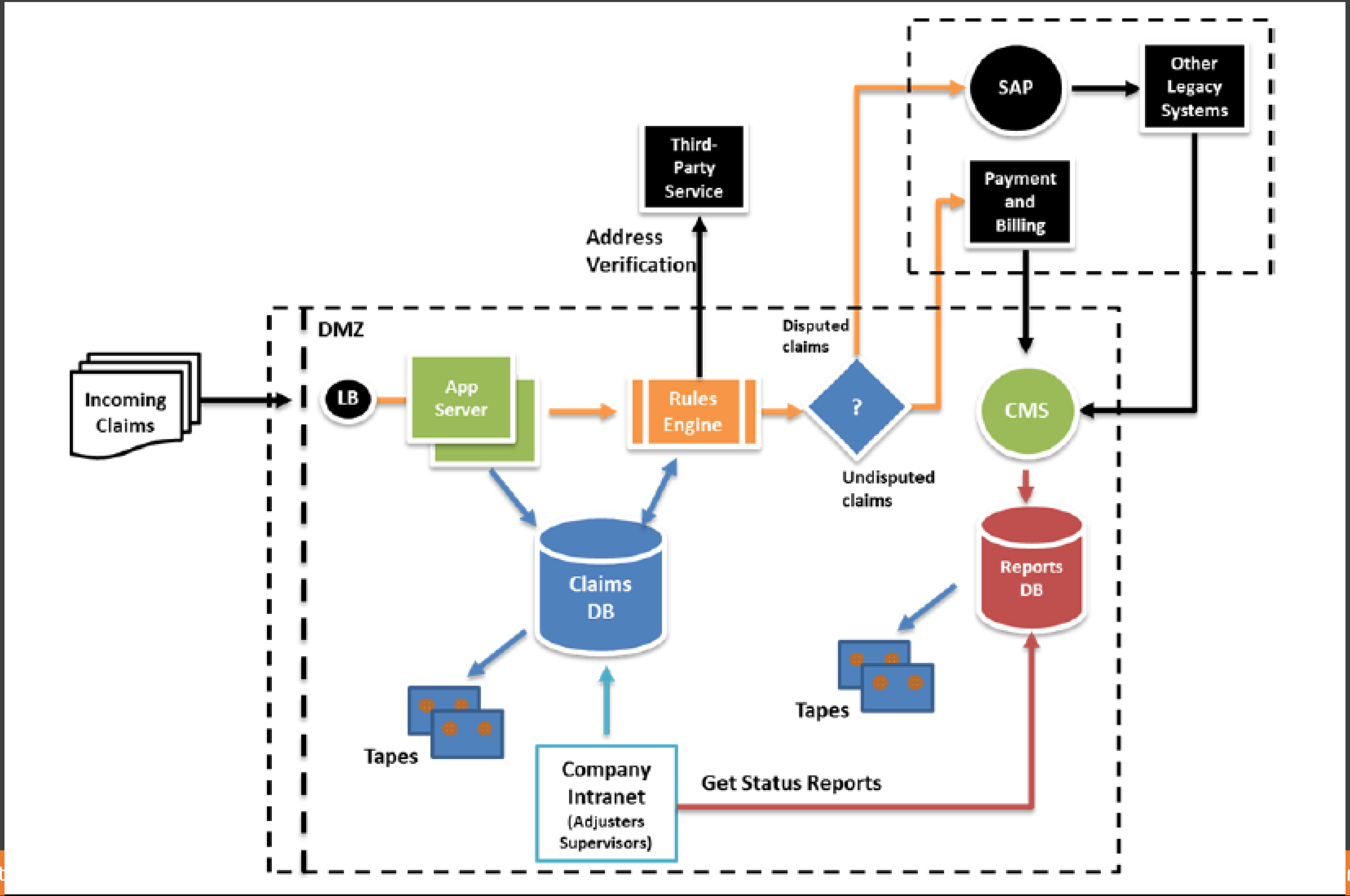
Batch Processing



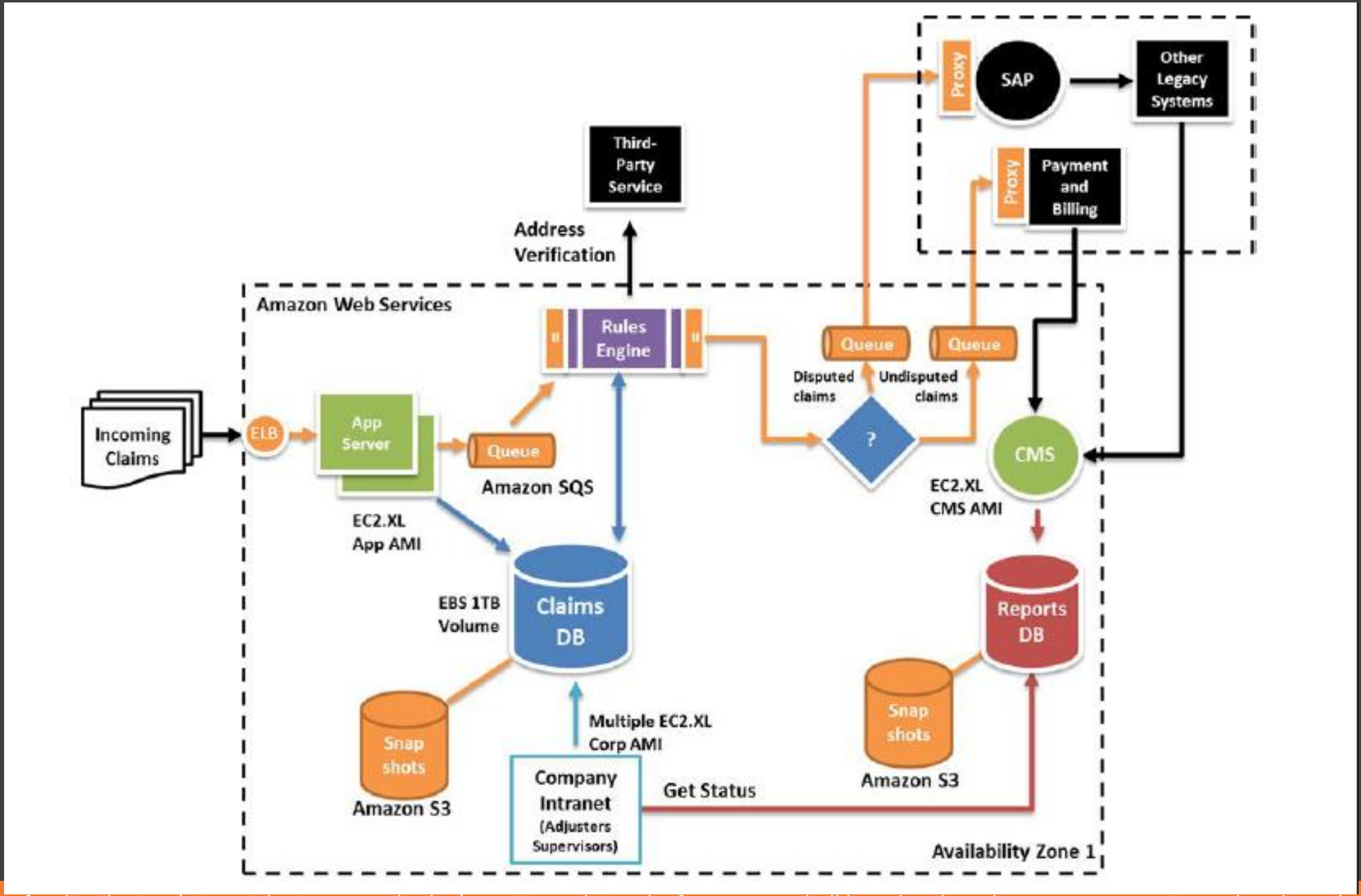
Batch Processing



Backend Processing



Backend Processing



Leverage AWS services

- Auto Scaling Service
- Amazon CloudFront
- Amazon Elastic MapReduce
- Automate Elasticity
  - Maintain Inventory of AMIs
  - Maintain a Golden AMI and fetch binaries on boot
  - Maintain a Just-Enough-OS AMI and a library of recipes or install scripts

Leverage AWS services

- Harden Security
  - Safeguard your AWS credentials
  - Restrict users to AWS resources
  - Protect your data by encrypting it at-rest (AES) and in-transit (TLS)
  - Adopt a recovery strategy
- Automate the In-cloud Software Development Lifecycle and Upgrade Process
- Create a Dashboard of your Elastic Datacenter to Manage AWS Resources

Leverage AWS services

- Create a Business Continuity Plan and Achieve High Availability (Leverage Multiple Availability Zones)
  - Data replication strategy (source, destination, frequency) of databases (Amazon EBS)
  - Data backup and retention strategy (Amazon S3 and Amazon RDS)
  - Creating AMIs with the latest patches and code updates (Amazon EC2)
  - Recovery plan to fail back to the corporate data center from the cloud post-disaster

Phase 6: Optimization Phase

- In this phase, you should focus on how you can optimize your cloud-based application in order to increase cost savings.
- Since you only pay for the resources you consume, you should strive to optimize your system whenever possible.
- In most cases, you will see immediate value in the optimizations.
- A small optimization might result in thousands of dollars of savings in your next monthly bill.

Phase 6: Optimization Phase

- How can I use some of the other AWS features and services in order to further reduce my cost?
- How can I improve the efficiency (and reduce waste) in my deployment footprint?
- How can I instrument my applications to have more visibility of my deployed applications? How can I set metrics for measuring critical application performance?
- Do I have the necessary cloud-aware system administration tools required to manage and maintain my applications?
- How can I optimize my application and database to run in more elastic fashion?

Understanding your Usage Patterns

- With the cloud, you don’t have to master the art of capacity planning because you have the ability to create an automated elastic environment.
- If you can understand, monitor, examine and observe your load patterns, you can manage this elastic environment more effectively.
- You can be more proactive if you understand your traffic patterns.
- For example, if your customer-facing website, deployed in AWS global infrastructure, does not expect any traffic from certain part of the world in certain time of the day, you can scale down your infrastructure in that AWS region for that time.
- The closer you can align your traffic to cloud resources you consume, the higher the cost savings will be.

Terminate the Under-Utilized Instances

- Inspect the system logs and access logs periodically to understand the usage and lifecycle patterns of each Amazon EC2 instance.
- Terminate your idle instances.
- Try to see whether you can eliminate under-utilized instances to increase utilization of the overall system.
- For example, examine the application that is running on an m1.large instance (1X $0.40/hour) and see whether you can scale out and distribute the load across to two m1.small instances (2 X $0.10/hour) instead.

Leverage Amazon EC2 Reserved Instances

- Reserved Instances give you the option to make a low, one-time payment for each instance you want to reserve and in turn receive a significant discount on the hourly usage charge for that instance.
- When looking at usage patterns, try to identify instances that are running in steady-state such as a database server or domain controller.
- You may want to consider investing in Amazon EC2 Reserved Instances (3 year term) for servers running above 24% or higher utilization. This can save up to 49% of the hourly rate.

Improve Efficiency

- The AWS cloud provides utility-style pricing. You are billed only for the infrastructure that has been used.
- You are not liable for the entire infrastructure that may be in place. This adds a new dimension to cost savings.
- You can make very measureable optimizations to your system and see the savings reflected in your next monthly bill.
- For example, if a caching layer can reduce your data requests by 80%, you realize the reward right in the next bill.
- Improving performance of the application running in the cloud might also result in overall cost savings.
- For example, if your application is transferring a lot of data between Amazon EC2 and your private data center, it might make sense to compress the data before transmitting it over the wire. This could result in significant cost savings in both data transfer and storage. The same concept applies to storing raw data in Amazon S3

Advanced Monitoring and Telemetry

- Implement telemetry in your cloud applications so it gives you the necessary visibility you need for your mission-critical applications or services.
- It is important to understand that end-user response time of your applications depends upon various factors, not just the cloud infrastructure – ISP connectivity, third-party services, browsers and hops, just to name a few.
- Measuring and monitoring the performance of your cloud applications will give you the opportunity to proactively identify any performance issues and help you diagnose the root causes so you take appropriate actions.
- For example, if an end-user accessing the nearest node of your globally hosted application is experiencing a lower response rate, perhaps you can try launching more web servers.
- You can send yourself notifications using Amazon Simple Notifications Service (HTTP/Email/SQS) if the metric (of a given AWS resource or an application) approaches an undesired threshold.

Track your AWS Usage and Logs

- Monitor your AWS usage bill, Service API usage reports, Amazon S3 or Amazon CloudFront access logs periodically

Maintain Security of Your Applications

- Ensure that application software is consistent and always up to date and that you are patching your operating systems and applications with the latest vendor security updates.
- Patch an AMI, not an instance and redeploy often; ensure that the latest AMI is deployed across all your instances.

Re-engineer your application

- Can you package and deploy your application into an AMI so it can run on an Amazon EC2 instance? Can you run multiple instances of the application on one instance, if needed? Or can you run multiple instances on multiple Amazon EC2 instances?
- Is it possible to design the system such that in the event of a failure, it is resilient enough to automatically relaunch and restart?
- Can you divide the application into components and run them on separate Amazon EC2 instances? For example, can you separate a complex web application into individual components or layers of Web, App and DB and run them on separate instances?
- Can you extract stateful components and make them stateless?
- Can you consider application partitioning (splitting the load across many smaller machines instead of fewer larger machines)?
- Is it possible to isolate the components using Amazon SQS?
- Can you decouple code with deployment and configuration?
- Can you containerize your application?
- Can you convert application in to Micro services Architecture?